Comunicazione bimodale nel web per mezzo di facce parlanti 3D

Giuseppe Riccardo Leone

dottorato in Ingegneria Informatica e Automatica - ciclo XIX,
Università di Firenze

Abstract

Computers are becoming an integral part of all activities in our daily lives and the simply natural interaction with them is one of the most vibrant research in the last decade. Life-like characters [92] are embodied agents living on the screens of computational devices that invite us to communicate with them in familiar, expressive/emotive multi-modal ways. Face to face communication is the main element of interaction between humans because both acoustic and visual signal simultaneously convey linguistic, extra linguistic and paralinguistic information. This is the reason why a realistic audio/visual synthesis is very important for virtual agents. This is a research topic since the early 70's and many different principles, models and animations have been proposed over the years [56].

There are many ways to control a synthetic talking face. Among them, geometric parameterization [70, 42], morphing between target speech shapes [19], muscle and pseudo-muscle models [30, 108], are the most attractive. Growing interest have encountered text to audiovisual systems [13, 55], in which acoustical signal is generated by a Text to Speech engine and the phoneme information extracted from input text is used to define the articulatory movements.

To generate realistic facial animation it is necessary to reproduce the contextual variability due to the reciprocal influence of articulatory movements for the production of following phonemes. This phenomenon, defined coarticulation [39], is extremely complex and difficult to model. A variety of co-articulation strategies are possible and different strategies may be needed for different languages [16].

In the late 90's a specification for efficient coding of shape and animation of human face was included in the MPEG-4 standard [75]. The focus was extended from traditional audio and video coding to other multimedia context including images, text, graphics, 3D scenes, animation and synthetic audio. Concerning Facial Animation the MPEG-4 standard defines the shape of the model (Facial Definition Parameters - FDPs) and a set of actions (Facial Action Parameters - FAPs); the animation is obtained by specifying

This PhD thesis, titled 'Bimodal Web Communication by means of 3D talking heads', is written in Italian with a short abstract in English.

a stream of numbers representing for each frame the values of the FAPs. Many implementations of this standard were developed [80] as stand alone applications built for research purpose. The ISTC-CNR of Padua developed in the years 2000-2006 LUCIA talking head, an open source facial animation framework [65].

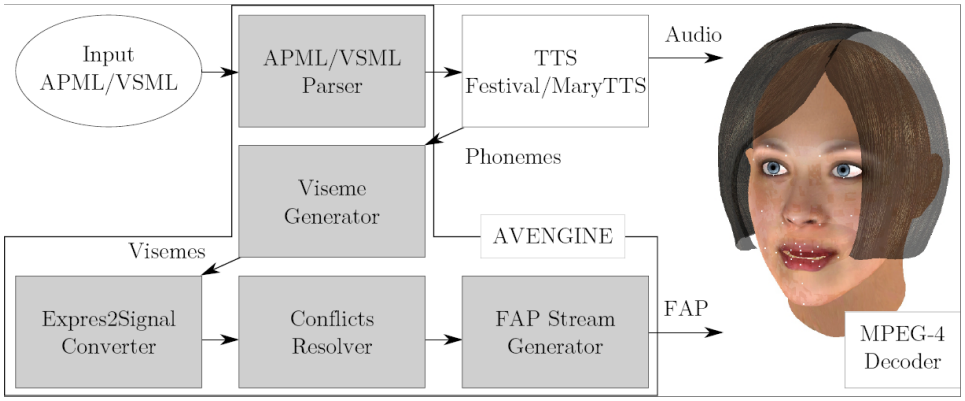


Figura 1: Audio-Visual synthesis work-flow of LUCIA talking head developed at ISTC-CNR

LUCIA is an MPEG-4 facial animation system working on standard Facial Animation Parameters and talks with the Italian version of FESTIVAL [29] and now, as a result of the present work, also with MaryTTS and Loquendo (that has better voice quality). To achieve an emotive/expressive talking head, LUCIA was built from real human data physically extracted by ELITE [40], an optic-tracking movement analyzer. LUCIA can copy a real human being by reproducing the movements of passive markers positioned on his face or can be driven by an emotional XML tagged input text, thus realizing true audio/visual emotive synthesis. In order to automate many of the operation needed for building-up the 3D avatar from the motion-captured data we used the INTERFACE Matlab tool [105]. The emotional speech is coded by means of the Affective Presentation Markup Language (APML) [22]. This XML text is the input of the Audio Video Engine module which creates the synchronized visual and audio streams used for the animation. The complete work-flow is depicted in figure 1.

The recent introduction of WebGL [113], which is 3D graphics in web browsers, opens the possibility to bring all these applications to the home computers of a very large number of persons and to burst this natural way of interaction with the machines.

The development of Lucia-WebGL (the WebGL software version of LUCIA for web browser and mobile devices) is the first result of this thesis.

Moreover, we developed a prototype of website integration with the avatar (it is a goal for the project Wikimemo.it which founded the research).

We organized a user's acceptance experiment with forty elderly people using the avatar as a personal assistant. Very early result confirm the theory that the technological divide can be reduced with friendly interfaces.

At last we present "Personal Avatar": it is a procedure to create a custom 3D model using the Microsoft Kinect sensor (a low cost range sensor) and a language training phase to create the acoustic and visual speech model using this markerless technology.

Data Acquisition

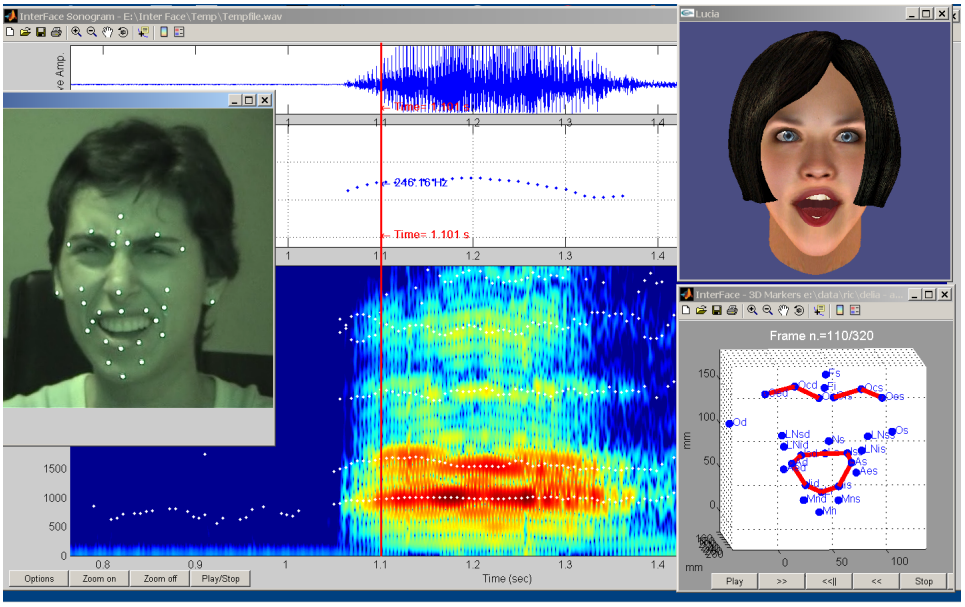


Figura 2: The temporal selection on the INTERFACE sonogram (or pitch or articulatory data) allows to synchronize the 3D reconstructed frame (n. 110/320) (bottom right), the Talking Face frame (top right), and the real movie frame (left)

LUCIA is totally based on true real human data collected by the use of ELITE [69, 68], a fully automatic movement analyzer for 3D kinematics data acquisition [40], which provides 3D coordinate reconstruction, starting from 2D perspective projections, by means of a stereo-photogrammetric procedure which allows a free positioning of the TV cameras.

Two different configurations have been adopted for articulatory data collection: the first one, specifically designed for the analysis of labial movements, considers a simple scheme with only 8 reflecting markers, while the second, adapted to the analysis of expressive and emotive speech, utilizes the full and complete set of 28 markers. All the movements of the 8 or 28 markers, depending on the adopted acquisition pattern, are recorded and collected, together with their velocity and acceleration, simultaneously with the co-produced speech which is usually segmented and analyzed by means of PRAAT [18], that computes also intensity, duration, spectrograms, formants, pitch synchronous F0, and various voice quality parameters in the case of emotive and expressive speech [20, 33].

The 3D data dynamic coordinates of passive markers (see fig. 2) are then used to create the articulatory models, to drive directly the talking face, and to study the phonetic articulatory phenomena.

In order to automate many of the operations needed for building-up the 3D avatar from the motion-captured data we used INTERFACE [105], a Matlab© software designed to integrate in an unique environment the multimodal, audio-visual, speech analysis and synthesis, the study on emotion expressions and the talking heads modeling.

Audio/video synthesis

The AVEngine software (see fig. 1) transforms an input text into audio and video (FAPs) streams.

An external text-to-speech synthesizer is used (LUCIA is able to use the Italian voice of FESTIVAL [29] and, as a result of this thesis, also with the Italian voice of MaryTTS and Loquendo).

The text-to-speech system provides an audio stream and a description of the phoneme stream (e.g., the phonemes that were synthesised along with their duration). AVengine exploits this information to synthesise a phoneme synchronized FAP stream to visually reproduce movements that an Italian speaker would perform to pronounce the phoneme stream.

To reproduce realistic facial animation in presence of co-articulation [39], a modified version of the Cohen-Massaro co-articulation model [28] has been adopted and a semi-automatic minimization technique, working on the real cinematic data, was used for training the dynamic characteristics of the model, in order to be more accurate in reproducing the true human lip movements.

Emotional synthesis

Audio Visual emotional rendering was developed working on true real emotional audio and visual databases whose content was used to automatically train emotion specific intonation and voice quality models to be included

in FESTIVAL, the Italian TTS system [104, 102, 34, 77] and also to define specific emotional visual rendering.

An emotion specific XML editor explicitly designed for emotional tagged texts was developed. The Affective Presentation Markup Language(APML) [22] for behavior specification permits to specify how to markup the verbal part of a dialog move so as to add to it the "meanings" that the graphical and the speech generation components of an animated agent need to produce the required expressions. So far, the language defines the components that may be useful to drive a face animation through the facial description language (FAP) and facial display functions.

In order to check and evaluate, by direct low-level manual/graphic instructions, various multi level emotional facial configurations we developed "EmotionPlayer", which was strongly inspired by the EmotionDisc of Zsofia Ruttkay [94]. It is designed for a useful immediate feedback and it is shown in figure 3.

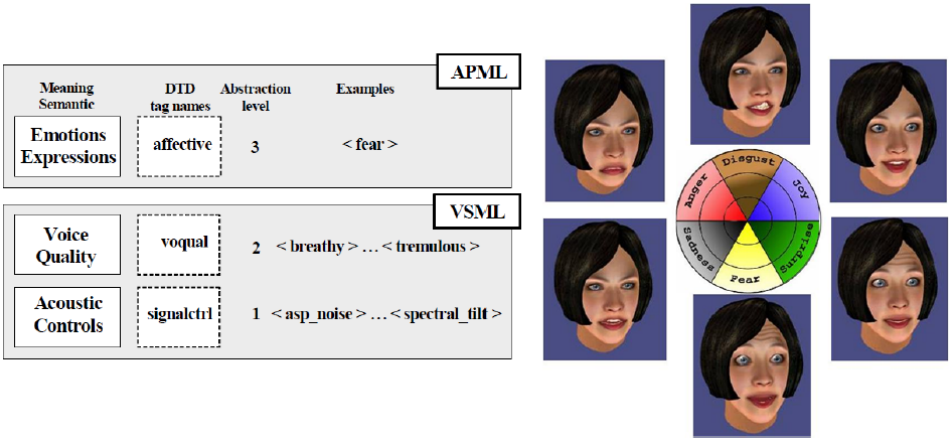


Figura 3: APML mark-up language extensions for emotive audio/visual synthesis and the Emotion Player

Lucia Talking Head

LUCIA is a MPEG-4 [59] facial animation engine implementing a decoder compatible with the "Predictable Facial Animation Object Profile". MPEG-4 specifies a set of Face Animation Parameters (FAPs), each corresponding to a particular facial action deforming a face model in its neutral state. A particular facial action sequence is generated by deforming the face model, according to the specified FAP values, indicating the magnitude of the corresponding action, for the corresponding time instant.

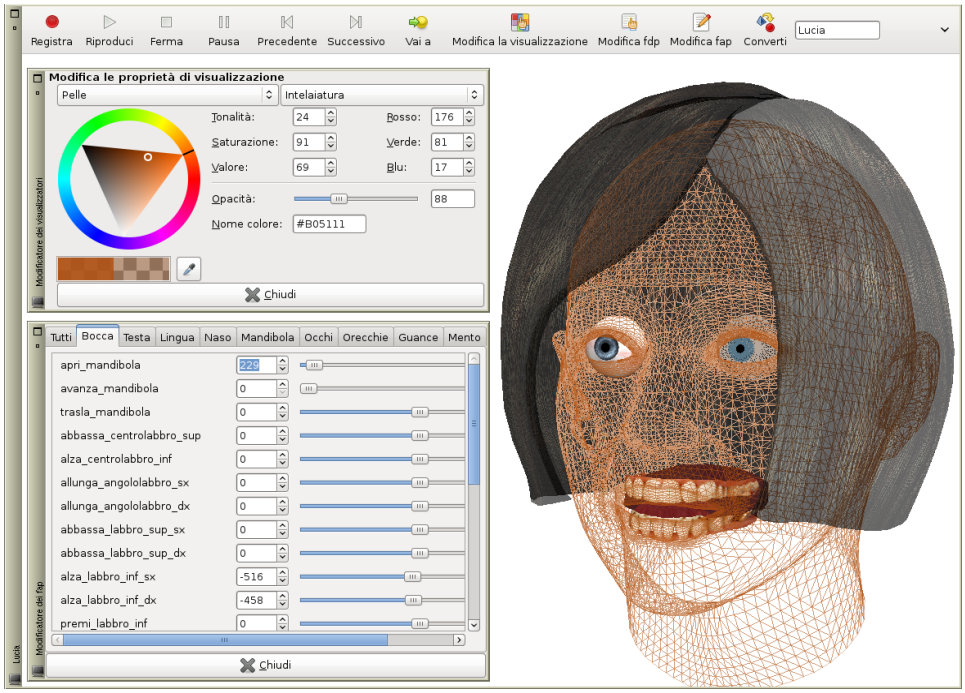


Figura 4: Lucia GUI, showing the head model wireframe

LUCIA implements the face model using a 3D mesh polygonal model and the animation is built in real-time by modifying this structure and rendering it onto screen. At the current stage of development, LUCIA is a textured young female 3D face model built with 22237 polygons, divided into 7 independent components: the skin (14116 polygons), the hair (4616 polygons), the two eyes (1120x2 polygons), the tongue (236 polygons) and the teeth (1029 polygons). All these components derived from a static VRML [7, 96] model with the exception of the eyes, whose models are dynamically generated at runtime, allowing to specify the desired level of detail.

The subdivision of the model in components is quite useful when animation is running, because it allows to divide the skin, whose reticule of polygons is directly driven by the pseudo-muscles and constitutes a continuous and unitary element, from the inner articulators, such as the tongue and the teeth, and all the other anatomical components that move themselves independently, following translations and rotations (for example the eyes rotate around their center). According to this strategy the polygons are distributed in such a way that the resulting visual effect is quite smooth with no rigid “jumps” over all the 3D model.

LUCIA emulates the functionalities of the mimic muscles, by the use of

specific displacement functions" and of their following action on the skin of the face. The activation of such functions is determined by specific parameters that encode small muscular actions acting on the face; these actions can be modified in time in order to generate the wished animation. Such parameters, in MPEG-4, take the name of Facial Animation Parameters and their role is fundamental for achieving a natural movement. The muscular action is made explicit by means of the deformation of a polygonal reticule built around some particular key points called *feature point* that correspond to the junction on the skin of the mimic muscles. Moving only the *feature point* is not sufficient to smoothly move the whole 3D model, thus, each one is related to a particular "influence zone" constituted by an ellipse that represents a zone of the reticule where the movement of the vertexes is strictly connected. Finally, after having established the relationship for the whole set of *feature point* and the whole set of vertexes, all the points of the 3D model can be simultaneously moved with a graded strength following a raised-cosine function rule associated to each *feature point* and FAP.

There are two versions of LUCIA: the "old" application so far described and now, as a result of this thesis, a new web animation framework implemented with the WebGL technology.

Lucia-WebGL

The recent introduction of the Web Graphic Library (WebGL) [113] opens new possibilities for our 3D avatar. WebGL brings native 3D graphics in the web browsers; there is no need anymore to download additional software or driver to interact with a 3D world.

Lucia-WebGL follows the common client-server paradigm as shown in fig. 5. The client (a web browser or a mobile app) connects to the server opening a web-page. The server answers with an HTML5 web-page where many new tag and multimedia contents have been introduced. One of these is the Canvas 3D, which is the place where all our 3D graphics lives. The typical WebGL application is composed by three parts: the standard html code, the main JavaScript program and a new shading language section. The html section is intended mainly for user interaction: check-box, input values and often a debugging zone for errors or warning messages during the execution. The JavaScript part is the core of the application: the graphic library itself, all the matrix manipulation, support and utility functions take place here. The input from the user is connected with JavaScript variables via ad-hoc event-driven procedures. The novelty is the third part which is the OpenGL Shading Language code (GLSL). This software runs on the Video Card and actually these are the instructions that calculate every pixel color value on the screen whenever the drawing function is called in the JavaScript main program. To be able to change the values of the GLSL variables from

the JavaScript, WebGL Application Program Interface implements special methods to connect them with JavaScript objects, arrays and variables. During the initialization of the WebGL page the shader code is compiled and copied to the memory of the Video Card ready to be executed on the Graphic Processing Unit. At the beginning of the connection model parts data are sent over the Internet using the lightweight data-interchange format JSON (JavaScript Object Notation) [48]. This is the only moment where you can wait for a while because of the amount of the data to transmit.

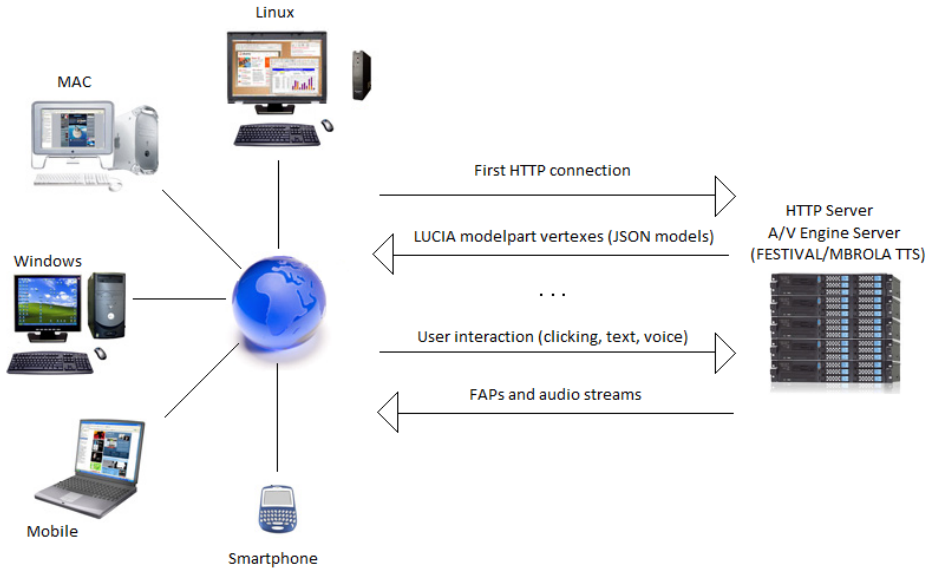


Figura 5: The new client-server architecture: WebGL allows any system, even smart-phone and P.D.A, to interact with LUCIA via standard web browsers. At the beginning of the connection the model-parts of Lucia are fetched from the server in the JSON format. Every following communication involves only FAPs and audio streams with a very low bandwidth consumption

The development of the client-server framework Lucia-WebGL is the very first result of this thesis.

The modified co-articulatory model is able to reproduce quite precisely the true cinematic movements of the articulatory parameters. The mean error between real and simulated trajectories for the whole set of parameters is lower than 0.3 mm. Labial movements implemented with the adopted model are quite natural and convincing especially in the production of bilabials and labiodentals and remain coherent and robust to speech rate variations. We presented Lucia-WebGL, the very first WebGL avatar in the world, at the Auditory-Visual Speech Processing International Conference in 2011 [58].

Real time is a demanding request expecially for low computational power machine such as mobile devices; we put much effort in the improvement of algorithms used for the animation. We used lookup table for goniometric functions and the subdivision of the face in very small areas of influence to minimize the number of loops and increase efficiency. We obtain an increase of 110% in the frame rate (from 9-11 fps to 22-25 fps on Samsung Galaxy Note smartphone).

Lucia-WebGL is the first WebGL Application ever published for the entire Apple world. This is because we were able to use an undocumented class officially enabled only for the advertisement library class (which is intended for placing ad-banners in applications). We were able to use this feature to visualize and animate our WebGL talking head. This result is published in the proceedings of the International Conference Web3D 2012 and it is very appreciated by the iOS community[11]. In figure Lucia-WebGL rendered in the Chrome web browser and in the Apple iPhone.

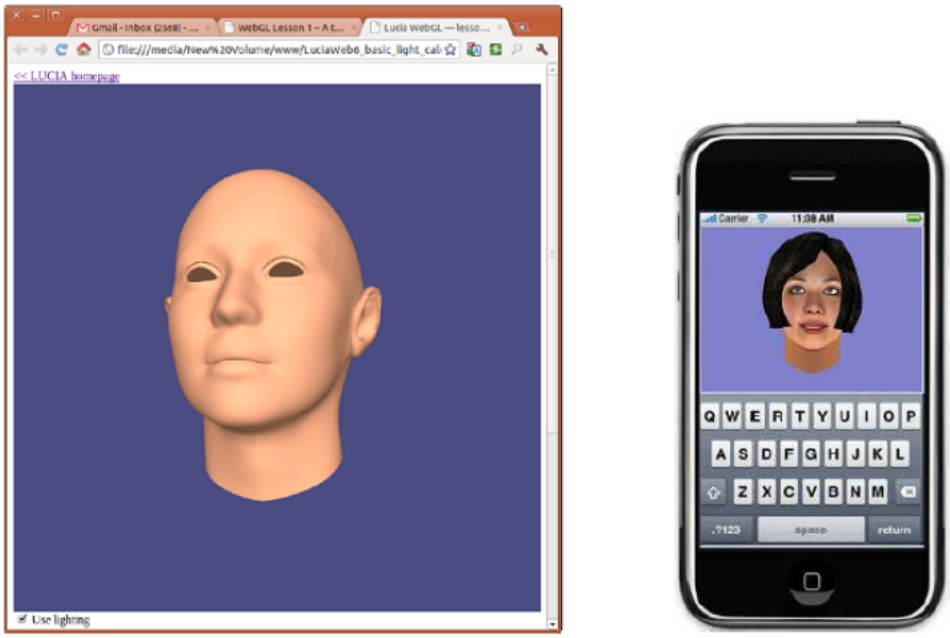


Figura 6: On the left LUCIA's skin element rendered in Chrome web-browser. Gouraud shading is adopted to obtain a very smooth and realistic curve face. On the right LuciaWebGl working on iOS (Apple iPhone).

Integration into website

This work is part of the project "Wikimemo.it, Il Portale della Lingua e della Cultura Italiana" (The portal of the Italian language and culture) funded by MIUR (Italian Ministry for University and Research) aiming at promoting the Italian culture through reading and listening. As reported by [21, p. 41], Italian language counts more than 60 million native speakers around the world and represents the 20th most spoken language worldwide, thanks also to 125 million Italian second language speakers and to the large emigrant communities still speaking Italian. The main objective of the project is to give the broad population of Italian users the opportunity to access Italian literature resources and providing people learning Italian examples of how words are uttered by native speakers.

We developed a prototype web interface (Figure 7) based on a previous work by Drioli [32] that allows users to browse the indexed content by searching for specific words and by listening how they are pronounced by native speakers within a context or to watch the visual speech with LUCIA-WebGL.

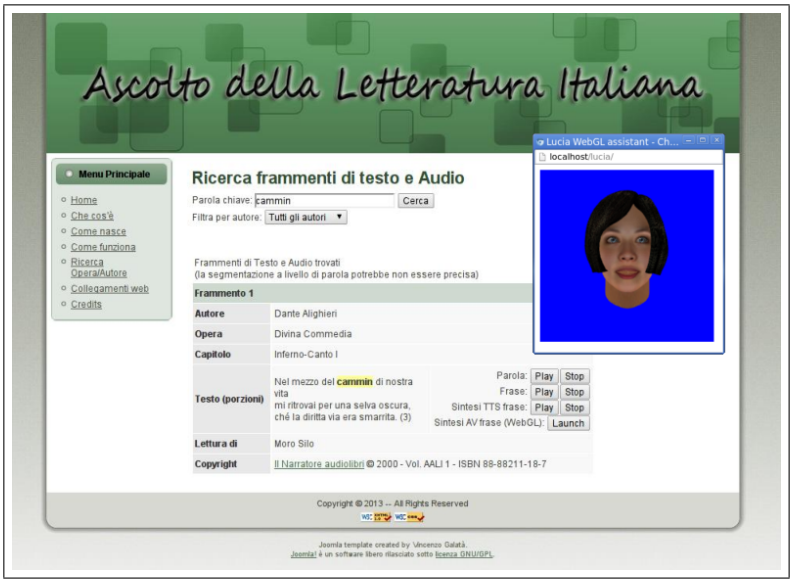


Figura 7: A screen-shot of the component showing the search result for the keyword "cammin" (highlighted in yellow): all the available information for the given search results is given in a table format. The buttons to play the results are visible on the right part of the table.

Evaluation of user's acceptability

Since 2012 the research unit ISTC-CNR is involved in the Speaky Acutattile project which aims to have an integrated domotic environment that can be controlled in a friendly way with LUCIA. In collaboration with Fondazione Ugo Bordoni of Rome we organized a set of experiments with forty elderly people using the avatar as a personal assistant.

Forty people in the age range of 65-80 participated in the test.

The main objective of this experiment has been the evaluation of user's acceptability of a very useful virtual assistant (it means very small time of response and very few "I don't understand" answer).

The test consisted in the oral communication with the avatar asking to performe twenty simple tasks (e.g. booking a train ticket, asking for a restaurant, setting the heating of the house or the alarm clock...). The user interface is depicted in fig. 8.

Figura 8: The visual user interface of the test. On the background there is a picture relative to the current task, while the communication beetwen the avatar and the human is made by natural language.

The 13 people we recorded in Padua after the test answered this way:

- 7% did not understand the speech (1 p)
- 23% finds the test hard (3 p)
- 69% enjoyed the test (9 p)
- 77% found nice the voice and the face of LUCIA (10 p)

- 84% found the assistant very useful and they would like to have it at home (11 p)
- 93% found it very easy to use (12 p)
- 100% would like to choose the aspect of the avatar

Personal avatar

Based on LUCIA's graphics and visual speech models, the possibility to create a custom 3D model using the Microsoft Kinect sensor (a low cost range sensor) has been exploited. The goal is to have an avatar which resembles to the recorded person and which talks with the same voice and similar articulators movements.

Therefore the system should provide three different outputs: the 3D static model (mesh and texture), the personal voice and the visual speech model. The last one is based on the kinematics of the speaker's articulators. There are four steps involved in the procedure:

- 3D static model building
- Data acquisition
- Voice model creation
- Co-articulation model calculation

The first goal is achieved using the software Faceshift [37] which is based on Hao-Li works [115, 114]. This model is able to reproduce very accurately the movements of the lips and the area surronding the mouth. The wireframe result is shown in fig. 9.

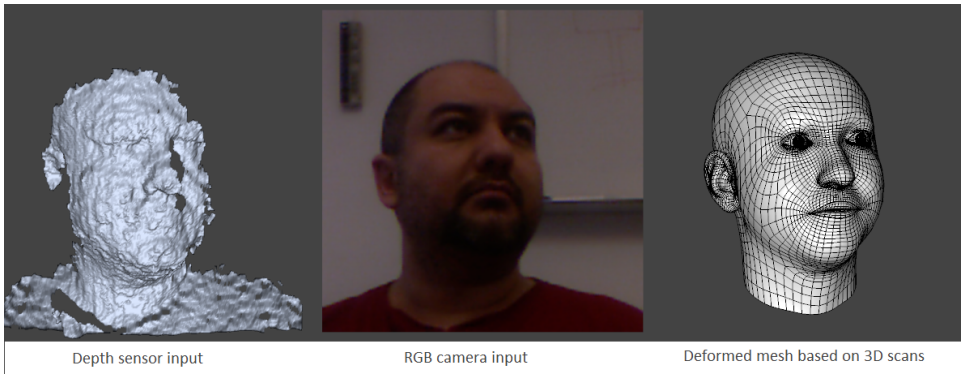


Figura 9: 3D static modeling with MS Kinect and Faceshift software

The next step involves data acquisition. The speaker reads a set of phrases with the aim to record two synchronized files: the speech and the

respective kinematics movements of the visible articulators. This set is choosen to have full cover of any audio and visual combination of the language (phonemes and visemes). The voice model and the visual speech model calculation relies on these data.

The voice model is obtained with the creation of a new italian voice with the MARYTTS synthesizer [97] (thanks to the work of Tesser from ISTC-CNR of Padua [103]).

The avatar personality is mainly based on the visual speech model. This takes into account the anatomical traits and their temporal evolution.

The model is a modified version of Coehn e Massaro co-articulation model [28]. This model describes articulators movements with pretty good realism in very different linguistic context. The introduction of the shape and resistance functions allows to overcome the limits of the original model.

The proposed procedure is depicted in fig. 10.

Conclusion and future work

LUCIA-WebGL is the first WebGL system able to use a 3D Avatar inside web pages compliant with the HTML5 standard. It is the improved WebGL version of LUCIA Talking Head. LUCIA is an MPEG-4 standard FAPs driven framework which provides several common facilities needed to create a real-time Facial Animation application. It has high quality 3D model and a fine co-articulatory model, which is automatically trained by real data, used to animate the face.

We developed a prototipe of website integration with the avatar (it is a goal for the project Wikimemo.it which funded the research).

We started the user's acceptability evaluation with a set of experiments with forty elderly people using the avatar as a personal assistant. Very early results confirm the theory that the technological divide can be reduced with this friendly interfaces.

We have a promising work in progress: Personal Avatar. Based on LUCIA speech model, we propose a standard procedure to create a custom audio/video speech model using the Kinect sensor (a low cost range sensor) which allows to replicates pretty well the movements of an actor's articulators.

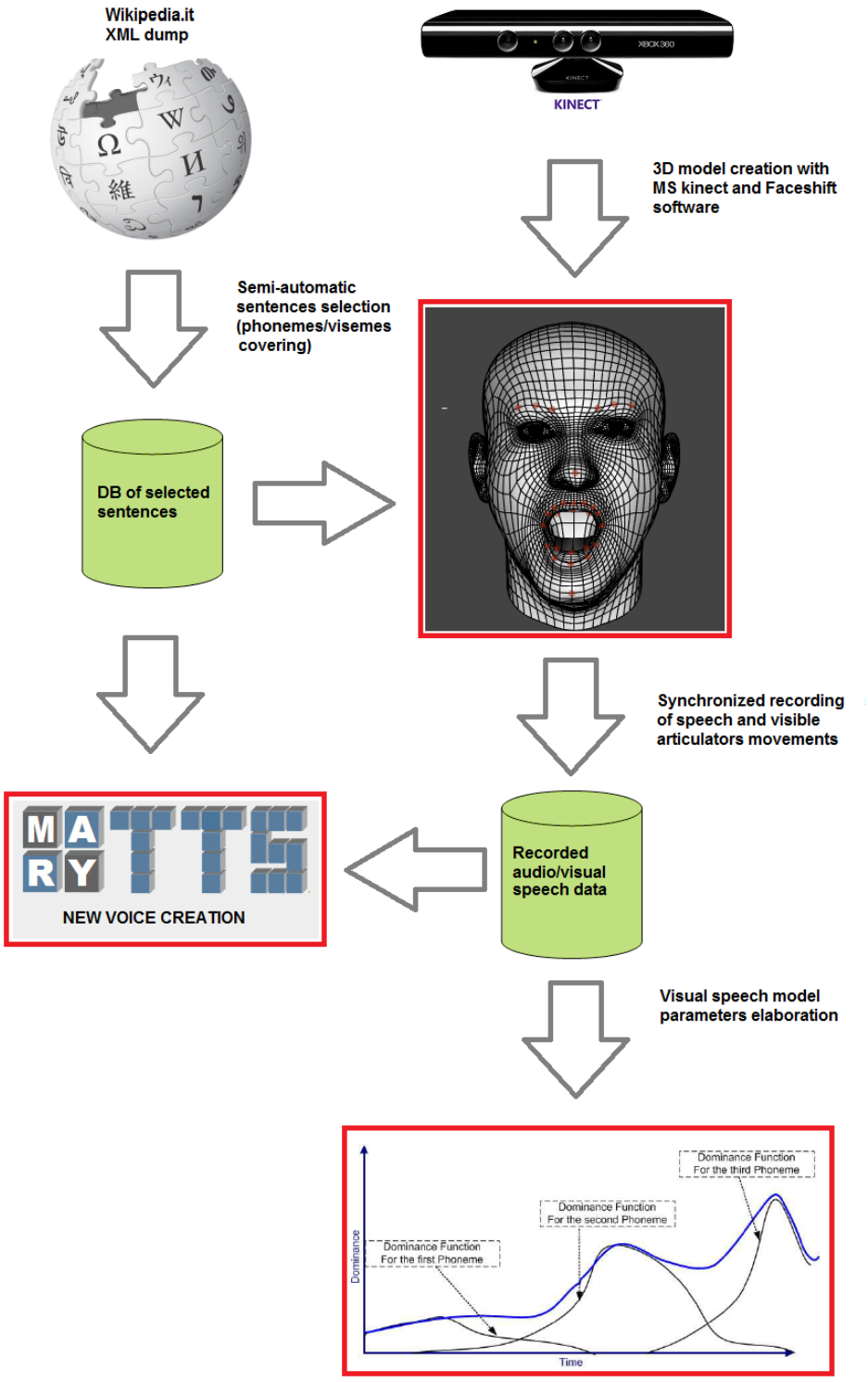


Figura 10: The functional schema of Personal Avatar creation procedure. The outputs of the system are in the red squares: the 3D model, la MARYTTS personal voice and the visual speech model.

Indice

Introduzione

Nell'ambito delle tecnologie di interazione uomo-macchina, i sistemi di sintesi bimodale audio-visiva, meglio conosciuti come "facce parlanti", costituiscono un campo di enorme interesse per la creazione di un'interfaccia amichevole. Essi conferiscono naturalezza, poiché riproducono un contesto estremamente familiare quale la comunicazione faccia-a-faccia, e, attraverso le espressioni facciali, possono essere arricchiti di emozioni che rendono l'interazione più naturale e determinano una maggiore attenzione da parte dell'utente. Le informazioni rivelate dal volto umano sono inoltre intimamente correlate e spesso complementari al messaggio uditivo, e in alcune situazioni assumono particolare importanza, come nel caso in cui il segnale acustico risulta attenuato, danneggiato o affetto da rumore. Benoit et al. [12] hanno dimostrato, ad esempio, che un sistema bimodale ha un grado di intellegibilità superiore al 60% di fronte ad un segnale con un Rapporto Segnale Rumore (SNR) di -24 dB rappresentante parole prive di senso. Nelle stesse condizioni il sistema unimodale acustico ha ottenuto valori di intelligibilità vicini allo zero. L'influenza dell'informazione visiva è così forte che se associamo la pronuncia della sillaba /b a/ all'immagine di una persona che pronuncia la sillaba /g a/, ciò che viene percepito è la sillaba /d a/. Questo risultato di fusione delle due informazioni, provato in un noto esperimento di McGurk e MacDonald [72], è noto come "effetto McGurk". Precisiamo quindi che l'utilizzo di sistemi audio-visivi migliora l'intelligibilità, a patto che i gesti facciali e il parlato siano coerenti e sincroni.

Scopo della tesi

Questa tesi è frutto del lavoro svolto presso la sezione di Padova dell'Istituto di Scienze e Tecnologie della Cognizione del CNR in relazione al progetto "Wikimemo.it: Il portale della lingua e della cultura italiana" finanziato dal M.I.U.R. per promuovere la cultura italiana attraverso la lettura e l'ascolto. Pensando al vasto numero di utenti italiani nel mondo (di cui ben 120 milioni di nati all'estero che usano l'italiano come seconda lingua) l'obiettivo principale del progetto è la realizzazione di un portale web multimediale per la consultazione delle risorse letterarie italiane. L'utente deve poter navigare

nei contenuti, effettuare ricerche di frasi e parole e sentire come sono pronunciate in un contesto specifico. I risultati di una query di ricerca possono essere ascoltati con la voce di un italiano nativo o con la voce sintetica di un avatar di supporto che ne mostra il movimento labiale.

Lo scopo principale di questa tesi è la realizzazione di una faccia parlante che sia facilmente integrabile in una pagina html e che fornisca il servizio di sintesi audio-visiva di un qualsiasi testo in lingua italiana. La produzione del parlato visuale deve essere di qualità elevata perché pensato come valido aiuto per una persona con difficoltà uditive (una persona capace di leggere le labbra).

Piano della tesi

L'animazione facciale è un campo di ricerca che nasce nei primi anni '70. Molti principi, modelli e tecniche di animazione differenti sono stati proposti negli anni. Negli ultimi anni '90 le specifiche per una codifica efficiente della forma e animazione della faccia umana è stata inclusa nello standard internazionale MPEG-4. Ne diamo una rassegna nel Capitolo 1.

Nel Capitolo 2 presentiamo LUCIA, la faccia parlante sviluppata a Padova dallo ISTC-CNR, un sistema di sintesi audio visiva basata su un modello facciale parametrico realizzato secondo lo standard MPEG-4 SNHC (Synthetic/Natural Hybrid Coding). Nel 2011 nasce la tecnologia WebGL che estende le capacità del linguaggio Javascript per generare grafica 3D in ogni browser compatibile. Tale tecnologia ha permesso di trasformare la precedente applicazione LUCIA in un moderno sistema client-server che permette di rispettare le caratteristiche di progetto: LUCIA-WebGL è la prima faccia parlante al mondo utilizzabile in ambito web. Il miglioramento della efficienza delle procedure di animazione hanno permesso l'utilizzo del client WebGL su dispositivi mobili a bassa potenza elaborativa. Un prototipo di portale del progetto Wikimemo è stato realizzato. Una valutazione dell'accettabilità del prodotto è stata effettuata mediante sessioni sperimentali con persone in una fascia d'età 65-80 anni.

Infine nel Capitolo 3 viene presentato un promettente *work in progress*: la personalizzazione dell'avatar nell'aspetto e nel parlato audio visivo per mezzo del sensore Microsoft Kinect. Si approfondisce in questa sede il funzionamento del modello di co-articolazione utilizzato anche da LUCIA, mediante il quale si raggiunge l'elevata qualità dei movimenti labiali.

Capitolo 1

L'animazione facciale

Sviluppare un modello facciale è un procedimento complesso, in cui, prima di tutto, è necessaria un'analisi approfondita dell'anatomia della testa e, successivamente, uno studio dei meccanismi che ne determinano forma, movimenti e caratteristiche superficiali. In secondo luogo, si deve stabilire, con precisione, la sua descrizione geometrica e la capacità di animazione della faccia in esame. Il passo successivo è costituito dall'animazione dello stesso: il modello facciale andrà deformato, nel tempo, per generare le espressioni volute che saranno presentate su di un display a costituire il risultato finale: un'animazione fluida ed espressiva di un volto umano.

1.1 Modellazione del volto umano

Una faccia ha una superficie tridimensionale deformabile molto complessa, ha variazioni di colore e caratteristiche superficiali, e, generalmente, presenta pieghe, rigonfiamenti e rughe. L'anatomia dettagliata della testa e del volto, quindi, è un insieme dinamico ed articolato di ossa, cartilagine, muscoli, tendini, vasi sanguigni, ghiandole, strati adiposi, tessuti connettivi e pelle.

La struttura geometrica della faccia influenza le sue potenzialità dal punto di vista dell'animazione: è essenziale dunque tener conto delle azioni che un modello dovrà compiere perché le scelte fatte al momento della modellazione geometrica del volto determinano a loro volta le capacità di animazione di quest'ultimo.

Le caratteristiche fisiche da tenere in considerazione sono estremamente importanti: la mandibola deve muoversi liberamente, le palpebre devono potersi aprire e chiudere, le guance hanno necessità di tendersi e la pelle, in generale, di comportarsi elasticamente alle sollecitazioni. Tenere conto di questi meccanismi oltre che a rendere realistico il modello contribuisce

anche a chiarire l'età e la personalità dell'attore virtuale.

Inoltre, in fase di creazione di un modello, bisogna tener presente che, sebbene tutte le facce abbiano simili strutture e comportamenti, c'è sempre un considerevole margine di variazione nelle caratteristiche, che determina quella che chiamiamo individualità di un modello; uno dei fini dell'animazione facciale è proprio sviluppare un modello che permetta queste variazioni, in maniera da rendere più realistico il risultato finale.

1.1.1 La rappresentazione geometrica

Un'animazione realistica e un *rendering* efficiente sono le caratteristiche imprescindibili della rappresentazione geometrica di un volto. Gli approcci più usati in letteratura sono:

1. rappresentazione volumetrica (come la *constructive solid geometry* utile nella costruzione di sistemi meccanici non adatta però per il progetto di strutture organiche);

2. array di elementi volumetrici (*voxels*) utili nelle rappresentazioni per fini medici perchè i dati possono essere acquisiti facilmente da strumenti che effettuano tomografie computerizzate, ma scomoda per ottenere animazioni;

3. aggregati di volumi come gli *octres*.

4. *boundary representation (B-Reps)* in cui l'oggetto viene descritto in termini delle sue superfici esterne; queste ultime devono permettere conformazioni e deformazioni come richiedono le varie espressioni del volto umano.

La rappresentazione geometrica preferita, nell'animazione facciale, è di gran lunga la *boundary representation*. Le possibili tecniche di descrizione delle superfici esterne della faccia si dividono in: superfici implicite, superfici parametriche, e superfici poligonali; le superfici parametriche, a loro volta, comprendono quelle di *Bézier, Catmull-Rom, Beta-spline, B-spline, hierachical B-spline (HB-Spline)* e *NURBS* [41]; le superfici poligonali, invece, comprendono reticoli (*meshes*) di poligoni regolari e di poligoni arbitrari.

La velocità di visualizzazione dovuta alla potenza elaborativa dell'hardware dedicato, fanno preferire a tutte queste modalità di rappresentazione quella poligonale; si tenga presente che, per come sono progettati i moderni sistemi grafici, anche le rappresentazioni prettamente non-poligonali passano attraverso una poligonalizzazione in fase di visualizzazione a schermo. Possiamo avere una mesh poligonale regolare oppure un reticolo di poligoni connessi di grandezza arbitraria (figura 1.1), questo dipende dal modo in cui

sono connessi i poligoni per formare la superficie. Una mesh poligonale regolare organizza i vertici in un array (tridimensionale) per poi connetterli a formare solo triangoli o quadrilateri. Un reticolo arbitrario di poligoni invece è costruito connettendo vertici posti, appositamente ed arbitrariamente, in maniera tale da generare la superficie voluta. Quest'ultima topologia permette una descrizione più precisa (le coordinate dei vertici sono libere da vincoli) ma in genere risulta meno efficiente.

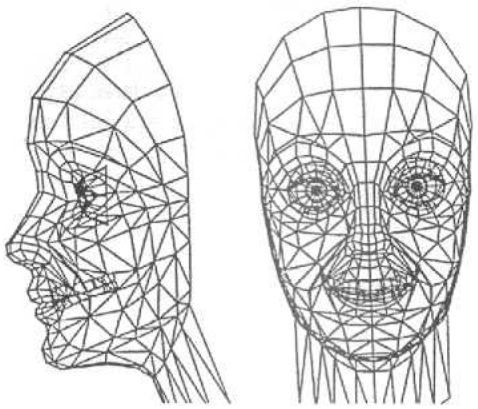


Figura 1.1: Esempio di modello facciale poligonale.

1.1.2 I Modelli

Di seguito diamo una breve panoramica sui principali modelli facciali adottati in letteratura. Si possono evidenziare tre macro-categorie:

- Modelli parametrici
- Modelli fisiologici
- Modelli pseudo-muscolari

In realtà in alcuni casi le distinzioni non risultano così nette come verranno trattate in seguito, in quanto si fa spesso un utilizzo contemporaneo di più approcci al fine di ottenere migliori risultati.

1.1.2.1 Modelli parametrici

L'idea di base consiste nell'identificare le proprietà fisiche di un elemento e nel determinare i parametri che le riescano a descrivere in modo appropriato. La combinazione dei vari parametri permette di realizzare un'ampia

gamma di possibilità ad un relativamente basso costo computazionale. Nel caso specifico dei modelli facciali sono state identificate due categorie di parametri: i parametri che determinano la conformazione topologia del viso ed i parametri che controllano le varie posizioni per generare le espressioni desiderate, incluse le diverse conformazioni della bocca. Il primo modello tridimensionale creato da Parke [81] era parametrico. Successivamente molti ricercatori hanno utilizzato questo modello come base per la realizzazione di facce parlanti per diverse lingue. La più famosa è sicuramente Baldi (figura 1.2), messa a punto presso il Perceptual Science Laboratory della University of California a S.Cruz (USCS) ed adattata alla lingua italiana dall'ISTC-CNR di Padova [26].

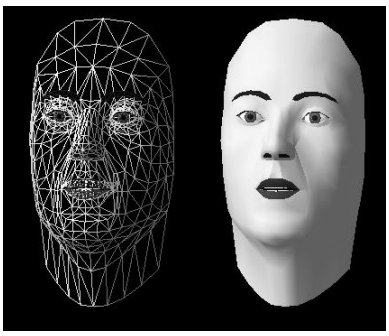


Figura 1.2: Il modello facciale Baldi.

Il modello parametrico è alla base di altre numerose facce parlanti tra le quali possiamo includere:

- la faccia parlante dell'Institut de la Communication Parlée (ICP) di Grenoble [42]
- Alf e August, elaborati al Departement of Speech Music and Hearing del KHT di Stoccolma [66]
- *J.O.E* sviluppata presso lo CSELT di Torino [53]
- *Mike e Oskar* messe a punto al DIST dell'Università di Genova [109]
- *Greta* implementata all'Università "La Sapienza" di Roma [82]
- *Lucia* realizzata dall'Istituto di Scienze e Tecnologie della Cognizione del CNR di Padova (descritta nel capitolo 2)
- *Xface* elaborata all'I.R.S.T. di Trento [6]

In particolar modo i modelli indicati in corsivo sono stati realizzati in base allo standard internazionale MPEG-4 SNHC che descriviamo nell'appendice 4.2.

1.1.2.2 Modelli fisiologici

Nei modelli basati sulla fisiologia del volto le proprietà della pelle e le azioni muscolari sono simulate tramite una maglia elastica che ne riproduce il comportamento. Molti di questi sistemi utilizzano uno schema notazionale denominato *Facial Action Coding System (FACS)* creato da Ekman e Friesen [35]. Esso descrive i movimenti facciali visibili ed è stato costruito basandosi su studi anatomici del volto umano. Le *Action Units (AU)*, unità base di questo sistema, descrivono le azioni prodotte da un muscolo o un gruppo correlato di muscoli. Lo scopo del FACS era di sviluppare un sistema completo che fosse capace di descrivere tutti i possibili, e visivamente distinguibili, movimenti facciali. In effetti FACS è concentrato, esclusivamente, solo su ciò che risulta chiaramente visibile nel volto umano, ignorando ogni invisibile o, comunque, poco percettibile cambiamento; FACS, inoltre, prende in considerazione solo i movimenti e non altri fenomeni rilevabili come cambiamenti di colore od altro.
Il FACS trae fondamento da un'analisi anatomica dei movimenti facciali e, dal momento che questi ultimi derivano da una qualche azione muscolare, definisce come ogni muscolo della faccia agisca per determinare un cambiamento effettivo e visibile sul volto, includendo tutte le azioni muscolari che possono essere indipendentemente controllate.
Il primo passo nello sviluppo del FACS è stato determinare quei muscoli che possono essere volontariamente e indipendentemente attivati e poi determinare come ognuno di questi cambia l'aspetto del volto. Il passo successivo è stato determinare se tutte le separate azioni muscolari potessero essere accuratamente distinte in maniera tale da poter essere considerate indipendenti dalle altre; data la difficoltà della differenziazione basandosi solo su quanto visibile dall'esterno, non c'è una completa corrispondenza uno ad uno tra *action units* e muscoli distinti, ed, ancora, più di una AU può essere associata allo stesso muscolo. Il FACS include 66 AU, la combinazione delle quali crea le varie espressioni facciali, e 14 AU che servono a descrivere i cambiamenti della posizione della testa.

I modelli che seguono questo tipo di approccio possono essere raggruppati in tre categorie che solitamente fanno riferimento a importanti lavori di una o più persone: i modelli strutturali, i modelli vettoriali ed i modelli stratificati.

Modelli strutturali I modelli strutturali sono correlati al primo sistema che simulava l'azione dei muscoli, creato da Platt [89]. Egli decompose la faccia in regioni e sotto-regioni strutturate gerarchicamente che possono contrarsi o rilassarsi sotto l'azione di una AU, oppure deformarsi a causa della propagazione del movimento da regioni confinanti. Una regione è definita dal punto di inserzione del muscolo e ad essa sono associate informazioni di tipo fisico (la conformazione spaziale della regione), funzionale (il modo in cui essa si modificherà sotto l'azione dell'AU) e di connessione (le regioni

AU	**Nome**	**AU**	**Nome**
AU1	Inner Brow Raiser	AU31	Jaw Clencher
AU2	Outer Brow Raiser	AU32	Lip Bite
AU4	Brow Lowerer	AU33	Cheek Blow
AU5	Upper Lid Raiser	AU34	Cheek Puff
AU6	Cheek Raiser	AU35	Cheek Suck
AU7	Lid Tightener	AU36	Tongue Bulge
AU8	Lips Toward Each Other	AU37	Lip Wipe
AU9	Nose Wrinkler	AU38	Nostril Dilator
AU10	Upper Lip Raiser	AU39	Nostril Compressor
AU11	Nasolabial Furrow Deepener	AU41	Lip Droop
AU12	Lip Corner Puller	AU42	Slit
AU13	Cheek Puffer	AU43	Eyes Closed
AU14	Dimpler	AU44	Squint
AU15	Lip Corner Depressor	AU45	Blink
AU16	Lower Lip Depressor	AU46	Wink
AU17	Chin Raiser	AU51	Head Turn Left
AU18	Lip Puckerer	AU52	Head Turn Right
AU19	Tongue Show	AU53	Head Up
AU20	Lip Stretcher	AU54	Head Down
AU21	Neck Tightener	AU55	Head Tilt Left
AU22	Lip Funneler	AU56	Head Tilt Right
AU23	Lip Tightener	AU57	Head Forward
AU24	Lip Presser	AU58	Head Back
AU25	Lips Part	AU61	Eyes Turn left
AU26	Jaw Drop	AU62	Eyes Turn right
AU27	Mouth Stretch	AU63	Eyes Up
AU28	Lip Suck	AU64	Eyes Down
AU29	Jaw Thrust	AU65	Walleye
AU30	Jaw Side To Side	AU66	Crosseye

Tabella 1.1: Lista delle Action Unit.

a cui viene propagato il movimento). Il vantaggio notevole in questo caso, è che si può cambiare la conformazione fisica della faccia, mantenendone inalterate le caratteristiche funzionali.

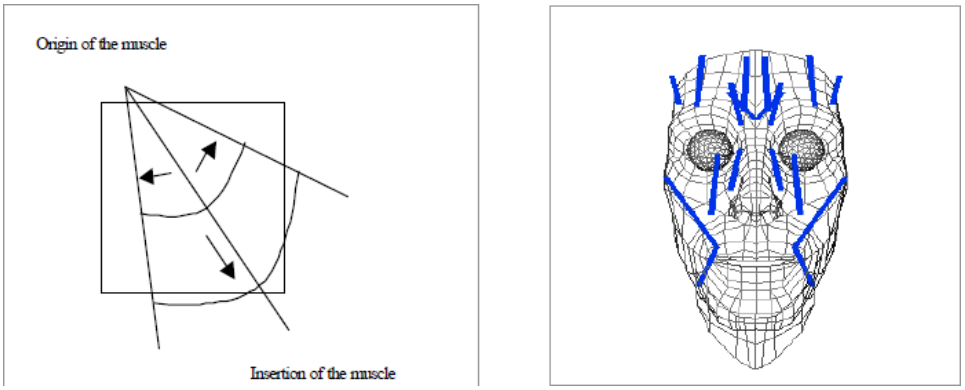


Figura 1.3: Il modello facciale vettoriale realizzato da Waters [112].

Modelli vettoriali Un modello che ha avuto molto successo (figura 1.3) è stato quello proposto da Waters [112]. Egli ha utilizzato un approccio vettoriale per descrivere in modo accurato le deformazioni della pelle: ogni muscolo è caratterizzato da un vettore che rappresenta una direzione, un'ampiezza ed una zona di influenza. In base alle caratteristiche dei muscoli egli ha distinto tre categorie: muscoli lineari, muscoli laminari e muscoli sfinterici. I muscoli lineari, che costituiscono la maggioranza, sono definiti da un'origine e da un punto di inserzione. Il loro movimento è lineare, dal punto di inserzione verso l'origine, con un'intensità che man mano decresce fino a diventare nulla nel punto d'origine. I muscoli laminari agiscono su zone ampie e sono modellati come un insieme parallelo di muscoli lineari. I muscoli sfinterici si muovono verso un centro. Per simulare l'azione muscolare delle varie espressioni Waters ha utilizzato il sistema FACS.

Il posizionamento dei muscoli lineari sul viso può essere un compito difficile: deve essere fatto manualmente per tentativi successivi e non vi è nessuna garanzia che esista una sistemazione efficiente. Nonostante questo inconveniente il successo del modello è legato alla sua indipendenza dalla struttura facciale e dal basso numero di dati necessari per animarlo.

Modelli stratificati Terzopulos e lo stesso Waters [30] svilupparono successivamente un modello della pelle a tre strati corrispettivi all'epidermide, al derma ed allo strato sottocutaneo. Le azioni muscolari si propagano attraverso gli strati creando così un'animazione estremamente realistica. Tuttavia per simulare le deformazioni tridimensionali dell'intera struttura è necessario un notevole carico computazionale.

1.1.2.3 Modelli pseudo-muscolari

I modelli pseudo-muscolari non prendono come punto di partenza la complessa struttura anatomica della faccia, ma piuttosto mirano ad ottenere un'animazione il più realistica possibile cercando di simulare l'azione dei muscoli attraverso l'utilizzo di un numero limitato di parametri. Tra essi possiamo sicuramente includere:

Modelli procedurali I modelli procedurali utilizzano procedure specializzate, chiamate *Abstract Muscle Action (AMA)* [67] che determinano i movimenti e le deformazioni della superficie in seguito ad una contrazione muscolare. Un'AMA agisce in una regione specifica del viso e corrisponde in modo approssimativo ad un muscolo. Questo metodo è stato sviluppato in stretta correlazione con la definizione dei FACS.

Modelli basati su deformazioni libere I modelli basati su deformazioni libere fanno uso di una struttura a forma di parallelepipedo che controlla una serie di punti chiave della zona di interesse. Il parallelepipedo si può stirare, schiacciare, curvare, espandere o contrarre influenzando conseguentemente i punti chiave cui è legato. Evoluzioni di questo modello utilizzano un cilindro al posto del parallelepipedo, permettendo così un grado di libertà più elevato nelle deformazioni.

Modelli basati su spline Tutti i modelli esposti in precedenza utilizzano una faccia modellata attraverso poligoni opportunamente smussati attraverso l'algoritmo di Henry Gouraud (un algoritmo di rendering molto veloce, sviluppato nel 1971; fornisce immagini in cui le superfici curve appaiono sfumate uniformemente, anche se la mesh originale è sfaccettata.). Le *spline* rappresentano un approccio alternativo a questo ed offrono maggiore flessibilità rispetto ai modelli poligonali. Sono stati, ad esempio creati dei sistemi che fanno uso di B-splines, in cui l'animazione è ottenuta muovendo gruppi di punti di controllo. I punti da coordinare in questo modo sono numerosi, rendendo così difficoltosa la gestione del modello. Con l'uso di spline gerarchiche si è riusciti a ridurre tale quantità. Wang et al. [110] infine hanno realizzato un sistema che associa le spline gerarchiche ad un sistema di pseudo-muscoli, con il quale sono riusciti a realizzare in modo molto realistico effetti come rigonfiamenti e rughe.

1.2 Animazione e tecniche di controllo

Per creare un'animazione dei modelli facciali appena trattati sono state sviluppate diverse tecniche automatiche di controllo; per le immagini bidimensionali generalmente si usano le tecniche di *key-framing* e *Morfing*; per i

modelli 3D i metodi principali sono tre: metodi basati su regole, metodi basati su analisi, metodi basati su esecuzioni.

1.2.1 Key-framing e interpolazione geometrica

La tecnica del *key-framing* utilizza delle specificazioni complete, denominate per l'appunto *key-frames*, che riproducono le espressioni desiderate. In questo caso non si necessita di alcuna informazione fisica o strutturale per quanto riguarda il modello. Per generare l'animazione, i *key-frames* vengono interpolati di solito in modo lineare. Le immagini sono generate variando i parametri della funzione di interpolazione che vanno a modificare direttamente la posizione dei vertici della faccia.

Un esempio lo possiamo trovare nel video *Tony de Peltrie* [36], in cui è stata presa come riferimento la faccia "neutrale" ottenuta dalla digitalizzazione di un modellino in argilla. A partire da essa i vari *key-frames* sono stati ottenuti basandosi su fotografie delle varie espressioni; l'animazione è stata ottenuta utilizzando B-Splines Bicubiche come metodo di interpolazione.

1.2.2 Morphing

Il morphing consiste in una trasformazione tra due immagini bidimensionali. Si basa sull'utilizzo contemporaneo di una funzione che trasforma la prima immagine nella seconda e di una funzione inversa per trasformare la seconda immagine nella prima, ottenendo l'effetto di una dissolvenza incrociata. Attraverso il morphing di immagini opportunamente scelte si possono ottenere animazioni con un grado elevato di realismo, anche se richiedono dei tempi ed un intervento umano eccessivo. Trattandosi poi di immagini bidimensionali, il punto visivo rimane fisso.

Per ovviare a questa limitazione Pighin et al. [88] hanno combinato il morphing bidimensionale con la deformazione di un modello tridimensionale.

1.2.3 Metodi basati su regole

Questi metodi utilizzano un insieme di regole per organizzare i legami tra espressioni, emozioni, intonazione, enfasi e contesto semantico. Tali regole possono lavorare a differenti livelli (frase, parola, fonema oppure attraverso un linguaggio di scripting che integra informazioni aggiuntive).

1.2.4 Metodi basati su analisi

Tale tecnica consiste nell'estrarre le informazioni da videoriprese dal vivo, che vengono elaborate in termini di contrazioni muscolari. Ricavare tali informazioni non è un compito facile, perché non risulta evidente la correlazione tra le deformazioni facciali e quelle muscolari. Waters [101] ha utilizzato un profilo dinamico posizionato manualmente su regioni prefissate

di immagini animate del volto. Il profilo dinamico è una *spline* che segue la deformazione nel tempo di una caratteristica facciale ed il cui dispiegamento viene interpretato come una contrazione muscolare. Le tecniche basate su *optical flow* individuano il movimento attraverso l'analisi della luminosità dei pixel in sequenze successive di immagini invece di ricercare caratteristiche specifiche del volto.

1.2.5 Metodi "performance driven"

Tale tecnica, più nota come *motion capture*, è estremamente efficace, ed è la più utilizzata nella realizzazione di film d'animazione. Sul soggetto vengono applicati speciali marcatori e i suoi movimenti vengono registrati da una o più videocamere. Nella prima immagine i marcatori vengono etichettati manualmente per facilitarne il riconoscimento; di seguito un software ne individua gli spostamenti in modo automatico. Sono molti gli esempi che vengono dal mondo cinematografico; una menzione particolare va fatta per la tecnica usata nel film "The audition"[83]: l'animazione della faccia caricaturale di un cane è basata sulle immagini registrate del viso di un attore posto tra due specchi, in modo tale da avere più punti di vista della *performance*. In genere vengono usati dei sistemi optoelettronici che utilizzano due telecamere per calcolare l'andamento tridimensionale dei marcatori. Il modello del parlato visivo dell'avatar LUCIA oggetto della presente tesi è stato ottenuto in questo modo. Presentiamo il sistema optoelettronico ELITE nell'appendice 4.1.

1.2.6 Controllo dei movimenti labiali

In questa sezione entriamo nel dettaglio per quanto riguarda le tecniche di controllo delle labbra. Le labbra possono generare movimenti molto complessi grazie ai numerosi muscoli localizzati attorno alla bocca. Il movimento labiale è associato alla produzione di una varietà di suoni differenti con la conseguenza che molte parole visivamente si assomigliano. Non è possibile collegare in modo biunivoco un singolo suono del parlato con un'unica forma labiale; risulta perciò difficile modellare le variabilità della forma degli articolatori labiali. Sono stati adottati diversi metodi, tra i quali nessuno sembra esser prevalso sugli altri. In realtà la scelta di un approccio rispetto ad un altro dipende dall'ambito di utilizzo, dal livello di realismo che si vuole raggiungere o da eventuali limiti (carico computazionale, costi hardware).

Una differenziazione tra vari metodi di controllo la si può individuare nella forma iniziale dalla quale si estraggono le informazioni. Possiamo quindi distinguere sistemi text-driven, speech-driven, image-driven, sistemi ibridi text-and-speech-driven, image-and-speech-driven e EMG-and-speech-driven.

Text-driven

L'idea principale è quella di decomporre un testo scritto in ingresso nella sua rappresentazione di base, tramite un programma che in modo automatico estrae i fonemi e ne stima l'estensione temporale. Successivamente il fonema è associato all'appropriato visema e vengono utilizzate tecniche di interpolazione o di morphing per unire visemi consecutivi e generare l'animazione. Un'altra possibilità è utilizzare i fonemi per comandare una serie di funzioni parametriche continue. Dal testo possono anche essere ricavate informazioni per quanto riguarda aspetti prosodici (e.g. intonazione) e alcuni aspetti semantici.

Speech-driven

Il segnale audio preregistrato viene analizzato per determinare le informazioni del parlato da associarsi ai visemi. Si vorrebbe in qualche modo determinare il ritmo e l'intonazione nonchè lo stato emozionale del soggetto. Tra i metodi utilizzati possiamo distinguere:

- **Tracking del volume.** L'apertura della bocca e la rotazione della mandibola vengono regolate in base all'intensità del segnale sonoro. Si tratta di un approccio semplicistico che produce risultati di scarsa qualità.

- **Riconoscimento automatico del parlato.** Attraverso dei sistemi di riconoscimento del parlato il segnale acustico viene analizzato per identificare i fonemi e la durata dei segmenti fonemici.

- **Analisi tramite un predittore lineare.** Lewis e Parke hanno descritto un procedimento per estrarre classi visemiche di fonemi grazie all'applicazione di un predittore lineare al segnale acustico [60].

- **Spectrum matching.** Al segnale acustico viene applicata una Trasformazione di Fourier Discreta (DFT) a successivi intervalli per calcolarne lo spettro, che viene interpretato come una funzione di densità di probabilità. Vengono calcolati i momenti di vario ordine i cui valori localizzano un punto specifico su superfici parametriche che predicono la posizione degli articolatori. Il metodo è stato sviluppato da McAllister et al. [71].

- **Analisi tramite ANN, TDNN o VQ.** Esistono metodi che non decompongono il segnale in classi significative, ma piuttosto cercano di associare il sonoro direttamente a parametri visuali continui. Tali metodi fanno uso di Articial Neural Networks (ANN) [74], Time Delay Neural Networks (TDNN) [54] o Vector Quantization (VQ) [116].

Text-and-speech-driven

Questa metodologia combina le informazioni ricavate da un segnale audio e dal corrispettivo testo scritto. In particolare il segnale audio viene utilizzato per identificare le vocali; dal testo invece si ottengono informazioni riguardanti le consonanti.

Image-and-speech-driven.

Attraverso sistemi optoelettronici o attraverso riprese cinematografiche viene rilevato l'andamento temporale di alcuni marcatori posti intorno alla zona labiale. Tale andamento viene riprodotto controllando analoghi punti sulla faccia sintetica. Attualmente si stanno evolvendo sistemi automatici di riconoscimento di immagini digitalizzate che si basano su metodi statistici. Fra questi metodi i più importanti sono quelli che operano senza avere una conoscenza a priori della struttura dei dati di partenza, che sfruttano proprietà statistiche della sorgente e che sono già state applicate al riconoscimento vocale (Articial Neural Networks, Principal Component Analysis, Hidden Markov Models).

EMG-and-speech-driven.

Gli elettrodi di un elettromiografo (EMG) vengono collocati sull'area labiale ed attorno ad essa. I dati rilevati sono utilizzati per estrarre informazioni sulle contrazioni muscolari che vengono utilizzate per guidare il modello.

1.2.7 Sincronizzazione

Una volta determinato l'andamento temporale del segnale visivo, sorge il problema della sincronizzazione con il segnale audio. Avere un buon grado di sincronismo è fondamentale per non avere spiacevoli effetti simili al ritardo che si può notare in alcune trasmissioni di videoconferenza evitando lo "effetto McGurck" descritto nell'introduzione. Alcuni studi stabiliscono che la differenza di sincronismo diventa evidente se il segnale audio è in anticipo di $130ms$ oppure se è in ritardo di $260ms$ [45]. Per suoni netti si applicano valori più piccoli (in un contesto fonetico le consonanti /p b m/ sono assimilabili a suoni netti). I valori per cui la differenza comincia ad essere percepita sono di $40ms$ e $60ms$ rispettivamente per audio in anticipo e in ritardo [100].

1.3 L'animazione facciale in MPEG-4

MPEG-4 è uno standard ISO/IEC [47] nato nel 1998 e costituisce le fondamenta della televisione digitale, delle applicazioni grafiche interattive e multimediali; esso ha assicurato una standardizzazione tecnologica a livello

di produzione e di diffusione di contenuti multimediali naturali e sintetici.

MPEG-4 è un sistema di codifica e decodifica di una scena audiovisiva. Tale scena è descritta mediante una struttura gerarchica chiamata *scene graph*. I nodi del grafo sono i componenti della scena e sono chiamati *media objects*. Tra i *media objects* utilizzabili vi sono gli oggetti sintetici ovvero caratteri virtuali che convivono nella stessa scena insieme ad elementi reali. Per questo MPEG-4 è utilizzato anche come sistema di animazione facciale.

Face and Body Animation - FBA è la sezione dello standard che si occupa di tale argomento. Essa è molto corposa e dettagliata dimostrando la notevole importanza che già negli anni Novanta questa classe di oggetti rivestiva e riveste tutt'oggi. Per quanto concerne la presente tesi siamo interessati alla parte che riguarda l'animazione dei volti.

MPEG-4 definisce una faccia netrale ed individua su di essa un insieme di *feature points* o punti fondamentali. Fornisce due set di parametri per l'animazione e la calibrazione dei *feature points*:

- ***FDP*** *(Facial Defintion Parameters)*: sono i parametri per la definizione della struttura della faccia; possono essere usati sia per modificare la forma e l'aspetto del modello facciale in uso oppure per trasmettere un nuovo modello completo, insieme ai criteri che devono essere applicati per animarlo.

- ***FAP*** *(Facial Animation Parameters)*: descrivono il movimento della faccia, sia a basso livello (spostamento di uno specifico, singolo punto chiave della geometria del volto) sia ad alto livello (riproduzione di espressioni ed emozioni). I FAPs costituiscono il vero e proprio *stream* di parametri che, associato al particolare *media object*, genererà l'animazione ed il comportamento nel tempo.

Abbiamo dedicato un approfondimento specifico dello standard MPEG-4 nell'appendice 4.2.

1.4 Il modello del parlato visivo

Nell'ambito del controllo labiale non è stato menzionato un aspetto fondamentale, che, data la sua rilevanza per questa tesi, è bene approfondire in questa sezione. L'influenza, infatti, del contesto in cui un fonema si trova è determinante per la conformazione finale degli articolatori. Tale fenomeno, comunemente definito come "coarticolazione", è stato ed è tutt'oggi oggetto

di numerosi studi. I risultati ottenuti, talvolta contrastanti, dimostrano l'incredibile complessità del meccanismo di produzione del parlato e la difficoltà nel creare un modello che ne riproduca il comportamento.

I sistemi basati sulla rappresentazione attraverso segmenti fonetici concatenati presentano due principali difficoltà. In primo luogo segmentare significa stabilire dei confini di demarcazione tra un fonema e l'altro, mentre il parlato è un processo continuo in cui non si possono determinare frontiere così nette. Come seconda ragione, un tale frazionamento suggerisce l'idea non corretta di indipendenza dell'unità fonetica, mentre in realtà essa è fortemente dipendente dal contesto in cui si trova. Ciò è dovuto al fatto che il movimento dei differenti articolatori per la produzione di successivi fonemi si sovrappone e interagisce col movimento articolatorio dei segmenti adiacenti. Questo fa capire perché gli studi sui fenomeni coarticolatori abbiano un grande rilievo. Da una parte si cercano delle teorie che ne spieghino l'origine, la natura ed il funzionamento, dall'altra si vuole creare dei modelli che ne predicano i dettagli. Gli studi coarticolatori riguardano due campi principali:

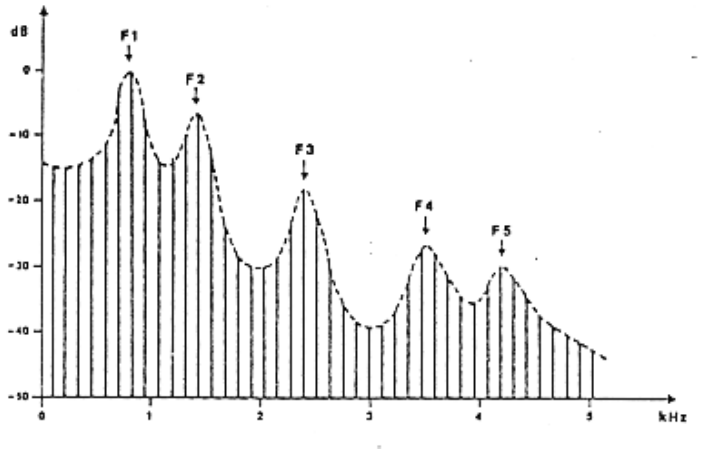


Figura 1.4: Spettro del segnale acustico e rispettive formanti.

- **la variabilità acustica:** essa è legata al concetto di target acustico, che si riferisce alla posizione e all'ampiezza delle frequenze di risonanza del suono dette "formanti" (vedi figura 1.4).
- **la variabilità articolatoria:** l'oggetto dell'analisi è la conformazione degli articolatori in termini di alcuni parametri costitutivi caratteristici (e.g. apertura e larghezza labiale, protrusione del labbro inferiore e superiore, rotazione mandibolare) o dell'attività muscolare rilevata con strumenti elettromiografici (EMG).

Le indagini sul processo che lega le invarianti unità fonemiche alla variabilità della conformazione acustica ed articolatoria possono essere ricondotte ai due seguenti aspetti fondamentali:

- **l'evoluzione temporale:** si cerca di determinare l'estensione e la direzione temporale dei movimenti coarticolatori che caratterizzano le unità di base del parlato.
- **l'emergere di gesti conflittuali:** l'obiettivo è definire cosa accade quando gesti contrastanti devono essere prodotti in successione dalla stessa struttura articolatoria; si vuole capire se la coarticolazione viene bloccata oppure continua la sua estensione con un grado diverso a seconda della costrizione imposta.

Grande rilievo è stato dato anche alle differenze tra le varie lingue. Il proposito è verificare se esistono caratteristiche universali che sottostanno al fenomeno coarticolatorio e di caratterizzare le diversità che contraddistinguono le varie lingue.

1.4.1 Teorie e modelli

Il tentativo di spiegare la variabilità degli aspetti coarticolatori ha dato vita a numerose teorie e modelli. Data la complessità dei comportamenti articolatori, gli studi da cui sono conseguite le varie teorie fanno in genere riferimento ad analisi di aspetti particolari e circoscritti. In questo paragrafo vogliamo dare una panoramica generale dei principali modelli con l'obiettivo di evidenziare le idee che stanno alla base dei modelli creati.

Considereremo sia l'ambito acustico sia quello articolatorio poiché si influenzano reciprocamente e molti studi si sono basati su entrambi gli aspetti. La trattazione che segue fa riferimento a quella proposta da Farnetani e Recaens [38]).

1.4.1.1 La variabilità adattativa

Alla base di questa teoria proposta da Lindblom [62] vi sono studi sulle variabilità fonetiche riguardanti le cause della riduzione delle formanti vocaliche nello Svedese. In precedenza la principale convinzione era che essa fosse la conseguenza dell'inerzia dei meccanismi di produzione verso target acustici successivi. Lindblom cercò di andare oltre questa visione ed elaborò la teoria della "variabilità adattativa" che afferma che la produzione acustica deriva dall'adattamento alle esigenze dettate dalla situazione comunicativa. In particolare il suono sarà un'interazione tra l'esigenza dell'ascoltatore di comprendere con successo il messaggio e la tendenza all'economia da parte del comunicante. L'atteggiamento economico si manifesta con un graduale

decremento del contrasto fonetico e con un aumento della coarticolazione fisiologica.

In una prima versione del modello, Lindblom trovò che, in sequenze CVC, la frequenza delle formanti delle vocali varia in funzione della durata e del contesto consonantico. Una motivazione di tale comportamento è che la riduzione dei target è la diretta risposta del sistema motorio ad un aumento della frequenza del numero di stimoli.

Successive ricerche hanno mostrato che non sempre questo accade descritto, anzi, in alcuni casi la riduzione può verificarsi anche a ritmi bassi. Ciò era dovuto a variazioni del contesto comunicativo cui conseguivano modifiche nella produzione del parlato. Lindblom, quindi, nella versione finale del modello, prese in considerazione non solo la durata contestuale delle unità fonetiche, ma anche aspetti che caratterizzassero lo stile del parlato.

1.4.1.2 Il modello di Öhman

Gli studi di Öhman [78, 79] si basano sulla caratteristiche acustiche ed articolatorie di sequenze VCV prodotte isolatamente in diverse lingue. Egli trovò che in tali sequenze, dove C è una consonante occlusiva, il valore delle formanti non dipende solamente dalla consonante e dalla adiacente vocale (transizione VC e CV), ma anche dalla transizione della prima vocale verso la seconda vocale. Egli, quindi propose un modello numerico in cui venivano considerati separatamente l'andamento vocalico da quello consonantico. La caratteristica finale veniva quindi generata dalla combinazione pesata dei due e alla consonante veniva associato un coefficiente che quantificava l'"emergenza" della consonante. I dettagli del modello vengono forniti in 1.4.3.

1.4.1.3 La fonologia generativa

Chomsky ed Halle [23] proposero una chiara differenziazione della coarticolazione da altre caratteristiche contestuali. Definirono quindi:

- **coarticolazione:** la coarticolazione, vista a livello fonetico, è data dalla transizione tra una vocale e la consonante adiacente, ovvero l'aggiustamento nella forma dei tratti vocalici risultanti come anticipo ad un movimento successivo. I principi alla base del meccanismo coarticolatorio si basano su regole universali, indipendenti dalla lingua considerata.

- **assimilazioni:** alle assimilazioni fanno riferimento le operazioni svolte sulle caratteristiche fonologiche ovvero sui costituenti classificatori minimi di un fonema. Esse sono giustificate da regole fonologiche che collegano regole lessicali con la loro rappresentazione fonetica e in que-

sto senso si riferiscono alla modificazione di aspetti che sono parte della grammatica e sono specifici della lingua utilizzata.

1.4.1.4 La teoria dell'estensione delle caratteristiche

Daniloff e Hammenberg [31] si allontanarono decisamente dall'idea proposta da Chomsky di caratteristiche universali.

I loro studi analizzarono l'anticipazione dell'arrotondamento delle labbra in presenza di una vocale arrotondata (/o, u/). Venne dimostrato che tale azione iniziava tre o quattro segmenti antecedenti, e indicava chiaramente che la coarticolazione non era dovuta ad un fenomeno inerziale ma ad un processo di estensione di caratteristiche specifiche. Gli aggiustamenti relativi alla coarticolazione non possono essere considerati inevitabili e universali; nel Russo, ad esempio, gli accomodamenti sembrano essere minimi.

Secondo la loro visione, quindi, la coarticolazione anticipatoria sarebbe spiegabile dalla grammatica di un linguaggio attraverso regole di estensione nel tempo delle caratteristiche fonologiche.

Il modello look-ahead A partire dai lavori di Henke [46] e di Kozhevnikov e Chistovich [52] sull'anticipazione coarticolatoria labiale e linguale, Daniloff e Hammenberg decisero di assegnare ai segmenti le seguenti caratteristiche:

- + per le vocali arrotondate;
- − per le vocali non arrotondate;
- 0 per segmenti che presentavano una caratteristica neutra rispetto all'arrotondamento.

Il modello da loro creato stabilisce che i segmenti neutri assumono la dicitura del primo segmento successivo non neutro. L'anticipazione quindi si estende lungo tutti i segmenti rappresentati dal segno +, mentre viene bloccata da segmenti contraddistinti da un −. Il movimento di arrotondamento viene infine visto come la traiettoria che collega i segmenti con caratteristica + con segmenti con caratteristica −(figura 1.5).

Gli aspetti fondamentali del modello sono:

- l'estensione dell'anticipazione è variabile e dipende dalla durata di tutti i segmenti neutrali precedenti. Alcuni studi su diverse lingue concordano con questo aspetto, mentre altri sono in disaccordo. Gli studi sul francese svolti da Benguerel [10], ad esempio, mostravano che il movimento anticipatorio delle vocali velari[1] non si estendeva oltre i due

[1]Le vocali velari sono quelle la cui articolazione avviene tra il dorso della lingua ed il velo palatino (/e,i/).

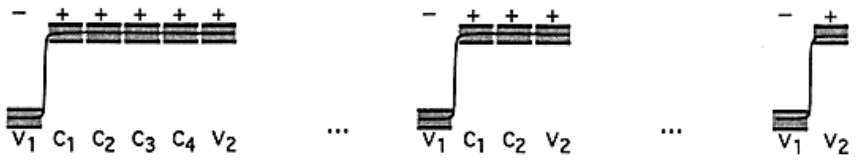


Figura 1.5: Il modello Look-Ahead di Daniloff e Hammemberg rappresentato con la struttura "a finestra" definita da Keating (vedi 1.4.1.6).

segmenti precedenti; Bell-Berti ed Harris [8] in seguito hanno trovato per l'inglese che il tempo di inizio di tale movimento era costante. Si può pensare che la caratteristica di tipo look-ahead sia circostanziale alla specifica lingua ed è in parte vero; tuttavia le divergenze dei risultati in alcuni casi riguardavano anche una stessa lingua.

- il movimento anticipatorio può iniziare solo dopo segmenti con caratteristica opposta rappresentati dal segno −. Benguerel, Cowan et al. [9] hanno precisato attraverso i loro studi che il movimento inizia "durante" piuttosto che "dopo" tale segmento, in particolar modo in presenza di un ridotto numero di segmenti neutrali frapposti. Inoltre, gli studi su sequenze VCV indicano che la prima vocale, anche se ha una caratteristica contrastante, è influenzata dalla seconda.

- i segmenti neutrali assumono la caratteristica dei segmenti + o −. In realtà l'influenza può essere solo parziale. Ciò può essere spiegato con il fatto che alcuni segmenti ritenuti neutrali possano avere una loro caratteristica articolatoria.

1.4.1.5 La resistenza alla coarticolazione

In seguito agli studi sulle caratteristiche articolatorie dei segmenti neutri, Bladon e Al-Bamerni [15] introdussero il concetto di resistenza coarticolatoria associando un indice numerico alle specifiche caratteristiche di un fono che rappresentasse una maggiore o minore resistenza alla coarticolazione. I valori di tali indici non sono da considerarsi universali, ma variano in base alla specifica lingua o al dialetto.

1.4.1.6 Il modello a finestra

Keating [49, 50] propose un nuovo modello articolatorio che tiene in considerazione i continui cambiamenti spaziali e temporali allontanandosi dai presupposti del modello ad estensione delle caratteristiche. Secondo questo modello si possono verificare le seguenti possibilità, in base a regole fonologiche specifiche della lingua considerata:

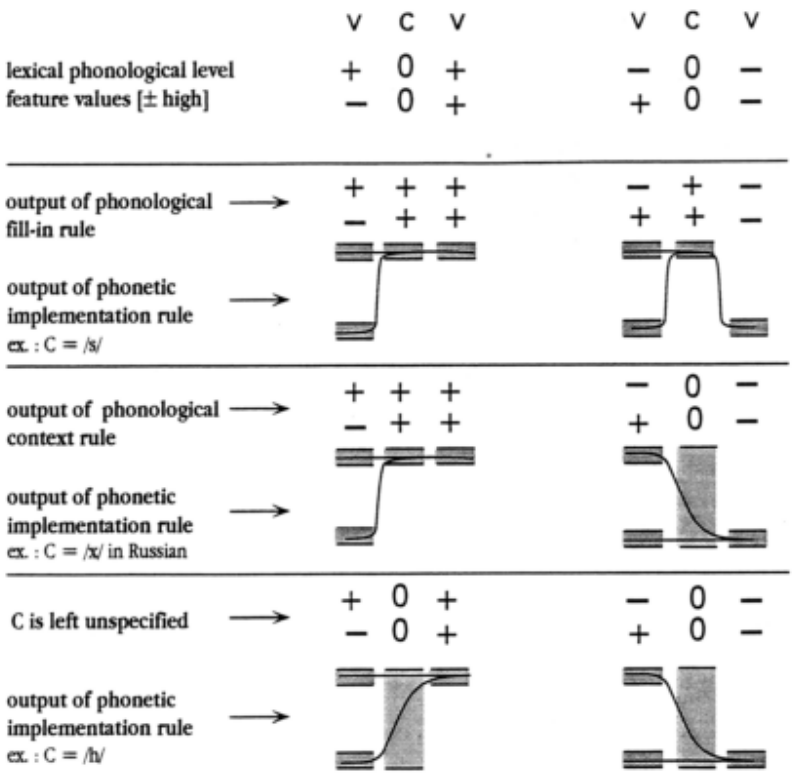


Figura 1.6: Il modello “a finestre” di Keating.

- Il segmento neutro acquisisce sempre la caratteristica arrotondata (vedi figura 1.6, consonante /s/).
- il segmento neutro acquisisce la caratteristica arrotondata o rimane neutro in base a specifiche situazioni (vedi figura 1.6, consonante /x/).
- Il segmento neutro rimane neutro (vedi figura 1.6, consonante /h/).

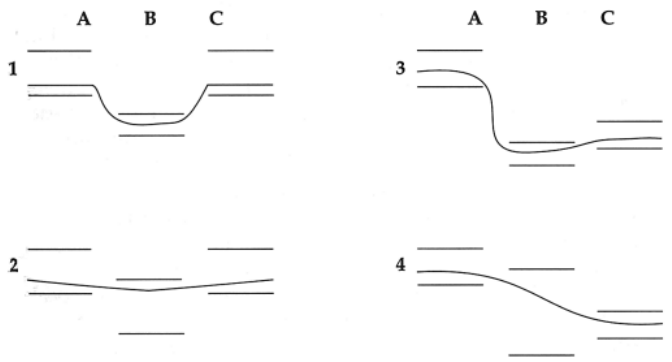


Figura 1.7: "Finestre" di diversa collocazione e ampiezza e la corrispettiva implementazione del movimento degli articolatori.

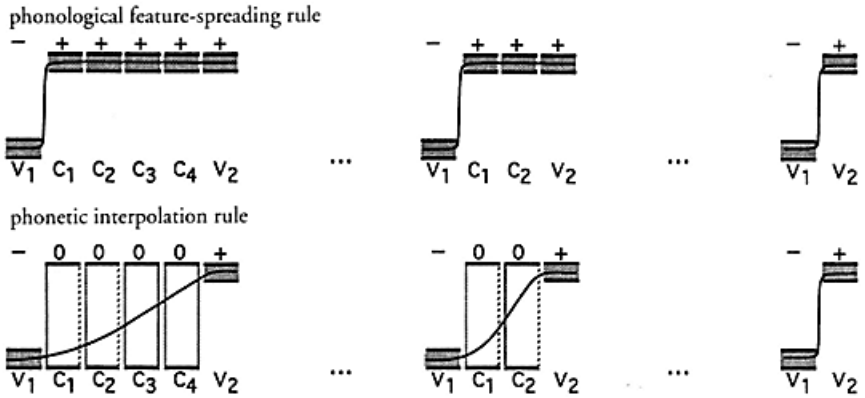


Figura 1.8: Il modello Look-Ahead secondo l'interpretazione di Keating.

A livello fonetico tali caratteristiche sono associate ad una finestra con una propria durata e ampiezza. L'ampiezza della finestra rappresenta tutti i possibili valori che un target articolatorio può assumere. Più è grande la finestra, più ad un segmento fonetico è associata la caratteristica di neutralità (figura 1.7). Il valore di questa ampiezza varia in base alla lingua e può

essere ottenuto osservando le variabilità contestuali. Man mano che l'ampiezza della finestra diminuisce si riesce a descrivere quei segmenti neutri che manifestano resistenza alla coarticolazione. Nelle regioni di transizione tra una finestra e l'altra si usano funzioni di interpolazione. Esse devono avere un impatto articolatorio minimo, in accordo con il principio di economia di Lindblom.

Il modello di Keating offre un ampiamento del modello look-ahead di Daniloff e Hammenberg. Se, infatti, i segmenti neutri non acquisiscono la caratteristica dell'arrotondamento, il movimento anticipatorio inizia prima possibile, immediatamente dopo la vocale non arrotondata, e si dirige gradualmente verso la vocale arrotondata con una durata che è più breve man mano che diminuisce il numero di segmenti neutrali (figura 1.8 in basso).

1.4.1.7 La coproduzione gestuale

Una forte critica al modello di Keating viene dal fatto che in esso le unità di rappresentazione fonologiche sono astratte e statiche e perciò necessitano di un processo che le trasformi nei diversi movimenti articolatori. Questo approccio non riesce a superare la diversità tra aspetti cognitivi e fisici perché sono ancora visti come appartenenti a domini diversi piuttosto che come due livelli di descrizione di uno stesso aspetto.

In una visione alternativa le unità fonologiche sono delle azioni pianificate, con una specifica dimensione temporale, che vengono definite "gesti fonetici". L'attivazione del gesto fonetico cresce fino a raggiungere il massimo valore durante il segmento cui si riferisce per poi decrescere fino ad annullarsi. La combinazione temporale dei gesti determina una loro sovrapposizione reciproca. Si distinguono in questa maniera due tipi di coarticolazione :

- **anticipatoria:** si estende da destra verso sinistra rispetto al segmento fonetico in questione; è dovuta al movimento degli articolatori verso la posizione desiderata durante la produzione di un suono precedente a quello in esame.

- **perseverativa:** si estende da sinistra verso destra ed è determinata dal rilascio del movimento del corrente suono durante la produzione del suono successivo.

In alcuni casi la sovrapposizione di sequenza di due gesti successivi si risolve in modo additivo, come mostrano gli studi svolti da Munhall e Löfqvist [76] (figura 1.9). Vi sono altri casi, poi, in cui i gesti successivi sono contrastanti. Bell-Berti e Harris [8] hanno proposto che tali conflitti siano risolti ritardando la pianificazione dell'inizio di attivazione dei gesti di modo che gli obbiettivi articolatori possano essere raggiunti. Saltzam e Munhall [95] hanno ripreso il concetto di resistenza coarticolatoria, affermando che gesti

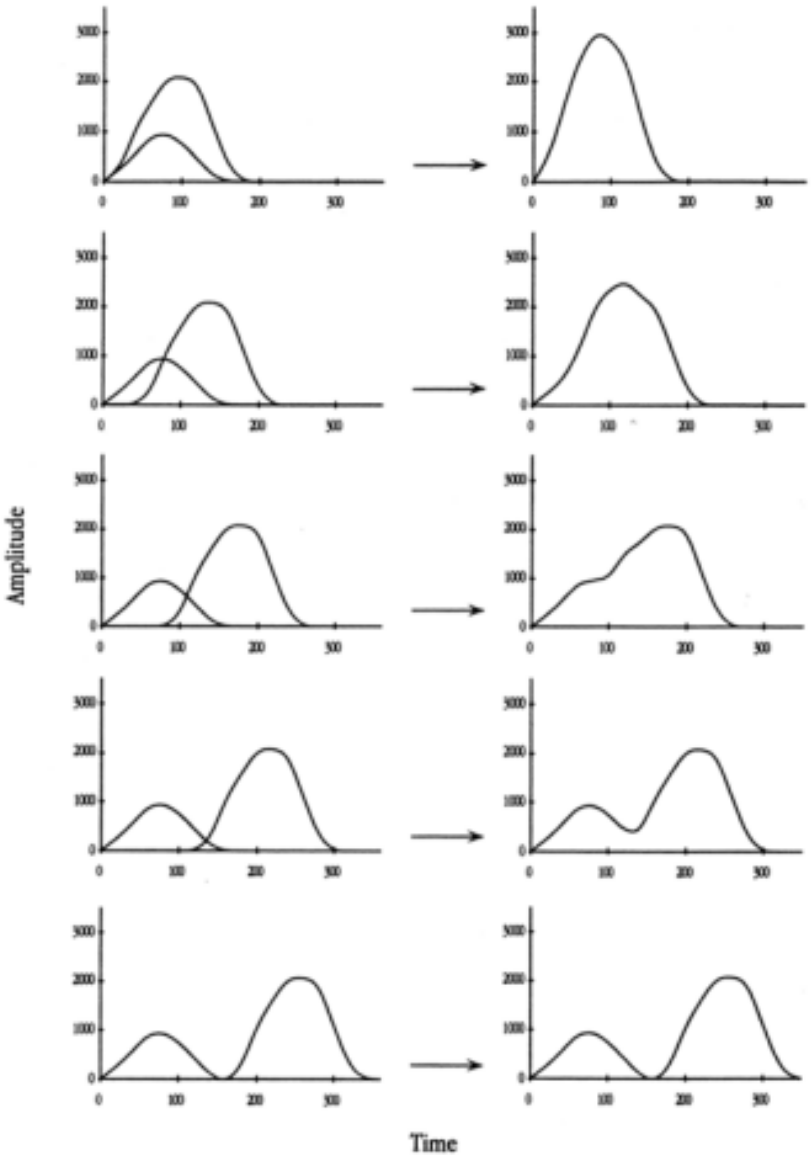


Figura 1.9: Sovrapposizone dei gesti articolatori secondo quanto emerso dagli studi di Munhall e Löfqvist

più "forti" bloccano l'influenza di gesti "deboli", mentre di fronte a due gesti con uguale forza il risultante sarà una media tra i due.

Secondo questo modello le differenze tra varie lingue sono dovute a diversità della caratterizzazione dei gesti fonetici.

Il modello time-locked Un gesto fonetico, secondo il modello della coproduzione, ha una caratteristica che rimane invariata e di conseguenza la sua estensione temporale deve essere costante. In base a quest'osservazione e alle conferme ottenute da dati sull'arrotondamento labiale e sull'abbassamento velare, Bell-Berti ed Harris [8] proposero il modello coarticolatorio "time-locked" in cui il movimento di un articolatore, indipendentemente dalla lunghezza del precedente fono, inizia in un tempo fisso (vedi figura 1.10).

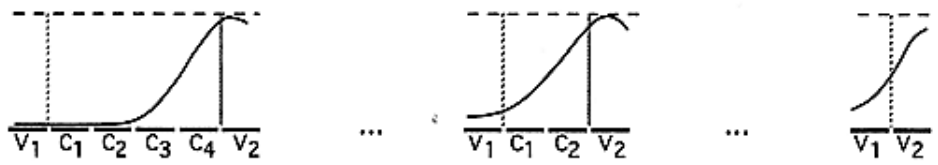


Figura 1.10: Il modello Time-Locked anche chiamato "Coproduction Model".

1.4.1.8 Il modello ibrido

Al-Bamerni e Bladon [3], in uno studio sulla coarticolazione velare, osservarono che alcuni soggetti presentavano un gesto di apertura a due fasi, la prima a bassa velocità e la seconda a forte velocità. Perkell [86] notò la stessa particolarità in studi sull'arrotondamento anticipatorio in un contesto /i C u/. L'inizio dell'arrotondamento coincideva con la fine del gesto della /i/ come predetto del modello Look-Ahead, mentre l'inizio della seconda fase, identificabile tramite il punto di estremo dell'accelerazione, è legato in maniera costante alla /u/. Gli autori allora proposero un modello, il "modello ibrido", che fosse un compromesso tra i modelli Look-Ahead e Time-Locked (figura 1.11) .

Matthies e lo stesso Perkell in uno studio successivo [87] videro che in alcuni soggetti non vi era una relazione fissa tra la seconda fase e la vocale arrotondata. Conclusero quindi che nessuno dei tre modelli anticipatori (look-ahead, time-locked, ibrido) poteva spiegare le relazioni temporali osservate nei dati.

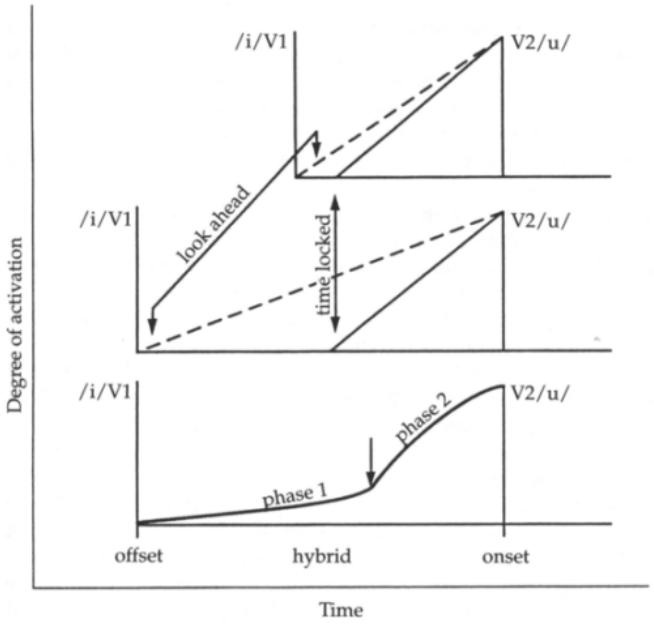


Figura 1.11: Rappresentazione schematica del modello ibrido proposto da Perkell.

1.4.1.9 L'espansione del movimento

Il modello anticipatorio proposto da Abry e Lallouache [1] basato sull'analisi di dati riguardanti il labbro superiore. In base ai risultati ottenuti stabilirono che il gesto vocalico di protrusione labiale può espandersi per un tempo proporzionale alla durata della consonante precedente. L'influenza però non può essere ridotta entro un intervallo minimo che è stato valutato in circa 140ms. L'ammontare dell'espansione varia da soggetto a soggetto, mentre la soglia minima rimane pressoché costante (figura 1.12).

1.4.2 Coarticolazione labiale

Faremo ora alcune precisazioni per quanto riguarda la coarticolazione labiale.

Possiamo distinguere le vocali in arrotondate (e.g. /o u/) e non arrotondate (e.g. /a e i/). Le implicazioni della caratteristica di arrotondamento in termini articolatori sono state oggetto di numerose studi. I risultati stabiliscono che il particolare distintivo dell'arrotondamento è dato dall'apertura orizzontale delle labbra. In particolare per l'italiano, poi, il movimento di arrotondamento si verifica contemporaneamente alla protrusione delle labbra, cosa che non avviene in altre lingue.

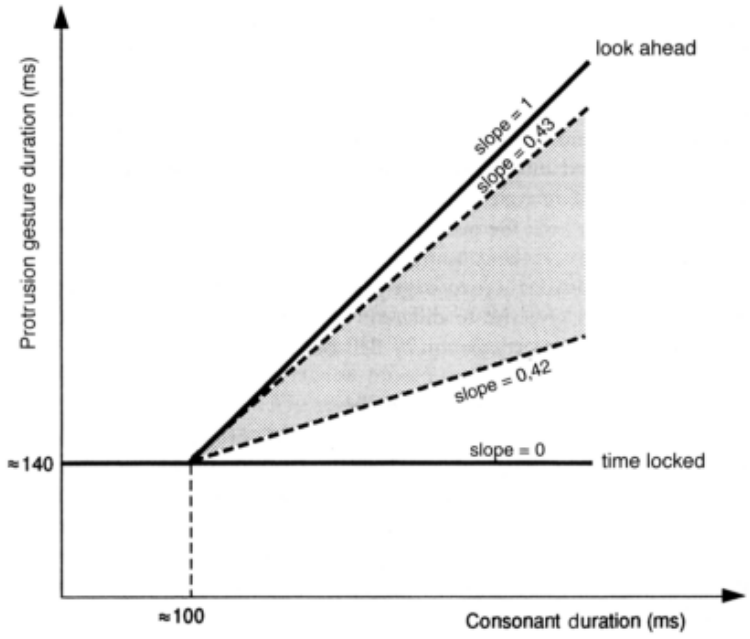


Figura 1.12: Il modello ad espansione del movimento

Per quanto riguarda le consonanti possiamo classificarle in **labiali**, dove le labbra sono l'articolatore primario (e.g /p b m f v/), **labializzate**, se le labbra agiscono come articolatore secondario (e.g. /s z/ per l'italiano), e **neutre**, quando la loro caratteristica labiale rimane non specificata (e.g. /n l/). Per quanto riguarda quest'ultime, è stato dimostrato che in realtà possono avere una loro conformazione labiale: tale aspetto è dipendente dalla lingua considerata e spesso anche dai singoli soggetti.

Molti studi sulla coarticolazione labiale sono effettuati tramite analisi dell'attività muscolare attraverso l'elettromiografia. I principali muscoli coinvolti nei movimenti labiali sono l'*orbicularis oris,* responsabile della protrusione e dell'arrotondamento, il *buccinator* ed il *risorius,* che determinano l'estensione delle labbra. Una serie di altri muscoli agiscono in sinergia con quelli appena citati: il *mentalis* ed il *depressor labii inferioris* che insieme all'orbicularis oris fanno abbassare e protrudere il labbro inferiore; il *levator labii* ed il *depressor anguli oris* possono agire con il *buccinator* per aprire ed estendere le labbra (figura 1.13). Gli studi elettromiografici nel loro complesso indicano che i movimenti labiali sono il risultato di una interazione tra i muscoli labiali i cui dettagli non sono ancora del tutto chiari.

È da notarsi infine che se da una parte è stata riservata grande attenzione alle analisi sulla coarticolazione anticipatoria, poco si sa per quanto riguarda quella perseverativa. Alcuni studi effettuati confermano la teoria dell'estensione delle caratteristiche e mostrano un movimento perseverativo

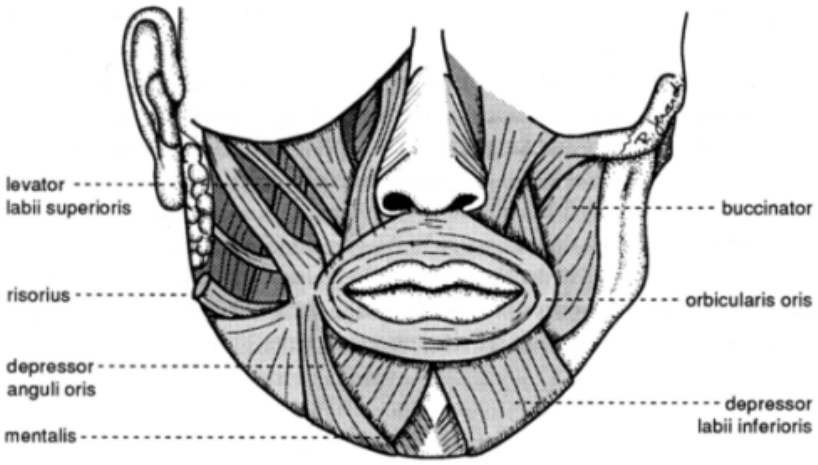


Figura 1.13: Struttura muscolare della zona orale

più lento e più lungo rispetto a quello anticipatorio. Altri risultati, tuttavia, sono in disaccordo e ne evidenziano una limitata estensione temporale.

1.4.3 Algoritmi e Modelli numerici

Abbiamo visto la diversità di approccio dei due sistemi ad estensione delle caratteristiche e di coproduzione gestuale.

Il primo vede le variazioni in modo statico ed in termini pratici si traduce nell'elaborazione di algoritmi per l'assegnazione di target articolatori che cambiano secondo il contesto. Possono essere utilizzati anche metodi come le ANN o le analisi spettrali ed altri già citati nel paragrafo 1.2.6. Pearce [84], ad esempio, ha elaborato una serie di regole che a partire dalla rappresentazione fonemica, calcolava la corrispettiva posizione delle labbra. Pelachaud [85] ha implementato una variante del modello look-ahead attraverso un algoritmo a tre passi. Nel primo vengono applicate le regole di coarticolazione anticipatoria e perseverativa a tutti gli obiettivi articolatori influenzati dal contesto. Nel secondo si considerano gli effetti della contrazione e del rilassamento muscolare nella produzione di posizioni consecutive. Nel terzo si prendono in esame le conseguenze legate alla successione di caratteristiche geometricamente contrastanti.

Il modello gestuale comporta l'elaborazione di un sistema dinamico guidato da una serie di funzioni. Mermelstein [73] ha proposto un modello che utilizza funzioni esponenziali per rappresentare il movimento di apertura e di chiusura e una funzione di smussamento che realizzi le transizioni intervocaliche. Coker [25], invece ha utilizzato una combinazione lineare dei modi di un'equazione differenziale per descrivere i movimenti articolatori. In par-

ticolare, in relazione agli studi elettromiografici viene solitamente usato un modello che si basa su un'equazione differenziale del secondo ordine del tipo:

$$m\ddot{x} + b\dot{x} + kx \tag{1.1}$$

dove m, b, k stanno rispettivamente per massa, viscosità ed elasticità. I muscoli delle labbra hanno una massa ridotta ed hanno forti caratteristiche elastiche attenuate nel tempo a causa della struttura della pelle cui sono attaccati. Questo metodo lavora bene con i modelli facciali basati su modelli fisiologici. Lucero e Smith et al. [64, 99] utilizzano la seguente serie di Fourier per controllare le labbra. Sia infatti $p(\theta)$ un generico parametro

$$p(\theta) = \sum_{n=1}^{6} \Re(C_n)\cos(n\theta) + \Im(C_n)\sin(n\theta) \tag{1.2}$$

in cui $\Re(C_n)$ e $\Im(C_n)$ sono assegnati e $\theta \in [0, 1]$ dipende dalla frequenza di clock ω_p e dalla funzione variabile $\gamma(t)$ secondo la relazione

$$\frac{d\theta}{dt} = \omega_p + \gamma(t). \tag{1.3}$$

Öhman [79] definisce un modello che descrive le transizioni VCV come un gesto vocalico in cui si inserisce quello consonantico nel modo seguente

$$p(x,t) = v(x,t) + k_c(t)w_c(x)\,[c(x) - v(x,t)] \tag{1.4}$$

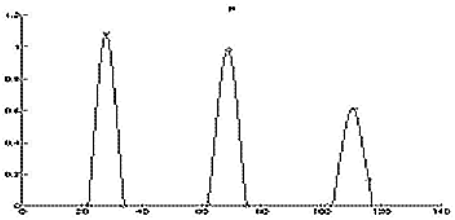


Figura 1.14: Esempio di funzione di emergenza k_c per la consonante /p/ nei contesti /'a p a/, /'i p i/ e /'u p u/ (tratto da [93]).

dove x identifica il parametro, t il tempo, $p(x,t)$ il valore del parametro, $v(x,t)$ il valore del parametro come puro gesto vocalico cioè in assenza dell'influenza consonantica, $c(x)$ il target consonantico, $k_c(t)$ l"'emergenza" della consonante e $w_c(x)$ un fattore di coarticolazione. Per capire il funzionamento dell'equazione (1.4), riscriviamola nella seguente forma

$$p(x,t) = v(x,t)(1 - k_c(t)w_c(x)) + k_c(t)w_c(x)c(x). \tag{1.5}$$

Possiamo allora comprendere che il modello di Öhman rappresenta la transizione dall'andamento vovalico descritto da $v(x,t)$ verso il target consonantico $c(x)$ in base alla funzione peso complessiva descritta da $k_c(t)w_c(x)$.

In figura 1.14 vediamo un esempio di funzioni di emergenza per la consonante /p/.

Il modello più interessante è sicuramente quello adottato da Cohen e Massaro [27]. Essi si basarono sul modello gestuale di Löfqvist in cui ad ogni singolo gesto articolatorio è associata una funzione di dominanza con le stesse caratteristiche dei gesti fonetici descritti in 1.4.1.7. Una funzione di dominanza è caratterizzata da una propria ampiezza, durata, e grado di attivazione. L'ampiezza determina l'importanza relativa del gesto per il segmento; la durata stabilisce l'estensione del movimento ed influisce sul grado di sovrapposizione che ne conseguirà; il grado di attivazione caratterizza il fatto che il gesto si avvia in modo più o meno graduale.

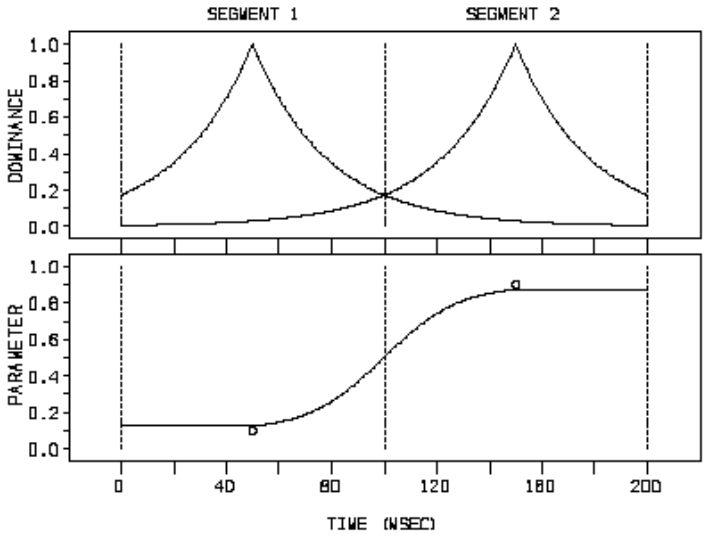


Figura 1.15: Dominanza di due segmenti nel tempo (in alto) e andamento risultante del parametro (in basso) secondo il modello di Cohen e Massaro (tratto da [27]).

Una versione modificata di questo modello di coarticolazione [28] è utilizzata dall'avatar realizzato in questa tesi e verrà approfondita nella sezione 3.4.1

Capitolo 2

La faccia parlante LUCIA

LUCIA è un sistema di animazione facciale sviluppato a Padova dallo Istituto di Scienze e Tecnologie della Cognizione del CNR. Si basa sullo standard MPEG-4 presentato nel capitolo precedente e in particolare sulla codifica dei parametri di animazione facciale (FAP - Facial Animation Parameters). LUCIA parla per mezzo della versione italiana di FESTIVAL TTS ed ora, grazie al lavoro svolto con questa tesi, anche con i più moderni MARYTTS e Loquendo (che presentano una qualità sonora più elevata). La sintesi del parlato visuale di LUCIA si basa su dati reali collezionati riprendendo un attore per mezzo di un sistema di motion capture denominato ELITE. LUCIA può imitare un essere umano riproducendo i movimenti dei markers passivi posizionati sulla sua faccia o può essere pilotata da un testo di input. Il parlato emotivo è codificato per mezzo di APML, una versione di XML pensata per rappresentare le emozioni. Questo testo costituisce l'input per il modulo AVEngine (Motore di sintesi audio/video) che crea i flussi audio/video sincronizzati con cui viene animato l'avatar. In figura 2.1 è mostrato lo schema logico del sistema.

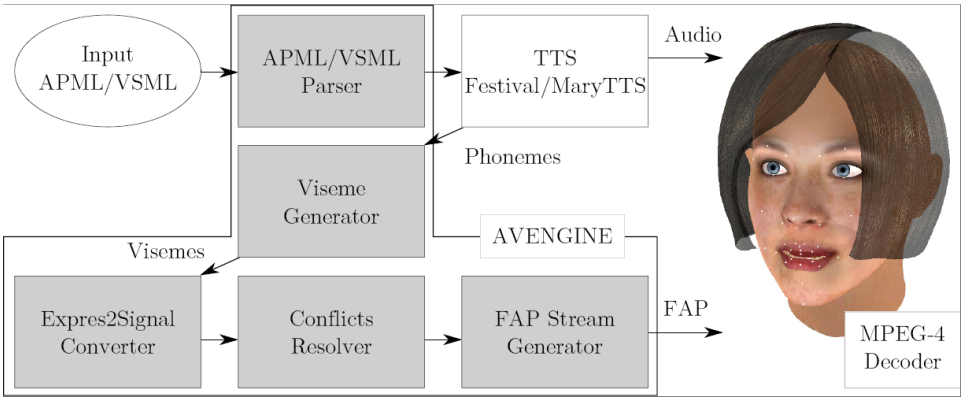


Figura 2.1: Lo schema logico del sistema per la sintesi audio-visiva della faccia parlante LUCIA

2.1 La creazione dei flussi audio/video

La sintesi audio/visiva del sistema LUCIA è basata su dati reali collezionati riprendendo attori umani [69, 68] per mezzo di ELITE, un analizzatore automatico di movimento per l'acquisizione dati di cinematica 3D [40]. Esso fornisce la ricostruzione delle coordinate 3D di alcuni punti dello spazio, per mezzo di una procedura di triangolazione stereo, utilizzando dei marcatori riflettenti passivi (che nel nostro caso vengono applicati sul volto della persona ripresa). Tutti i movimenti dei marcatori sono registrati in tempo reale (posizione, velocità e accelerazione) simultaneamente con il parlato prodotto. Due configurazioni differenti sono state utilizzate per l'acquisizione dei movimenti articolatori: la prima, specificamente pensata per l'analisi dei movimenti labiali, considera solo 8 marcatori, mentre la seconda, adatta per l'analisi della espressività e del parlato emotivo, utilizza il set completo composto da 28 unità. Una descrizione più dettagliata del dispositivo e dei parametri articolatori è presente nell'appendice 4.1.

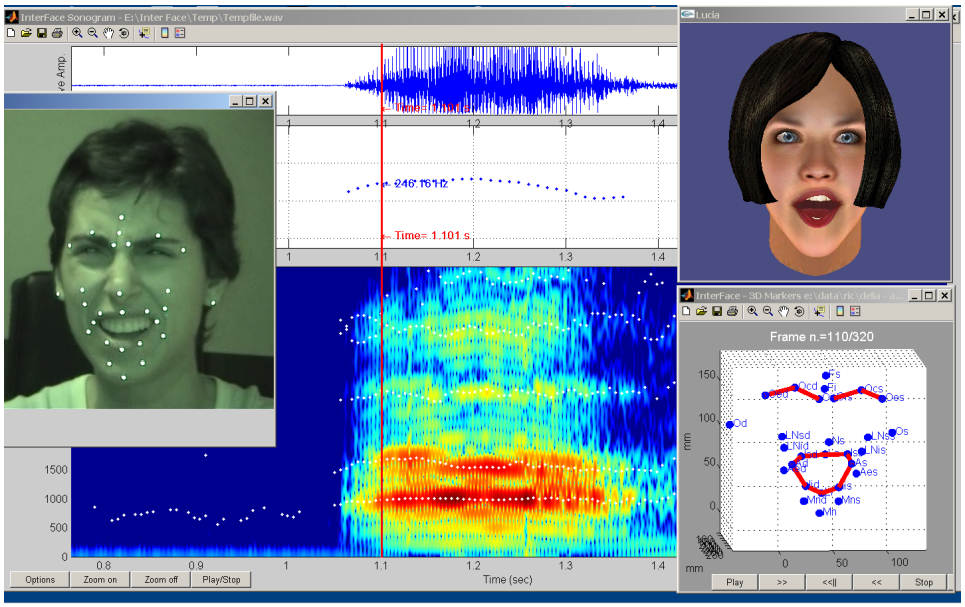


Figura 2.2: Il software INTERFACE permette la sincronizzazione del frame 3D ricostruito (n. 110/320 nell'esempio). In alto a destra LUCIA Talking Head riproduce la posa dell'attore (A sinistra con i markers sul volto)

2.1.1 Il modello del parlato visivo

Per la generazione di facce parlanti naturali, espressive e realistiche è necessario riprodurre fedelmente la variabilità contestuale dovuta alla reciproca

influenza dei movimenti articolatori durante la produzione del segnale verbale ("coarticolazione"); abbiamo descritto il fenomeno nel capitolo precedente.

Il database realizzato con ELITE consiste nelle coppie sincronizzate dei file audio e dei file dei movimenti articolatori. Grazie ad esso è stato possibile calcolare i coefficienti del modello di coarticolazione adottato: una versione modificata di quello proposto da Coehn e Massaro [28]. Il modello, l'elaborazione dei dati e il calcolo dei coefficienti vengono mostrati in dettaglio nella sezione 3.4 del prossimo capitolo.

2.1.2 AVEngine e la sintesi emotiva

Il modulo software AVEngine (al centro in figura 2.1) trasforma il testo in input in flussi audio/video utilizzando.

La sintesi audio è ottenuta per mezzo di un sistema text-to-speech esterno. Originariamente era possibile utilizzare solo la versione italiana di FESTIVAL [29] e ora, come risultato di questo lavoro, anche la voce italiane di MARYTTS [97] e Loquendo [106]. La qualità sonora di quest'ultimo in particolare è di gran lunga migliore rispetto agli altri due.

Il sistema TTS fornisce l'audio e il corrispondente stream di fonemi (con la loro durata temporale). AVEngine utilizza queste informazioni e il modello del parlato per trasformare la sequenza dei fonemi in un flusso di FAPs che riproduce visivamente i movimenti labiali che un parlatore italiano eseguirebbe per pronunciare tale sequenza.

Oltre alla sintesi del movimento labiale è possibile effetture la sintesi audio visiva degli stati emotivi eventualmente presenti nel testo in input. Il testo da sintetizzare infatti è scritto con il linguaggio APML (Affective Presentation Markup Language) [22] che permette di distinguere le parti verbali del dialogo da pronunciare con l'espressione emotiva desiderata. Anche questo obiettivo è raggiunto basandosi su basi di dati audio-visive i cui contenuti sono stati usati per allenare intonazioni della voce e i modelli presenti in FESTIVAL [104, 102, 34, 77]. Anche il nuovo MARYTTS supporta le voci emotive e l'input in linguaggio APLM. E' questo il motivo per cui è stato scelto come motore TTS di default per il sistema.

Per valutare in modo rapido gli effetti della sintesi emotiva con vari livelli di intensità è stato sviluppato EmotionPlayer, fortemente ispirato dall'Emotion Disk di Z. Ruttkay [94]. E' mostrato in figura 2.3.

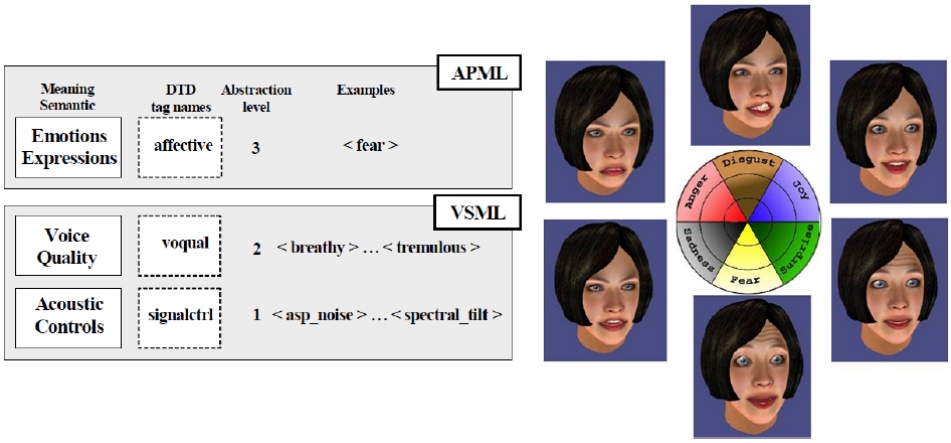


Figura 2.3: Il linguaggio APML per la sintesi emotiva e lo strumento Emotion Player

2.2 Il motore di animazione facciale

Un motore di animazione facciale (*Facial Animation Engine - F.A.E.*) è in grado di animare il wireframe di un volto sulla base della sua descrizione geometrica e di un'informazione semantica ad esso associata. È basato su procedure software che compongono una interfaccia ad alto livello per l'implementazione e l'animazione di modelli 3D di facce sintetiche. LUCIA è un motore di animazione facciale che implementa un decoder compatibile con *MPEG-4 / Simple Facial Animation Object Profile* che importa da un file vrml la struttura facciale del modello proprietario e la anima in base al FAPs stream in input.

Il rispetto delle specifiche dello standard MPEG-4 e di tempo reale hanno influenzato la scelta del modello e delle tecniche di animazione. Abbiamo già accennato, parlando dei tipi di rappresentazione dei modelli 3D nel paragrafo 1.1.1, che per i modelli facciali la rappresentazione più adatta è quella mediante superfici parametriche (NURBS) o superfici poligonali. LUCIA usa un modello poligonale per tre motivi principali:

- I dispositivi mobili ancora non forniscono la potenza di calcolo sufficiente per gestire un'animazione con un numero di superfici tale da descrivere accuratamente un volto umano.

- Costruire, modificare e trasportare un modello in NURBS è molto più complesso rispetto ad un modello poligonale; quest'ultimo può essere ottenuto da uno scanner 3D, può essere modificato con i più comuni applicativi di grafica tridimensionale, può essere rappresentato con

formati diffusi e di semplice utilizzo (essenzialmente a lista di vertici)

- utilizzare un modello in NURBS avrebbe creato incompatibilità con *MPEG-4 / Predictable Facial Animation Object Profile*, in cui si prevede la trasmissione di un modello facciale da remoto sotto forma di mesh poligonale; la scelta delle superfici parametriche avrebbe precluso l'estensione del nostro profilo a quelli più complessi dello standard MPEG-4 (realizzato con il presente lavoro, si veda il paragrafo 2.3.4)

LUCIA usa un approccio pseudo-muscolare per ottenere animazioni in tempo reale. L'azione muscolare è simulata mediante degli opportuni operatori che, comandati dai FAP in input, modificano la struttura geometrica del modello facciale 3D, generando, al variare dei parametri nel tempo, l'animazione desiderata.

2.2.1 La rappresentazione VRML

LUCIA importa da un file di testo di tipo *vrml* la struttura del modello facciale. La faccia è stata concepita come un insieme di oggetti indipendenti l'uno dall'altro.

Il file VRML è un file di testo in cui sono memorizzate, per ogni oggetto caricato, le coordinate dei vertici, la mappatura di questi sulla textures e gli indici con cui viene costruita la struttura del wireframe.

Nella fattispecie il motore di animazione facciale prevede la possibilità di caricare 6 oggetti distinti: occhio destro, occhio sinistro, pelle, capelli, denti, lingua.

Il FAE dopo la lettura del file di input costruisce nello spazio 3D gli oggetto in questione. L'appendice 4.5 mostra un esempio di file sorgente VRML e la rappresentazione grafica degli oggetti che compongono la testa nella versione wireframe e in quella definitiva con texture.

Nelle figure 2.4, 2.5 e 2.6 si può osservare l'avatar LUCIA nella visualizzazione wireframe e con texture.

2.2.2 L'animazione

L'input del sistema è costituito dalla coppia di file audio e FAP prodotti dal modulo AVEngine descritto in precedenza.

Per quanto riguarda l'audio si è adottata una semplificazione rispetto allo standard: la componente sonora dell'animazione realizzata dal TTS esterno, è contenuta in un file audio di tipo *.wav*. In MPEG-4 (Appendice 4.2) la

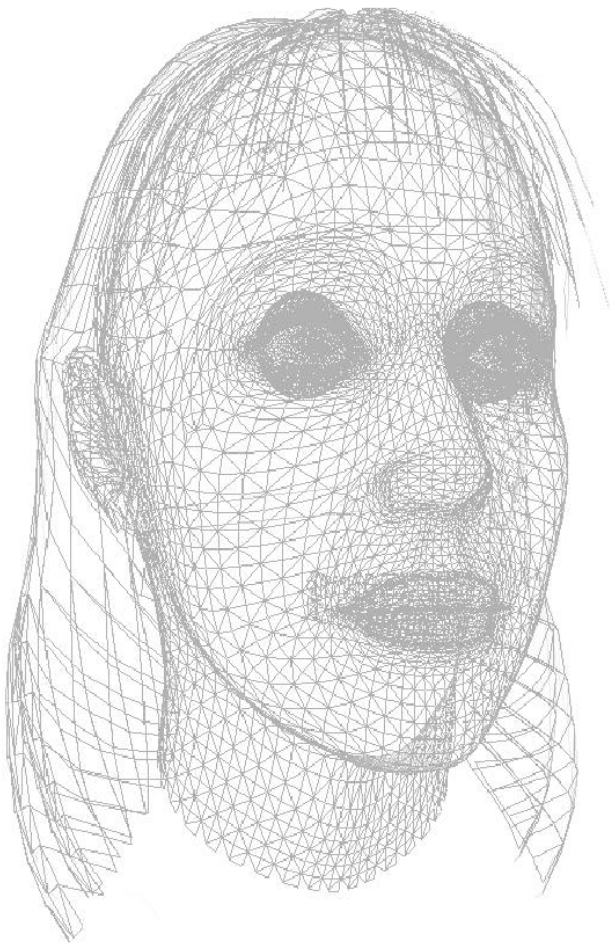


Figura 2.4: Il motore di animazione facciale LUCIA (vista in wireframe)

Figura 2.5: Il motore di animazione facciale LUCIA (vista frontale)

Figura 2.6: Il motore di animazione facciale LUCIA (vista laterale)

componente sonora è un altro *media-object* presente nello *scene-graph* tipicamente un audio registrato o sintetico che viene compresso e codificata in maniera opportuna dall'encoder.

Lo stream di FAPs (vedi Appendice 4.3) definisce il movimento dei *feature points* della faccia neutrale. I *feature points* sono solo dei singoli vertici della mesh poligonale che costituisce il volto di un modello 3D. Non è sufficiente muovere esclusivamente questi punti per avere un'animazione facciale nel modello; ogni punto chiave è collegato a tutti i vertici che si trovano all'interno di una certa area di influenza; il risultato finale dell'applicazione di un FAP sul modello consiste nel movimento di un gruppo di vertici della mesh poligonale, secondo un'area e una intensità da definire.

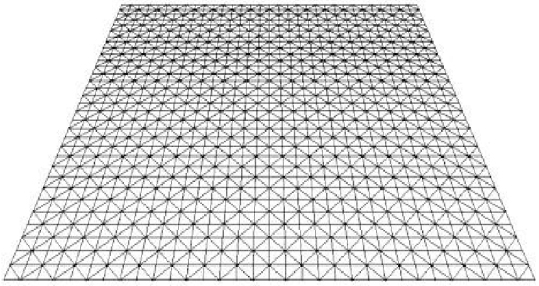


Figura 2.7: Un reticolo poligonale

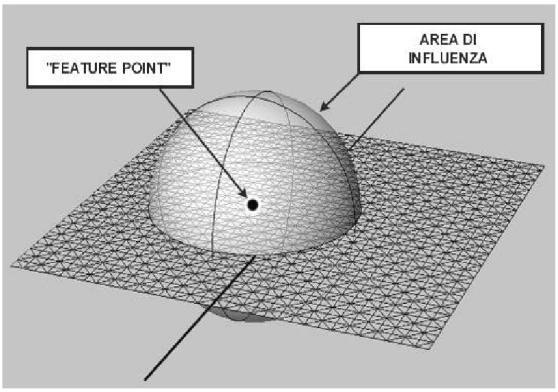


Figura 2.8: L'area di influenza del *feature point* entro cui cadono i vertici.

2.2.2.1 Area di influenza di un *feature point*

Per semplificare la trattazione non consideriamo direttamente il modello facciale 3D del nostro decoder, ma un reticolo poligonale piano. L'area di influenza è la regione del reticolo in cui i vertici sono influenzati dal movimento del *feature point* (imposto dal rispettivo FAP). Quest'area è costituita da un'ellissoide, centrato sul punto chiave, in cui i tre assi principali sono orientati secondo gli assi cartesiani del sistema di coordinate attuale; variando le dimensioni dell'ellissoide si può definire in maniera adeguata l'area di influenza per meglio circoscrivere i vertici del reticolo che si vogliono muovere (figura 2.8).

La funzione che effettua lo spostamento dei vertici intorno al *feature point* agisce, quindi, solo sui punti che si trovano all'interno dell'area di influenza e non su quelli esterni.

Per effettuare il test di appartenza bisogna calcolare la distanza del punto generico della mesh poligonale dal punto chiave (che vogliamo che generi la deformazione sotto l'impulso di un particolare FAP). Per rendere più veloce questo calcolo si è pensato di suddividere il volto di LUCIA in due grandi zone corrispondenti alla parte superiore e a quella inferiore (figura 2.9).

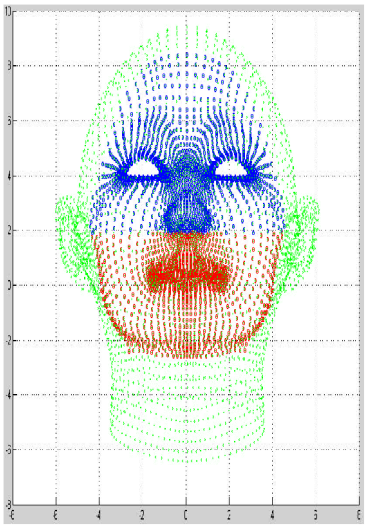


Figura 2.9: Suddivisione del volto di LUCIA in due zone specifiche

In questo modo si prendono in considerazione i soli punti che caratterizzano esclusivamente il volto trascurando quelli che appartengono a zone della testa che non verranno mai interessate dal movimento dei muscoli facciali.

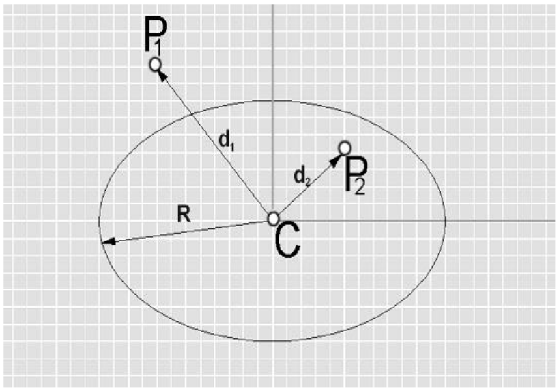


Figura 2.10: P_1 e P_2 sono due punti generici del reticolo poligonale: il punto P_1 è all'esterno dell'ellissoide e non fa parte dell'area di influenza mentre il punto P_2 lo è e sarà sottoposto alla funzione di spostamento che ne stabilisce l'intensità e la direzione.

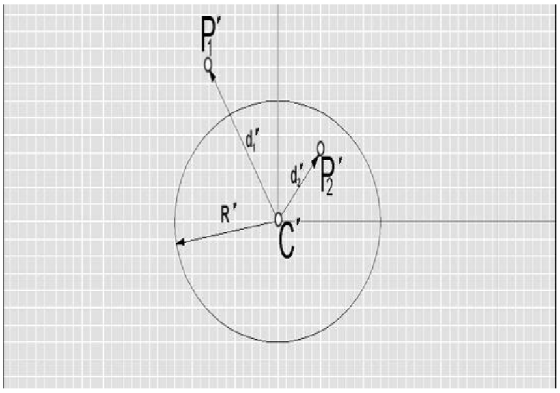


Figura 2.11: Il test di appartenenza dei punti all'area di influenza dopo il cambio di sistema di riferimento.

Siano P_1 $(x_1,\ y_1,\ z_1)$ e P_2 $(x_2,\ y_2,\ z_2)$ due punti generici del reticolo poligonale su cui effettuare il test di appartenenza (figura 2.10). Per rendere il test più agevole si effettua un cambio di sistema di riferimento per ricondurre il generico ellissoide ad una sfera (Operazione di *Scale* lungo i tre assi cartesiani). Si consideri un nuovo sistema di coordinate x', y', z' ove $x' = S_x * x$, $y' = S_y * y$, $z' = S_z * z$ (con S_x, S_y ed S_z dipendenti dagli assi dell'ellissoide). Abbiamo quindi: $P_1'(x_1', y_1', z_1'), P_2'(x_2', y_2', z_2'), C'(x_0', y_0', z_0')$.

La distanza d'_1 del punto P'_1 da C' è data da:

$$d'_1 = \sqrt{(x'_1 - x'_0)^2 + (y'_1 - y'_0)^2 + (z'_1 - z'_0)^2}$$

Se la distanza d'_1 risulta minore del raggio R' il punto viene sottoposto all'azione della funzione di spostamento.

2.2.2.2 La funzione di spostamento dei vertici

Il FAP è pensato come un'azione muscolare applicata nel punto chiave; tutta la zona circostante, quindi, deve comportarsi come la pelle sottoposta ad una trazione esercitata da un muscolo mimico, con attaccatura proprio nel *feature point*. È necessaria una funzione che, dato un certo numero di punti (i vertici che cadono nell'area di influenza) nell'intorno di un *feature point* e, dato un FAP che implica lo spostamento di quest'ultimo, sposti la posizione dei punti dell'area di influenza in maniera armoniosa (per simulare l'azione muscolare sulla pelle) e conforme a quanto indicato nel parametro.

Consideriamo per ora uno spostamento monodimensionale, in quanto un FAP in generale determina la variazione solo di una singola coordinata di un punto chiave secondo quanto stabilito dalle specifiche di MPEG-4 (si veda l'Appendice 4.4). La variazione, dovuta ad un FAP (indichiamo FAP_x un FAP che indica una traslazione lungo l'asse x), della coordinata x_j di un punto $P_j(x_j, y_j, z_j)$ della mesh poligonale in esame, in generale si può esprimere così:

$$\Delta x_j = W_j * FAP_x \text{ con } W_j = 0 \text{ se } P_j \notin I$$

Ove Δx_j è la variazione della coordinata x_j del punto P_j e I è l'insieme dei punti appartenenti all'area di influenza. La funzione di spostamento deve individuare il valore del fattore W_j, che trasforma il parametro FAP_x in spostamenti Δx_j. Nel vertice che corrisponde al *feature point* il peso W_j è pari ad 1 mentre negli altri punti dell'area di influenza esso deve decrescere in relazione alla distanza fino ad annullarsi per i punti che appartengono alla frontiera dell'area di influenza (questi, infatti, non si devono muovere). La trattazione per gli incrementi lungo le coordinate y e z è analoga.

La funzione che è risultata più indicata a simulare un'azione simile a quella dei muscoli mimici che effettuano una trazione sulla cute, è il "coseno rialzato" (figura 2.12).

La legge che descrive il peso W_j è la seguente:

$$\begin{cases} W_j = 0.5 * (1 + \cos(d''_j)) & \text{, per } 0 \leq d''_j \leq \pi \\ W_j = 0 & \text{, altrove} \end{cases}$$

dove d''_j è la distanza del punto P_j dal *feature point* normalizzata nell'intervallo $0 - \pi$.

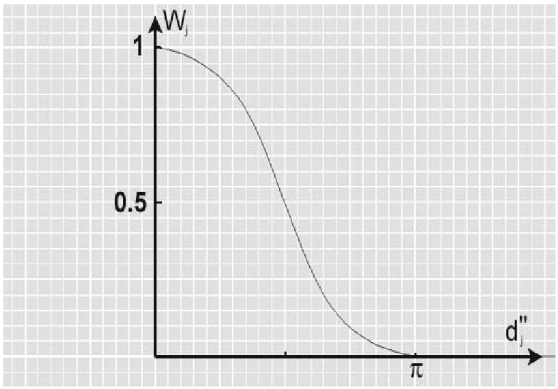


Figura 2.12: Funzione "coseno rialzato"

2.2.2.3 Applicazione dei FAPs

In questo paragrafo descriviamo le procedure relative ai FAP che riguardano la pelle del volto e non i componenti anatomici interni; questi ultimi infatti sono strutture rigide che subiscono solo traslazioni e rotazioni di valore proporzionale al parametro considerato.

Ora dobbiamo estendere quanto visto nel caso del reticolo poligonale semplice del paragrafo precedente; la geometria della struttura poligonale di un tratto del volto 3D è più complessa e sicuramente non planare, ma, in ogni caso, i concetti applicati sono gli stessi.

In ogni frame i FAPs indicano (per un solo asse), uno spostamento del *feature point* a cui afferiscono. Lo spostamento dei punti secondo gli altri due assi cartesiani è lasciato al libero arbitrio del progettista del decoder, che può decidere sia che rimanga completamente nullo sia che avvenga secondo precise leggi che rendano più realistiche le deformazioni del volto sintetico. Ad esempio i FAP 19 e 20 (close_t_l_eyelid e close_t_r_eyelid) che si occupano dell'abbassamento delle palpebre superiori, descrivono solo un movimento verticale (lungo l'asse y) di queste; è compito del progettista implementare il movimento dei punti nelle altre due coordinate (la coordinata z in particolare) in maniera che questi si muovano su di una superficie sferica, seguendo idealmente il profilo del bulbo oculare.
Nel FAE LUCIA ad ogni FAP corrisponde una particolare procedura che incrementa le coordinate dei punti nell'intorno di un *feature point*; l'incremento può riguardare la sola coordinata indicata dal FAP o anche tutte e tre, soprattutto in zone particolarmente significative come gli occhi e la bocca.

2.2.2.4 La calibrazione dell'area di influenza

Per semplificare questo compito si è usato il programma *Face Player* (figura 2.13). Questo software realizzato in MATLAB permette, rispettando le specifiche fornite dallo standard MPEG-4, di creare un fap stream con *frame rate* e durata temporale impostabili dall'utente (par. 4.3).

Il programma permette di settare gli andamenti temporali corrispondenti ad un numero di *feature point* illimitato visualizzandoli con curve di colori diversi sullo stesso grafico (con in ascissa il tempo e in ordinata dei valori assoluti).

In base al FAP che viene selezionato le ampiezze della relativa curva vengono riscalate per la FAP Unit associata secondo quanto indicato nella tabella 4.1 e visualizzato in figura 4.3.

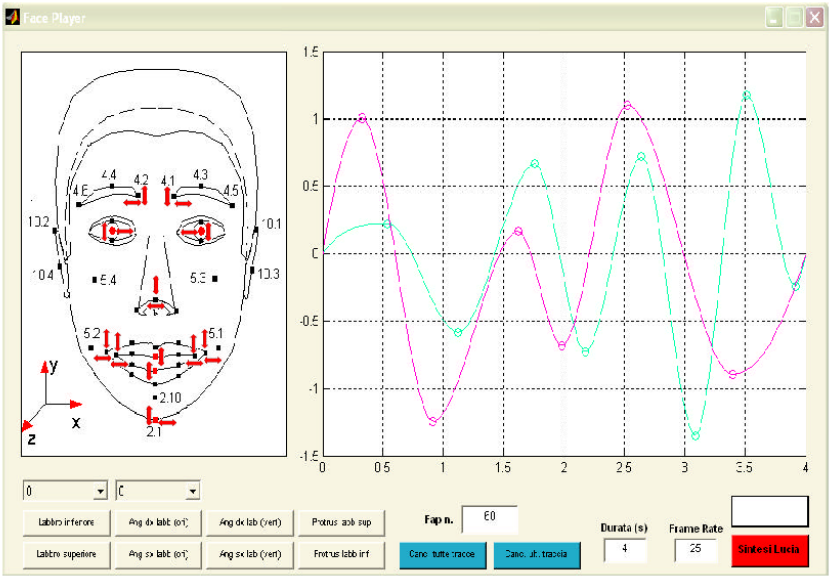


Figura 2.13: L'applicazione Face Player e gli andamenti temporali di alcuni FAPs

Le informazioni così create sono memorizzate in una matrice utilizzata in fase di sintesi per creare il fap stream.

L'utilizzo del Face Player ha permesso di calibrare nel modo opportuno il motore di animazione facciale durante il suo sviluppo per evitare collisioni e artefatti grafici.

2.3 LUCIA-WebGL

LUCIA-WebGL è parte del progetto di ricerca "Wikimemo.it Il portale della lingua e della cultura italiana" finanziato dal M.I.U.R. per promuovere la cultura italiana attraverso la lettura e l'ascolto. L'obiettivo principale del progetto è dare al vasto numero di utenti italiani (di cui ben 120 milioni di nati all'estero che usano l'italiano come seconda lingua) l'opportunità di accedere alle risorse letterarie italiane e fornire a scopo didattico gli esempi di come vengono pronunciate le parole dai nativi italiani. Il nostro compito nel progetto è realizzare una faccia parlante che sia facilmente integrabile in una pagina html e che fornisca i servizi di guida virtuale del portale. Per fare questo ci siamo basati sulla faccia parlante LUCIA (vista nel capitolo precedente) e abbiamo trasformato una applicazione monolitica in una applicazione client-server che si basa sulla nuova tecnologia WebGL.

2.3.1 WebGL: The Web Graphic Library

Nel febbraio 2011 è stata rilasciata la versione 1.0 della tecnologia WebGL [113], che permette di includere la grafica 3D nei *web-browser* senza dover installare plug-in aggiuntivi e potenzialmente pericolosi. WebGL estende la capacità del linguaggio Javascript per generare grafica interattiva 3D in ogni *browser* compatibile. Il WebGL *working group* include Apple, Google, Opera e Mozilla ovvero tutti i produttori dei principali browser ad oggi disponibili (tranne Microsoft). WebGL è basato sulla libreria OpenGL Embedded System 2.0, la libreria grafica per sistemi portatili. Questo significa non disporre di tutte le funzioni presenti nella versione per *desktop*, ma ci si assicura una piú vasta compatibilità da parte dei moderni *smartphone*. Qualsiasi applicazione può essere eseguita in ogni piattaforma che supporti il nuovo standard.

La tipica applicazione webGL è composta da tre parti:

- la sezione shading: è la vera novità introdotta da WebGL che permette l'accelerazione 3D. E' un linguaggio chiamato GLSL ed ha una sintassi simile al C. Queste istruzioni hanno il compito di calcolare il valore del colore di tutti i pixel della parte di schermo dedicata alla rappresentazione della scena 3D ogni volta che la funzione principale di disegno è invocata nel programma Javascript. Attraverso le API WebGL si possono connettere variabili ed array Javascript agli attributi GLSL. Durante l'inizializzazione della pagina WebGL il codice di shading è compilato e copiato nella memoria della scheda grafica per essere eseguito dall'Unità di Processo Grafico.

- il programma Javascript è il cuore dell'applicazione: la libreria grafica, la matematica matriciale e tutte le funzioni di supporto. Gli input dell'utente sono connessi con le variabili Javascript mediante eventi ad-hoc.

- la sezione html è usata principalmente per l'input/outup con l'utente: checkbox, valori e spesso anche una zona di debug per segnalare errori di esecuzione.

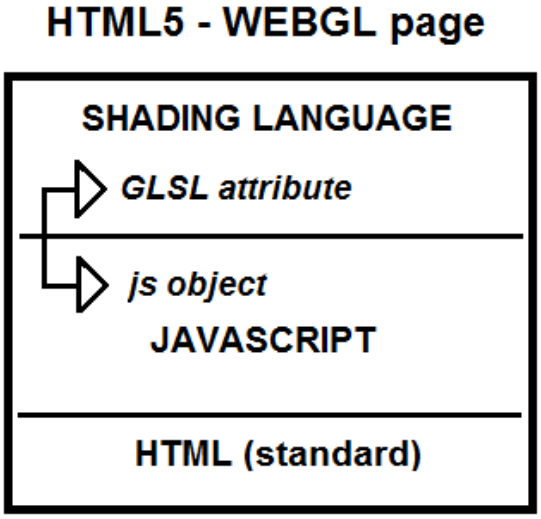


Figura 2.14: La tipica applicazione WebGL è composta da tre parti: la sezione HTML, il programma principale Javascript e la sezione di Shading il cui codice è compilato e copiato nella sceda video per permettere l'accelerazione 3D

2.3.2 Il client WebGL

Nell'ambito del progetto Wikimemo.it si è scelto di utilizzare la tecnologia WebGL appena descritta per realizzare l'assistente virtuale che deve accompagnare l'utente nella visita del portale.

Ci siamo basati sulle stesse logiche dell'applicazione LUCIA descritta nel capitolo 2 riprogettando tutto il sistema per trasformare una applicazione monolitica in una moderna applicazione web.

LUCIA-WebGL funziona con il classico paradigma client-server (fig. 2.3.2). Il client (un browser web o una App per gli smartphone) si connette al server chiedendo una pagina web. Le pagine fornite dal server sono state scritte utilizzando il nuovo standard HTML5 che permette l'utilizzo di nuovi tag e contenuti multimediali. Uno di questi è il Canvas 3D, il luogo in cui vive l'animazione grafica come illustrato nella sezione 2.3.1.

All'inizio della connessione vengono trasmessi tutti i dati che compongono in modello utilizzando il formato di scambio dati JSON (Javascript Objec Notation) [48]. Questo è l'unico momento in cui si è in attesa qualche secondo a causa della quantità di dati trasmessa. Una volta che il modello

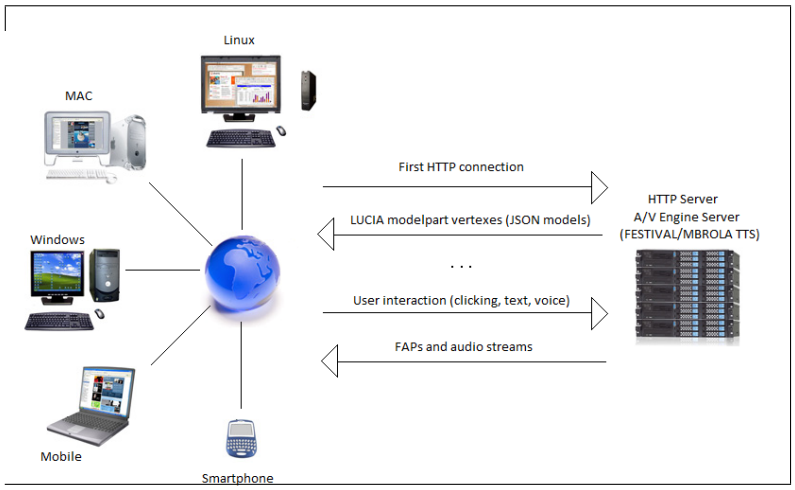


Figura 2.15: La nuova architettura client-server: WebGL permette a qualsiasi browser compatibile, anche quelli di smartphone e PDA, di interagire con LUCIA attraverso il protocollo http. All'inizio della connessione il modello di LUCIA viene trasmesso dal server utilizzando il formato JSON. Nei successivi scambi dati solo i flussi audio e FAP sono necessari per l'animazione.

3D è stato caricato e renderizzato nella pagina web in posizione neutrale tutte le comunicazioni tra il client e il server sono a banda molto limitata. Questo deriva direttamente dallo standard MPEG-4 a dal meccanismo di animazione. Lo stream audio e quello dei FAP sono le sole informazioni necessarie per animare la faccia parlante.

La tecnica di animazione adottata in LUCIA-WebGL è la stessa utilizzata da LUCIA (descritta precedentemente nel paragrafo 2.2.2).

2.3.2.1 Efficienza dell'animazione

La velocità di elaborazione è un parametro critico del progetto perché si richiede l'utilizzo dell'avatar anche su dispositivi mobili a bassa potenza elaborativa. I vertici che conpongono il volto di LUCIA sono migliaia e le operazioni che calcolano il loro movimento sono la percentuale principale del tempo di elaborazione richiesto. Molta attenzione è stata posta al miglioramento dell'efficienza di queste sezioni di codice. Per migliorare l'efficienza degli algoritmi abbiamo fatto uso intensivo di lookup-tables delle funzioni goniometriche e abbiamo effettuato la suddivisione del volto in aree specifiche. Nel corso dell'animazione, i *feature points* vengono mossi secondo quanto indicato dai FAP (come abbiamo visto in 2.2.2.3) e il reticolo dei poligoni delle zone della pelle circostante viene deformato di conseguenza. Per ridurre il numero di vertici da prendere in considerazione la struttura poligonale, che descrive la pelle superficiale della faccia 3D, è stata scom-

posta in 68 aree specifiche. Le aree specifiche sono particolari porzioni del reticolo poligonale in corrispondenza dei *feature points*; esse sono zone che identificano univocamente l'insieme di poligoni che viene sottoposto all'azione deformante di un determinato FAP. Si noti che i poligoni che compongono ogni area specifica possono essere nella zona di influenza di uno o più *feature points*.

Le migliorie algoritmiche effettuate hanno permesso di incrementare del 100% il frame rate dei dispositivi mobili a bassa potenza elaborativa (da 11fps a 22fps su Android Samsung Galaxy Note)

LUCIA-WebGL è stata la prima applicazione WebGL in assoluto prodotta per l'intero mondo Apple. Questo perché siamo riusciti ad utilizzare una funzione riservata, non ancora ufficialmente supportata, per attivare la grafica 3D anche per tali dispositivi. Tale risultato è stato presentato alla conferenza internazionale Web3D 2012 [11].

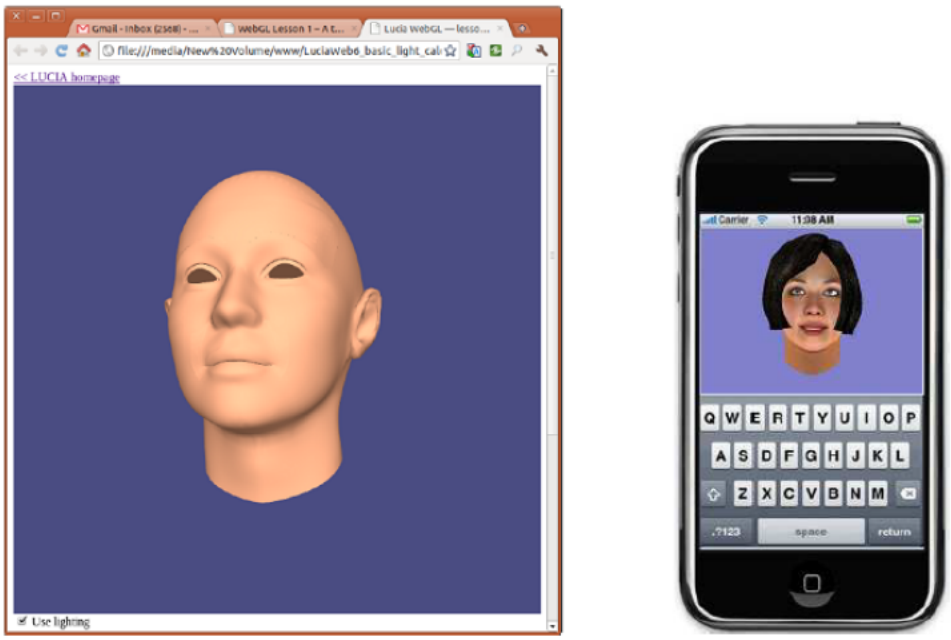


Figura 2.16: Sulla sinistra l'elemento pelle renderizzato in Chrome Web Browser. Sulla destra LUCIA-WebGL in funzione su un dispositivo iOS (Apple).

2.3.3 Il prototipo Wikimemo.it

Il prototipo realizzato per il progetto Wikimemo serve principalmente a dimostrare i meccanismi di integrazione dell'avatar con la pagina web. Presenta una interfaccia molto semplice che attraverso una ricerca indicizza una

base dati di opere di letteratura. Il sistema permette all'utente di navigare nei contenuti, effettuare ricerche di frasi e parole e sentire come sono pronunciate in un contesto specifico. I risultati di una query di ricerca possono essere ascoltati con la voce di un italiano nativo o con la voce sintetica di un avatar di supporto che ne mostra il movimento labiale (si veda figura 2.3.3). Il paper "Discovering the Italian literature: interactive access to audio indexed text resources" di cui sono co-autore descrive più approfonditamente le parti del sistema. Sarà presentato alla conferenza internazionale LREC 2014.

Figura 2.17: Il componente del prototipo che mostra i risultati della ricerca della parola cammin (evidenziata in giallo). Tutte le informazioni vengono restituite in forma tabellare. I bottoni per ascoltare i risultati sono sulla parte destra della tabella

2.3.4 La calibrazione del modello

In una scena creata tramite lo standard MPEG-4, una faccia sintetica è un nodo nella struttura ad albero che definisce la scena stessa. La forma, la texture e le espressioni del volto sono generalmente controllate da bitstream che contiene le istanze per settare i valori dei FDP e dei FAPs. Ogni decoder, dunque, deve avere un predefinito modello facciale da poter animare e visualizzare tramite le operazioni di setup e calibrazione degli FDP.

L'applicazione LUCIA soddisfa le specifiche del *Simple Facial Animation Object Profile* di MPEG-4 ovvero possiede un modello proprietario che anima

con i FAPs. Non è in grado di importare modelli esterni, ne di effettuare una calibrazione di quello esistente.

In Lucia-WebGL questa importante feature è stata aggiunta. Lo schema in figura 2.18, esprime il diagramma a blocchi del nuovo F.A.E.

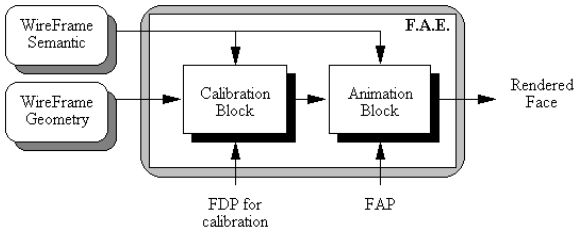


Figura 2.18: Schema a blocchi del FAE LUCIA-WebGL

Il *Calibration Block* ha il compito di adattare il modello proprietario secondo gli FDP ricevuti in modo da fargli assumere le sembianze di un volto specifico.

L' *Animation Block* realizza i movimenti del volto sintetico mediante gli spostamenti definiti dai parametri FAPs.

Il vantaggio offerto da questo tipo di implementazione è la possibilità di operare su modelli differenti, qualunque sia la loro geometria.

Ora dunque abbiamo la possibilità di cambiare i parametri definizione FDP e animare qualsiasi avatar che rispetti il modello poligonale che abbiamo scelto. Nel prossimo capitolo esploreremo la possibilità di creare un modello personalizzato usando il sensore Microsoft Kinect (uno scanner 3D a bassissimo costo).

2.4 Valutazione dell'accettabilità

Dal 2012 lo ISTC-CNR di Padova partecipa al progetto di ricerca Speaky Acutattile (Industria 2015) che prevede l'utilizzo dell'avatar LUCIA in un ambiente domotizzato pensato per abbattere il digital divide delle persone non vedenti e degli anziani.

In collaborazione con la Fondazione Ugo Bordoni di Roma (FUB) sono state effettuate delle registrazioni audio video per testare l'usabilità dell'interfaccia Avatar. I soggetti totali esaminati sono stati 80, 40 non vedenti e 40 persone con più di 65 anni d'età. Per avere un campione variegato e non influenzato da pronunce o modi di dire locali le registrazioni sono avvenute

in quattro città italiane differenti: Torino, Padova, Roma e Palermo.A Padova abbiamo effettuato 23 sedute: 10 non vedenti e 13 over 65.

L'obiettivo principale di questo esperimento è stato valutare l'accettabilità di un assistente virtuale da parte di un anziano o un non vendente supponendo che esso sia di reale utilità ovvero che permetta, attraverso l'uso della sola voce, di eseguire una ventina di compiti di quotidiana routine. Alcuni esempi: Impostare la radiosveglia, Comprare un biglietto del treno, Programmare una registrazione tv, Prenotare un ristorante, Prenotare un taxi, Fare la spesa online.

Per raggiungere tale obiettivo si è simulato un sistema automatico all'insaputa del soggetto in esame. Attraverso un teleoperatore (il mago) che da posizione remota ascolta le domande del soggetto l'avatar fornisce risposte attinenti e articolate. Esse possono essere risposte standard preregistrate oppure risposte ad hoc sintetizzate al momento. Lo strumento utilizzato è un software di telecontrollo chiamato Wizard of Oz (realizzato da FUB) che permette la visualizzazione di "scenari" diversi a seconda del tipo di compito da svolgere mentre l'operatore remoto sceglie le risposte da fornire interfacciandosi solo con l'avatar LUCIA (figura 2.4).

Figura 2.19: L'interfaccia utente di Wizard Of Oz. Sullo sfondo sono presenti scenari inerenti al compito che si sta richiedendo di svolgere, mentre la comunicazione avviene in modo naturale per mezzo di un dialogo tra l'utente e l'avatar.

Dal campione di Padova di 13 persone over 65 si evince che il:

- 7% non capiva bene (1p)
- 23% si è stancato (3 p)
- 69% si è divertito (9 p)
- 77% ha trovato piacevole la voce e la faccia dell'avatar (10 p).
- 84% ha trovato utile l'assistente e gli piacerebbe averlo a casa (11p),
- 93% ha trovato semplice l'uso (12 p),
- 100% piacerebbe avere un avatar con un volto e una voce personalizzata

La persona che non capiva bene aveva problemi di udito e le 3 persone che si sono stancate erano le più anziane di 78, 80 e 83 anni. Possiamo decisamente affermare che un assistente personale utile, affidabile e semplice da usare avrebbe una accoglienza calorosa tra le persone nella fascia 65-75.

La conferma che la personalizzazione dell'avatar riscuote successo è una ottima introduzione per l'argomento presentato nel prossimo capitolo: la procedura di creazione di un avatar personale.

Capitolo 3

Personal Avatar

Quest'ultimo capitolo della tesi è la presentazione di un promettente *work in progress*: una procedura semiautomatica per la creazione di un avatar personalizzato. Utilizzando il motore di animazione facciale descritto nel capitolo precedente vogliamo realizzare un carattere virtuale che abbia le stesse sembianze e parli con la stessa voce e con lo stesso parlato visivo di un attore umano. La procedura è stata sviluppata sulla base del funzionamento di LUCIA ed è dunque relativa al set di fonemi/visemi della lingua italiana.

Il tutto vuole essere realizzato con hardware standard a bassissimo costo e con diffusione sempre crescente per via del suo utilizzo nel campo ludico delle consolle: la Microsoft Kinect [51], un sensore di profondità basato su un proiettore di infrarossi e un ricevitore monocromatico CMOS (in figura 3.1 il dispositivo e le sue componenti). Questa tecnologia non risente delle variazioni di luce ambientale e questo è un enorme punto di forza rispetto ad altri dispositivi stereoscopici basati su telecamere. La Kinect può essere considerato a tutti gli effetti uno scanner 3D molto rumoroso e noi lo utilizzeremo per sostituire il costoso dispositivo optoelettronico ELITE utilizzato per LUCIA (appendice 4.1). A discapito di una perdita di precisione otteniamo la possibilità di effettuare il tracciamento delle traiettorie cinematiche di marcatori virtuali *(marker-less motion capture)*.

E' notizia recente che sensori simili sono già stati integrati in alcuni smartphone e tablet di ultima generazione rendendo potenzialmente disponibili milioni di utenti per una applicazione che fin'ora è stata confinata nei laboratori di ricerca.

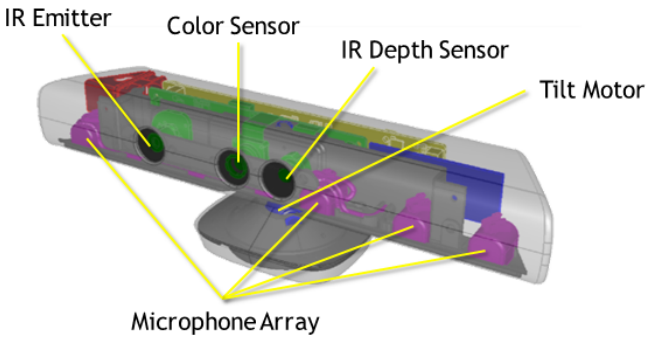


Figura 3.1: La Microsoft Kinect è un sensore di profondità basato su un proiettore di infrarossi e un ricevitore monocromatico CMOS. Sono presenti inoltre una telecamera RGB e un array di 4 microfoni.

3.1 La procedura

La procedura di personalizzazione (illustrata in figura 3.2) deve produrre tre output: il modello 3D (mesh e texture), una voce personalizzata e un modello di parlato visivo che sia basato sulla cinematica degli articolatori del soggetto da avatarizzare.

Vi sono quattro fasi ben distinte da considerare:

- Creazione del modello 3D
- Acquisizione dati
- Creazione del modello vocale
- Calcolo dei coefficienti del modello di coarticolazione

La prima caratteristica che l'avatar deve avere è un aspetto quanto più simile al soggetto ripreso. Inoltre il modello creato deve riprodurre in modo soddisfacente i movimenti della bocca, la zona interessata nel tracciamento dei parametri articolatori. Raggiungiamo questo obiettivo per mezzo del software Faceshift [37] ottenendo una mesh poligonale texturizzata con le sembianze desiderate. I dettagli nella sezione 3.3.

Per effettuare l'aquisizione dati il soggetto deve leggere un set di frasi per produrre delle coppie sincronizzate di file con il parlato audio e i movimenti cinematici degli articolatori. Il set è scelto opportunamente per fornire la copertura di tutte le possibili combinazioni audio e cinematiche (fonemi e visemi) che servono a calcolare i parametri del modello vocale e del modello del parlato visivo.

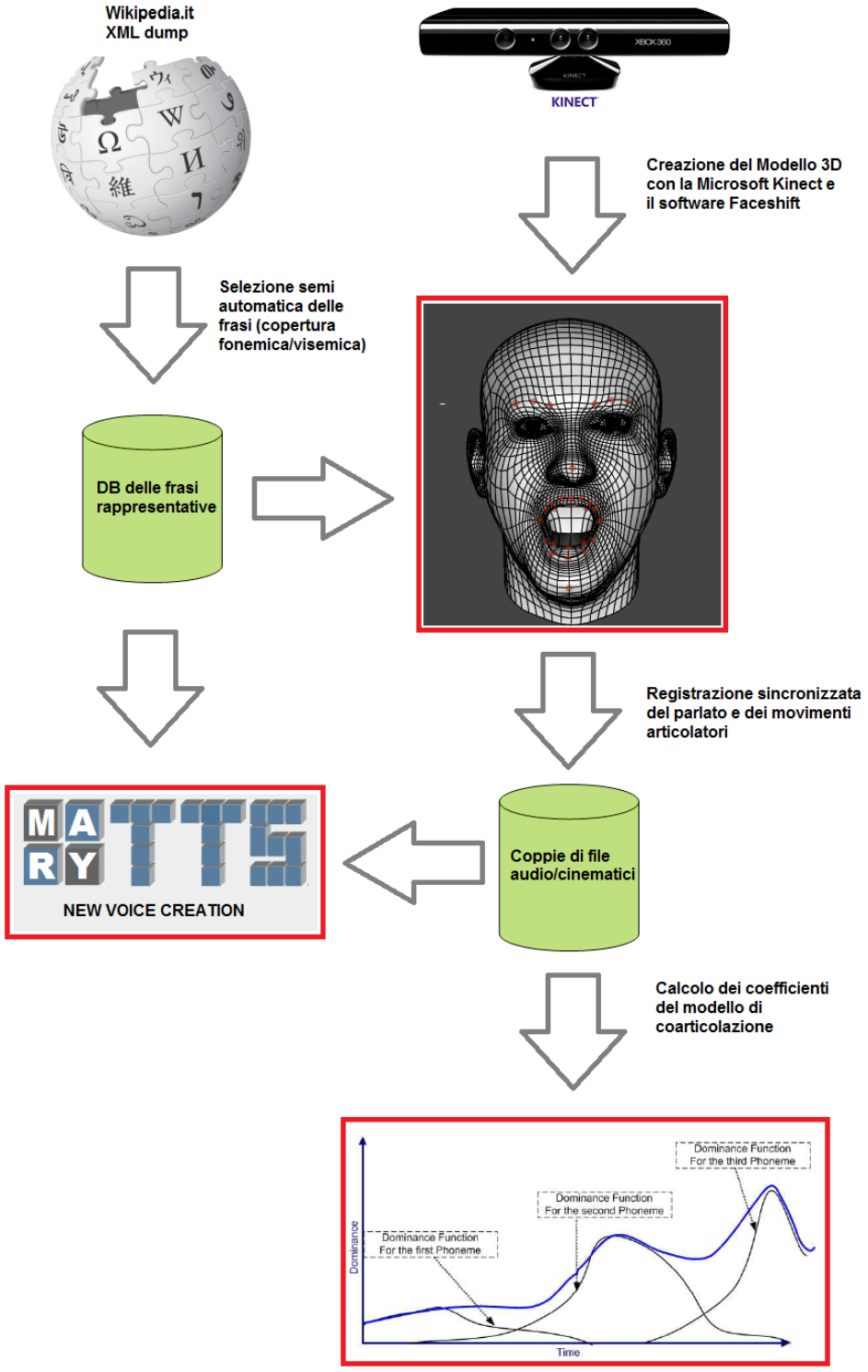


Figura 3.2: Lo schema funzionale della procedura di creazione del modello personale. Nei riquadri in rosso gli output del sistema: il modello 3D, la voce personalizzata per il sintetizzatore MARYTTS e il modello del parlato visivo.

Il modello vocale è ottenuto mediante la creazione di una nuova voce italiana con il sintetizzatore MARYTTS [97], lavoro implementato allo ISTC-CNR di Padova da Tesser [103]. MARYTTS è il sintetizzatore utilizzato nella nuova versione dell'avatar realizzata nell'ambito di questa tesi e mostrata nel capitolo precedente. Nella sezione 3.2 forniamo maggiori dettagli.

La personalità dell'avatar è data in particolare dal modello del parlato visivo che costituisce la peculiarità di questo sistema. Per creare un modello della produzione del parlato occorre prendere in considerazione non solo gli aspetti anatomici e fisiologici statici, ma anche la loro evoluzione temporale. La tematica viene affrontata in dettaglio nella sezione 3.4.

3.2 Il modello vocale

Le tecnologie all'avanguardia nel campo dei sistemi *Text To Speech* utilizzano una modalità *corpus based* a differenza della modalità basata su difoni utilizzata negli anni 80-90. Per creare una nuova voce per un sistema TTS tramite questa modalità è necessario un corpus contenente la registrazione di varie frasi lette da parte del parlante.

La selezione dei testi da leggere è una parte molto importante della procedura: per ottenere un sistema in grado di sintetizzare in maniera accurata ogni sequenza fonetica/prosodica di una lingua, è necessario che all'interno del corpus vi siano tali suoni. Per selezionare un set di frasi con queste proprietà il sistema MARYTTS [97] utilizza una procedura che, partendo dai testi di Wikipedia, individua il set di frasi ottimale per questo compito.

Per l'italiano sono stati fatti alcuni esperimenti [103] e ne è risultato che le prime 72 frasi selezionate dall'algoritmo contengono la copertura fonetica semplice. La copertura difonica totale non si può ottenere perché ci sono alcune sequenze di suoni, che non sono pronunciabili nella lingua italiana. Si può stimare il raggiungimento si un valore di copertura dopo il quale la copertura difonica non cresce più tanto attorno alle 1000/1500 frasi. Per la copertura prosodica il ragionamento è analogo, ma ancora più complesso, perché la copertura dipende dal sistema di prosodia simbolica adottata dal sistema. Anche in questo caso non si può raggiungere la copertura totale e tale limite, nel caso del nostro esempio, è spostato ancora più in alto, attorno alle 4000 frasi.

Dopo aver registrato tutte le frasi è possibile creare il modello vocale in grado di riprodurre timbro e prosodia presenti nel corpus. A seconda della tecnologia utilizzata il modello conterrà la forma d'onda vera e propria delle

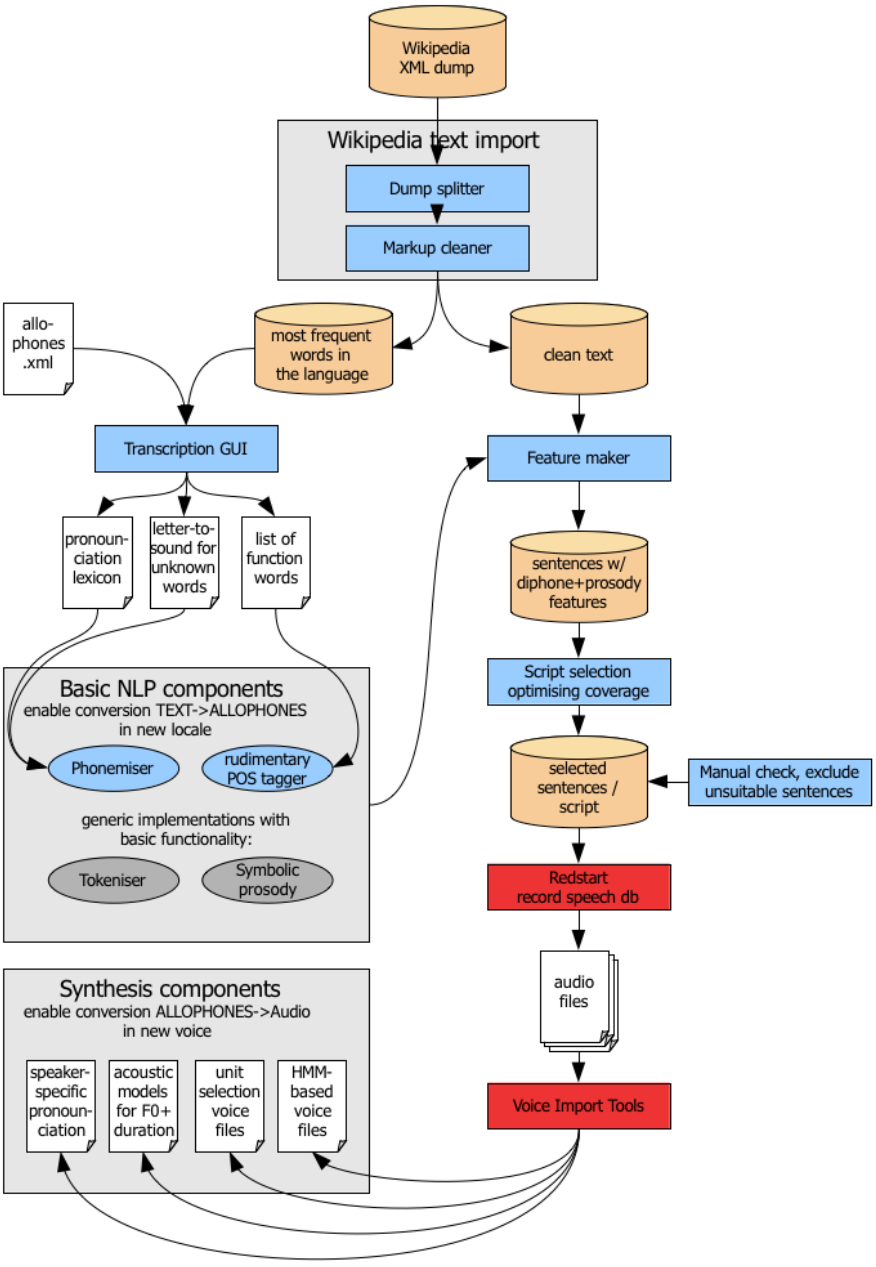


Figura 3.3: La procedura utilizzata da Tesser [103] per creare una nuova lingua (Italiano) e una nuova voce per il sintetizzatore MARYTTS. Il set di circa 1000 frasi opportunamente scelto permette la totale copertura fonemica e articolatoria.

unità acustiche presenti nel corpus (Sintesi Unit Selection [14]) oppure sarà un modello statistico dei parametri timbrici e prosodici (Sintesi Statistica Parametrica [117]). La procedura completa è illustrata in figura 3.3.

3.3 Modello 3D e acquisizione dati

Modellare la geometria 3D con annessa informazione cromatica di un volto specifico è una problematica molto importante in animazione facciale. I range scanner (la Kinect ricade in questa categoria) o la disparità stereo possono misurare le coordinate tridimensionali producendo le cosiddette nuvole di punti. Questi dati grezzi sono spesso poco adatti per l'animazione: mancano informazioni sulle strutture facciali, i vertici del modello sono mal distribuiti e gli errori di misurazione producono artefatti; inoltre, molti metodi di misurazione producono modelli incompleti, privi di capelli, orecchie, occhi. Una post-elaborazione dei dati di misura è sempre necessaria.

Un approccio molto interessante è basato su un set precostruito di database facciali [17]. In primo luogo, attraverso un processo di scansione, si raccoglie un gran numero di facce per popolare un database. Questi modelli facciali 3D di esempio costituiscono la base dello spazio vettoriale di ogni eventuale volto umano in termini di geometria e texture: nuovi volti ed espressioni possono essere rappresentati come combinazione lineare degli esempi.

Il metodo più utilizzato in letteratura consiste nel partire da un modello generico che possieda tutte le necessarie informazioni di struttura e di animazione e poi deformarlo sulla base delle misurazioni 3D. L'adattamento geometrico facilita anche il trasferimento della texture se questa viene catturata contemporaneamente alle misurazioni (è proprio quello che succede con Kinect). Processi di modellazione e di adattamento utilizzano vari approcci come interpolazioni di dati sparsi [107, 44] e proiezioni in coordinate cilindriche [57]. Le funzioni a base radiale sono in grado di approssimare o interpolare con poco scarto ipersuperfici regolari [91] come la forma del volto umano. Il processo di morphing del volume 3D considera innanzitutto i punti di riferimento biologicamente significativi che vengono selezionati manualmente: contorno degli occhi, del naso, delle labbra e perimetro del volto. Questi punti di riferimento definiscono i coefficienti del kernel della funzione a base radiale usati per il morph del volume. I punti nella mesh generica vengono interpolati utilizzando tali coefficienti. Il successo del morphing dipende fortemente dalla selezione dei punti di riferimento. Con questo approccio il "morph" non richiede lo stesso numero di vertici nelle mesh coinvolte dato che i punti mancanti sono interpolati e la mesh trasformata si avvicina al meglio alla mesh bersaglio se vengono selezionate

corrispondenze appropriate [90].

La maggior parte dei metodi richiede una supervisione nella creazione del modello. Un processo totalmente automatico non esiste, ma alcuni metodi utilizzano un approccio semi-automatizzato in cui l'intervento umano è molto limitato e non influenza direttamente il modello.

E' questo il caso dell'eccellente lavoro di Hao-Li e Weise [115, 114] che presenta un sistema di modellazione e tracking facciale basato sulla ottimizzazione delle corrispondenze di superfici non rigide ottenute per mezzo di sensori di profondità [61]. Tutti i parametri dell'ottimizzazione sono scelti automaticamente senza nessun intervento manuale. Questa ricerca ha dato vita al software commerciale Faceshift [37] che è utilizzato per la realizzazione degli scopi di questa tesi.

Per la costruzione del modello vengono registrate 19 pose differenti che servono a coprire le più comuni deformazioni facciali: per ognuna il soggetto deve ruotare lentamente la testa prima a destra e poi a sinistra per permettere la ricostruzione di ogni parte della vista frontale del volto. Le pose sono:

- Neutrale (posizione di riposo come in MPEG-4)
- Brows Up (alzare le ciglia)
- Brows Down (abbassare le ciglia)
- Sneer (abbassare le ciglia e alzare gli zigomi)
- Smile (sorridere)
- Mouth Left (spingere la bocca a sx)
- Mouth Right (spingere la bocca a dx)
- Open (aprire al massimo la bocca)
- Funnel (pronunciare la vocale O in modo prolungato)
- Kiss (baciare, massima protrusione)
- Puff (gonfiare le guance)
- Teeth (mostrare i denti)
- Lip Down (spostare in giù il labbro inferiore)
- Dimple (serrare le labbra)
- M (emettere il suono M)

- Jaw Left (ruotare la mandibola a sx)
- Jaw Right (ruotare la mandibola a dx)
- Jaw Front (spingere la mandibola in avanti)
- Chew (spostare la mandibola in basso senza aprire la bocca)

Nelle figure 3.4 e 3.5 si possono osservare le espressioni Smile e Kiss. Oltre alla mesh deformata e alla lettura del sensore di profondità è mostrato l'errore (differenza tra il modello e i dati di profondità) e la texture applicata alla lettura grezza (purtroppo la telecamera RGB è di bassa qualità). L'elenco visivo completo di tutte le espressioni è presente nell'appendice 4.6.

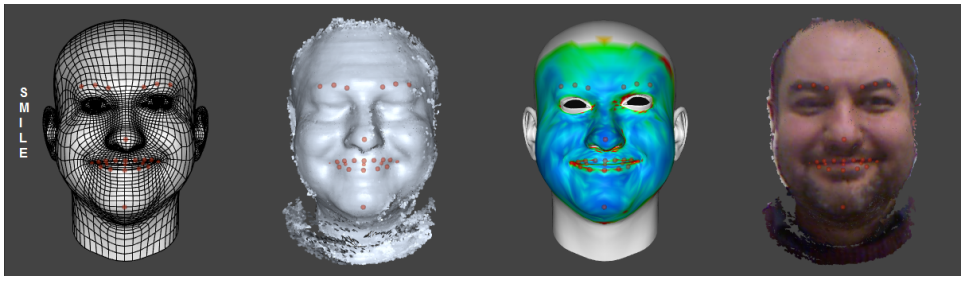


Figura 3.4: L'espressione Smile si ottiene mimando un sorriso; è la massima estensione laterale degli angoli della bocca.

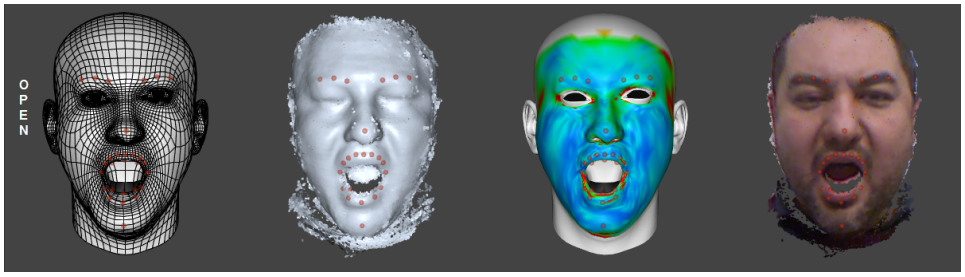


Figura 3.5: L'espressione Open si ottiene mimando un grido forte; è la massima apertura della bocca.

Il set di registrazioni viene utilizzato per un processo di training che permette di deformare la generica mesh di partenza in una che riproduce fedelmente la geometria e l'aspetto dell'attore (vedi figura 3.6).

Ottenuto un modello che permetta il tracciamento degli articolatori si può iniziare la fase di acquisizione dati; di tutti i marcatori virtuali mostrati in figura 3.7 se ne usano solo 6 nella stessa configurazione usata con il sistema optoelettronico ELITE illustrato in figura 4.1. Il soggetto (seduto

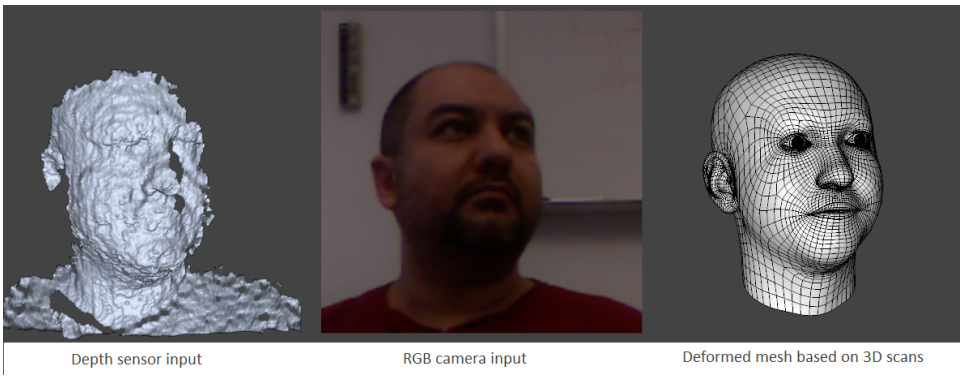


Figura 3.6: Modellazione 3D con la Microsoft Kinect e il software FaceShift. Da sinistra: la lettura rumorosa del sensore, l'output della telecamera RGB e la mesh poligonale ottenuta per mezzo della fase di training.

di fronte alla Kinect ad una distanza di 60cm circa con il sensore ad altezza occhi) deve leggere un set composto da circa 1000 frasi cercando di mantenere la stessa posizione spaziale e rendendo ben visibili i movimenti labiali. Questo set è scelto opportunamente per fornire la copertura di tutte le possibili combinazioni audio e cinematiche che servono a calcolare i parametri del modello vocale e del modello del parlato visivo. Ogni frase deve essere registrata più volte (un minimo di 5) per dare validità statistica ai movimenti (si effettua una media delle registrazioni eliminando quelle palesemente difformi). Una sessione di registrazione con il programma Faceshift è mostrata in figura 3.8.

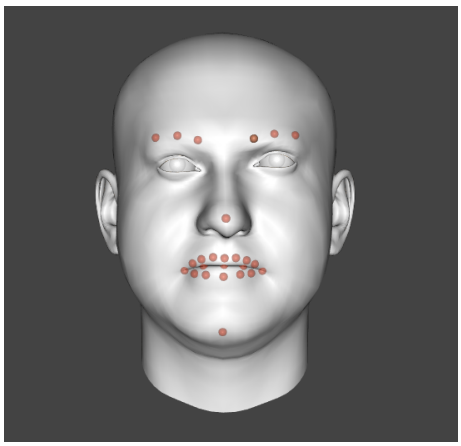


Figura 3.7: Ottenuto un modello che permetta il tracciamento degli articolatori si può iniziare la fase di acquisizione dati.

Grazie al preciso tracking facciale è possibile tracciare in tempo reale i markers virtuali. Una sessione di registrazione produce delle coppie di file

sincronizzate con il parlato audio e i movimenti cinematici degli articolatori del parlante.

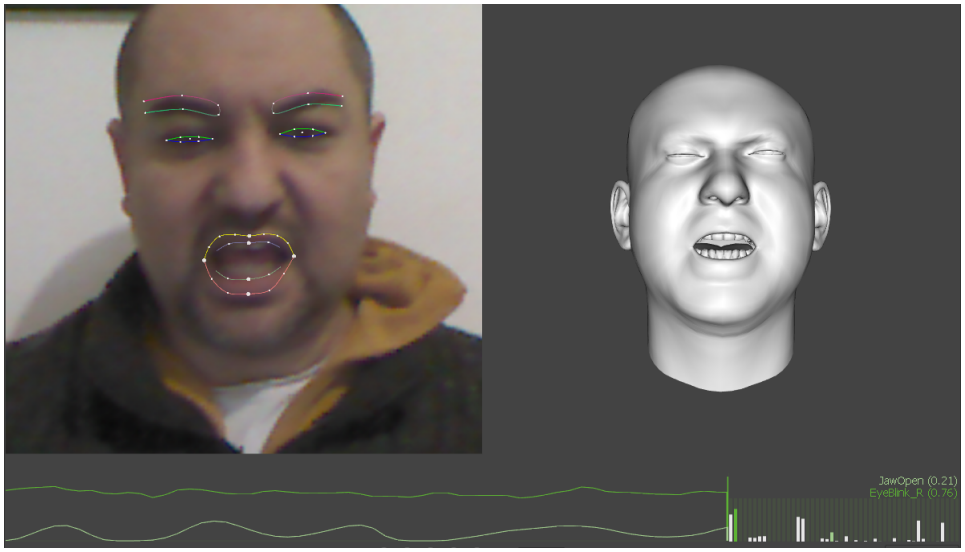


Figura 3.8: Una sessione di registrazione con Faceshift: il tracking facciale permette di tracciare in tempo reale i markers virtuali.

I parametri articolatori presi in considerazione sono gli stessi utilizzati per la creazione del modello del parlato di LUCIA:

1. Labbro inferiore (LI)
2. Labbro superiore (LS)
3. Arrotondamento (AR)
4. Protrusione Labbro Inferiore (PI)
5. Protrusione Labbro Superiore (PS)
6. Emiapertura Mandibolare (EM)

Per una spiegazione dettagliata si consulti l'Appendice 4.1. Il database audio/cinematico prodotto costituisce la base di partenza per la creazione del modello del parlato visivo che approfondiamo nella prossima sezione.

3.4 Il modello del parlato visivo

Un metodo per realizzare la descrizione degli andamenti cinematici degli articolatori visibili è quello di identificare le unità di base del segnale sonoro, chiamate fonemi, a cui associare i visemi, che sono le analoghe unità visive,

determinati attraverso test di intellegibilità. Tuttavia un fonema non ha sempre una stessa rappresentazione visiva. Il suo aspetto dipende da alcuni fattori quali enfasi, intonazione e velocità del parlato e soprattutto dal particolare contesto determinato dai fonemi adiacenti (co-articolazione). Abbiamo ampiamente discusso di questo fenomeno nella sezione 1.4 mostrando i vari approcci presenti in letteratura. Nella prossima sezione analizziamo in dettaglio il modello utilizzato nel nostro sistema.

3.4.1 Modello di coarticolazione

LUCIA e Personal Avatar adottano una versione modificata del modello di coarticolazione proposto da Coehn e Massaro [28]. Esso riesce a descrivere i movimenti articolatori con un buon grado di realismo, a partire dalla rappresentazione fonetica. La variazione dei coefficienti che caratterizzano il gesto articolatorio permette di realizzare differenti strategie di produzione rendendo il modello adattabile a diversi contesti linguistici. La modifica della funzione di dominanza mediante l'introduzione dei fattori di resistenza e di forma permette di superare i limiti presentati dal modello originale.

3.4.1.1 La funzione di dominanza

Ad ogni singolo gesto articolatorio è associata una funzione di dominanza. Una funzione di dominanza è caratterizzata da una propria ampiezza, durata, e grado di attivazione. L'ampiezza determina l'importanza relativa del gesto per il segmento; la durata stabilisce l'estensione del movimento ed influisce sul grado di sovrapposizione che ne conseguirà; il grado di attivazione caratterizza il fatto che il gesto si avvia in modo più o meno graduale. Per rappresentare una dominanza che caratterizzasse in modo differente la coarticolazione anticipatoria e perseverativa, è stata utilizzata una funzione esponenziale asimmetrica del tipo:

$$D(\tau) = \begin{cases} \alpha\, e^{-\theta_{bw}|\tau|^c} & \text{se } \tau < 0 \\ \alpha\, e^{-\theta_{fw}|\tau|^c} & \text{se } \tau > 0 \end{cases} \tag{3.1}$$

in cui τ rappresenta la distanza temporale dal centro del segmento fonetico, α l'ampiezza della funzione di dominanza; θ_{bw} e θ_{fw} caratterizzano rispettivamente estensione anticipatoria ed estensione perseverativa ed infine c indica il grado di attivazione del gesto. Queste funzioni vengono combinate nel tempo con il valore stimato del target articolatorio e normalizzate per ottenere l'andamento complessivo:

$$F(t) = \frac{\sum_{i=1}^{N} T_i D_i(t - t_i)}{\sum_{i=1}^{N} D_i(t - t_i)} \tag{3.2}$$

dove N si riferisce al numero totale di segmenti, T_i e $D_i(t)$ rappresentano target e funzione di dominanza del segmento i-esimo centrato in t_i.

Tre coefficienti definiscono la conformazione di una funzione di dominanza, due per determinarne l'estensione temporale e uno che ne rappresenta il grado di attivazione. Nella versione originale si utilizza un valore costante del coefficiente di attivazione. In base al confronto con andamenti cinematici reali si è visto che un valore unitario di c fornisce i migliori risultati. È opportuno tuttavia considerare che il grado di attivazione del gesto articolatorio vari a seconda del fonema e che possa essere diverso per l'estensione anticipatoria e per quella perseverativa. Considereremo da ora in poi, quindi, una funzione di dominanza del tipo:

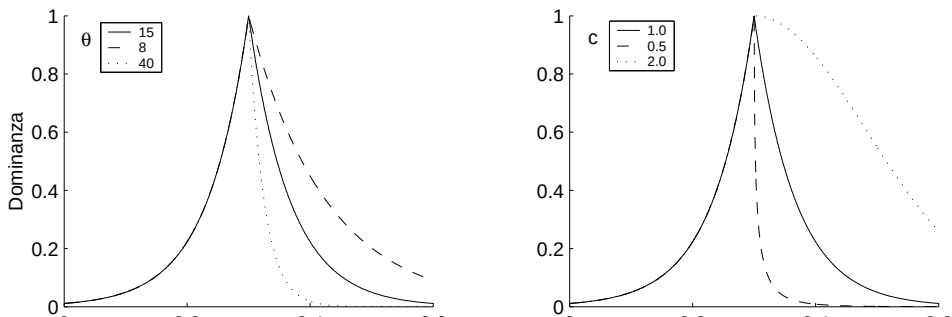
$$D(\tau) = \begin{cases} \alpha\, e^{-\theta_{bw}|\tau|^{c_{bw}}} & \text{se } \tau < 0 \\ \alpha\, e^{-\theta_{fw}|\tau|^{c_{fw}}} & \text{se } \tau > 0 \end{cases} \tag{3.3}$$

in cui c_{bw} rappresenta ancora il grado di attivazione, mentre c_{fw} può essere interpretato come grado di rilascio del movimento articolatorio. Facciamo notare, come appare in figura 3.9, che per diversi valori del coefficiente c si ottiene una variazione dell'estensione della dominanza. Questa è una diretta conseguenza della variazione dell'andamento qualitativo delle curve, che corrisponde a differenti gradi di concavità o convessità. Una variazione del coefficiente θ, invece, è unicamente legata alla modifica dell'estensione del gesto articolatorio e non comporta variazioni delle proprietà qualitative delle curve della dominanza.

Figura 3.9: Andamento della funzione di dominanza per diversi valori dei coefficienti θ_{fw} (con $c = 1$) e c_{fw} (con $\theta = 15$).

3.4.1.2 La funzione coarticolatoria

L'utilizzo di valori di attivazione e di rilascio differenti ci permette già di dare rilievo ad importanti proprietà del modello. Vogliamo innanzitutto analizzare le caratteristiche di base della funzione coarticolatoria attraverso un modello semplificato a due target. Se consideriamo l'equazione (3.2) possiamo fare le seguenti osservazioni:



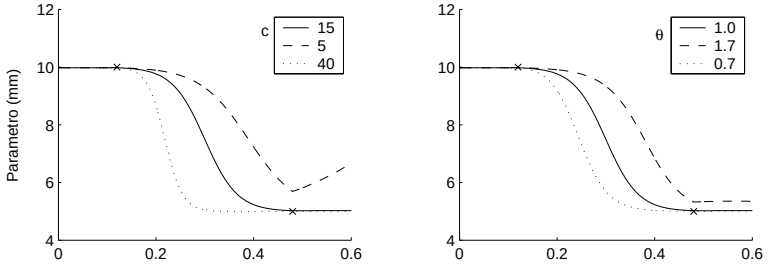
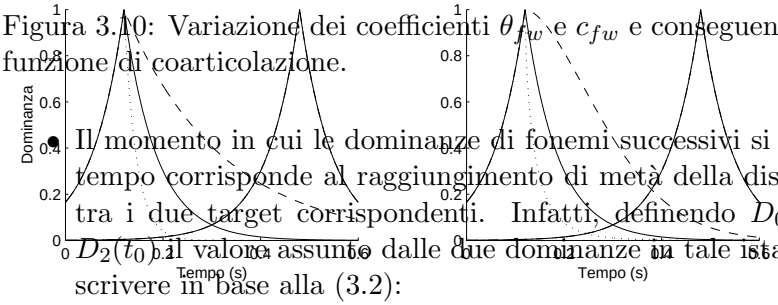


Figura 3.10: Variazione dei coefficienti θ_{fw} e c_{fw} e conseguente effetto sulla funzione di coarticolazione.



- Il momento in cui le dominanze di fonemi successivi si incrociano nel tempo corrisponde al raggiungimento di metà della distanza spaziale tra i due target corrispondenti. Infatti, definendo $D_0 = D_1(t_0) = D_2(t_0)$ il valore assunto dalle due dominanze in tale istante, possiamo scrivere in base alla (3.2):

$$F_0 = \frac{T_1 D_0 + T_2 D_0}{2D_0} = \frac{T_1 + T_2}{2} \tag{3.4}$$

  In questo caso non consideriamo l'eventuale contributo di altri fonemi al di fuori di quelli considerati che potrebbero modificare questa situazione.

- Nella situazione in cui in t_o i valori delle dominanze sono diversi, rispettivamente $D_1 = D_1(t_0)$ e $D_2 = D_2(t_0)$ per il primo e secondo target, il valore finale della funzione coarticolatoria sarà dato da

$$F_0 = \frac{T_1 D_1 + T_2 D_2}{D_1 + D_2}. \tag{3.5}$$

  Consideriamo il caso $T_1 > T_2$; possiamo allora scrivere:

$$F_0 = T_2 + \frac{D_1}{D_1 + D_2}(T_1 - T_2) \tag{3.6}$$

  Da ciò possiamo dedurre che, fissati i target, il valore finale rimane invariato a patto che il rapporto $D_1/(D_1 + D_2)$ rimanga costante.

- A parità di differenza tra le dominanze $D_1 - D_2$, più piccolo sarà il valore di D_1, maggiormente ci avviciniamo al primo target (se $D_1 > D_2$) o al secondo target (se $D_2 > D_1$).

Quanto appena detto appare in figura 3.10 in cui viene descritto il diverso comportamento della funzione coarticolatoria variando distintamente i coefficienti θ e c a partire dai valori base $\theta = 15$ e $c = 1$. In essa si è cercato di mantenere pressoché uguale il punto di incontro delle dominanze per coppie di curve ($\theta = 5$ a sinistra con $c = 1.7$ a destra; $\theta = 40$ a sinistra con $c = 0.7$ a destra). La differenza sostanziale degli andamenti nelle curve può essere osservata a partire dall'istante dell'incrocio delle dominanze in poi: un cambiamento del coefficiente θ ha maggiori effetti in prossimità del secondo target di quanto non si abbia cambiando il coefficiente c. In particolare nel caso di $\theta = 5$ l'influenza del primo target si mantiene forte sul secondo, mentre per $\theta = 40$ essa decade rapidamente. Con la variazione del coefficiente c si ha invece un comportamento più regolare.

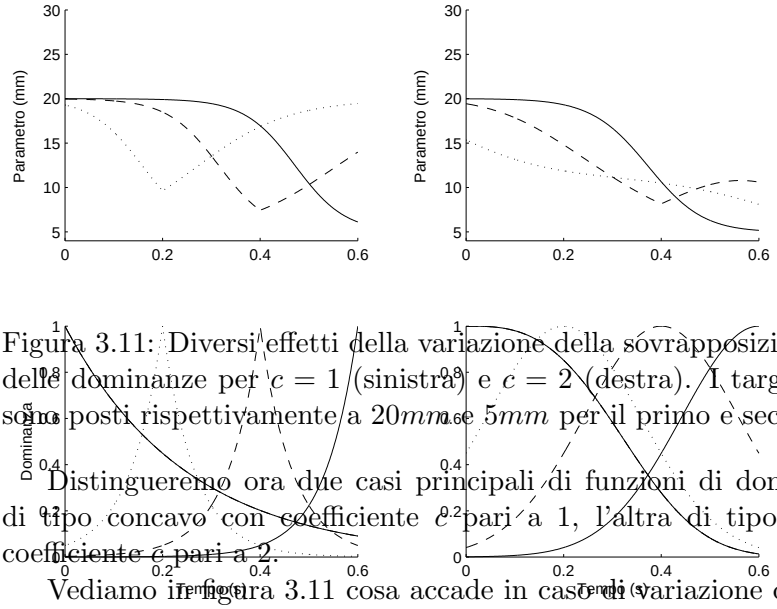


Figura 3.11: Diversi effetti della variazione della sovrapposizione temporale delle dominanze per $c = 1$ (sinistra) e $c = 2$ (destra). I target articolatori sono posti rispettivamente a $20mm$ e $5mm$ per il primo e secondo target.

Distingueremo ora due casi principali di funzioni di dominanza, l'una di tipo concavo con coefficiente c pari a 1, l'altra di tipo convesso con coefficiente c pari a 2.

Vediamo in figura 3.11 cosa accade in caso di variazione della posizione temporale tra funzioni di dominanza. Il fatto di avere valori "alti" dell'incrocio delle dominanze comporta una transizione più diretta tra i target.

Ciò vale fino a che l'estensione delle dominanza non va oltre il massimo dei target adiacenti. In questo caso l'andamento finale in prossimità di un target risente dell'influenza di quello adiacente ed il valore del target non viene raggiunto. L'utilizzo di funzioni di dominanza convesse fa sì che la situazione di incrocio "alto" venga raggiunta in più breve tempo. Tuttavia ciò vale anche per la situazione di perdita del target a causa della reciproca influenza delle dominanze.

In figura 3.12 è rappresentato un modello a tre target che illustra cosa accade quando facciamo variare l'ampiezza della dominanza. L'influenza del segmento centrale decresce man mano che l'ampiezza della sua funzione di dominanza si abbassa. In questo caso l'uso di funzioni di dominanza convesse rende più regolare il comportamento della funzione finale nella zona centrale.

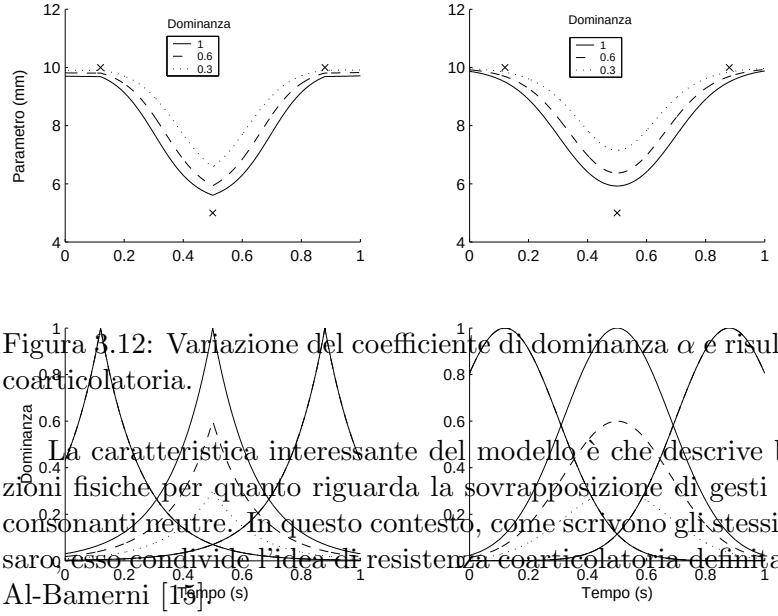


Figura 3.12: Variazione del coefficiente di dominanza α e risultante funzione coarticolatoria.

La caratteristica interessante del modello è che descrive bene le transizioni fisiche per quanto riguarda la sovrapposizione di gesti in presenza di consonanti neutre. In questo contesto, come scrivono gli stessi Cohen e Massaro, esso condivide l'idea di resistenza coarticolatoria definita da Bladon ed Al-Bamerni [15].

Altrettanto non si può dire in presenza di gesti con alta resistenza che entrano in contrasto. In particolare con le consonanti bilabiali (/p b m/) e labiodentali (/f v/) la grossa problematica è la perdita del target articolatorio a speech rate alti, in modo analogo a quanto descritto in figura 3.11.

Per superare questi limiti modifichiamo la funzione di dominanza originale mediante l'introduzione dei fattori di forma e di resistenza che introduciamo nei prossimi paragrafi.

3.4.1.3 L'approssimazione locale

Le considerazioni che faremo di seguito prendono spunto dagli studi relativi ai modelli locali per l'approssimazione di funzioni, ed in particolare alle ricerche effettuate da Cleveland [24] e Atkenson [5] nell'ambito della "regressione locale pesata".

Creare un modello locale significa a livello basilare:

- separare lo spazio di origine dei dati in regioni;
- associare ad ogni regione una caratteristica che approssimi l'andamento desiderato.

A livello elementare possiamo pensare che la caratteristica sia costante e che le regioni siano contigue. Avremo allora un andamento a gradini del tipo mostrato in figura 3.13. Immaginiamo, nella nostra situazione, che le regioni rappresentino i segmenti fonetici e che la caratteristica corrisponda al target articolatorio.

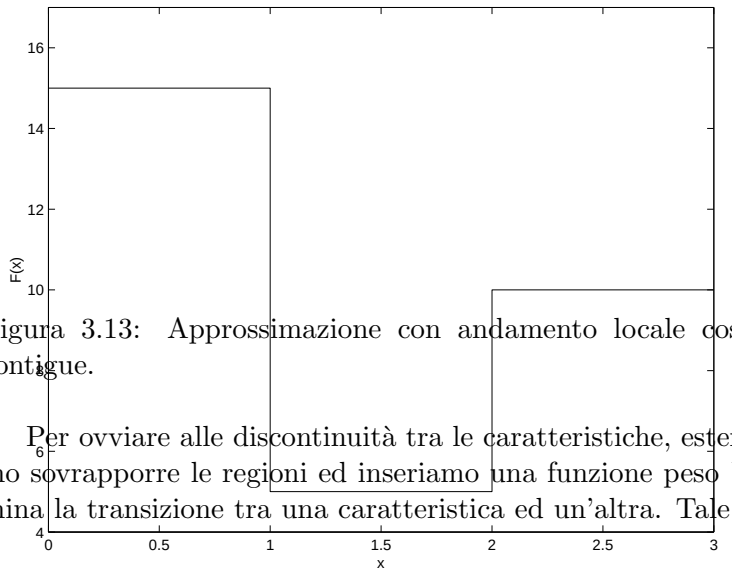


Figura 3.13: Approssimazione con andamento locale costante e regioni contigue.

Per ovviare alle discontinuità tra le caratteristiche, estendiamo e facciamo sovrapporre le regioni ed inseriamo una funzione peso $W(x)$ che determina la transizione tra una caratteristica ed un'altra. Tale funzione avrà il

massimo valore nella zona in cui la caratteristica è dominante, mentre decadrà rapidamente a zero nelle zone di transizione adiacenti. Definiamo quindi l'attivazione della caratteristica i-esima centrata in x_i come il peso $W_i(x)$ normalizzato dalla somma dei pesi delle altre caratteristiche. Poniamo, poi, che l'estensione delle regioni e delle sovrapposizioni possa cambiare per le varie caratteristiche. L'attivazione, se il numero di regioni è pari a N, sarà quindi

$$A(x)_i = \frac{W_i\left(\frac{x-x_i}{h_i}\right)}{\sum_{i=1}^{N} W_i\left(\frac{x-x_i}{h_i}\right)} \tag{3.7}$$

in cui h_i determina l'estensione della regione i-esima.

La funzione finale sarà la somma delle varie attivazioni moltiplicate per la caratteristica costante P_i

$$F(x) = \sum_{i=1}^{N} P_i A(x)_i = \frac{\sum_{i=1}^{N} P_i W_i\left(\frac{x-x_i}{h_i}\right)}{\sum_{i=1}^{N} W_i\left(\frac{x-x_i}{h_i}\right)} \tag{3.8}$$

Se poniamo $P_i = T_i$, $h_i = \frac{1}{\theta_i}$ e $W_i(x) = D_i(t)$ riotteniamo una funzione simile a quella descritta da Cohen e Massaro (vedi formula 3.2) e, se consideriamo una funzione peso di tipo gaussiano, il comportamento della (3.8) è del tutto analogo a quello descritto nel paragrafo precedente. La funzione peso $W(x)$ viene definita comunemente con il nome di *Kernel.* Nell'ambito della regressione locale, stiamo infatti parlando di *Kernel Regression* e la (3.8) prende il nome di *stimatore di Nadaraya-Watson.*

Un'approssimazione locale costante può tuttavia essere debole. Un caso più generale può essere ottenuto considerando una caratteristica locale che possa variare, ad esempio, all'interno di una famiglia di funzioni parametriche $P(x)$. Possiamo allora scrivere

$$F(x) = \frac{\sum_{i=1}^{N} P_i(x) W_i\left(\frac{x-x_i}{h_i}\right)}{\sum_{i=1}^{N} W_i\left(\frac{x-x_i}{h_i}\right)} \tag{3.9}$$

Solitamente si utilizza la famiglia dei polinomi di secondo grado per cui

$$P_i(x) = a_2(x - x_i)^2 + a_1(x - x_i) + a_o \tag{3.10}$$

Questa famiglia di polinomi è sufficiente per ottenere delle buone approssimazioni, ma si può generalizzare parlando della famiglia di polinomi di grado p. Possiamo vedere un esempio dell'effetto di tale variazione in figura 3.14 dove si è utilizzata una funzione del tipo

$$Y(x) = a_1 |x| + 1 \tag{3.11}$$

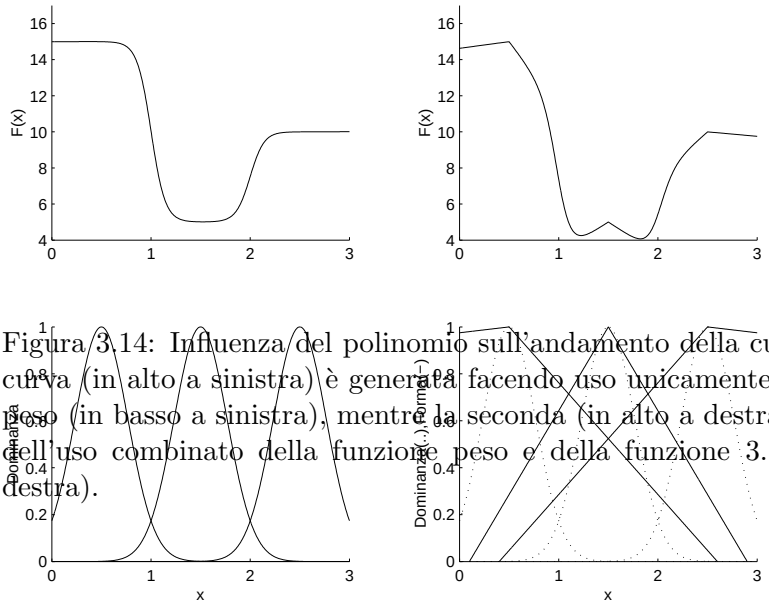


Figura 3.14: Influenza del polinomio sull'andamento della curva. La prima curva (in alto a sinistra) è generata facendo uso unicamente della funzione peso (in basso a sinistra), mentre la seconda (in alto a destra) è il risultato dell'uso combinato della funzione peso e della funzione 3.11 (in basso a destra).

Cleveland fa notare che un cambiamento, ad esempio, da un grado 1 ad un grado 2 può significare un cambiamento sostanziale dei risultati, per cui consiglia considerare la famiglia dei polinomi misti, nei quali il grado non è più un numero naturale, ma un numero reale positivo. Egli definisce polinomio misto di grado reale positivo non intero $p = m + c$, dove m è un intero e $0 < c < 1$, la media pesata dei polinomi di grado m e $m + 1$, con peso $1 - c$ per il primo e c per il secondo.

3.4.1.4 Evoluzione: la funzione di forma

In base a quanto visto finora, per migliorare la precisione dell'approssimazione, integriamo nel modello coarticolatorio una funzione che riprende il concetto espresso da Cleveland. Vogliamo che tale funzione esprima il concetto di polinomio di grado reale e che allo stesso tempo sia il più semplice possibile. L'espressione che utilizzeremo sarà quindi del tipo

$$Y(x) = a\ |x|^p + 1 \tag{3.12}$$

con p reale positivo.

Per il nostro modello in particolare utilizzeremo una forma che caratterizza in modo diverso l'andamento del polinomio secondo la direzione di estensione della coarticolazione:

$$S_{TL}(\tau) = \begin{cases} \beta_{bw}\,|\tau|^{p_{bw}} + 1 & \text{se } \tau < 0 \\ \beta_{fw}\,|\tau|^{p_{fw}} + 1 & \text{se } \tau > 0 \end{cases} \tag{3.13}$$

dove TL sta per "Time-Locked", in quanto la sua caratteristica è indipendente dal posizionamento dei fonemi precedenti e successivi, analogamente a ciò che accade nel modello di Bell-Berti e Harris [8].

Chiameremo "Funzione di Forma" tale espressione, perché il suo effetto principale è quello di modellare l'andamento del target articolatorio in prossimità del suo massimo rilievo. Avremo quindi un target articolatorio non più discreto, ma che si evolve nel tempo con una propria caratteristica.

Si può anche pensare ad una forma di tipo "Look-Ahead" in cui l'influenza della funzione di forma sia proporzionale alla distanza con il target successivo o antecedente. Possiamo allora scrivere

$$S_{LA}(\tau) = \begin{cases} \beta_{bw} \left|\frac{\tau}{h_{bw}}\right|^{p_{bw}} + 1 & \text{se } \tau < 0 \\ \beta_{fw} \left|\frac{\tau}{h_{fw}}\right|^{p_{fw}} + 1 & \text{se } \tau > 0 \end{cases} \tag{3.14}$$

in cui h_{bw} e h_{fw} rappresentano fattori proporzionali alla distanza dai target precedenti o successivi.

L'utilizzo delle funzione di forma risulta utile nel riprodurre andamenti con caratteristiche particolari come ad esempio la pendenza rilevata nella produzione della vocale /u/ in alcuni contesti consonantici (figura 3.15).

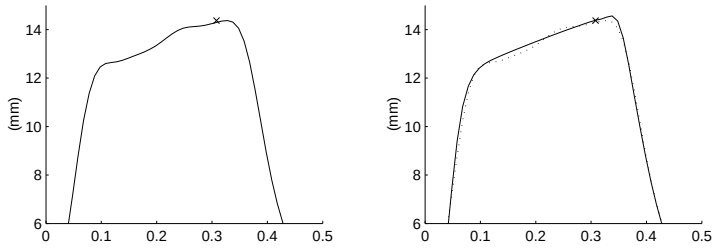
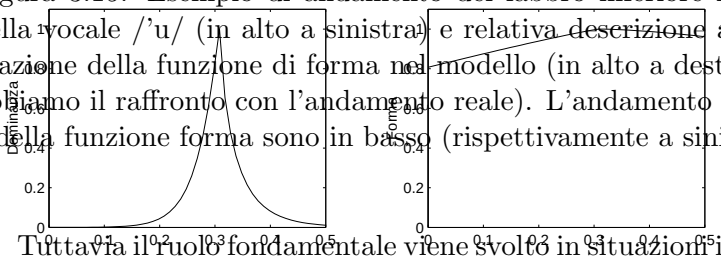


Figura 3.15: Esempio di andamento del labbro inferiore nella produzione della vocale /'u/ (in alto a sinistra) e relativa descrizione attraverso l'integrazione della funzione di forma nel modello (in alto a destra, in tratteggio abbiamo il raffronto con l'andamento reale). L'andamento della dominanza e della funzione forma sono in basso (rispettivamente a sinistra e destra).



Tuttavia il ruolo fondamentale viene svolto in situazioni in cui si necessita di un rapido smorzamento come, ad esempio, nel caso del rilascio del gesto alla fine di una frase (figura 3.16). In questo caso particolare senza l'utilizzo di tale funzione non sarebbe possibile in alcun modo riprodurne l'andamento.

Tra le due funzioni mostrate la scelta per il nostro modello ricade su $S_{LA}(\tau)$ perché da prove empiriche su dati reali abbiamo rilevato un comportamento migliore rispetto a $S_{TL}(\tau)$.

3.4.1.5 Evoluzione: la resistenza temporale

Per ovviare al problema della perdita del target consonantico estendiamo temporalmente il concetto di resistenza coarticolatoria prima espresso tramite la variazione dell'ampiezza della funzione di dominanza. Vogliamo avere la possibilità di bloccare il gesto articolatorio precedente e successivo di modo che la loro influenza si annulli cosicché il target possa essere raggiunto. Per ottenere ciò, affianchiamo alla dominanza una funzione, che chiameremo di "Resistenza Temporale", con un'andamento simile alla dominanza, ma con estensione variabile in base alla collocazione dei fonemi precedenti o successivi ed al loro grado di resistenza. Dopo alcune prove su

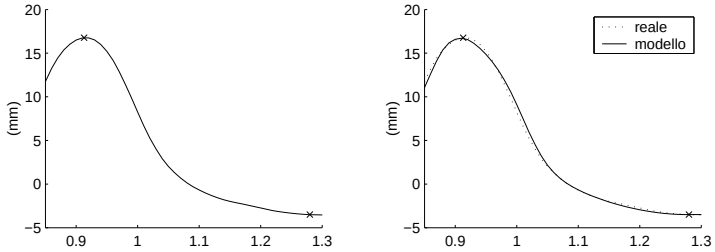


Figura 3.16: Esempio di rilascio del gesto vocalico (in alto a destra) e descrizione tramite il modello coarticolatorio (in alto a sinistra). Sotto sono rappresentate le dominanza e la funzione forma utilizzate; è stata rappresentata anche la funzione di forma del secondo target perché gioca un ruolo importante per l'andamento finale.

dati reali abbiamo scelto una funzione con andamento esponenziale del tipo

$$R(\tau) = \begin{cases} e^{-6\left|\frac{\tau}{h_{bw}}\right|^4} & \text{se } \tau < 0 \\ e^{-6\left|\frac{\tau}{h_{fw}}\right|^4} & \text{se } \tau > 0 \end{cases} \tag{3.15}$$

in cui h_{bw} e h_{fw} hanno un significato analogo a quelli della funzione forma.

Abbiamo associato ad ogni fonema un *coefficiente di resistenza* k_R attraverso il quale possiamo calcolare il valore di h_{bw} e h_{fw} con la seguente procedura ricorsiva (ci riferiamo al caso del calcolo di h_{fw}; h_{bw} si ottiene prendendo in considerazione i segmenti antecedenti al posto di quelli successivi):

Passo 1 Se Il coefficiente di resistenza k_R del segmento successivo è pari a 1, allora h_{fw} è pari alla distanza tra il target attuale e quello successivo, altrimenti si salta al passo 2.

Passo 2 h_{fw} è uguale alla distanza tra il target attuale e quello successivo sommato al valore di h_{fw} del segmento successivo calcolato con la medesima procedura.

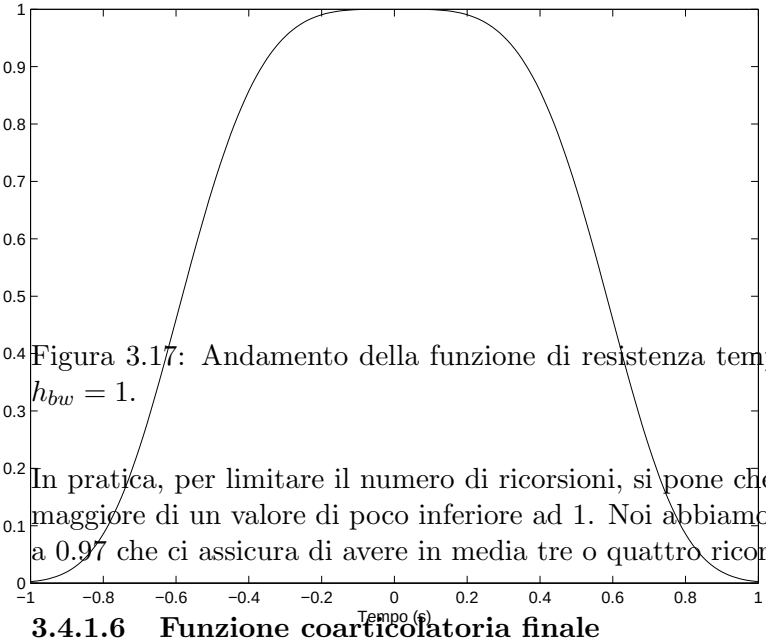


Figura 3.17: Andamento della funzione di resistenza temporale per $h_{fw} = h_{bw} = 1$.

In pratica, per limitare il numero di ricorsioni, si pone che k_R debba essere maggiore di un valore di poco inferiore ad 1. Noi abbiamo posto tale limite a 0.97 che ci assicura di avere in media tre o quattro ricorsioni.

3.4.1.6 Funzione coarticolatoria finale

Riassumiamo, infine le modifiche apportate precedentemente in un'unica espressione. La funzione coarticolatoria da noi utilizzata è quindi

$$F(t) = \frac{\sum_{i=1}^{N} T_i S_{LA_i}(t-t_i) R_i(t-t_i) D_i(t-t_i)}{\sum_{i=1}^{N} R_i(t-t_i) D_i(t-t_i)} \tag{3.16}$$

Il termine $S_{LA_i}(t-t_i)$ non compare a denominatore, analogamente a quanto accade nella (3.9), mentre essendo $R_i(t-t_i)$ in stretta relazione con la dominanza entrerà anch'esso a far parte del termine di normalizzazione.

Vogliamo far notare, infine, che con l'introduzione della funzione di resistenza temporale, il ruolo svolto dai coefficienti di attivazione e rilascio diventa fondamentale. Si pensi, ad esempio, a due fonemi successivi entrambi con resistenza unitaria. Solo una variazione consistente del coefficiente θ comporta variazioni significative nell'andamento finale. Ciò non vale per i valori di c_{bw} e c_{fw}. Possiamo allora capire che è agendo principalmente su di essi che riusciamo a diversificare le caratteristiche cinematiche in presenza di fonemi con alta resistenza.

3.4.2 Elaborazione dei dati

L'insieme di dati di partenza è costituito dalle registrazioni complete di tre soggetti che indichiamo con le stringhe AN,MA e PA. Ognuno di essi ha registrato circa mille frasi in lingua italiana (par. 3.3). Ogni frase è stata registrata cinque volte (cinque prove) per dare valenza statistica (non ci si può basare sulla prova singola). Ad ogni frase è associata una coppia di file che rappresenta la registrazione audio e i relativi movimenti degli articolatori del parlante.

Su questi dati grezzi bisogna effettuare delle pre-elaborazioni: essi vanno prima segmentati perché per la stima dei coefficienti del modello coarticolatorio servono tutte le sequenze Vocale-Consonante-Vocale (VCV) presenti nelle frasi registrate. V rappresenta una tra le vocali /a, e, i, o, u/, in cui la prima vocale è tonica, e C è uno tra i 21 fonemi consonantici italiani definiti dal sistema SAMPA [2]. É inoltre necessaria una normalizzazione temporale e di ampiezza per poter confrontare le differenti prove.

Da una prima analisi degli andamenti cinematici dei gesti articolatori si riscontra un'estrema variabilità tra soggetti diversi e anche tra prove relative ad un unico soggetto. Alcuni studi evidenziano, ad esempio, che in alcune persone l'apertura labiale e quella mandibolare possono variare fino al 50% rispetto alla media. La variabilità articolatoria, indipendentemente dal contesto linguistico, è data da una serie di fattori tra cui possiamo includere:

- **Speech rate:** la durata del segmento fonemico corrente e di quelli successivi e precedenti varia in funzione della velocità del parlato e del contesto fonemico ed ha conseguenze fondamentali sull'andamento articolatorio. A speech rate alti in particolar modo, i gesti hanno un'alta sovrapposizione che comporta una variabilità maggiore e una grado minore di articolazione.

- **Intensità sonora:** vi è una stretta relazione tra l'intensità sonora prodotta e la conformazione degli articolatori. Generalmente ad una maggiore intensità sonora corrisponde un'ampiezza maggiore dei gesti.

- **Precisione:** per precisione si intende il differente grado di articolazione conseguente all'intento di comprensibilità da parte del parlante; in una situazione di parlato naturale si riferisce al compromesso dinamico tra economia ed esigenze comunicative già espresso in 1.4.1.1.

- **Strategia articolatoria:** gli studi hanno individuato variabilità tra soggetto e soggetto per quanto riguarda la strategia articolatoria; ciò comporta ad esempio che alcuni soggetti si comportino secondo quanto

previsto dal modello look-ahead, mentre altri abbiano una strategia anticipatoria di tipo time-locked.

- **Variabilità prosodica:** l'influenza prosodica è il risultato dell'intento linguistico come l'intonazione. Essa ha un grosso rilievo a livello espressivo ed è difficile da modellare perché richiede una comprensione semantica di ciò che si esprime. Nell'ambito dei dati in nostro possesso non vi è influenza di tipo prosodico, nè è nostro intento esprimere nel modello questo tipo di variabilità.

3.4.2.1 Segmentazione

La prima operazione da effettuare sui dati forniti dal sistema di acquisizione (coppie sincronizzate di file audio/cinematica) riguarda la segmentazione del segnale acustico per l'identificazione dei fonemi. Questo passo è stato effettuato in prima istanza utilizzando un sistema automatico di riconoscimento con un'architettura ibrida basata su Hidden Markov Models (HMM) e Artificial Neural Networks (ANN) [29]. Il sistema, sviluppato dall'ISTC-CNR di Padova, è stato allenato sul database APASCI [4] contenete circa 2500 frasi lette da 100 soggetti, 50 maschili e 50 femminili. Esso è in grado di riconoscere i 38 principali fonemi italiani del sistema S.A.M.P.A. (Speech Assessment Methods Phonetic Alphabet) tra cui distingue anche vocali accentate da quelle non accentate, e vocali aperte da quelle chiuse [2].

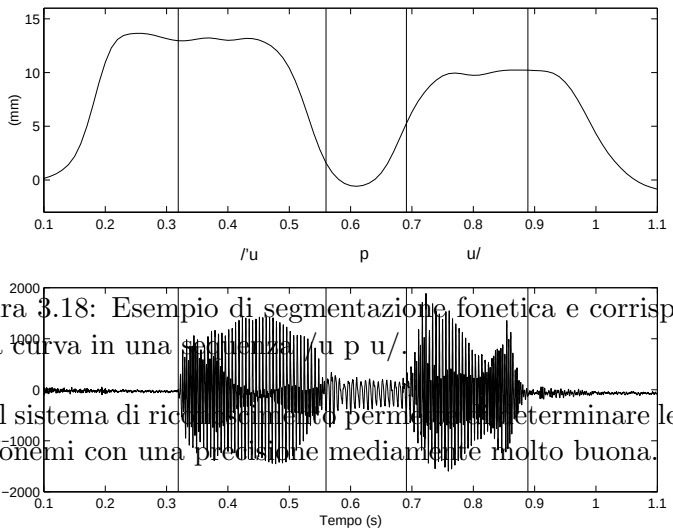


Figura 3.18: Esempio di segmentazione fonetica e corrispettivo andamento della curva in una sequenza /u p u/.

Il sistema di riconoscimento permette di determinare le durate temporali dei fonemi con una precisione mediamente molto buona.

I valori della segmentazione automatica sono controllati ed eventualmente corretti a mano con l'ausilio del software INTERFACE [105] un ambiente integrato Matlab© sviluppato ad hoc per rendere questo compito meno gravoso.

3.4.2.2 Normalizzazione delle ampiezze

Una volta segmentati i dati, ci si trova di fronte al problema di dover analizzare delle prove dislocate diversamente nello spazio per una variabilità da parte del soggetto stesso che effettua le prove in momenti temporali diversi.

Per superare questo ostacolo si effettuano preliminarmente delle acquisizioni in condizione di riposo che vengono utilizzate per normalizzare le ampiezze del resto dei dati acquisiti. In particolare si vuole ottenere per ogni parametro il valore medio di riposo m_k calcolato su tutto il periodo di acquisizione dove $k = 1, \ldots, 6$ rappresenta il parametro k-esimo.

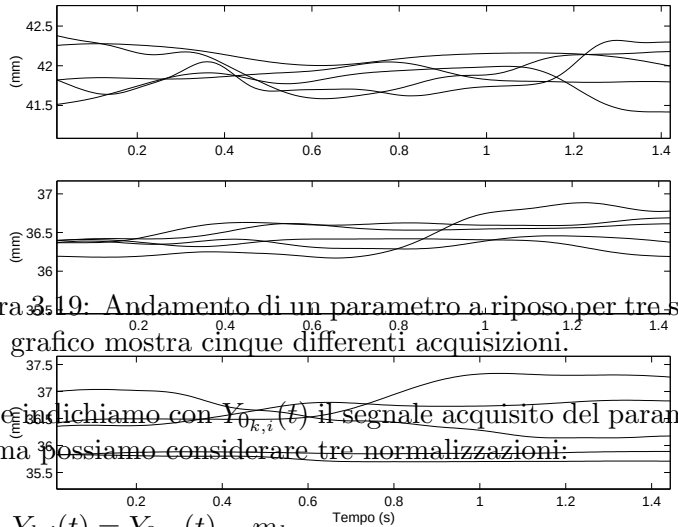


Figura 3.19: Andamento di un parametro a riposo per tre soggetti differenti. Ogni grafico mostra cinque differenti acquisizioni.

Se indichiamo con $Y_{0_{k,i}}(t)$ il segnale acquisito del parametro k alla prova i-esima possiamo considerare tre normalizzazioni:

1. $Y_{k,i}(t) = Y_{0_{k,i}}(t) - m_k$
2. $Y_{k,i}(t) = \frac{Y_{0_{k,i}}(t)}{m_k}$
3. $Y_{k,i}(t) = \frac{\left(\frac{Y_{0_{k,i}}(t)}{m_k} - 1\right)}{Y_{MAX_k}}$

La prima consiste nella sottrazione della posizione a riposo; le altre due sono indici di variazione del movimento dalla posizione di riposo i cui valori di partenza sono rispettivamente 1 e 0. L'ultimo in particolare rappresenta una caratteristica indipendente dalla morfologia del soggetto per la presenza del termine Y_{MAX_k} che indica il valore massimo che può assumere il dato parametro. Tuttavia la stima di tale valore non è semplice e si rivela poco pratica.

Abbiamo scelto la semplice sottrazione perché l'uso di valori spaziali ci permette di poter rapportare i risultati ottenuti per la collocazione dei target con altri studi statistici per l'italiano in particolare con quelli effettuati da Magno-Caldognetto e Zmarich [69].

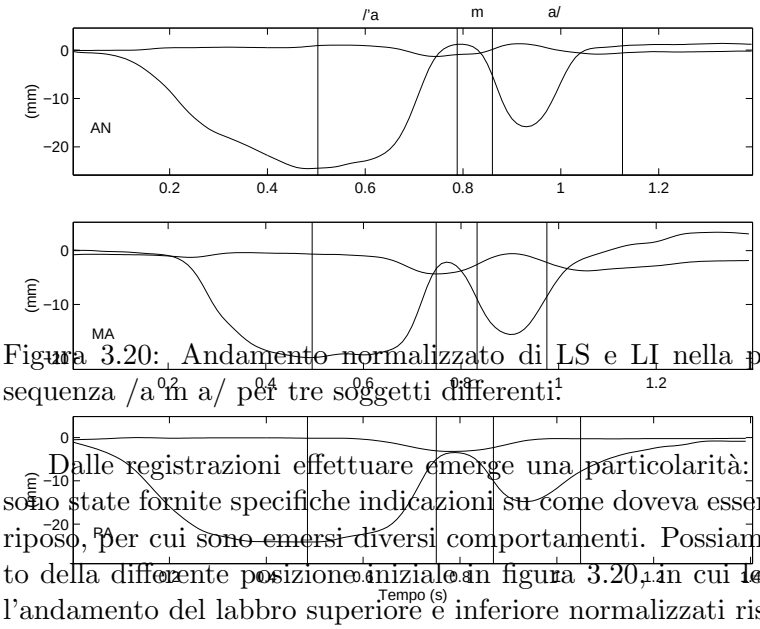


Figura 3.20: Andamento normalizzato di LS e LI nella produzione della sequenza /a m a/ per tre soggetti differenti.

Dalle registrazioni effettuare emerge una particolarità: ai soggetti non sono state fornite specifiche indicazioni su come doveva essere la posizione a riposo, per cui sono emersi diversi comportamenti. Possiamo vedere l'effetto della differente posizione iniziale in figura 3.20, in cui le curve indicano l'andamento del labbro superiore e inferiore normalizzati rispetto alla posizione di riposo in una acquisizione della sequenza /a m a/. Ricordiamo che i marcatori sono stati collocati sul bordo esterno delle labbra, per cui una sovrapposizione nelle ampiezze delle due curve normalizzate corrisponde ad una compressione labiale, ma solo nel caso in cui le labbra nella posizione di riposo siano chiuse.

Per i primi due soggetti è evidente la sovrapposizione del picco centrale

corrispondente alla massima chiusura durante la produzione del fonema /m/. Guardando la parte finale delle curve notiamo che presenta anch'essa una zona di sovrapposizione. Possiamo fare quindi le seguenti considerazioni:

- La zona finale nel caso del primo soggetto (in alto) presenta un andamento costante. Se le labbra fossero compresse sicuramente non avremo una tale caratteristica, perché i muscoli non riescono a controllare la compressione con una tale precisione. La situazione farebbe pensare piuttosto a delle labbra solamente chiuse. Inferiamo che il soggetto ha una posizione iniziale a labbra semiaperte e finale a labbra chiuse;

- L'andamento della parte finale del secondo soggetto (al centro) presenta una caratteristica tipica da compressione, ma l'entità della stessa è talmente elevata da risultare innaturale. In questo caso la posizione iniziale è a labbra semiaperte, posizione finale a labbra chiuse compresse;

- Il terzo soggetto (in basso) presenta la situazione più realistica in cui visivamente le curve si incontrano senza incrociarsi nel punto di massima chiusura e non vi è sovrapposizione nè nella parte iniziale e finale. Posizione iniziale e finale a labbra chiuse.

Nel calcolo finale delle posizioni base degli articolatori bisogna prendere in considerazione questa variabilità ed effettuare le opportune correzioni.

3.4.2.3 Normalizzazione temporale

Il primo problema che bisogna affrontare nel raffrontare le curve cinematiche consiste nel fatto che l'inizio dell'acquisizione dei dati non coincide con l'inizio del movimento fono-articolatorio di interesse e quindi quest'ultimo si trova posizionato ad istanti diversi all'interno dell'intervallo di acquisizione. Si rende necessario quindi un riallineamento dei dati.

Un altro problema è rappresentato dalle diverse durate delle prove. Una tra le più comuni semplificazioni consiste nel restringere le analisi a caratteristiche fondamentali che riassumono il movimento delle curve, come le durate ed i relativi valori spaziali dei movimenti di apertura e chiusura, che poi vengono sottoposti ad analisi statistiche. Nel nostro caso tuttavia abbiamo bisogno di analizzare la curva nella sua interezza e di conseguenza abbiamo bisogno di una trasformazione temporale che ci permetta di rapportare le prove di diversa durata. Tre tipi di metodologie possono essere seguite:

- **Normalizzazione lineare:** si tratta di stabilire inizio e fine del gesto di ogni prova ed applicare una trasformazione lineare per rapportarle

ad una durata comune. Tale procedimento porta a scarsi risultati perché non considera le variabilità interne alla prova stessa.

- **Normalizzazione lineare a tratti:** consiste nell'identificare dei momenti significativi all'interno delle prove, come ad esempio i massimi e minimi relativi al gesto di apertura e chiusura labiale oppure l'attraversamento dell'asse delle ascisse (*zero crossing*), e poi allinearli tramite una trasformazione lineare. L'inconveniente di questo metodo è che si deve decidere a priori quali sono i punti di riferimento di interesse e può capitare che in alcune delle curve tali punti manchino.

- **Non lineare:** Lucero et al. [64] propongono di scalare le forme d'onda dei movimenti tramite delle funzioni di *warping* per deformarle in modo non lineare. Il loro obiettivo è quello di preservare il più possibile le caratteristiche di riferimento degli andamenti, e usano come punto di partenza le accelerazioni per trovare la media delle curve. Il metodo è molto interessante e presenta buoni risultati. Tuttavia non è chiaro il motivo per cui i picchi nelle accelerazioni siano da considerarsi importanti aspetti da preservare, magari a scapito di altre particolarità che vengono modificate dalla funzione di warping. Probabilmente in un modello di tipo muscolare si può pensare ad una forte corrispondenza tra accelerazione e attività muscolare.

Il metodo di normalizzazione scelto è quello lineare a tratti; i valori della segmentazione fonetica sono utilizzati come punti di riferimento. Ciò ha il vantaggio di mantenere intatti gli andamenti caratteristici all'interno del segmento fonetico e lo svantaggio di introdurre deformazioni nell'istante di passaggio tra un segmento ed i successivo. Esse tuttavia risultano trascurabili in quanto le compressioni o espansioni si ripercuotono sui segmenti in misura simile.

Per ricavare l'andamento medio e la deviazione standard per ogni singola sequenza, i dati normalizzati devono essere ricampionati. La scelta del metodo di ricampionamento non comporta particolari difficoltà in quanto l'errore finale è simile sia con filtri interpolatori che con spline bicubiche o con la sintesi di Fourier [111]. Noi abbiamo scelto le spline bicubiche. L'unico problema riscontrato consiste in distorsioni nei punti di discontinuità all'inizio e alla fine del segnale; esso può essere facilmente aggirato introducendo dati contestuali all'inizio e alla fine che vengono rimossi una volta effettuato il ricampionamento.

3.4.3 Risultati

Con i dati segmentati e normalizzati possiamo determinare il modello personalizzato con i seguenti passi:

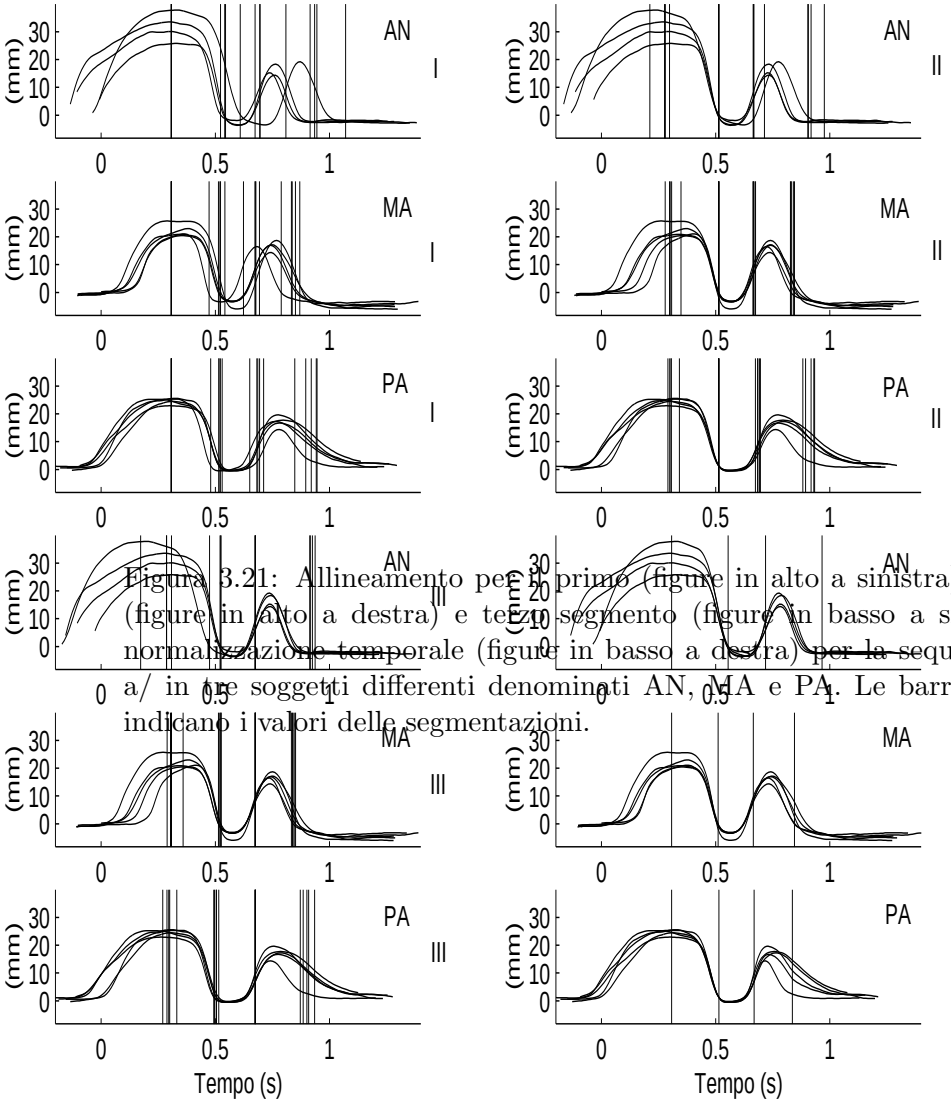


Figura 3.21: Allineamento per il primo (figure in alto a sinistra), secondo (figure in alto a destra) e terzo segmento (figure in basso a sinistra), e normalizzazione temporale (figure in basso a destra) per la sequenza /a b a/ in tre soggetti differenti denominati AN, MA e PA. Le barre verticali indicano i valori delle segmentazioni.

- Determinazione dei target articolatori e identificazione dei visemi[1]
- Caratterizzazione dei target all'interno del segmento fonemico
- Calcolo dei coefficienti del modello

Nella prossima sezione analizzeremo i target articolatori, che verranno successivamente utilizzati per definire i visemi. Verificheremo quindi se i fonemi cui è stato associato uno stesso visema hanno un andamento cinematico simile (sezione 3.4.3.2). Nel caso in cui ciò si verifichi, potremo in seguito caratterizzare ogni visema attraverso un insieme di *valori base*. La differenziazione dei singoli fonemi si avrà tramite la modifica di alcuni dei valori del visema di appartenenza. Questo tipo di approccio ha il vantaggio di poter adattare il livello di dettaglio dei movimenti in base alle esigenze qualitative della faccia parlante.

Nella paragrafo 3.4.3.3 mostriamo i risultati ottenuti dall'ottimizzazione dei coefficienti sulle singole curve e sulle curve medie. è opportuno mettere in evidenza che il processo di ottimizzazione riguarda anche la collocazione dei target. Ciò risulta determinante nelle situazioni in cui non possiamo far corrispondere il target ad un massimo o ad un minimo dell'andamento del parametro in considerazione (e.g. nel caso di dominanza bassa), oppure quando il parametro mantiene il suo valore per un certo tempo, per cui non è chiaro ove posizionare il target all'interno di questo intervallo temporale.

La precisione nella determinazione dell'andamento cinematico dei parametri assume particolare rilievo in quelle consonanti per cui le labbra costituiscono l'articolatore primario. Più precisamente, nelle consonanti bilabiali una minima variazione delle caratteristiche può implicare diversità visuali significative, costituite dalla maggiore o minore compressione delle labbra. Per questo motivo daremo maggior rilievo ai risultati relativi alle consonanti bilabiali e labiodentali.

3.4.3.1 Variabilità dei target articolatori

Nell'analisi dei diversi contesti si è cercato innanzitutto di capire se si poteva definire un target consonantico che fosse indipendente dalla vocale, ottenendo la variazione della posizione finale dell'articolatore durante la produzione della consonante come diretta conseguenza delle caratteristiche del gesto articolatorio vocalico descritto tramite il nostro modello. Tale soluzione presenta grosse difficoltà che emergono immediatamente esaminando le consonanti bilabiali e labiodentali (figure 3.22 e 3.23). Notiamo infatti che per il

[1] Ricordiamo che il visema costituisce l'elemento di base visivo della conformazione articolatoria, analogamente a quanto accade per i fonemi in ambito acustico. Generalmente ci sono meno visemi che fonemi, in quanto suoni differenti possono avere una simile conformazione degli articolatori visibili.

labbro inferiore ad un abbassamento del target per la produzione di differenti vocali corrisponde generalmente un innalzamento del target consonantico. Si deve poi ricordare che nel caso delle bilabiali il target consonantico è in stretta relazione con il target del labbro superiore in modo da poter effettuare la compressione delle labbra. Nel nostro modello l'unico modo per avere tali caratteristiche è considerare target differenti per i diversi contesti. Gli studi effettuati da Löfqvist e Gracco [63] su sequenze /'a p V/ e /'a b V/ in cui V è una delle vocali /a, i, u/, fanno presupporre che la collocazione della consonante sia variabile e dipende fortemente dal soggetto considerato. Nel nostro caso adottiamo l'ipotesi semplificativa per cui l'ampiezza T e la posizione temporale t della consonante tra le due vocali V_1e V_2 sia data dalla media delle posizioni dei corrispettivi contesti simmetrici:

$$t_{V_1CV_2} = \frac{t_{V_1CV_1} + t_{V_2CV_2}}{2} \tag{3.17}$$

$$T_{V_1CV_2} = \frac{T_{V_1CV_1} + T_{V_2CV_2}}{2} \tag{3.18}$$

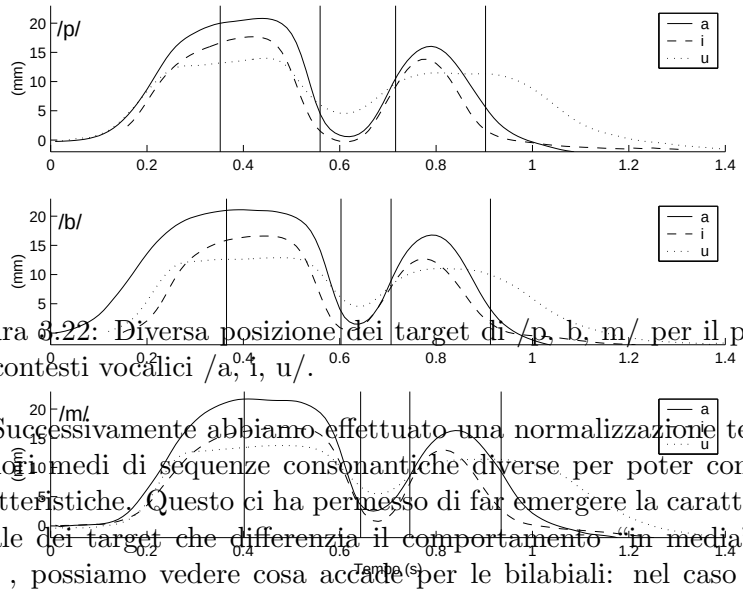


Figura 3.22: Diversa posizione dei target di /p, b, m/ per il parametro LI nei contesti vocalici /a, i, u/.

Successivamente abbiamo effettuato una normalizzazione temporale tra i valori medi di sequenze consonantiche diverse per poter confrontarne le caratteristiche. Questo ci ha permesso di far emergere la caratteristica temporale dei target che differenzia il comportamento "in media". In figura 3.24 , possiamo vedere cosa accade per le bilabiali: nel caso della vocale

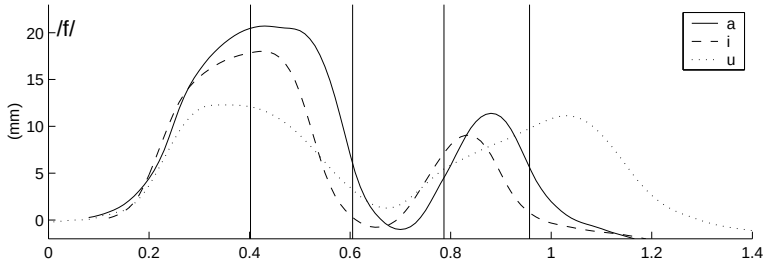
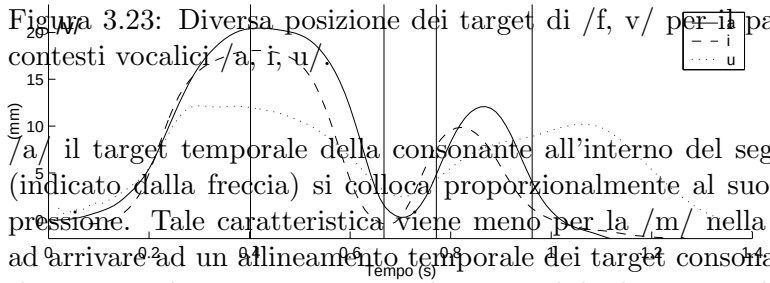


Figura 3.23: Diversa posizione dei target di /f, v/ per il parametro LI nei contesti vocalici /a, i, u/.



/a/ il target temporale della consonante all'interno del segmento fonetico (indicato dalla freccia) si colloca proporzionalmente al suo grado di compressione. Tale caratteristica viene meno per la /m/ nella vocale /i/ fino ad arrivare ad un allineamento temporale dei target consonantici nella /u/. Il motivo può essere imputato ad un possibile diverso grado di costrizione esercitato dalle vocali sulla consonante che aumenta in relazione al grado di arrotondamento.

I dati che utilizzeremo per la collocazione temporale dei target di ogni specifica consonante o vocale nei diversi contesti saranno espressi in relazione alla durata del segmento fonetico ($t_{end} - t_{start}$) secondo la relazione

$$t_i = t_{start_i} + k_i(t_{end_i} - t_{start_i}) \qquad 0 < k_i < 1 \tag{3.19}$$

dove t_{start_i} e t_{end_i}rappresentano l'inizio e la fine del segmento fonetico i-esimo, mentre k_i indica l'intervallo normalizzato, ricavato dai dati medi, che intercorre tra l'inizio del segmento fonetico ed il valore temporale t_i del target.

3.4.3.2 Determinazione dei visemi

Un confronto tra le consonanti per il parametro LI appare in figura 3.25. Esse sono state classificate in cinque gruppi cui faremo riferimento per definire i visemi. La determinazione dei visemi è importante perché ci permette

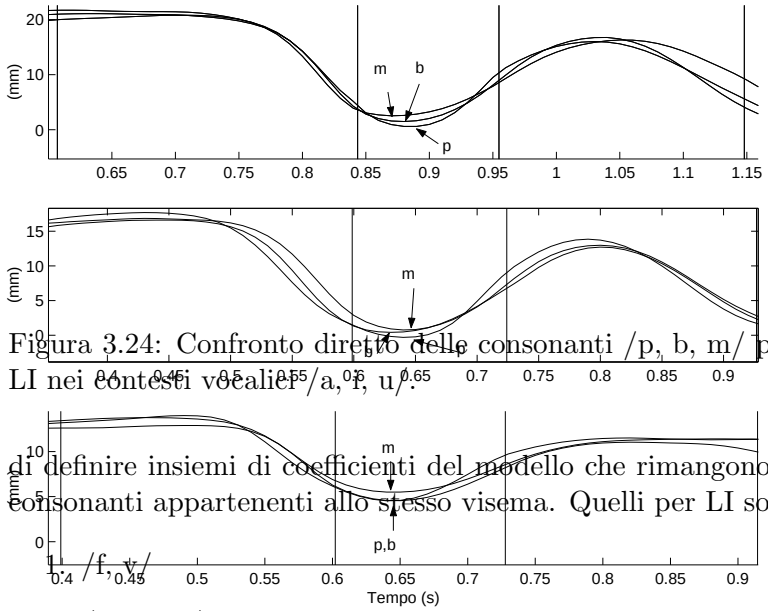


Figura 3.24: Confronto diretto delle consonanti /p, b, m/ per il parametro LI nei contesti vocalici /a, i, u/.

di definire insiemi di coefficienti del modello che rimangono costanti per le consonanti appartenenti allo stesso visema. Quelli per LI sono quindi:

1. /f, v/
2. /p, b, m/
3. /z, s, ts, dz, t, d/
4. /tS, dZ, N/
5. /L, S, g, k, n, r, l/

I risultati sono in accordo con gli analoghi gruppi visemici per l'apertura labiale ottenuti negli studi statistici effettuati da Magno-Caldognetto e Zmarich [69] tranne per il fonema /L/ (evidenziato in figura). Da un confronto con gli altri due soggetti appare che l'ampiezza dei target per questa consonante si discosta significativamente dalle altre prove. Verrà quindi considerata a parte e i suoi valori articolatori verranno fatti corrispondere al fonema cui dovrebbe assomigliare maggiormente, ossia a /N/.

Per gli altri parametri risulta più difficile determinare graficamente i visemi come abbiamo fatto per LI a causa del minore dispiegamento delle labbra e della maggiore variabilità del segnale. Ci rifacciamo allora agli studi statistici già citati in precedenza da cui abbiamo potuto ricavare i seguenti visemi consonantici finali:

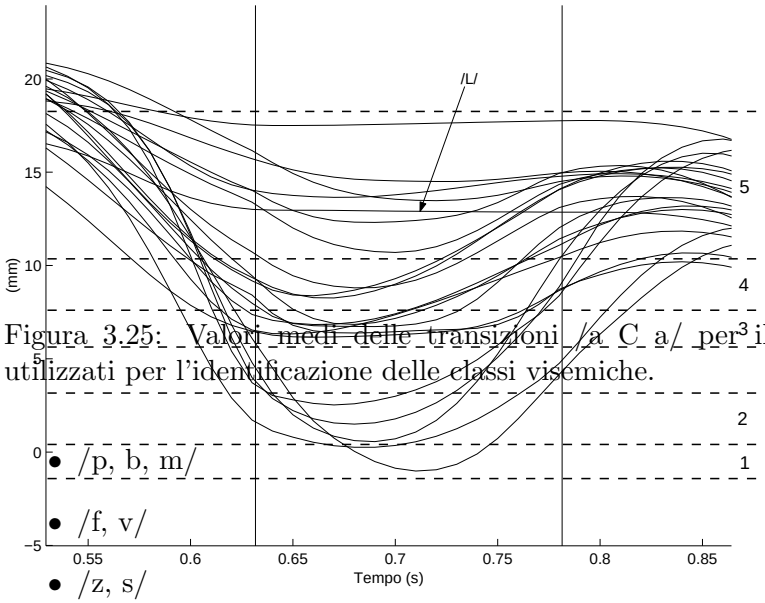


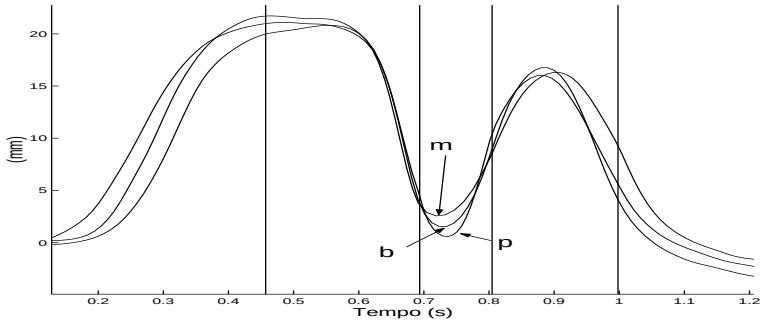
Figura 3.25: Valori medi delle transizioni /a C a/ per il parametro LI utilizzati per l'identificazione delle classi visemiche.

- /p, b, m/
- /f, v/
- /z, s/
- /ts, dz/
- /t, d/
- /tS, dZ/
- /L, N/
- /S/
- /g, k/
- /l, n, r/

Possiamo ottenere una conferma della validità dei visemi definiti considerando le caratteristiche delle curve separatamente per ogni gruppo (figure 3.26 e 3.27). Si evidenzia quindi un andamento cinematico molto simile per i gruppi separati con l'eccezione delle consonanti /t, d/. La diversità è stata rilevata anche negli altri soggetti. Dai grafici, inoltre, risulta che /t/ ha una caratteristica somigliante a /ts, dz/, mentre /d/ è analoga a /tS, dZ/.

Un altro aspetto importante da notare è che per consonanti appartenenti allo stesso gruppo visemico corrisponde lo stesso grado di ampiezza e durata del gesto di apertura. Possiamo quindi affermare che esiste un effetto coarticolatorio della consonante sulla successiva vocale, che possiamo distinguere in base al visema di appartenenza.

I fonemi /S, g, k, l, n, r/ non compaiono nei grafici in quanto sulla loro caratteristica prevale il comportamento del contesto vocalico, e per cui presentano una conformazione incostante, tale da avere difficilmente un riscontro visivo sulla validità dei visemi cui appartengono.



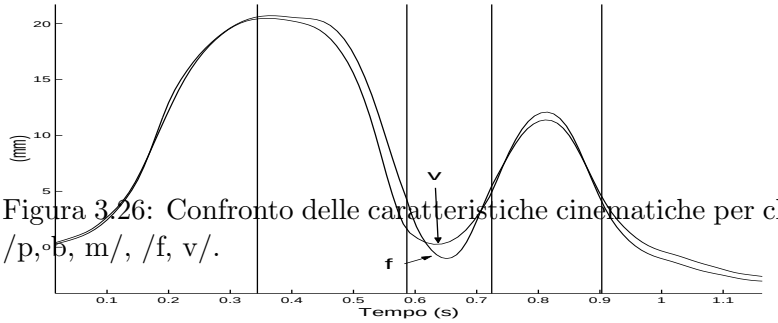
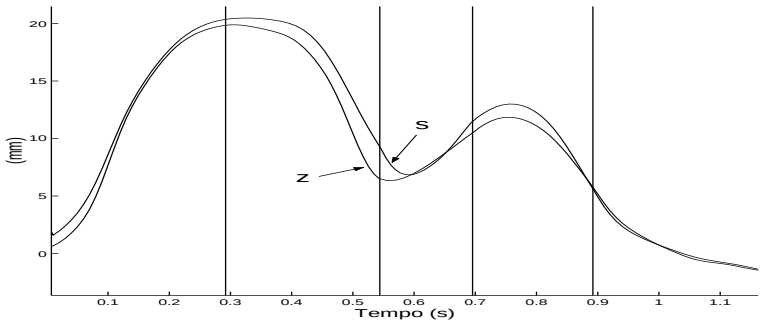


Figura 3.26: Confronto delle caratteristiche cinematiche per classi visemiche /p,ᵒb, m/, /f, v/.



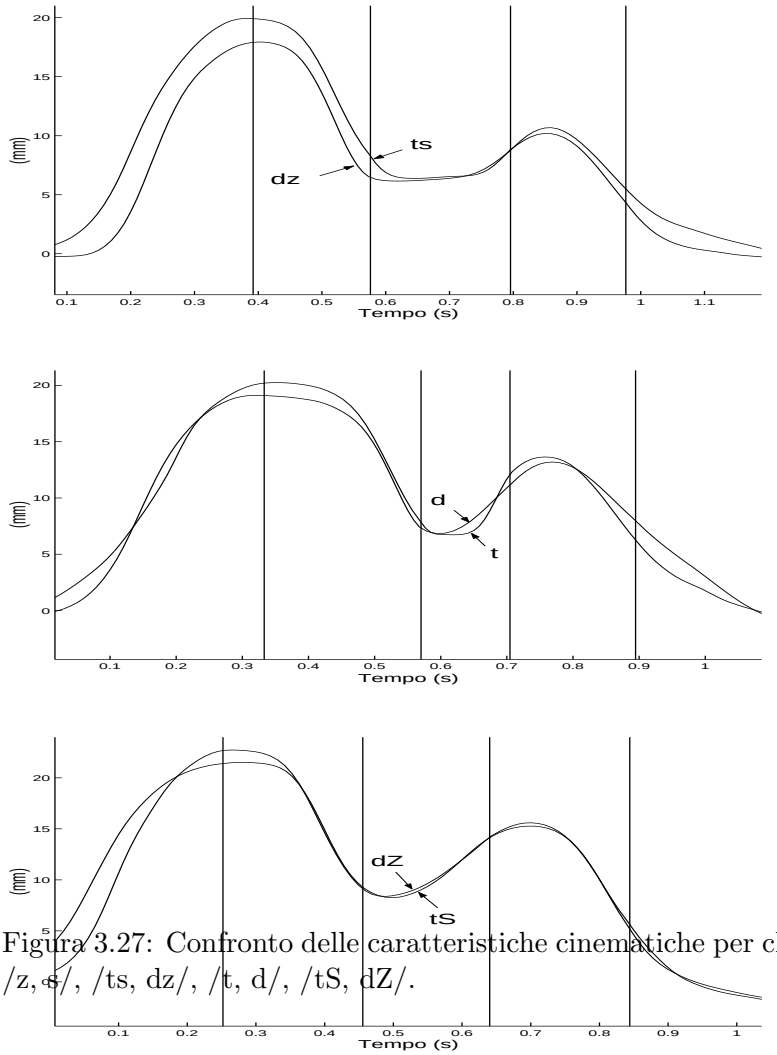


Figura 3.27: Confronto delle caratteristiche cinematiche per classi visemiche /z, s/, /ts, dz/, /t, d/, /tS, dZ/.

3.4.3.3 Calcolo dei coefficienti del modello

Il calcolo delle caratteristiche dinamiche (i coefficienti del modello) viene effettuato minimizzando l'errore quadratico, ossia minimizzando la funzione costo:

$$Z(\mathbf{x}) = \sum_{n=1}^{N} (Y_k(n) - F(n, \mathbf{x}))^2 \tag{3.20}$$

in cui k rappresenta il numero della prova, N il numero totale dei campioni all'interno della prova, $Y_k(n)$ il valore dell'n-esimo campione acquisito nella

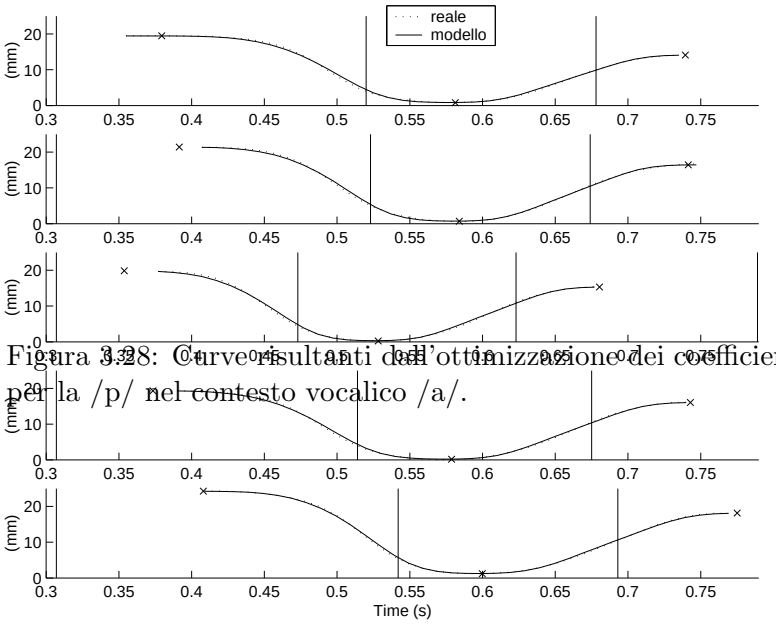


Figura 3.28: Curve risultanti dall'ottimizzazione dei coefficienti del modello per la /p/ nel contesto vocalico /a/.

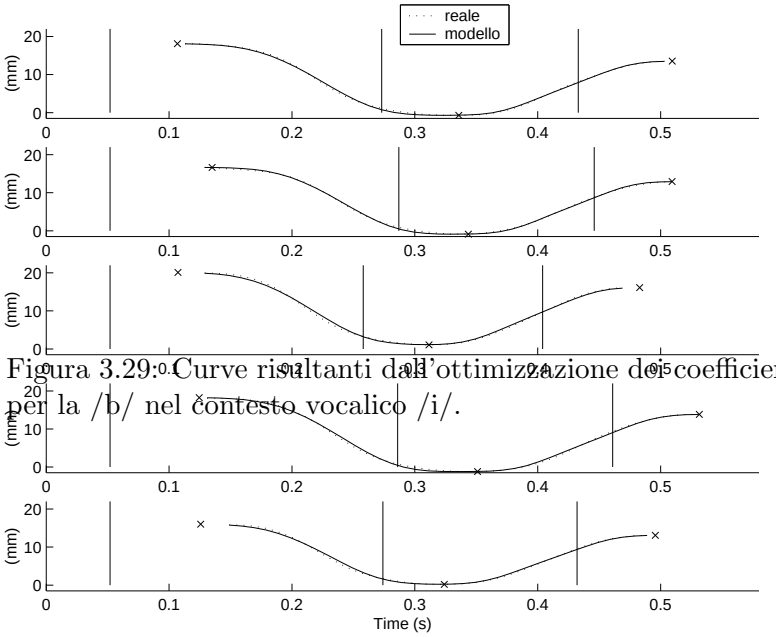


Figura 3.29: Curve risultanti dall'ottimizzazione dei coefficienti del modello per la /b/ nel contesto vocalico /i/.

k-esima prova, $\mathbf{x}$ il vettore dei coefficienti per l'intero insieme di fonemi coinvolto: se siamo in presenza di tre fonemi, $\mathbf{x}$ è costituito da tre sequenze di coefficienti ordinati come in tabella 3.1.

Il calcolo dei coefficienti viene fatto a passi successivi che combinano analisi manuali e tecniche automatiche di ottimizzazione. Non è possibile trattare i dati in modo completamente automatico in quanto per il modello da noi utilizzato, a differenza di quanto accade nel lavoro di Le Goff [42], la funzione costo presenta più minimi globali e deve essere guidata verso gli opportuni valori. Si sono riscontrati, poi, casi di minimi locali indesiderati verso cui l'algoritmo converge e a cui si è potuto rimediare attraverso una correzione dei valori di partenza.

Particolare attenzione è stata rivolta alla selezione del metodo di ottimizzazione perché il numero di parametri in gioco è molto alto. Per avere un algoritmo che avesse una convergenza verso il minimo in poche iterazioni è stato scelto un metodo di tipo *Trust Region* con approssimazione del passo di aggiornamento in un sottospazio a due dimensioni che contenga il cammino calcolato secondo il metodo *Dogleg.* Tale metodo ha infatti una forte convergenza e garantisce, nel nostro caso, una buona approssimazione del minimo già in 10-15 iterazioni. I dettagli di tale procedimento sono contenuti in [98].

N.	**Nome Coeff.**	**Descrizione**
1	t	Dislocazione del target all'interno del segmento
2	T	Valore del target
3	α	Ampiezza della funzione di dominanza
4	k	Coefficiente di resistenza
5	θ_{bw}	Estensione anticipatoria
6	θ_{fw}	Estensione posticipatoria
7	c_{bw}	Coefficiente di attivazione
8	c_{fw}	Coefficiente di rilascio
9	p_{bw}	Fattore di forma anticipatoria
10	p_{fw}	Fattore di forma perseverativa
11	β_{bw}	Pendenza forma anticipatoria
12	β_{fw}	Pendenza di forma posticipatoria

Tabella 3.1: Coefficienti del modello come compaiono nell'algoritmo di ottimizzazione.

Facendo riferimento ai coefficienti in tabella 3.1, il procedimento iterativo di calcolo può essere descritto nella seguente maniera:

- **Inizializzazione:** i valori iniziali possono essere scelti in due modi diversi

  - Se non si hanno valori di riferimento per le curve è consigliabile iniziare con valori bassi di β (e.g. $\beta_{fw} = \beta_{fw} = 0.025$) di modo che l'influenza della funzione forma sia minima. Ciò non vale nei casi di forte smorzamento della curva in cui è consigliabile alzare il valore di β a 1, e partire con un valore di p pari a 2. I valori di partenza di α, k, θ variano secondo il contesto consonantico. Più precisamente se siamo nel caso di consonanti con alta dominanza partiamo con α, k prossimi ad 1 mentre θ sarà alto (tra 15 e 30 nel nostro caso); se ci troviamo di fronte a consonanti con bassa dominanza, α e k sono vicini allo zero e θ è basso (tra 4 e 9). Per quanto riguarda il valore di c, generalmente è opportuno partire con un valore pari ad 1; nel caso di transizioni dirette e di bassa curvatura tra un target ed il successivo è meglio alzare tale valore a 2 facendo attenzione che anche il fonema contrapposto abbia un valore di c simile.
  - In caso si possiedano i valori dei coefficienti di una caratteristica analoga a quella che si cerca è meglio utilizzare quei valori come punto di partenza.

- **Prima ottimizzazione:** il numero di iterazioni di questo passo e del successivo può essere relativamente basso (e.g. 10).

- **Verifica ed eventuale correzione dei valori:** Non esiste un modo standard per modificare i valori: individuare quali sono i coefficienti che maggiormente influenzano l'andamento della curva e modificarli in modo coerente.

- **Seconda ottimizzazione**: l'obbiettivo della seconda ottimizzazione è quello di rifinire i valori così che l'errore scenda sotto una certa soglia. Le iterazioni dell'algoritmo in questo caso sono maggiori: $20 - 30$ iterazioni in genere raggiungono il valore desiderato.

Il procedimento di calcolo dei coefficienti è stato applicato ai gesti di chiusura e apertura della transizione VCV (figura 3.30). Nei casi più complessi i gesti sono stati considerati separatamente e poi i risultati sono stati verificati sull'andamento globale.

Sono stati caratterizzati con buona approssimazione anche il primo tratto, dalla posizione di riposo all'apertura della prima vocale, e quello finale con il rilascio del gesto fino alla posizione di riposo.

Nel calcolo dei coefficienti è utile considerare sia i valori medi, sia le prove singole. Infatti, da una parte i valori medi ci permettono di raggiungere rapidamente dei coefficienti adeguati, dall'altra le prove singole ci permettono di verificarli ed eventualmente migliorarli in un contesto che ha al suo interno un certo grado di variabilità.

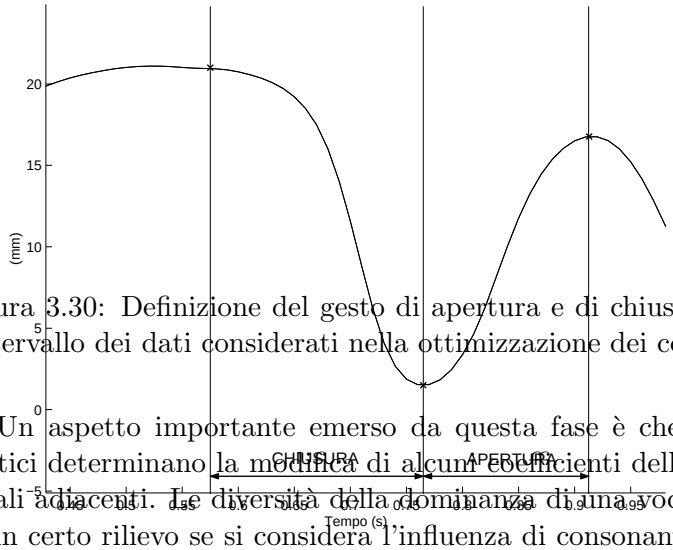


Figura 3.30: Definizione del gesto di apertura e di chiusura che determina l'intervallo dei dati considerati nella ottimizzazione dei coefficienti.

Un aspetto importante emerso da questa fase è che i contesti consonantici determinano la modifica di alcuni coefficienti della dominanza delle vocali adiacenti. Le diversità della dominanza di una vocale sono in genere di un certo rilievo se si considera l'influenza di consonanti associate a visemi differenti; sono invece minime se ci muoviamo all'interno di uno stesso visema.

Mostriamo innanzitutto come esempio una sequenza di cinque prove per il calcolo della caratteristica della /p/ nel contesto vocalico /a/ (figura 3.28) e della /b/ nel contesto vocalico /i/ (figura 3.29). Il modello utilizzato riesce a descrivere in maniera estremamente precisa le curve a tal punto da essere difficilmente distinguibile dall'andamento reale.

Vediamo ora il caso critico della vocale /u/ in cui l'ottimizzazione ci ha permesso di calcolare con precisione il valore temporale del target. Mostriamo due esempi di andamenti cinematici di sequenze /u p u/ (fig. 3.31) e /u b u/ (fig. 3.32) in cui possiamo vedere le due tipiche conformazioni della vocale: la prima con andamento centrale costante e la seconda con una pendenza, che varia in base all'ampiezza del movimento e alla durata del fonema. Notiamo innanzitutto come il modello riesca a caratterizzare bene i due comportamenti; tuttavia per descriverli devono essere utilizzati coefficienti differenti per la vocale e in parte anche per la consonante successiva. La diversità importante tra le due rappresentazioni sta nel posizionamento del target. Nella figura 3.31 esso si colloca in prossimità della zona di transizione tra la vocale e la consonante, mentre nella figura 3.31 si trova al centro del tratto costante. Il problema in questo caso è decidere quale delle due descrizioni rappresenti in modo adeguato la vocale in questione. La scelta

ricade sulla curva del contesto /u p u/ per i seguenti motivi:

- la caratteristica con pendenza iniziale nelle bilabiali è più frequente e ciò vale per tutti i tre soggetti considerati
- i valori della consonante successiva sono conformi a quelli ottenuti in altri contesti vocalici
- una caratteristica del tipo descritto nella figura in basso può anche essere ottenuta con la collocazione del target in prossimità della zona di transizione e con gli stessi coefficienti del caso sopra a patto che si annulli il fattore di forma della /u/, che serve a realizzare la pendenza della vocale

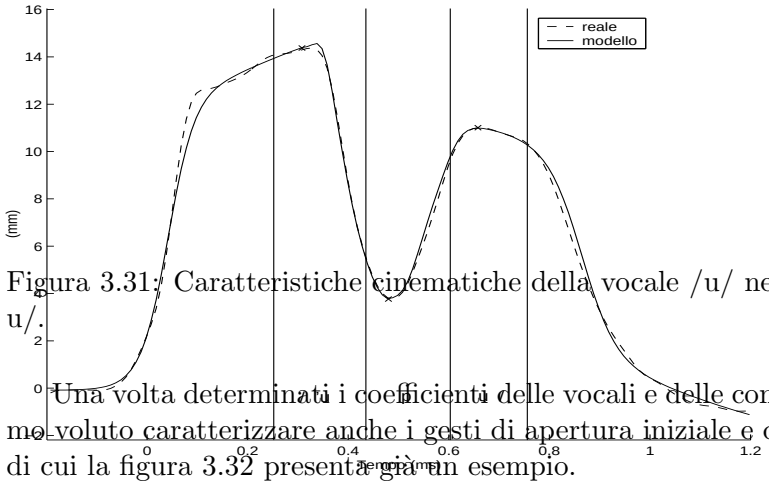


Figura 3.31: Caratteristiche cinematiche della vocale /u/ nel contesto /u p u/.

Una volta determinati i coefficienti delle vocali e delle consonanti, abbiamo voluto caratterizzare anche i gesti di apertura iniziale e di rilascio finale di cui la figura 3.32 presenta già un esempio.

L'errore medio per le consonanti bilabiali e labiodentali risulta inferiore in un contesto vocalico /a, i/ rispetto alla /u/, a causa della variabilità di conformazione di questa vocale.

In figura 3.33 si può osservare il movimento completo ottenuto per le consonanti /p, b, m/ nel contesto vocalico /a/ e la relativa tabella 3.2 mostra i valori dei coefficienti utilizzati. Possiamo rilevare alcune caratteristiche di queste consonanti: la dominanza è unitaria come ci si sarebbe aspettati ed il coefficiente k, anch'esso unitario dimostra che la consonante blocca l'estensione del gesto articolatorio vocalico anticipatorio e perseverativo; la caratteristica che contraddistingue le consonanti in questione sono i valori

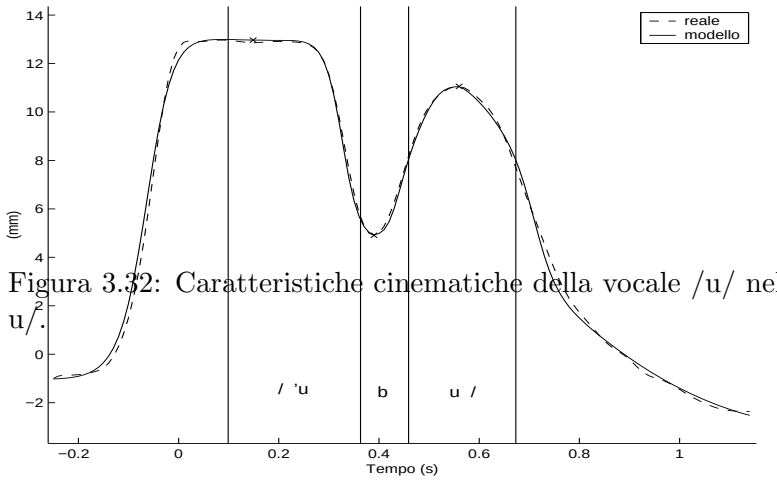


Figura 3.32: Caratteristiche cinematiche della vocale /u/ nel contesto /u b u/.

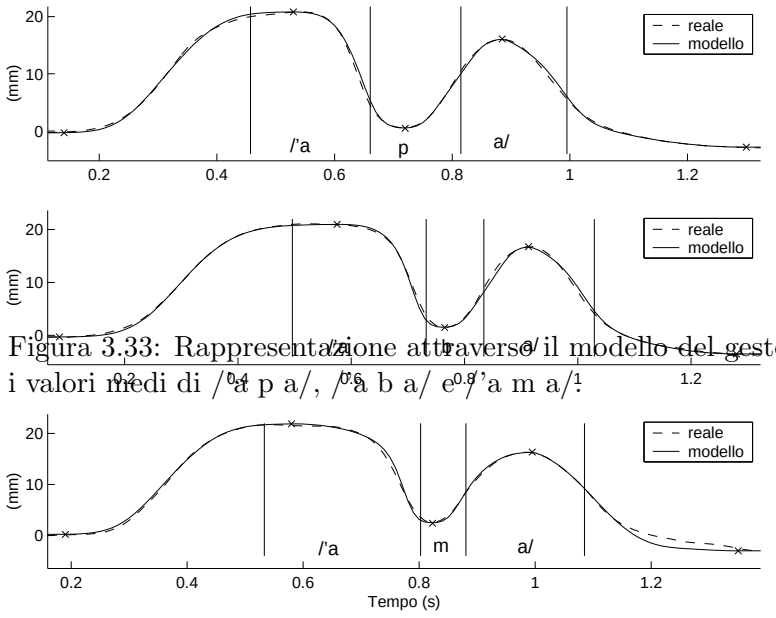


Figura 3.33: Rappresentazione attraverso il modello del gesto completo per i valori medi di /'a p a/, /'a b a/ e /'a m a/.

di attivazione e di rilascio (coefficienti c_{bw} e c_{fw}), mentre i rimanenti valori rimangono identici. La funzione di dominanza relativa alle tre consonanti compare in figura 3.34.

N.	**Nome Coeff.**	/p/	/b/	/m/
1	t	0.330	0.260	0.140
2	T	-2.000	-1.000	-0.200
3	α	1.000	1.000	1.000
4	k	1.000	1.000	1.000
5	θ_{bw}	8.907	8.907	8.907
6	θ_{fw}	3.000	3.000	3.000
7	c_{bw}	0.881	0.584	0.386
8	c_{fw}	2.072	2.072	0.595
9	p_{bw}	1.000	1.000	1.000
10	p_{fw}	1.000	1.000	1.000
11	β_{bw}	0.025	0.025	0.025
12	β_{fw}	0.025	0.025	0.025

Tabella 3.2: Valore dei coefficienti del parametro LI per le consonanti bilabiali nel contesto vocalico /a/.

Il grado di attivazione è in stretta relazione con la compressione labiale, mentre quello di rilascio si dimostra identico per /p/ e /b/ ed è minore per la /m/.

Per quanto riguarda gli altri contesti vocalici, mentre i valori degli altri parametri rimangono costanti, i coefficienti di attivazione e rilascio vanno man mano aumentando passando dalla /a/ alla /i/ fino alla /u/. La spiegazione probabilmente è da cercare nel fatto che essi debbano compensare una maggiore costrizione esercitata dalle vocali, in accordo con quanto ipotizzato precedentemente per la diversa collocazione temporale dei target delle bilabiali.

Per quanto riguarda i rimanenti fonemi, abbiamo rilevato valori delle dominanze leggermente inferiori all'unità per quanto riguarda /f, v/, fino ad un valore nullo per la /l/. Il modello è riuscito anche in questi casi a descrivere con accuratezza l'andamento cinematico caratteristico, persino in presenza di dominanza consonantica nulla come accade per il fonema /n/.

Alcune considerazioni finali sulle tre principali tipologie di movimento:

- **chiusura-apertura:** è l'andamento più comune e che abbiamo già trattato a lungo parlando delle consonanti bilabiali. Notiamo il caso particolare della /d/ (figura 3.35), in cui nel movimento di chiusura-apertura il target della consonante si colloca temporalmente in modo diverso rispetto al minimo della curva. In figura sono rappresentate

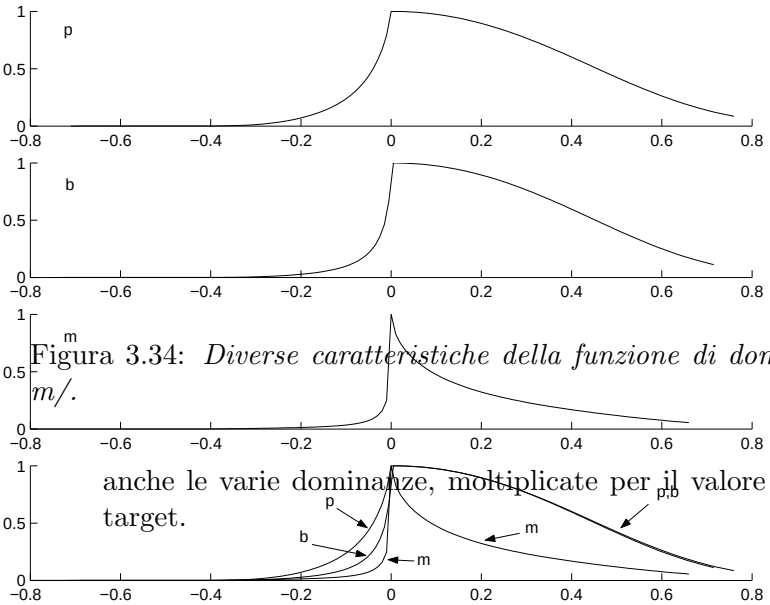


Figura 3.34: *Diverse caratteristiche della funzione di dominanza di /p, b, m/.*

anche le varie dominanze, moltiplicate per il valore del corrispettivo target.

- **chiusura-plateau-apertura:** una volta raggiunto il massimo della chiusura, per alcuni fonemi accade che la posizione acquisita si mantenga per un certo intervallo di tempo prima di iniziare il gesto di apertura. La consonante in questo caso assume una particolare conformazione "appiattita", come nel caso della /dz/ (figura 3.36). La collocazione della dominanza al di sotto del minimo del target consonantico ci fa capire che anche in questo caso, come nel precedente, la dominanza è inferiore all'unità.

- **glide:** nelle consonanti a dominanza quasi nulla capita che si verifichi un unico movimento dal massimo della prima vocale al massimo della seconda. In pratica la caratteristica della consonante scompare del tutto o quasi del tutto come succede per la /l/ (figura 3.37).

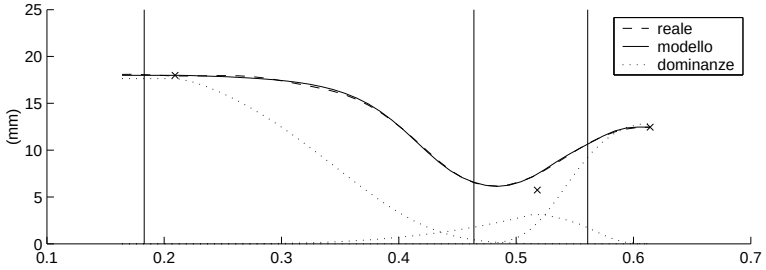
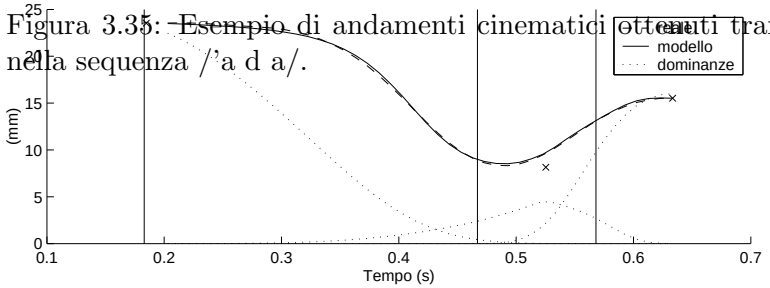


Figura 3.35: Esempio di andamenti cinematici ottenuti tramite il modello nella sequenza /'a d a/.



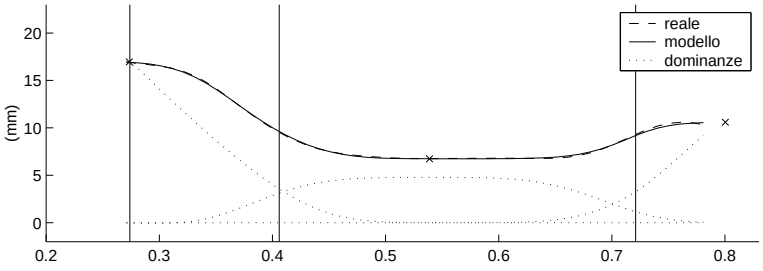
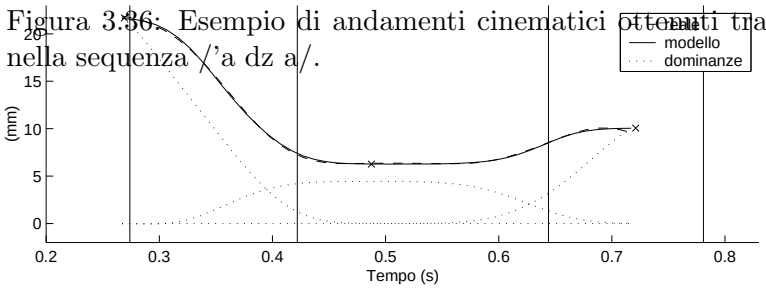


Figura 3.36: Esempio di andamenti cinematici ottenuti tramite il modello nella sequenza /'a dz a/.



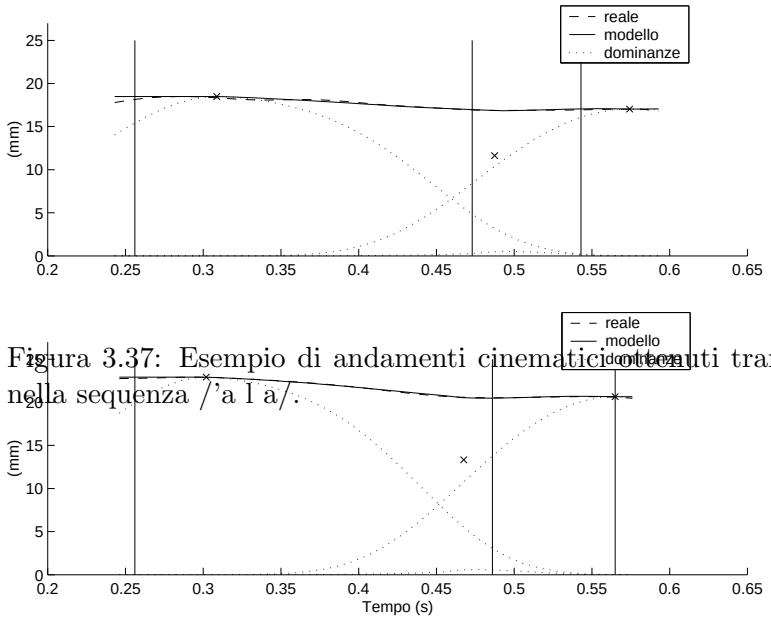


Figura 3.37: Esempio di andamenti cinematici ottenuti tramite il modello nella sequenza /'a l a/.

Conclusioni

Lo scopo principale di questa tesi è stata la realizzazione di una faccia parlante facilmente integrabile in una pagina html e che fornisca il servizio di sintesi audio-visiva di un qualsiasi testo in lingua italiana con una produzione del parlato visuale deve essere di elevata qualità.

Il framework realizzato si chiama LUCIA-WebGL. È una versione completamente reingegnerizzata dell'applicazione LUCIA Talking Head (sviluppata presso lo ISTC-CNR di Padova negli anni 2003-2006) utilizzando la nuova tecnologia WebGL.

LUCIA e LUCIA-WebGL sono sviluppate con tecnologie differenti, ma condividono gran parte delle logiche di progettazione e animazione. Sono entrambe totalmente basate sullo standard MPEG-4 SNHC (Synthetic/Natural Hybrid Coding) che fornisce le funzionalità per creare una animazione facciale in tempo reale guidata dai Parametri di Animazione Facciale (FAPs) con l'emulazione delle funzionalità dei muscoli mimici e la loro influenza sulla pelle della faccia. L'azione muscolare è resa esplicita per mezzo della deformazione del reticolo poligonale costruito attorno ai punti chiave FPs (*feature points*) che corrispondono alla giunzione dei muscoli mimici con la pelle. Tale deformazione dipende dal valore dei Parametri di Animazione Facciale FAPs (*Facial Animation Parameters*) che afferiscono ai movimenti dei singoli punti chiave.

Il calcolo degli streams audio/video è effettuata lato server dal modulo software AVEngine. L'input è un testo scritto con il linguaggio APML che permette di marcare le parti verbali del dialogo da pronunciare con l'espressione emotiva desiderata e di realizzare dunque anche una sintesi emotiva. Per riprodurre fedelmente la variabilità contestuale dovuta alla reciproca influenza dei movimenti articolatori durante la produzione del segnale verbale (coarticolazione) si utilizza una versione modificata del modello di Cohen-Massaro basata sui dati reali precedentemente acquisiti ed elaborati.

La sintesi audio è ottenuta per mezzo di un sistema text-to-speech esterno. Originariamente era possibile utilizzare solo la versione italiana di FE-

STIVAL e ora, come risultato di questo lavoro, anche le voce italiane di MARYTTS e Loquendo. La qualità sonora di quest'ultimo in particolare è di gran lunga migliore rispetto agli altri due.

LUCIA-WebGL soddisfa le specifiche del *Predictable Facial Animation Object Profile* di MPEG-4 ovvero è in grado di importare modelli esterni per mezzo della ricezione dei Parametri di Definizione FDP (*Facial Definition Parameters*). Questa possibilità non è presente in LUCIA che può animare solo il suo modello proprietario.

Il sistema è stato integrato nel prototipo realizzato per il progetto 'Wikimemo.it: Il portale della lingua e della cultura italiana' finanziato dal M.I.U.R. per promuovere la cultura italiana attraverso la lettura e l'ascolto. Il sistema permette all'utente di navigare nei contenuti, effettuare ricerche di frasi e parole e sentire come sono pronunciate in un contesto specifico. I risultati di una query di ricerca possono essere ascoltati con la voce di un italiano nativo o con la voce sintetica di LUCIA-WebGL che ne mostra il movimento labiale.

Abbiamo cominciato una serie di esperimenti volti a valutare l'accettabilità di un assistente virtuale da parte di un anziano supponendo che esso sia di reale utilità ovvero che permetta, attraverso l'uso della sola voce, di eseguire una ventina di compiti di quotidiana routine. I soggetti che hanno partecipato sono 40 persone con più di 65 anni d'età. Le registrazioni sono avvenute in quattro città italiane differenti: Torino, Padova, Roma e Palermo. Dai risultati dei questionari possiamo decisamente affermare che un assistente personale utile, affidabile e semplice da usare avrebbe una accoglienza calorosa tra le persone nella fascia 65-75.

Stiamo esplorando la possibilità di creare un modello personalizzato usando il sensore Kinect. La personalità dell'avatar sarà data in particolare dal modello del parlato visivo che replicherà in modo adeguato il modo di parlare dell'attore. E' prevista anche la creazione di una voce personalizzata con il sistema text-to-speech MARYTTS.

Un ulteriore miglioramento da considerare per gli sviluppi futuri riguarda i movimenti della lingua. Essa svolge un ruolo fondamentale per l'intellegibilità di alcune consonanti. Il modello utilizzato ora non considera questo aspetto; un nuovo modello della lingua, i cui movimenti possono essere controllati attraverso la stessa tipologia del modello di coarticolazione, incrementerebbe ancora di più la leggibilità del labiale.

Capitolo 4

Appendici

4.1 Il setting sperimentale

4.1.1 ELITE

Il database degli andamenti temporali degli articolatori per il modello del parlato visivo utilizzato in LUCIA è stato costruito utilizzando un sistema denominato EL.I.TE. (ELaboratore di Immagini TElevisive), realizzato al Centro di Bioingegneria del Politecnico di Milano [40].

ELITE consente la rilevazione del movimento di punti specifici dello spazio contraddistinti da speciali marcatori. Tali marcatori sono piccoli supporti emisferici di plastica rivestiti di materiale che riflette la luce emessa da sorgenti ad infrarosso: si tratta di elementi passivi molto leggeri, che non necessitano di alimentazioni o connessioni elettriche e che perciò non influenzano i movimenti del soggetto sotto test. Le sorgenti luminose sono costituite da due corone circolari di LED posizionate in modo coassiale attorno all'obbiettivo di altrettante telecamere sensibili allo spettro dell'infrarosso. I LED emettono luce entro questo spettro e vengono accesi in sincronismo con l'otturatore elettronico delle telecamere per un tempo pari a un decimo di secondo del periodo di campionamento delle telecamere. In questo modo si ottiene un effetto stroboscopico, utile per eliminare le distorsioni dell'immagine dovute a movimenti molto rapidi. Le telecamere utilizzate per riprendere le immagini sono dispositivi con frequenza di campionamento pari a $100\,Hz$.

Nei $10\,ms$ intercorrenti tra una acquisizione e la successiva, le immagini vengono elaborate per identificare i marcatori e i valori ottenuti memorizzati. In un secondo momento tali dati vengono utilizzati per svolgere le seguenti operazioni:

- **Etichettatura dei marcatori:** durante le acquisizioni il calcolatore riceve le coordinate bidimensionali del baricentro di ogni marcatore secondo l'ordine di scansione delle telecamere, cioè dall'alto in basso e

da sinistra verso destra dell'immagine. Poiché a causa del loro movimento i marcatori possono cambiare la loro posizione relativa e sono indistinguibili l'uno dall'altro, l'assegnazione di un'etichetta permette di identificare l'andamento di un singolo marcatore. A tale scopo viene definito un modello per ogni diverso movimento analizzato: ogni modello richiede il numero di marcatori, i collegamenti e la priorità di ognuno di questi. Per priorità si intende la probabilità che un marcatore, durante il suo spostamento, vada a mascherarne un altro. Si associa quindi un'alta priorità a quei marcatori che si suppone non vengano mai nascosti.

- **Stima dei dati mancanti:** i marcatori possono non essere visibili dalle telecamere in qualche istante. Si pensi ad esempio a dei marcatori piazzati sui lobi delle orecchie che a causa degli spostamenti della testa vengono nascosti, oppure ai marcatori sulle labbra che a causa di un'elevata protrusione non sono più nella posizione adatta a riflettere la luce. Per ovviare a questo problema, i dati mancati vengono stimati attraverso tecniche di interpolazione.

- **Ricostruzione dell'andamento tridimensionale:** i dati delle due telecamere vengono combinati per ottenere le coordinate tridimensionali dei marcatori. La trasformazione geometrica che mette in relazione un punto dello spazio tridimensionale con le coordinate bidimensionali della sua proiezione prospettica sul sistema di riferimento del sensore è descritta da equazioni di collinearità. La risoluzione di tali equazioni richiede, per ciascuna telecamera, la stima di nove parametri geometrici che definiscono l'orientamento delle telecamere nel sistema di riferimento assoluto, le coordinate assolute del centro di prospettiva delle telecamere, le coordinate del punto di intersezione dell'asse ottico con il piano dell'immagine e la lunghezza focale. Tali parametri vengono stimati a partire dalle coordinate dei marcatori appartenenti a un reticolo collocato in tre diverse posizioni note dello spazio. Dopo aver ottenuti i parametri il sistema può calcolare le coordinate tridimensionali dei marcatori. L'accuratezza di tale procedura di ricostruzione risulta pari a 1/2500 del campo di ripresa, in pratica 0.4 mm in un campo di ripresa di 1 m.

- **Filtraggio e calcolo di misure angolari, calcolo di velocità, accelerazioni:** il filtraggio ricopre un ruolo particolarmente importante in quanto i dati sono affetti da rumore. L'algoritmo di filtraggio comprende i seguenti passi:

  - Modellizzazione del segnale di entrata attraverso un modello AR;
  - Stima della densità dello spettro di potenza mediante trasformata Z del modello AR;

– Valutazione della frequenza di taglio per il filtraggio passabasso: si assume tale valore pari al 50% del SNR;
– Estrapolazione del segnale prima e dopo la registrazione per evitare distorsioni ai bordi dovute alla ripetizione periodica del segnale nel calcolo della trasformata di Fourier;
– Trasformazione del segnale nel dominio della frequenza;
– Filtraggio attraverso un filtro FIR;
– Calcolo delle velocità e accelerazioni nel dominio della frequenza;
– Trasformazione inversa nel dominio del tempo;

4.1.2 I parametri articolatori

I punti del viso sui quali posizionare i marcatori sono stati scelti in modo da fornire dati sui movimenti labiali e della mandibola, movimenti che ben caratterizzano l'articolazione del parlato.

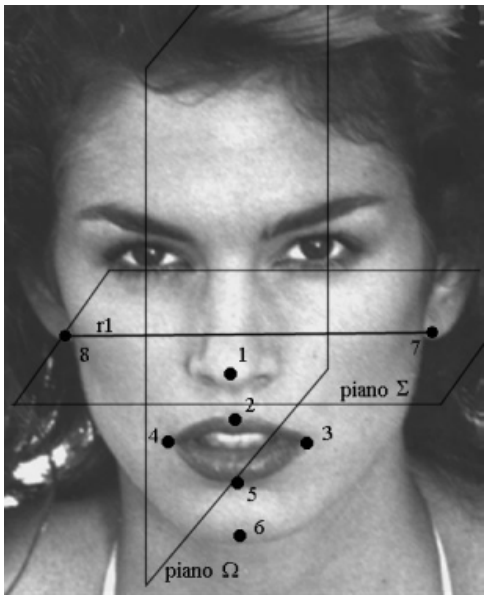


Figura 4.1: Posizionamento dei marcatori e piani di riferimento per il calcolo delle distanze

Come si nota dalla figura 4.1 i marcatori sono stati posizionati nei seguenti punti :

1. Punta del naso;
2. Punto centrale del bordo superiore del labbro superiore;
3. Angolo sinistro della bocca;

4. Angolo destro della bocca;
5. Punto centrale del bordo inferiore del labbro inferiore;
6. Punto centrale del mento;
7. Lobo dell'orecchio sinistro;
8. Lobo dell'orecchio destro

Il marcatore n. 1 è assunto come origine di una terna di assi cartesiani di riferimento (x,y,z) nello spazio. Il semiasse positivo delle z è diretto verso l'esterno del viso in direzione frontale; il semiasse positivo delle x è diretto verso la sinistra del soggetto; il semiasse positivo delle y è diretto verso l'alto. I parametri articolatori sono rappresentati da segmenti dello spazio tridimensionale, e risultano pertanto indipendenti da eventuali movimenti della testa del soggetto. La lunghezza $D_{i,j}$ del segmento che unisce due punti $P_i(x_i, y_i, z_i)$ e $P_j(x_j, y_j, z_j)$ dello spazio è data dalla:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{4.1}$$

Vengono di seguito descritti i 10 parametri articolatori:

- Apertura Labiale $AL = D_{2,5}$
- Apertura Mandibolare $AM = D_{1,6}$
- Arrotondamento $Arr = D_{3,4}$

$D_{2,5}$ rappresenta la distanza tra il marcatore 2 e il 5. La protrusione del labbro superiore è calcolata come la distanza tra il marcatore n. 2 e il punto $P_a(xa, ya, za)$ ottenuto come intersezione della retta $r1$, passante per i punti P_7 e P_8 ed il piano γ_1 perpendicolare a $r1$ e passante per P_2 (figura 4.1).

Analogamente, la protrusione del labbro inferiore è calcolata come la distanza tra il marcatore n° 5 e il punto $P_b(xb, yb, zb)$ ottenuto come intersezione della retta r1, ed il piano γ_2 perpendicolare a $r1$ e passante per P_5.

- Protrusione Labbro Superiore $PS = D_{2,a}$
- Protrusione Labbro Inferiore $PI = D_{2,b}$

L'emiapertura del labbro superiore è data dalla distanza del punto P_2 con il piano Σ passante per i punti P_1, P_7, P_8.
L'emiapertura del labbro inferiore è data dalla distanza dello stesso piano Σ con il punto P_5.
Sia P_α la proiezione ortogonale di P_2 su Σ (figura 4.1) :

- Emiapertura Labbro Superiore $LS = D_{2,\alpha}$
- Emiapertura Labbro Inferiore $LI = D_{5,\alpha}$

L'emiarrotondamento sinistro della bocca è dato dalla distanza tra il punto P_3 e il piano Ω passante per il punto P_1 e perpendicolare alla retta $r1$. Analogamente, l'emiarrotondamento destro della bocca è dato dalla distanza tra il punto P_4 e il piano Ω, mentre l'emiapertura mandibolare è data dalla distanza tra il punto P_6 e il piano Σ.

- Emiarrotondamento Sinistro $EAS = D_{3,\beta}$
- Emiarrotondamento Destro $EAD = D_{4,\delta}$
- Emiapertura Mandibolare $EM = D_{6,\gamma}$

I punti P_β, P_δ e P_γ sono le proiezioni dei punti P_3, P_4 sul piano Ω e P_6 sul piano Σ.

Per guidare il nostro modello utilizzeremo solo quei parametri necessari a comandare il movimento. L'apertura labiale, ad esempio, ha una forte validità per statistica, ma serve relativamente a descrivere i dettagli del movimento perché labbro superiore e inferiore hanno andamenti differenti. I parametri del modello saranno quindi, ordinati nel seguente modo:

1. Labbro inferiore (LI);
2. Labbro superiore (LS);
3. Arrotondamento (AR);
4. Protrusione Labbro Inferiore (PI);
5. Protrusione Labbro Superiore (PS);
6. Emiapertura Mandibolare (EM).

4.2 MPEG-4 FBA

Face and Body Animation - FBA è la sezione dello standard MPEG-4 che si occupa di definire le regole per la descrizione e l'animazione di caratteri virtuali. Per quanto concerne le facce sintetiche MPEG-4 fornisce tre set di parametri:

- ***FDP*** *(Facial Defintion Parameters)*: sono i parametri per la definizione della struttura della faccia; possono essere usati sia per modificare la forma e l'aspetto del modello facciale già disponibile a livello del decoder, sia per codificare l'informazione necessaria a trasmettere un nuovo modello completo, insieme ai criteri che devono essere applicati per animarlo. Gli FDP, tipicamente, vengono utilizzati solo quando viene iniziata una nuova sessione.

- ***FAP*** *(Facial Animation Parameters)*: descrivono il movimento della faccia, sia a basso livello (spostamento di uno specifico, singolo punto chiave della geometria del volto) sia ad alto livello (riproduzione di espressioni ed emozioni). I FAPs costituiscono il vero e proprio stream di parametri che, associato al particolare *media object*, genererà l'animazione ed il comportamento nel tempo.

- ***FAP Interpolation Table*** *(FIT)*, usato per ridurre il *bitrate* dei FAP. Sfruttando le conoscenze a priori sulla simmetria e la dinamica del volto umano è possibile dedurre il valore di alcuni FAP basandosi sulla conoscenza di alcuni altri.

I parametri di definizione (FDP) e di animazione (FAP) facciale definiti dallo standard MPEG-4 sono progettati allo scopo di codificare e trasmettere informazione sulla forma e texture di un volto umano, per consentirne l'animazione con riproduzione fedele dell'espressione e del movimento sincrono delle labbra col parlato. L'esistenza di un insieme di parametri standard esula infatti il decodificatore dalla necessità di "scaricare" dalla rete ogni volta un modello facciale diverso. L'uso corretto dei FAP conduce a risultati ragionevolmente simili, in termini di espressione e di movimento labiale, su diversi modelli facciali senza la necessità di inizializzare o calibrare il modello stesso. Gli FDP permettono, ove trasmessi, la definizione di una particolare struttura facciale e, opzionalmente, della texture associata consentendo una migliore riproduzione dei movimenti del volto. Utilizzando una conversione da fonemi a FAP è anche possibile interfacciare il modello facciale con sistemi *Text-To-Speech*.

4.2.1 La faccia neutrale e i *features points*

La faccia nel suo stato neutrale è definita così: sguardo fisso e in direzione dell'asse z; tutti i muscoli della faccia sono rilassati; le palpebre sono tangenti all'iride; le labbra sono chiuse, la linea delle labbra è orizzontale ed alla stessa altezza degli angoli delle labbra; la bocca è chiusa ed i denti superiori toccano l'arcata dentaria inferiore, la lingua è piatta, orizzontale con la sua punta che tocca il confine tra denti superiori ed inferiori.

I *feature points* rappresentano i punti somatici rilevanti della faccia nel suo stato neutrale; essi sono suddivisi in gruppi dipendenti dalla particolare regione della testa alla quale appartengono. Ognuno dei *feature points* è, quindi, catalogato con un numero che identifica il particolare gruppo di appartenenza, e un indice progressivo che lo identifica ulteriormente all'interno del gruppo.

La posizione dei vari punti sul volto è mostrata in figura 4.2; bisogna tener presente che al momento della calibrazione con gli FDPs, la posizione di questi può essere mutata per conformare il modello a quanto indicato dall'encoder. Inoltre, alcuni *feature points* (quelli indicati in nero in figura) sono affetti direttamente dai FAPs, e durante la fase di animazione dovranno muoversi secondo quanto indicato da questi ultimi.

4.2.2 Facial Animation Parameters (FAP)

I Parametri per l'Animazione Facciale (FAPs) sono basati sullo studio delle "minime azioni facciali" e sono strettamente correlati alle azioni muscolari. Rappresentano un insieme completo di movimenti facciali di base, e pertanto permettono la rappresentazione delle espressioni più naturali. Una particolare espressione facciale è generata deformando il modello facciale nel suo stato neutrale secondo i valori di FAP specificati per l'istante di tempo corrispondente. L'animazione è ottenuta da una sequenza di tali valori (un set per ogni frame) esattamente come avviene per una pellicola cinematografica, ma con molti meno dati da trassmettere.

Tutti i parametri che riguardano movimenti traslazionali sono espressi in termini di unità di parametro di animazione facciale (*Facial Animation Parameter Units - FAPU*). Tali unità sono definite in maniera tale da permettere l'interpretazione dei FAPs in modo univoco su qualunque modello, producendo risultati ragionevoli sia in termini di espressione mimica che di sincronizzazione labiale. Le FAPU sono definite come frazioni di distanze tra punti caratteristici della faccia (figura 4.3 e tabella 4.1). Queste caratteristiche come distanza tra gli occhi, distanza occhio-naso, distanza tra naso e bocca, ed ampiezza della bocca, sono definite per la faccia nel suo stato

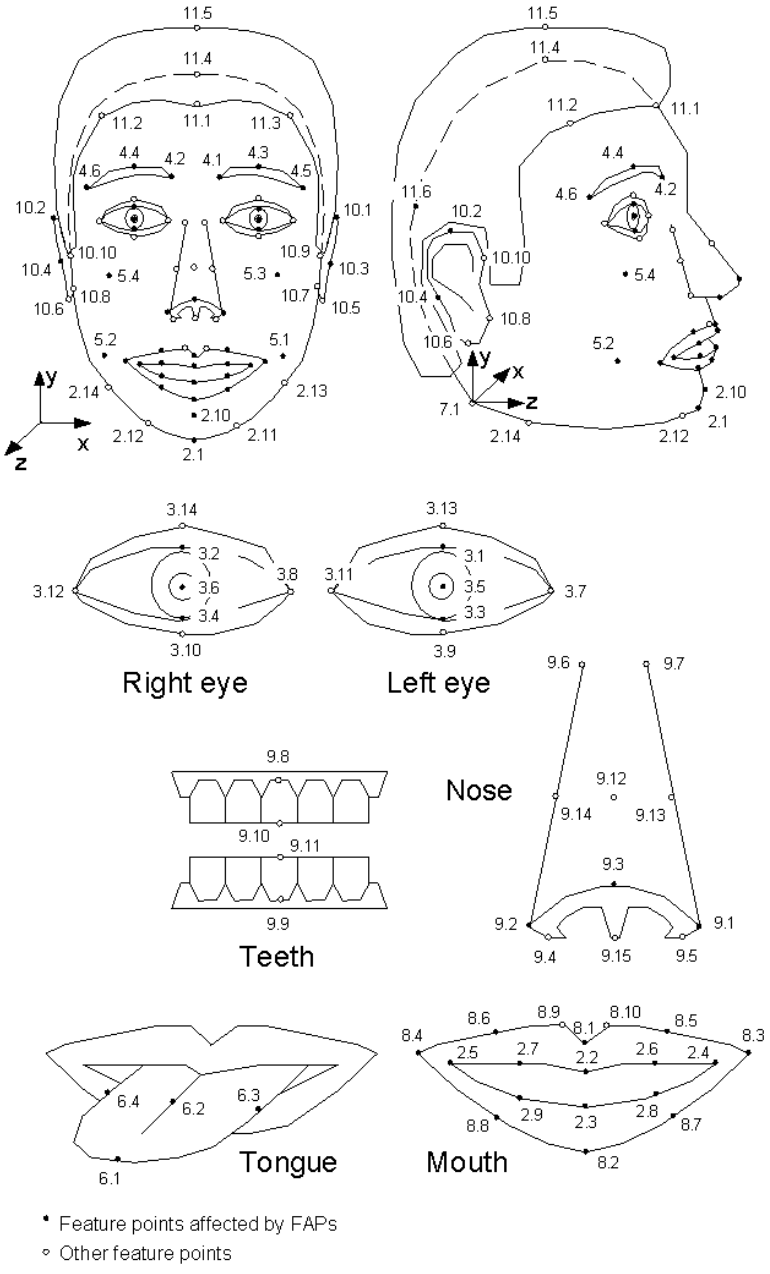


Figura 4.2: I *features points* della faccia neutrale (l'immagine è cortesia di [47])

neutrale. Permettono l'interpretazione dei FAPs su ogni tipo di modello facciale in modo consistente.

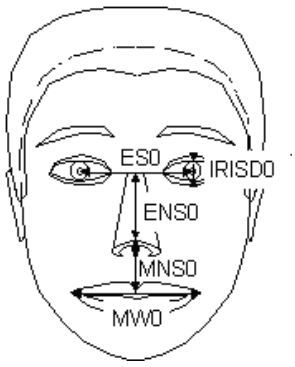


Figura 4.3: I FAPs sono definiti in termini di FAPU (FAP Unit) per rendere omogenei i movimenti su modelli differenti.

Descrizione	**Valore FAPU**
"IRIS Diameter" (per definizione è la distanza tra palpebra inferiore e superiore nella faccia neutrale)	IRISD = IRISD0/1024
"Eye Separation"	ES = ES0/1024
"Eye - Nose Separation"	ENS = ENS0/1024
"Mouth – Nose Separation"	MNS = MNS0/1024
"Mouth – Width Separation"	MW = MW0/1024

Tabella 4.1: Definizione esatta delle FAPU

Le unità frazionali usate sono scelte per garantire una sufficiente precisione. Inoltre le *Angular Units (AU)*, cioè le unità di misura angolari necessarie per le rotazioni, sono definite come frazioni di angolo rispetto al sistema di assi destrorso in cui è orientata la faccia neutrale. Il valore numerico dei FAPs è espresso in termini di FAPU ed è compito del decoder trasformare questi valori relativi in valori numerici reali corrispondenti al sistema di coordinate in cui "vive" il modello facciale proprietario.

I FAPs sono in tutto 68. La classificazione completa è mostrata nell'appendice 4.4.

I primi 2 FAPs sono parametri di alto livello: il parametro FAP1, denominato "visema", codifica la postura della bocca associata alla pronuncia di un particolare fonema senza doverlo esprimere tramite parametri di basso

livello e il parametro FAP2, denominato "espressione", codifica la mimica del volto associata ad alcuni tipici "stati emotivi" e alle possibili interpolazioni fra coppie di essi. La descrizione dei fonemi è data secondo lo standard SAMPA [2] che definisce i fonemi attraverso caratteri ASCII. SAMPA risulta quindi leggibile al calcolatore e permette la trascrizione dei linguaggi naturali. In fase di definizione solo quattordici visemi sono stati codificati tramite il FAP1. Poiché tuttavia il campo *viseme select* è costituito da sei bit, è possibile aggiungere ulteriori visemi in estensioni proprietarie dello standard. Il parametro "espressione" permette di scegliere da un insieme di espressioni facciali con 64 livelli di intensità dove il valore 0 si riferisce alla posizione neutrale. La ricerca nel campo delle espressioni facciali ha dimostrato che esistono sei categorie universali di espressioni facciali riconosciute da tutte le culture: rabbia, disgusto, paura, felicità, tristezza e sorpresa.

Gruppi	**Numero di FAP**
1: visemi ed espressioni	2
2: mandibola,mento,labbra interne	16
3: occhi,pupille,palpebre	12
4: sopracciglia	8
5: guance	4
6: lingua	5
7: rotazione della testa	3
8: labbra esterne	10
9: naso	4
10: orecchie	4

Tabella 4.2: I raggruppamenti di FAPs

I rimanenti FAPs, dal FAP3 al FAP68, sono parametri di basso livello che agiscono su uno o più *feature points* del volto e sono organizzati in gruppi in funzione delle parti della faccia che animano. La tabella 4.2 definisce i gruppi e il numero di parametri FAPs appartenenti ad ognuno di essi. Il valore nullo di un FAP significa mantenere il corrispondente *feature point* nella posizione che occupa sulla faccia neutrale.

4.2.3 Facial Definition Parameters (FDP)

Gli FDP sono utilizzati per personalizzare il modello base del decoder mediante la calibrazione oppure per effettuare il download opzionale di un modello esterno e delle specifiche per animarlo.
Generalmente, gli FDP sono trasmessi una sola volta per sessione, seguiti successivamente da uno stream di FAP compressi. Comunque, se il decoder

non riceve gli FDP, mediante l'uso delle FAPU, è in grado di interpretare il *FAP stream* e di generare comunque un'animazione con il suo modello proprietario; questo garantisce un funzionamento minimale in applicazioni broadcast o di teleconferenza.
Il set di FDP consiste nel *FDP node*, le cui specifiche sono espresse con sintassi BIFS, e opzionalmente una *Facial Animation Table*, che descrive al decoder come animare ed effettuare il *downloading* del modello facciale:

- La descrizione formale del *FDP node* è la seguente:

```
{
field  SFNode  featurePointsCoord NULL
field  SFNode  textureCoord NULL
field  SFBool  calibrationOnly NULL
exposedField  SFNode  sceneGraph NULL
}
```

  Il campo `featurePointsCoord` contiene le coordinate dei *feature points*, ordinate secondo il numero delle etichette corrispondenti ai punti come in figura 4.2.
  Il campo `textureCoord` contiene, se effettivamente lo `scenegraph` prevede delle texture da utilizzare, le coordinate delle texture dei *feature points*. Il numero delle *texture coordinates* deve essere pari a quello dei *feature points*.
  Il campo `calibrationOnly` specifica se il *FDP node* può essere usato esclusivamente per la calibrazione del modello proprietario del decoder, altrimenti, se `calibrationOnly==FALSE`, il *FDP node* contiene un modello facciale completo nel campo `sceneGraph` che può essere utilizzato al posto del modello del decoder. Il campo `sceneGraph`, come anticipato, contiene l'eventuale *scene graph* di un nuovo modello facciale, pronto per essere calibrato ed animato. Tale *scene graph* può consistere in una semplice mesh polygonale senza texture e colore (*wireframe*) oppure in un modello facciale con texture, colori e una gerarchia di trasformazioni. Lo *scene graph* di un modello *downloaded* può inoltre contenere due nodi che possono essere affetti dai FAP: l'*IndexedFaceSet* e il *Transform* (la cui descrizione si può trovare nei paragrafi 3.2.10 e 3.2.11 di [43]).

- La *Facial Animation Table*, a sua volta, viene trasmessa direttamente dopo il BIFS bitstream del *FDP node* se il campo *calibrationOnly* del *FDP node* è posto FALSE. La *Facial Animation Table* ha una lista di tutti i FAP che animano il modello facciale. I FAP, infatti, animeranno il *downloaded model* effettuando aggiornamenti opportuni nei

nodi *Transform* o *IndexedFaceSet* dello *scenegraph*; per ogni FAP della lista, la *Facial Animation Table* descrive quali nodi sono animati da questi FAP e, soprattutto, come sono animati. Saranno poi questi nodi a definire le trasformazioni e le deformazioni da applicare al *downloaded model*; quindi, la variazione, nel tempo, dei valori nei nodi *Transform* e *IndexedFaceSet* genera trasformazioni e deformazioni, nel tempo, del modello, producendo l'animazione facciale desiderata.

4.3 Esempio di file F.A.P.

Il *FAP stream* d'ingresso da cui LUCIA legge le informazioni per muovere i features points è ricavato da un file strutturato in maniera conforme a quanto indicato dallo standard MPEG-4. Il file in questione (la cui estensione è *.fap*) è un file di testo il cui header può contenere alcune righe di commento (il cui inizio deve essere fissato con il caratttere cancelletto) e successivamente una sequenza, su di una singola riga, che descrive alcune sue caratteristiche, costituita da:

1. un *float* che descrive la versione del file (ad esempio "`2.1`")
2. una stringa di caratteri che indica il nome del file (ad esempio "`Lucia`")
3. un intero che indica il *frame rate* in termini di *fps* (ad esempio "`30`")
4. un intero che indica il numero totale di frame dell'animazione (ad esempio "`2156`")

Finito l'header del file inizia la presentazione, nelle righe successive, dei FAP e le *FAP mask* che saranno espressi per ogni frame componente l'animazione. Ogni frame (il numero totale è indicato nell'header) dà origine a due righe, la prima riga è quella della *FAP mask* la seconda è quella dei valori effettivi dei FAP:

- la prima riga contiene una serie di interi che possono essere esclusivamente o "`0`" o "`1`"; il numero di "`0`" ed "`1`" complessivamente deve essere pari al numero di FAP del modello (68). Uno "`0`" indica che, nella riga successiva, il valore del FAP corrispondente al numero d'ordine indicato dalla posizione dello "`0`" non sarà fornito, e quindi mancherà del tutto (ad esempio uno "`0`" nella 17-esima posizione della *FAP mask* indica che non sarà fornito il valore del FAP 17). Un "`1`" indica che, nella riga successiva, il valore del FAP corrispondente al numero d'ordine indicato dalla posizione del numero "`1`" sarà debitamente fornito (ad esempio un "`1`" nella 17-esima posizione della *FAP mask* indica che sarà fornito nella riga successiva il valore del FAP 17). Quindi nella riga seguente, cioè quella dei valori dei FAP, saranno presenti solo i valori corrispondenti agli "`1`" nella riga *FAP mask*, ordinati in maniera crescente rispetto al numero d'ordine dei FAP;

- la seconda riga contiene una serie di interi; il primo numero della riga, che deve essere necessariamente presente, indica il numero del frame cui corrispondono i successivi valori dei FAP; gli, eventuali, interi successivi indicano i valori dei FAP, presentati nell'ordine con cui si presentano gli "`1`" nella precedente riga *FAP mask*.

```
    # Face animation parameters listed below in ascii
# Each record is a line of ascii with spaces delimiting the parameters.
# at the beginning of a line indicates a comment
# HEADER DESCRIPTION # # Header record :
# Version number:2.1
# Sequence base name:  note:face feature filename= <basename>.fff,
# video filename= <basename>.vid, separate image filenames=
# <basename>.nnnnn.img, audio filename= <basename>.aud,
# FAP filename = <basename>.fap, FDP filename = <basename>.fdp,
# BAP filename = <basename>.bap, BDP filename = <basename>.bdp
# Frame rate:  fps (note that this may be equal to the video field
# rate since interlace has no meaning here)
# Sequence length:  Nframe
# FAP mask:  one 0 or 1 for each of the 68 FAPs (masked or unmasked) indicating
# whether it is used.  The FAP mask precedes each FAP data line on a separate line.
# FAP DATA DESCRIPTION #
# Each frame is described as one line of integer FAP values starting with the
# frame number.  The FAP mask in the header indicates which
# FAP groups are included.  The 68 parameter values are defined
# in MPEG doc N1454 (Table 2-1).  A zero value for any
# FAP indicates a neutral face.
# Example:
# <frame number> <viseme> <expression> <open_jaw>...<pull_r_ear>
# Header description
2.1 G:\Animation\ItalianLucia\bin\aba 30 2156
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -27 -27 -13 -13 -6 -6 6 6 3
2 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -30 -30 -15 -15 -7 -7 7 7 3
2 0 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0
0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -47 -47 -23 -23 -11 -11 11 11
4 2 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -73 -73 -36 -36 -18 -18 18 18
7 3 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1
0 0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 -12 -12 0 0 0 0 0 0 0 0 24 24 0 24 24 6 5 0 0 0 0 0 0 0 0 -108 -108 -54 -54 -27
-27 27
```

```
27 9 4 0 0 0
.
.
.
.
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2152 187 145 -374 21 -5 148 142 -328 -309 -240 -228 0 0 4 4 0 4 4 -456 -500 -6650 -6650
11138
11138 0 0 0 0 -6 -6 -3 -3 -1 -1 1 1 -226 -250 -10992 6641 -9366
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2153 166 122 -332 -8 -5 137 139 -290 -316 -227 -193 0 0 26 26 0 26 26 -484 -438 -6645
-6645 11022
11022 0 0 0 0 -22 -22 -11 -11 -5 -5 5 5 -236 -223 -10977 6640 -9186
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2154 137 99 -275 -39 -2 114 125 -236 -278 -245 -185 0 0 13 13 0 13 13 -487 -381 -6657
-6657 10940
10940 0 0 0 0 -12 -12 -6 -6 -3 -3 3 3 -237 -199 -10887 6653 -8355
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2155 117 86 -234 -89 -78 91 109 -193 -228 -155 -141 0 0 157 157 0 157 157 -344 -327 -6663
-6663
10736 10736 0 0 0 0 -128 -128 -64 -64 -32 -32 32 32 -179 -174 -10694 6692 -11479
```

4.4 Tabella dei FAPs

Elenco e descrizione dei Facial Animation Parameters. Le due ultime colonne indicano i *feature points* che vengono influenzati.

#	FAP name	FAP description	FAP Units	Uni- or Bidir	Posit. FAP Mo- tion	Grp	FDP Sub Grp Num
1	Viseme	Set of values determining the mixture of two visemes for this frame (e.g. pbm, fv, th)	na	na	na	1	Na
2	Expression	A set of values determining the mixture of two facial expression	na	na	na	1	Na
3	open_jaw	Vertical jaw displacement (does not affect mouth opening)	MNS	U	down	2	1
4	lower_t_midlip	Vertical top middle inner lip displacement	MNS	B	down	2	2
5	raise_b_midlip	Vertical bottom middle inner lip displacement	MNS	B	up	2	3
6	stretch_l_cornerlip	Horizontal displacement of left inner lip corner	MW	B	left	2	4
7	stretch_r_cornerlip	Horizontal displacement of right inner lip corner	MW	B	right	2	5
8	lower_t_lip_lm	Vertical displacement of midpoint between left corner and middle of top inner lip	MNS	B	down	2	6
9	lower_t_lip_rm	Vertical displacement of midpoint between right corner and middle of top inner lip	MNS	B	down	2	7
10	raise_b_lip_lm	Vertical displacement of midpoint between left corner and middle of bottom inner lip	MNS	B	up	2	8
11	raise_b_lip_rm	Vertical displacement of midpoint between right corner and middle of bottom inner lip	MNS	B	up	2	9
12	raise_l_cornerlip	Vertical displacement of left inner lip corner	MNS	B	up	2	4
13	raise_r_cornerlip	Vertical displacement of right inner lip corner	MNS	B	up	2	5
14	thrust_jaw	Depth displacement of jaw	MNS	U	forward	2	1
15	shift_jaw	Side to side displacement of jaw	MW	B	right	2	1
16	push_b_lip	Depth displacement of bottom middle lip	MNS	B	forward	2	3

#	FAP name	FAP description	FAP Units	Uni- or Bidir	Posit. FAP Motion	Grp	FDP Sub Grp Num
17	push_t_lip	Depth displacement of top middle lip	MNS	B	forward	2	2
18	depress_chin	Upward and compressing movement of the chin (like in sadness)	MNS	B	up	2	10
19	close_t_l_eyelid	Vertical displacement of top left eyelid	IRISD	B	down	3	1
20	close_t_r_eyelid	Vertical displacement of top right eyelid	IRISD	B	down	3	2
21	close_b_l_eyelid	Vertical displacement of bottom left eyelid	IRISD	B	up	3	3
22	close_b_r_eyelid	Vertical displacement of bottom right eyelid	IRISD	B	up	3	4
23	yaw_l_eyeball	Horizontal orientation of left eyeball	AU	B	left	3	5
24	yaw_r_eyeball	Horizontal orientation of right eyeball	AU	B	left	3	6
25	pitch_l_eyeball	Vertical orientation of left eyeball	AU	B	down	3	5
26	pitch_r_eyeball	Vertical orientation of right eyeball	AU	B	down	3	6
27	thrust_l_eyeball	Depth displacement of left eyeball	ES	B	forward	3	5
28	thrust_r_eyeball	Depth displacement of right eyeball	ES	B	forward	3	6
29	dilate_l_pupil	Dilation of left pupil	IRISD	B	growing	3	5
30	dilate_r_pupil	Dilation of right pupil	IRISD	B	growing	3	6
31	raise_l_i_eyebrow	Vertical displacement of left inner eyebrow	ENS	B	up	4	1
32	raise_r_i_eyebrow	Vertical displacement of right inner eyebrow	ENS	B	up	4	2
33	raise_l_m_eyebrow	Vertical displacement of left middle eyebrow	ENS	B	up	4	3
34	raise_r_m_eyebrow	Vertical displacement of right middle eyebrow	ENS	B	up	4	4
35	raise_l_o_eyebrow	Vertical displacement of left outer eyebrow	ENS	B	up	4	5
36	raise_r_o_eyebrow	Vertical displacement of right outer eyebrow	ENS	B	up	4	6
37	squeeze_l_eyebrow	Horizontal displacement of left eyebrow	ES	B	right	4	1
38	squeeze_r_eyebrow	Horizontal displacement of right eyebrow	ES	B	left	4	2
39	puff_l_cheek	Horizontal displacement of left cheeck	ES	B	left	5	1

#	FAP name	FAP description	FAP Units	Uni- or Bidir	Posit. FAP Motion	Grp	FDP Sub Grp Num
40	puff_r_cheek	Horizontal displacement of right cheeck	ES	B	right	5	2
41	lift_l_cheek	Vertical displacement of left cheek	ENS	U	up	5	3
42	lift_r_cheek	Vertical displacement of right cheek	ENS	U	up	5	4
43	shift_tongue_tip	Horizontal displacement of tongue tip	MW	B	right	6	1
44	raise_tongue_tip	Vertical displacement of tongue tip	MNS	B	up	6	1
45	thrust_tongue_tip	Depth displacement of tongue tip	MW	B	forward	6	1
46	raise_tongue	Vertical displacement of tongue	MNS	B	up	6	2
47	tongue_roll	Rolling of the tongue into U shape	AU	U	concave upward	6	3, 4
48	head_pitch	Head pitch angle from top of spine	AU	B	down	7	1
49	head_yaw	Head yaw angle from top of spine	AU	B	left	7	1
50	head_roll	Head roll angle from top of spine	AU	B	right	7	1
51	lower_t_midlip _o	Vertical top middle outer lip displacement	MNS	B	down	8	1
52	raise_b_midlip_o	Vertical bottom middle outer lip displacement	MNS	B	up	8	2
53	stretch_l_cornerlip_o	Horizontal displacement of left outer lip corner	MW	B	left	8	
54	stretch_r_cornerlip_o	Horizontal displacement of right outer lip corner	MW	B	right	8	4
55	lower_t_lip_lm _o	Vertical displacement of midpoint between left corner and middle of top outer lip	MNS	B	down	8	5
56	lower_t_lip_rm _o	Vertical displacement of midpoint between right corner and middle of top outer lip	MNS	B	down	8	6
57	raise_b_lip_lm_o	Vertical displacement of midpoint between left corner and middle of bottom outer lip	MNS	B	up	8	7

#	FAP name	FAP description	FAP Units	Uni- or Bidir	Posit. FAP Motion	Grp	FDP Sub Grp Num
58	raise_b_lip_rm_o	Vertical displacement of midpoint between right corner and middle of bottom outer lip	MNS	B	up	8	8
59	raise_l_cornerlip_o	Vertical displacement of left outer lip corner	MNS	B	up	8	3
60	raise_r_cornerlip_o	Vertical displacement of right outer lip corner	MNS	B	up	8	4
61	stretch_l_nose	Horizontal displacement of left side of nose	ENS	B	left	9	1
62	stretch_r_nose	Horizontal displacement of right side of nose	ENS	B	right	9	2
63	raise_nose	Vertical displacement of nose tip	ENS	B	up	9	3
64	bend_nose	Horizontal displacement of nose tip	ENS	B	right	9	3
65	raise_l_ear	Vertical displacement of left ear	ENS	B	up	10	1
66	raise_r_ear	Vertical displacement of right ear	ENS	B	up	10	2
67	pull_l_ear	Horizontal displacement of left ear	ENS	B	left	10	3
68	pull_r_ear	Horizontal displacement of right ear	ENS	B	right	10	4

4.5 Lucia VRML Model

```
#VRML V2.0 ascii

Separator
{
    ShapeHints
    {
        creaseAngle    2.0
    }
    Material
    {
        ambientColor   1.0 1.0 1.0
        diffuseColor   0.5 0.5 0.5
        specularColor  0 0 0
    }
    Coordinate3
    {
        point
        [
            88.8424 160.27 173.634,
            92.1787 160.101 174.393,
            92.1128 159.52 174.385,
            91.9233 158.931 174.31,
            91.6298 158.368 174.196,

                    .
                    .
                    .

            91.2391 157.869 174.063,
            97.5271 166.774 156.502,
            100.998 162.239 157.801,
            100.862 161.44 157.778,
            101.256 160.754 157.904
        ]
    }
    Texture2
    {
        filename "FacciaFaceGen00.jpg"
    }
    TextureCoordinate2
    {
        point
        [
            0.499314 0.528691,
            0.462186 0.371322,
            0.494789 0.367172,
            0.431316 0.380585,
            0.40381 0.394932,
            0.381298 0.414312,

                    .
                    .
                    .

            0.681677 0.205372,
            0.643967 0.213374,
            0.62051 0.190268,
            0.580956 0.202874,
            0.547285 0.183792
```

```
        ]
    }
    IndexedFaceSet
    {
        coordIndex
        [
            325, 0, 323, 322, -1,
            323, 0, 327, 326, -1,
            327, 0, 330, 329, -1,
            330, 0, 333, 332, -1,
            333, 0, 336, 335, -1,
            336, 0, 339, 338, -1,

                    .
                    .
                    .

            1340, 318, 1276, 1341, -1,
            1276, 319, 1342, 1341, -1,
            1342, 319, 1279, 1343, -1,
            1279, 320, 1344, 1343, -1,
            1344, 320, 1281, 1345, -1,
            1281, 289, 1283, 1345, -1
        ]
        textureCoordIndex
        [
            325, 0, 323, 322, -1,
            323, 0, 327, 326, -1,
            327, 0, 330, 329, -1,
            330, 0, 333, 332, -1,

                    .
                    .
                    .

            1267, 316, 1336, 1335, -1,
            1276, 319, 1342, 1341, -1,
            1342, 319, 1279, 1343, -1,
            1279, 320, 1344, 1343, -1,
            1344, 320, 1281, 1345, -1,
            1281, 290, 1283, 1345, -1
        ]
    }
}
```

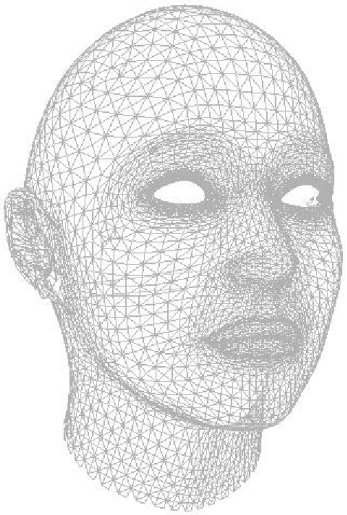


Figura 4.4: Wireframe dell'oggetto *pelle*

Figura 4.5: Oggetto *pelle*

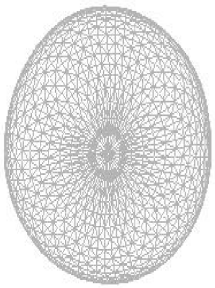


Figura 4.6: Wireframe dell'oggetto *occhio*

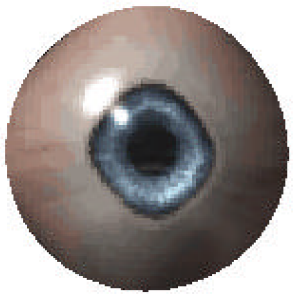


Figura 4.7: Oggetto *occhio*

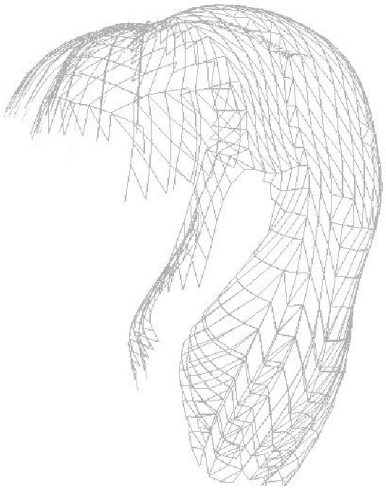


Figura 4.8: Wireframe dell'oggetto *capelli*

Figura 4.9: Oggetto *capelli*

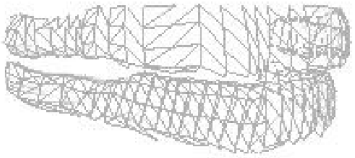


Figura 4.10: Wireframe dell'oggetto *denti*

Figura 4.11: Oggetto *denti*

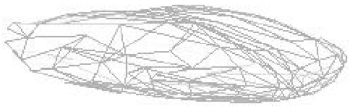


Figura 4.12: Wireframe dell'oggetto *lingua*

Figura 4.13: Oggetto *lingua*

Tutte le texture utilizzate sono state prese con l'uso di scanner 3D. Nelle figure 4.14 e 4.15 se ne riportano alcune.

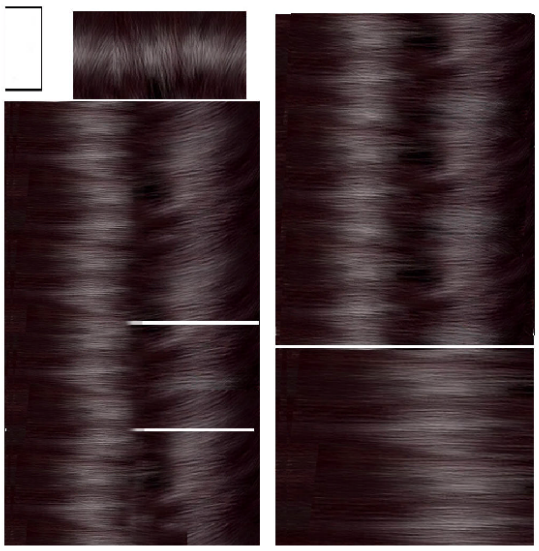


Figura 4.14: Texture dei capelli

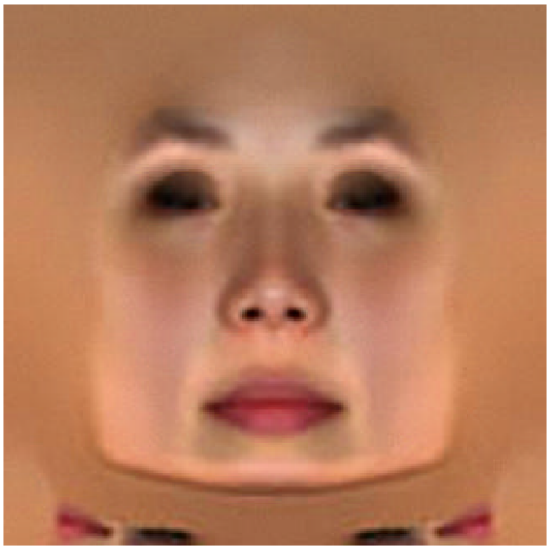
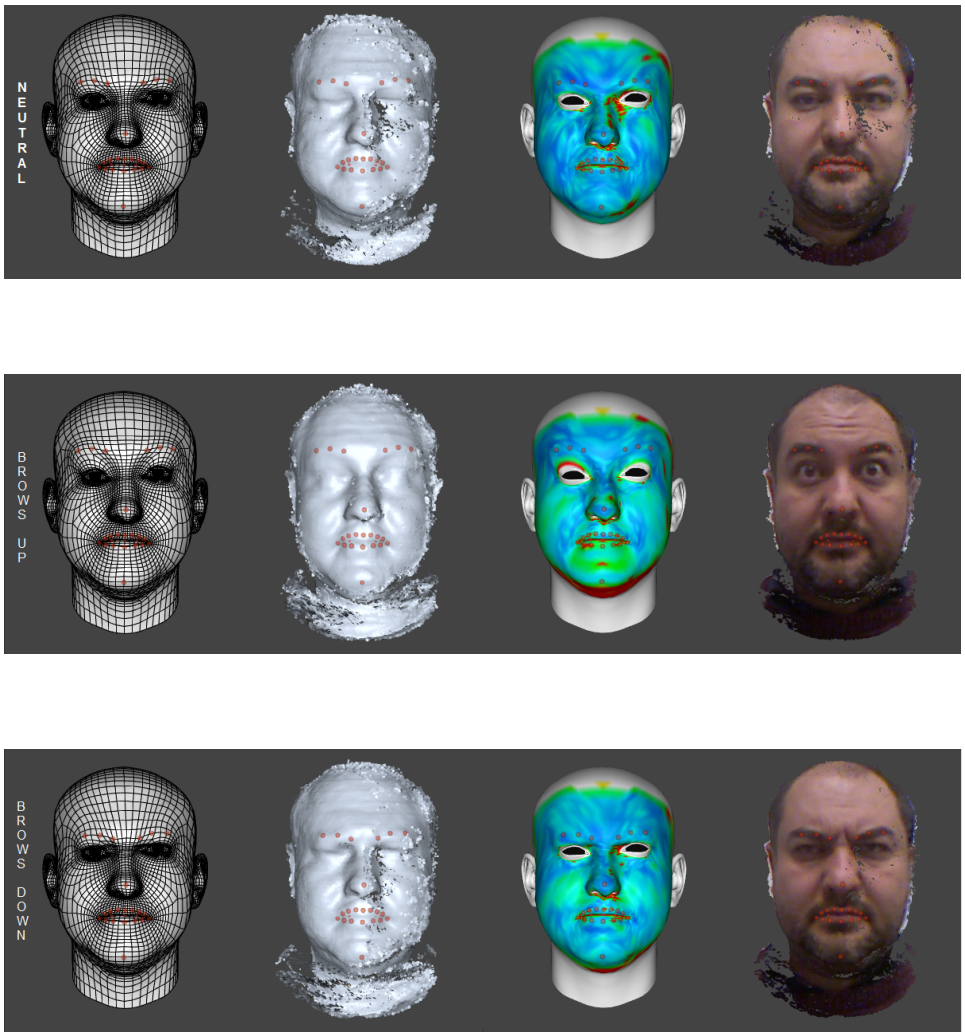
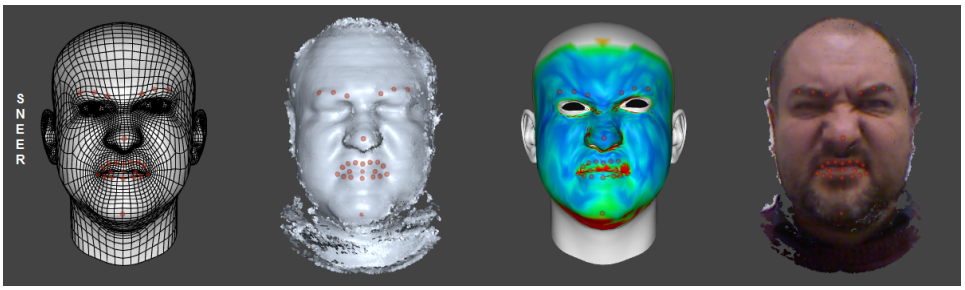


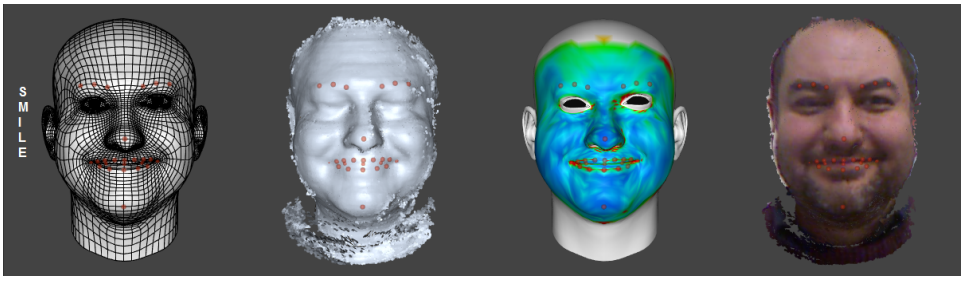
Figura 4.15: Texture della pelle

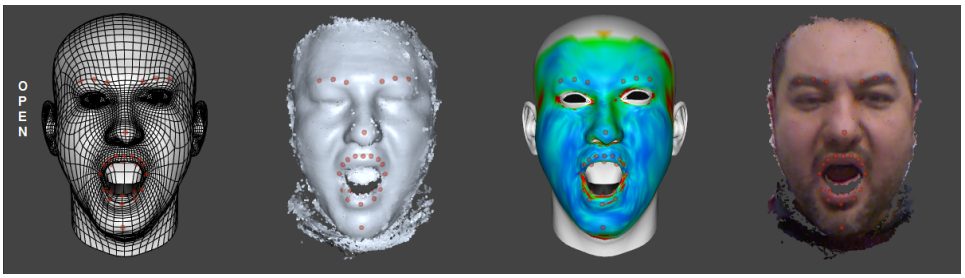
4.6 Le pose generative del modello 3D

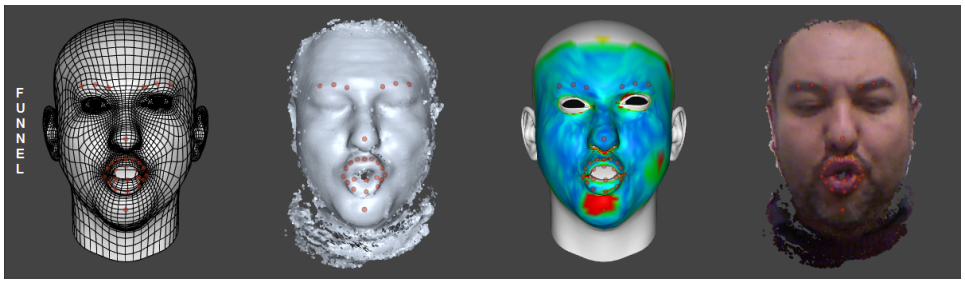
L'insieme visivo completo delle espressioni utilizzate per la costruzione del modello 3D. Nella prima colonna è visualizzato il modello in *wireframe*, nella seconda la lettura del sensore kinect, nella terza la discrepanza tra le prime due e nella quarta la texture relativa.

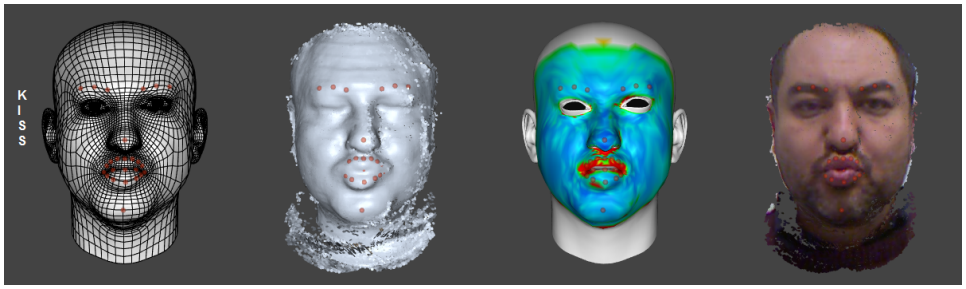


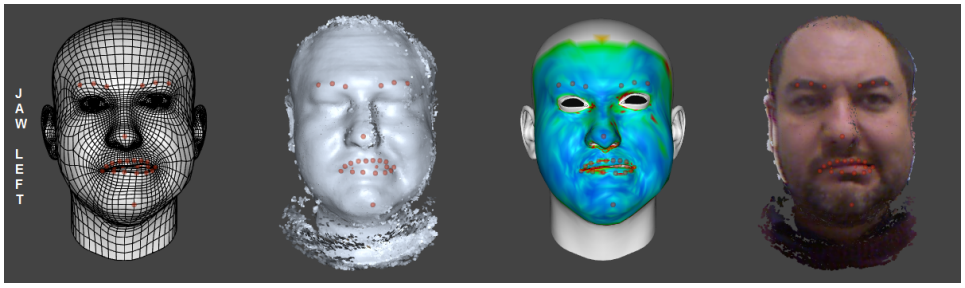

SNEER

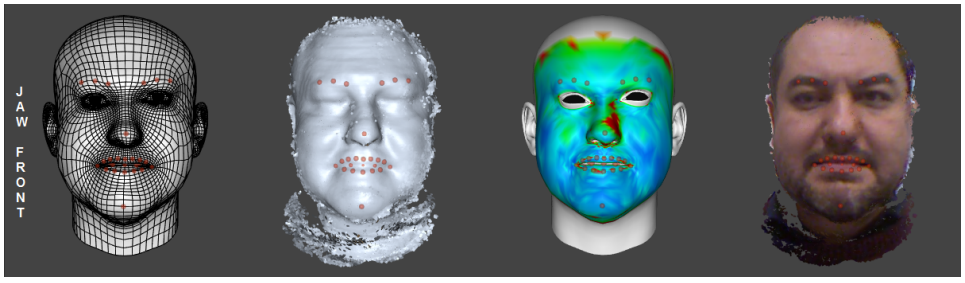

SMILE

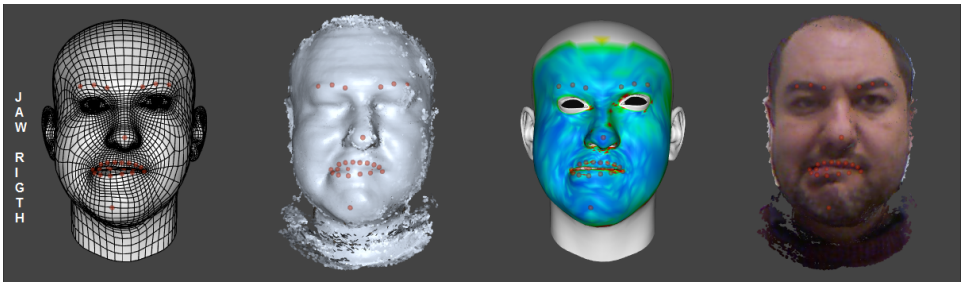

OPEN

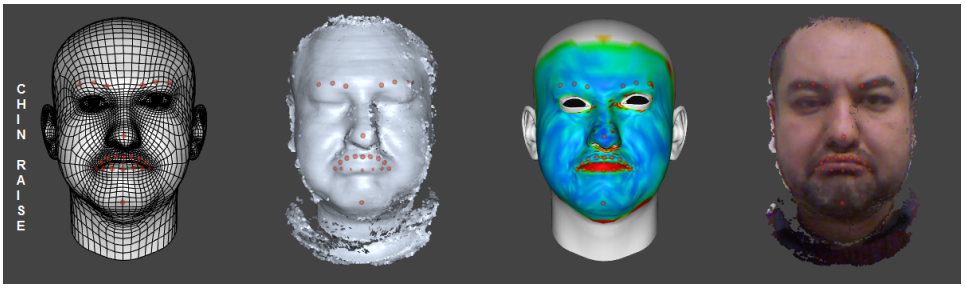

FUNNEL

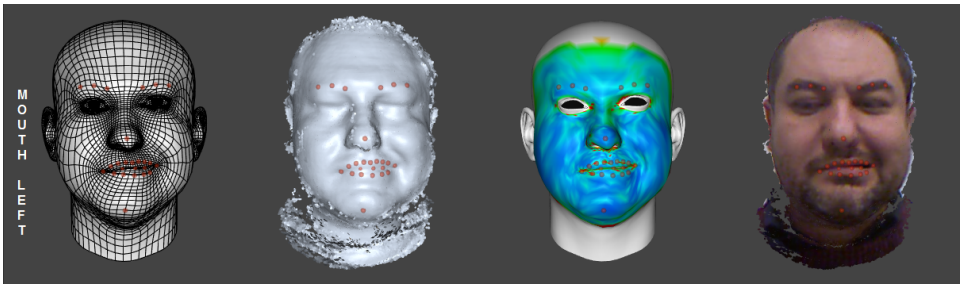

KISS

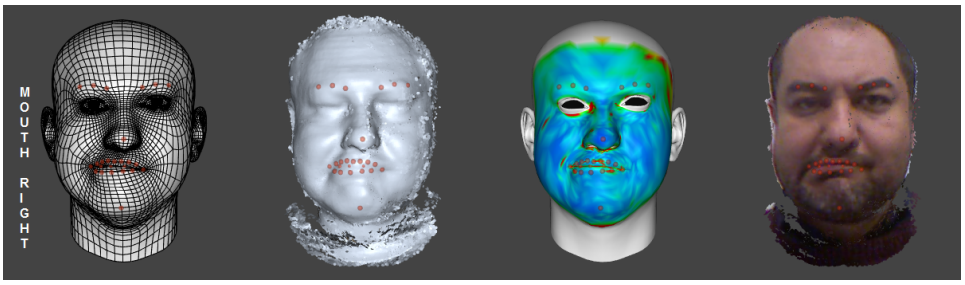

JAW LEFT

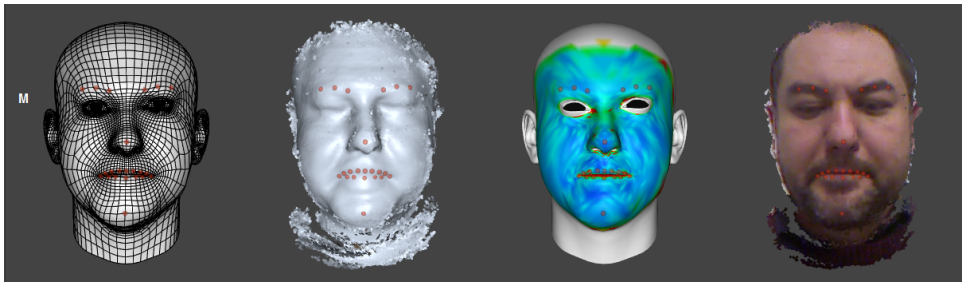

JAW FRONT

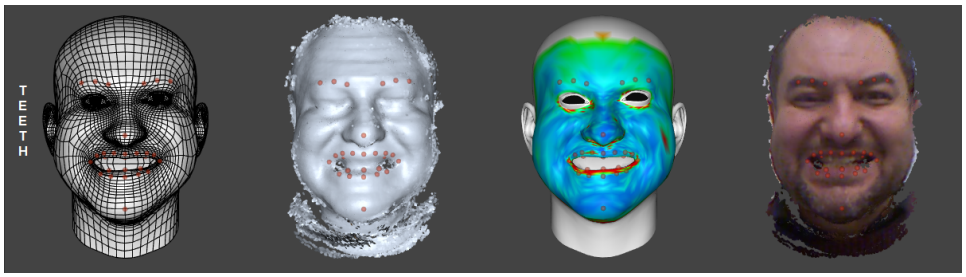

JAW RIGTH

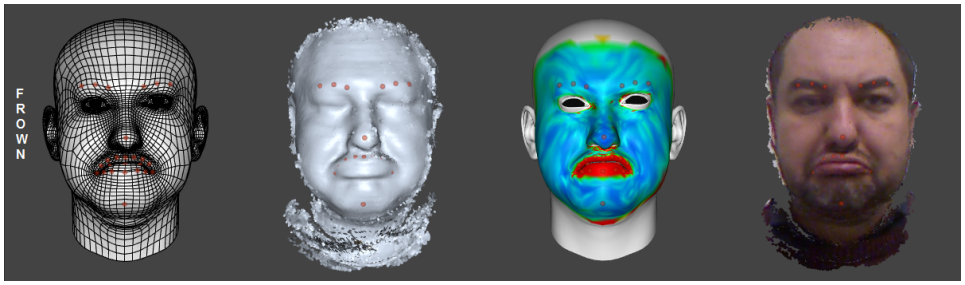

CHIN RAISE

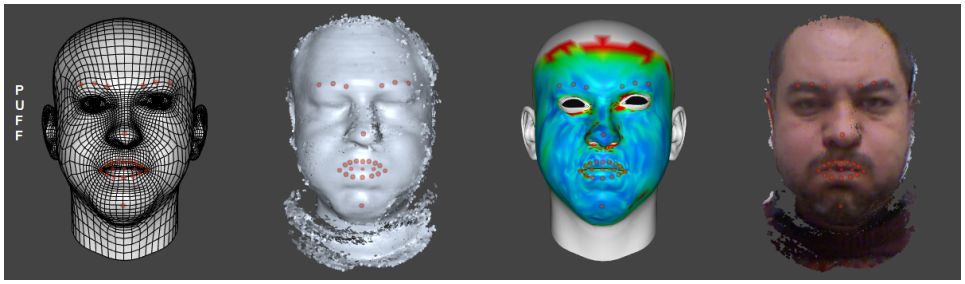

MOUTH LEFT

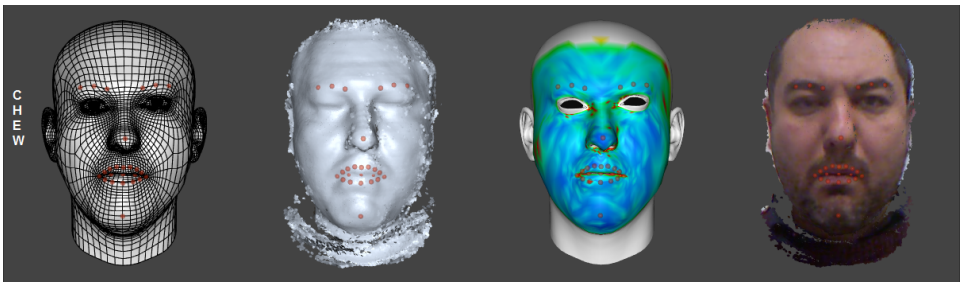

MOUTH RIGHT

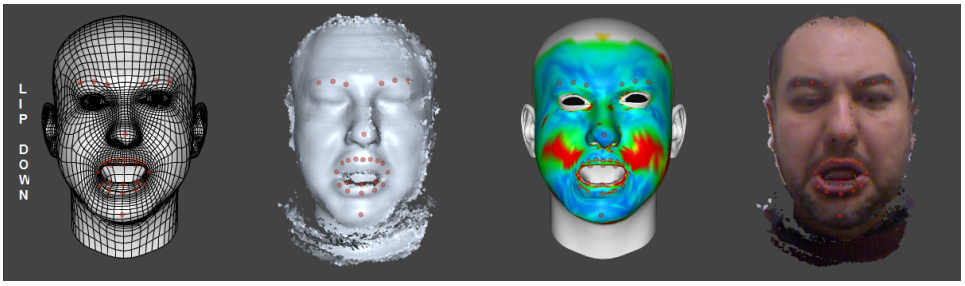

M

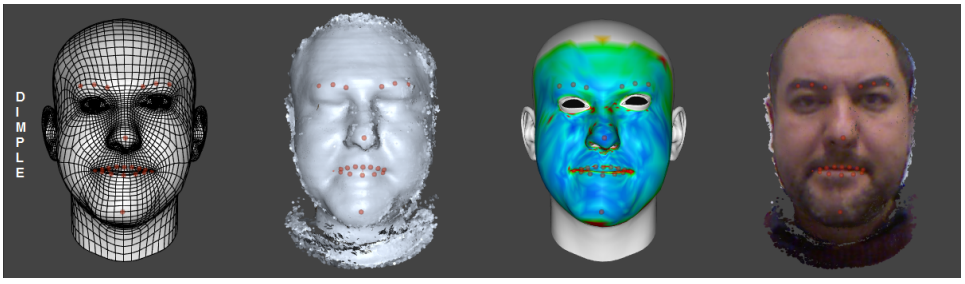

TEETH


FROWN


PUFF


CHEW


LIP DOWN


DIMPLE

Ringraziamenti

Questa tesi è stata resa possibile grazie al progetto WIKIMEMO.IT (Il portale della lingua e della Cultura Italiana, FIRB Project, RBNE078K93, Ministero dell'Istruzione, Università e Ricerca scientifica). E' stata parzialmente supportata dal progetto EU-FP7 ALIZ-E (ICT-248116).

Ringrazio di cuore il mio responsabile scientifico Piero Cosi e tutto il gruppo di ricerca di Padova dello ISTC-CNR per l'aiuto che mi hanno sempre dato senza riserve.

Ringrazio i miei tutor, i professori Giovanni Soda e Simone Marinai dell'università di Firenze per i consigli, la gentilezza e la disponibilità dimostrata negli anni.

Ringrazio la mia famiglia che mi ha sempre supportato e in particolare mio padre, che in questi anni ha sempre insistito perché portassi a termine questo percorso.

Ringrazio il mio quinquilino Vincenzo che mi ha preparato tante cene mentre lottavo contro il tempo per consegnare la tesi.

Ringrazio la mia compagna Ramona senza la quale non avrei raggiunto questo traguardo.

Giuseppe Riccardo Leone

Bibliografia

[1] C. Abry and M.T. Lallouache, *Does increasing representational complexity lead to more speech variability?*, Tech. Report 14: 1-5, Phonetic Experimental Research at the Institute of Linguistics, University of Stockholm, 1991.

[2] W. Barry A.J. Fourcin, G. Harland and W. Hazan Eds, *Speech input and output assessment, multilingual methods and standards*, Books in Information Technology, Ellis Horwood, 1989.

[3] A. Al-Bamerni and A. Blandon, *One-stage and two-stage patterns of velar coarticulation*, Journal of the Acoustical Society of America **72** (1982), no. 104.

[4] B. Angelini, F. Brugnara, and D. Falavigna, *A baseline of a speaker independent continuous speech recognizer of italian*, Proceedings of Eurospeech 93, 1993, pp. 847–850.

[5] C.G. Atkenson, A.W. Moore, and S. Schaal, *Locally weighted learning*, Artificial Intelligence Review (1985).

[6] K. Balci, *Xface: Mpeg-4 based open source toolkit for 3d facial animation*, Working Conference on Advanced Visual Interfaces, 2004.

[7] Gavin Bell, Anthony Parisi, and Mark Pesce, *The virtual reality modeling language*, `http://www.web3d.org/x3d/specifications/vrml/VRML1.0`, 1995.

[8] F. Bell-Berti and K.S. Harris, *A temporal model of speech production*, Phonetica **38** (1981), 9–20.

[9] A.P. Benguerel and H. Cowan, *Coarticulation of upper lip protrusion in french*, Phonetica **30** (1974), 41–55.

[10] A.P. Benguerel, H. Hirose, M. Sawaschima, and T. Ushijima, *Velar coarticulation in french: an electromyographic study*, Journal of Phonetics **5** (1997), 159–167.

[11] Alberto Benin, G. Riccardo Leone, and Piero Cosi, *A 3d talking head for mobile devices based on unofficial ios webgl support*, Proceedings of the 17th International Conference on 3D Web Technology (New York, NY, USA), Web3D '12, ACM, 2012, pp. 117–120.

[12] C. Benoit, S. Kandel, and T. Mohamadi, *Effects of phonetic context on audio-visual intelligibility of french*, Journal of Speech and Hearing Research **37** (1994), no. 3, 1195–1203.

[13] Jonas Beskow, *Rule-based visual speech synthesis.*, EUROSPEECH, ISCA, 1995.

[14] A. Black and N. Campbell, *Optimising selection of units from speech databases for concatenative synthesis*, Eurospeech, 1995.

[15] A. Bladon and A. Al-Bamerni, *Coarticulation resistance in english*, Journal of Phonetics **4** (1976), 137–150.

[16] R. Bladon and A. Al-Bamerni, *Coarticulation resistance in english*, Phonetics (1976), 135–150.

[17] Volker Blanz and Thomas Vetter, *A morphable model for the synthesis of 3d faces*, Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[18] P. Boersma, *Praat, a system for doing phonetics by computer*, Glot International (1996), 341–345.

[19] Christoph Bregler, Michele Covell, and Malcolm Slaney, *Video rewrite: Driving visual speech with audio*, Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 353–360.

[20] Emanuela Magno Caldognetto, Piero Cosi, Carlo Drioli, Graziano Tisato, and Federica Cavicchio, *Coproduction of speech and emotions: Visual and acoustic modifications of some phonetic labial targets*, AVSP 2003-International Conference on Auditory-Visual Speech Processing, 2003.

[21] N. Calzolari, B. Magnini, C. Soria, and M. Speranza, *La Lingua Italiana nell'Era Digitale – The Italian Language in the Digital Age*, META-NET White Paper Series. Georg Rehm and Hans Uszkoreit (Series Editors), Springer, 2012.

[22] B. De Carolis, C. Pelachaud, I. Poggi, and M. Steedman, *Apml, a mark-up language for believable behavior generation*, Life-Like Characters (2004), 65–85.

[23] N. Chomsky and M. Halle, *The sound pattern of english*, Harper and Row, New York, NY, 1968.

[24] W.S. Cleveland and C. Loader, *Smoothing by local regression: principles and methods*, Tech. report, AT&T Bell Laboratories: Murray Hill, NJ, 1995.

[25] C.H. Cocker, *A model of articulatory dynamics and control*, Proceedings of the IEEE **64** (1976), no. 4, 452–460.

[26] M.M. Cohen, J. Beskow, and D.W. Massaro, *Recent developments in facial animation: an inside view*, D. Burnam, J. Roberts-Ribes, E. Vatikiotis-Bateson, (Eds.), Proceedings of AVSP '98, Terrigal-Sidney, 1994, pp. 201–206.

[27] M.M. Cohen and D.W. Massaro, *Modeling coarticulation in synthetic visual speech*, M. Magnenat-Thalmann and D. Thalmann, (Eds.) Models and Techniques in Computer Animation, Springer-Verlag, 1993, pp. 139–155.

[28] P. Cosi and G. Perin, *Labial coarticulation modeling for realistic facial animation*, Proceedings of ICMI 2002 (Pittsburgh, USA), ICMI, 2002, pp. 505–510.

[29] P. Cosi, F. Tesser, R. Gretter, and C. Avesani, *Festival speaks italian!*, Proceedings of Eurospeech 2001 (Aalborg, Denmark), Eurospeech, 2001, pp. 509–512.

[30] Terzopoulos D. and Waters K., *Physically-based facial modelling, analysis, and animation*, Journal of Visualization and Computer Animation **1** (1990), no. 2, 73–90.

[31] R. Daniloff and K. Moll, *On defining coarticulation*, Journal of Speech and Hearing Research **1** (1973), 239–248.

[32] C. Drioli and P. Cosi, *Audio indexing for an interactive italian literature management system*, INTERSPEECH 2008, 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia, September 22-26, 2008, 2008, p. 2170.

[33] C. Drioli, P. Cosi, F. Tesser, and G. Tisato, *Emotions and voice quality: Experiments with sinusoidal modeling*, Proceedings of Voqual 2003 (Geneva, Switzerland), ISCA, 2003, pp. 127–132.

[34] Carlo Drioli, Fabio Tesser, Graziano Tisato, Piero Cosi, Enrico Marchetto, and Padova ISTC-CNR, *Control of voice quality for emotional speech synthesis*, Proceedings of AISV (2004), 789–798.

[35] P. Ekman and W. Friesen, *Facial action coding system*, Consulting Psychologists Press, Inc. Palo Alto, Ca, 1978.

[36] A. Emmett, *Digital portfolio: Tony de peltrie*, Computer Graphics World **8** (1985), no. 10, 72–77.

[37] FaceShift, `http://www.faceshift.com/`.

[38] E. Farnetani and D. Recaens, *Coarticulation models in recent speech production theories*, W.J. Hardcastle (eds), Coarticulation in Speech Production, Cambridge University Press, Cambridge, 1999.

[39] E. Farnetani and D. Recasens, *Coarticulation models in recent speech production theories*, Coarticulation in Speech Production (1999).

[40] G. Ferrigno and A. Pedotti, *Elite: A digital dedicated hardware system for movement analysis via real-time tv signal processing*, IEEE Transactions on Biomedical Engineering (1985), 943–950.

[41] J. D. Foley, S. K. Feiner A. Van Dam, and J. F. Hughes, *Computer graphics: Principles and practice.*, Addison-Wesley, MA, 1990. Second Edition.

[42] B. Le Goff and C. Benoit, *A french-speaking synthetic head*, C. Benoit and R. Campbell, (eds), Proceedings of AVSP '97, Rhodes, Greece, 1997, pp. 145–148.

[43] AA.VV. System Group, *System verification model 4.0*, ISO/IEC JTC1/SC29/WG11 N1693, Bristol, April 1997.

[44] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Fredric Pighin, *Making faces*, Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA), SIGGRAPH '98, ACM, 1998, pp. 55–66.

[45] T. Guiard-Marigny, *Modelisation tridimensionelle des articulateurs de la parole pour l'animation faciale: Implementation temps reel et mesure d'intelligibilite bimodale*, Ph.D. thesis, Institute National Polytechnique, Grenoble, France, 1996.

[46] W.L. Henke, *Dynamic articulatory model of speech production using computer simulation*, Unpublished doctoral dissertation, MIT Cambridge, Ma, 1966.

[47] ISO/IEC, *Information technology - coding of audio-visual objects - visual*, second ed., ISO/IEC 14496, vol. 2, International Standard Organization, Geneva, Switzerland., 2001.

[48] JSON, `http://www.json.org/`.

[49] P.A. Keating, *The window model of coarticulation: articulatory evidence*, UCLA Working Papers in Phonetics **69** (1988), 3–29.

[50] ———, *The window model of coarticulation: articulatory evidence*, M.E. Beckam, (eds.), Papers in Laboratory Phonetics I: between the grammar and the physics of speech, Cambridge University Press, 1990, pp. 451–470.

[51] Microsoft Kinect, `http://en.wikipedia.org/wiki/Kinect`.

[52] V. Kozhevnikov and L. Chistovich, *Speech: Articulation and perception*, 534, vol. 30, Joint Publications Research Service, Washington, DC, 1965.

[53] C. Lande and G. Francini, *An mpeg-4 facial animation system driven by synthetic speech*, Multimedia Modeling, Lousanne, October 12-15 1998.

[54] F. Lavagetto, *Time-delay neural networks for estimating lip movements from speech analysis: a useful tool, in audio-video synchronization*, IEEE Transaction on Circuits and System for Video Technology **7** (1997), no. 5, 786–800.

[55] B. Le Goff and C. Benoit, *A text-to-audiovisualspeech synthesizer for french.*, Proceedings of I.C.S.L.P. 1996, Philadelphia, USA, 1996, pp. 2163–2166.

[56] Y.C. Lee, D. Terzopoulos, and K. Waters, *Computer facial animation*, A. K. Peters, Ltd., Natick, MA, USA, 1996.

[57] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters, *Realistic modeling for facial animation*, Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA), SIGGRAPH '95, ACM, 1995, pp. 55–62.

[58] G.R. Leone and P. Cosi, *Lucia-webgl: A web based italian mpeg-4 talking head*, Proceedings of the International Conference on Audio-Visual Speech Processing 2011 (Volterra, Italy) (Giampiero Salvi, Jonas Beskow, Olov Engwall, and Samer Al Moubayed, eds.), 2011, pp. 123 – 126.

[59] G.R. Leone, G. Paci, and P. Cosi, *Lucia: An open source 3d expressive avatar for multimodal h.m.i.*, INTETAIN 2011, Intelligent Technologies, for Interactive Environments (Gualtiero Volpe Antonio Camurri, Cristina Costa, ed.), 2011, pp. 1 – 10.

[60] J. Lewis and F. Parke, *Automated lip-sync and speech synthesis for character animation*, Proceedings of CHI87, ACM, New York (1987), 143–147.

[61] Hao Li, Robert W. Sumner, and Mark Pauly, *Global correspondence optimization for non-rigid registration of depth scans*, Proceedings of the Symposium on Geometry Processing (Aire-la-Ville, Switzerland, Switzerland), SGP '08, Eurographics Association, 2008, pp. 1421–1430.

[62] B. Lindblom, *On vowel reduction*, Tech. Report 29, The Royal Institute of Technology, Speech Transmission Laboratory, Stockholm, 1963.

[63] A. Löfqvist and V. Gracco, *Lip and jaw kinematics in bilabial stop consonant production*, Journal of Speech and Hearing Research **40** (1997), no. 3, 877–893.

[64] J.C. Lucero, J.O. Ramsay, K.G. Munhall, and V.L. Gracco, *On the registration of time and the patterning of speech movements*, Journal of Speech Language and Hearing Research **40** (1997), 1111–1117.

[65] LUCIA, `http://www2.pd.istc.cnr.it/LUCIA/`.

[66] M. Lundeberg and J. Beskow, *Developing a 3d-agent for the august dialogue system*, Proceedings of the International Conference on Audio-Visual Speech Processing (Santa Cruz, USA) (D.W. Massaro, ed.), 1999, pp. 51–61.

[67] N. Magnenat-Thalmann, N.E. Primeau, and D. Thalmann, *Abstract muscle actions procedure for human face animation*, Visual Computer **3** (1988), no. 5, 290–297.

[68] E. Magno-Caldognetto, C. Zmarich, and P. Cosi, *Italian consonantal visemes: Relationships between spatial/temporal artculatory characteristics and coproduced acoustic signal*, C. Benoit and R. Campbell, (eds), Proceedings of AVSP '97, 1997, pp. 5–8.

[69] E. Magno-Caldognetto, C. Zmarich, and P. Cosi, *Statistical definition of visual information for italian vowels and consonants*, D. Burnham J., Robert-Ribes E., Vatikiotis-Bateson, (eds), Proceedings of AVSP '99, Univ. California, Santa Cruz, 1999, pp. 135–140.

[70] Dominic W. Massaro, Michael M. Cohen, Jonas Beskow, Sharon Daniel, and Ronald A. Cole, *Developing and Evaluating Conversational Agents*, (2000).

[71] D.F. McAllister, R.D. Rodman, D.L Bitzer, and A.S Freeman, *Lip synchronization of speech*, C. Benoit and R. Campbell, (eds), Proceedings of AVSP '97, 1997, pp. 133–136.

[72] H. McGurk and J. MacDonald, *Hearing lips and seeing voices*, Nature **264** (1976), 746–748.

[73] P. Mermelstein, *Articulatory model for the study of speech production*, Journal of the Acoustical Society of America **53** (1973), no. 4, 1070–1082.

[74] S. Morshima, *Real-time talking head driven by voice and its application to communication and entertainment*, Proc. AVSP 98 (1998).

[75] MPEG, `http://mpeg.chiariglione.org/standards/mpeg-4`.

[76] K.G. Munhall and A. Lofqvist, *Gestural aggregation in speech: laryngeal gestures*, Journal of Phonetics **20** (1992), 111–126.

[77] Mauro Nicolao, Carlo Drioli, and Piero Cosi, *Voice gmm modelling for festival/mbrola emotive tts synthesis.*, INTERSPEECH, 2006.

[78] S. Ohman, *Coarticulation in vcv utterances: spectrographic measurements*, Journal of acoustical Society of America **39** (1966), 151–168.

[79] _____, *Numerical model of coarticulation*, Journal of Acoustical Society of America **41** (1967), 310–320.

[80] Igor S. Pandzic and Robert Forchheimer (eds.), *Mpeg-4 facial animation: The standard, implementation and applications*, John Wiley & Sons, Inc., New York, NY, USA, 2003.

[81] F.I. Parke, *Computer generated animation of faces*, Master's thesis, University of Utah, UTEC-CSC-72-120, Salt Lake City, June 1972.

[82] S. Pasquariello, *Modello per l'animazione facciale in mpeg-4*, Ph.D. thesis, Universitá degli Studi di Roma la Sapienza, 2000.

[83] E.C Patterson, P.C. Litwinowicz, and N. Greene, *Facial animation by spatial mapping*, N. Magnenat-Thalmann and D. Thalmann, (Eds) Computer Animation '91, Springer-Verlag, 1991, pp. 45–58.

[84] A. Pearce, B. Wyvill, and D.R. Hill, *Speech and expression: a computer solution to face animation*, Graphics and Vision Interface '86 (1986), 136–140.

[85] C. Pelachaud, N.I. Badler, and M. Steedman, *Linguistic issues in facial animation*, N. Magnenat-Thalmann and D. Thalmann, editors, Computer Animation '91, Springer-Verlag, 1991, pp. 15–30.

[86] J.S. Perkell, *Testing theories of speech production: implications of same detailed analyses of variable articulatory data*, Hardcastle W.J., Marchal A.: Speech Production and Speech Modelling, Kluwer Publications, Dordrecht, 1990, pp. 263–288.

[87] J.S. Perkell and M. Matthies, *Temporal measures of anticipatory labial coarticulation for the vowel/u/: within-and cross-subject variability*, Journal of the Acoustical Society of America **91** (1992), 2911–25.

[88] F. Pighin, J. Auslander, and D. Lischinski, *Realistic facial expressions animations using image-based 3d morphing*, Tech. Report UW-CSE-97-01-03, Washington University, 1997.

[89] S.M. Platt, *A structural model of the human face*, Dept. of computer and information science, University of Pennsylvania, Philadelphia, PA, 1985.

[90] Tomaso Poggio and Federico Girosi, *A theory of networks for approximation and learning*, Laboratory, Massachusetts Institute of Technology **1140** (1989).

[91] M. J. D. Powell, *Algorithms for approximation*, Clarendon Press, New York, NY, USA, 1987, pp. 143–167.

[92] Helmut Prendinger and Mitsuru Ishizuka (eds.), *Life-like characters: Tools, affective functions, and applications*, Springer, Berlin, 2004.

[93] L. Reveret, G. Bailly, and P.Badin, *Mother: a new generation of talking heads providing a flexible articulatory control for video-realistic speech animation*, Proc. of the 6th Int. Conference of Spoken Language Processing, ICSLP'2000, Beijing, China, 2000, pp. 755–758.

[94] Z. Ruttkay, H. Noot, and P. Hagen, *Emotion disc and emotion squares: tools to explore the facial expression space*, Computer Graphics Forum, 2003, pp. 49–53.

[95] E.L. Salzman and K.G. Munhall, *A dynamical approach to gestural patterning in speech production*, Ecological Psychology **1** (1989), no. 4, 333–382.

[96] Daniel K. Schneider and Sylvere Martin-Michiellot, *Vrml primer and tutorial*, `http://tecfa.unige.ch/guides/vrml/vrmlman/vrmlman.html`, 1998.

[97] M. Schröder and J. Trouvain, *The german text-to-speech synthesis system mary: A tool for research, development and teaching*, Internation J. of Speech Technology, 2003.

[98] G.A. Schultz, R.S. Schnabel, and R.H. Byrd, *A family of trust-region-based algorithms for unconstrained optimization with strong global convergence properties*, SIAM Journal on Numerical Analysis **22** (1985), 47–67.

[99] A. Smith, M. Johnson, C. McGillem, and L. Goffman, *On the assessment of stability and patterning of speech movements*, Journal of Speech Language and Hearing Research **43** (2000), 277–286.

[100] Q. Summerfeld, *Lip-reading and audio-visual speech perception*, Philosophical transactions of the royal society of London **335** (1992), 71–78.

[101] D. Terzopoulos and K. Waters, *Analysis and synthesis of facial image sequences using physical and anatomical models*, IEEE Transactions on Pattern Analysis and Machine Intelligence **15** (1993), no. 6, 569–579.

[102] F. Tesser, P. Cosi, C. Drioli, and G. Tisato, *Emotional festival mbrola tts synthesis*, INTERSPEECH, 2005, pp. 505–508.

[103] F. Tesser, G. Paci, G. Sommavilla, and P. Cosi, *A new language and a new voice for MARY-TTS*, 9th national congress, AISV (Associazione Italiana di Scienze della Voce) (Venice, Italy), 2013.

[104] Fabio Tesser, Piero Cosi, Carlo Drioli, and Graziano Tisato, *Prosodic data driven modelling of a narrative style in festival tts*, Fifth ISCA Workshop on Speech Synthesis, 2004.

[105] G. Tisato, C. Drioli, P. Cosi, and F. Tesser, *Interface: a new tool for building emotive/expressive talking heads*, Proceedings of INTERSPEECH 2005 (Lisbon, Portugal), INTERSPEECH, 2005, pp. 781–784.

[106] Loquendo TTS, `http://bit.ly/1hd8jBj`.

[107] F. Ulgen, *A step toward universal facial animation via volume morphing*, In 6th IEEE International Workshop on Robot and Human communication, 1997.

[108] E. Vatikiotis-Bateson, K.G. Munhall, M. Hirayama, Y. Kasahara, and H. Yehia, *Physiology-based synthesis of audiovisual speech.*, Proceeding of 4th Speech Production Seminar: Models and Data, 1996.

[109] F. Vignoli and C. Braccini, *A text-speech synchronization technique with applications to talking heads*, D. Burnaham J., Robert-Ribes E., Vatikiotis-Bateson, (eds), Proceedings Of AVSP '99, 1999, pp. 128–132.

[110] C.L.Y. Wang and D.R. Forsey, *Langwidere: A new facial animation system*, Proceedings of Computer Animation (1994), 59–68.

[111] D. Ward and S. Arnfield, *Linear and nonlinear analysis of the stability of gestural organization in speech movement sequences*, Journal of Speech, Language, and Hearing Research **44** (2001), 108–117.

[112] K. Waters, *A muscle model for animating three-dimensional facial expressions*, Computer Graphics **21** (1987), no. 4, 17–24.

[113] WebGL, `http://www.khronos.org/webgl/`.

[114] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly, *Realtime performance-based facial animation*, ACM SIGGRAPH 2011 Papers (New York, NY, USA), SIGGRAPH '11, ACM, 2011, pp. 77:1–77:10.

[115] Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly, *Face/off: Live facial puppetry*, Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (New York, NY, USA), SCA '09, ACM, 2009, pp. 7–16.

[116] E. Yamamoto, S. Nakamura, and K. Shikano, *Lip movement synthesis from speech based on hidden markov models*, Speech Communication **105-115** (26).

[117] H. Zen, K. Tokuda, and A.W. Black, *Statistical parametric speech synthesis*, Speech Communication, 2009.