UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

INTERAZIONE EMOTIVA CON ROBOT AIBO

Relatore: Prof. Nunzio Alberto BORGHESE

Correlatori: Prof.ssa Paola CAMPADELLI

Prof. Massimiliano GOLDWURM

Ing. Iuri FROSIO

Tesi di Laurea di:

Isabella Cattinelli

Matr. Nr. 670715

ANNO ACCADEMICO 2004-2005

a Daniele, la mia stella sicura.

Indice

Abstract

Con la diffusione dei dispositivi robotici in settori sempre più ampi della società, cresce l'esigenza di nuove interfacce uomo-robot in grado non solo di rendere agevole l'utilizzo di questi strumenti ad un pubblico non specializzato, ma anche di favorirne una progressiva integrazione nella comunità umana. Ciò è particolarmente rilevante se pensiamo ai progetti di assistenza robotica ad anziani o malati: è essenziale che il paziente accetti la presenza e l'aiuto del robot, non ne sia spaventato e collabori attivamente con esso per il proprio benessere; d'altro canto, il robot dovrebbe avere la capacità di comprendere le necessità del paziente, sfruttando diversi canali di comunicazione. Dotare il robot, da una parte, degli strumenti necessari a interpretare lo stato emotivo dell'utente, tramite la lettura del suo linguaggio non verbale, e, dall'altra, della capacità di simulare un *proprio* stato emotivo consente di avvicinare entrambi gli obiettivi: la macchina assumerà tratti più umani e, quindi, maggiormente graditi e comprensibili all'essere umano, e potrà indovinare i bisogni dell'utente senza che questi debba esplicitarli tramite un insieme rigido e necessariamente limitato di comandi standard.

Il presente lavoro intende muoversi in questa direzione, esplorata da un numero sempre crescente di studi nel campo dell'*affective computing*. L'obiettivo consiste nel modellare un'**interazione emotiva** (vale a dire un mutuo scambio di emozioni) tra un essere umano e un robot, nella fattispecie AIBO, il cane robot di Sony pensato per l'intrattenimento [AIBO, 1999]. Ciò ha richiesto, da una parte, la progettazione di un

modulo dedicato all'analisi delle espressioni del volto dell'utente, da cui estrarre informazioni circa il suo stato d'animo, e, dall'altra, la ricerca di un modello appropriato per la descrizione della dinamica emotiva.

Per quanto riguarda il modulo di riconoscimento delle emozioni, sono state adottate tecniche base di elaborazione delle immagini (segmentazione sul piano colore YCbCr, filtri derivativi, block matching, ecc.) per determinare il movimento a cui ciascuna feature del volto è stata sottoposta nel passaggio dall'espressione neutrale a quella emotiva; i movimenti elementari del viso sono stati codificati secondo il diffuso Facial Action Coding System (FACS) [Ekman e Friesen, 1978] e impiegati come base per la composizione delle 6 emozioni fondamentali secondo Ekman [Ekman, 1992]: felicità, tristezza, sorpresa, rabbia, paura, disgusto.

L'emozione riconosciuta nell'utente diventa l'input al modello di interazione, costituito da un *automa a stati finiti probabilistico* [Rabin, 1963] [Paz, 1971]: in tale modello, gli stati dell'automa coincidono con i possibili stati emotivi del robot e le transizioni di stato sono regolate da probabilità determinate congiuntamente dai due concetti fondamentali di *personalità* e *comportamento*. Con il termine *personalità* ci riferiamo al set iniziale di probabilità di transizione: tale set sarà infatti diverso a seconda che la personalità che si intenda modellare sia amichevole (nel qual caso saranno privilegiate, generalmente, le transizioni verso lo stato di gioia), piuttosto che scontrosa. Tali probabilità subiscono un'evoluzione nel corso dell'interazione, in base alla qualità finora osservata degli input emotivi forniti dall'utente; il criterio secondo cui avviene l'aggiornamento delle probabilità prende il nome di *comportamento*. In caso di comportamento imitativo, per esempio, AIBO modificherà le probabilità di transizione in modo da uniformare il proprio atteggiamento a quello tenuto dall'utente.

Il riconoscimento delle emozioni e il successivo calcolo dello stato emotivo di AIBO avvengono in tempo reale, consentendo un'interazione fluida e maggiormente credibile; gli algoritmi utilizzati in sede di analisi delle immagini sono stati scelti al fine di minimizzare il tempo di calcolo e l'occupazione di memoria, due risorse critiche sulla

piattaforma robotica adottata. Sebbene tali tecniche risultino sensibili alle condizioni di illuminazione e richiedano pertanto l'introduzione di vincoli circa le modalità di presentazione del soggetto, risultano, per i nostri scopi, più che appropriate.

Il principale risultato di questa tesi consiste nello sforzo di modellizzazione dell'interazione emotiva uomo-robot: il carattere probabilistico del modello consente di ottenere interazioni sempre diverse, a parità di input; la combinazione dei due concetti di personalità e comportamento rende il dialogo emotivo più verosimile, ricco e dinamico.

Un tale modello può facilmente essere esteso ad includere un maggior numero di stati e di input ed una varietà di personalità e di comportamenti, in modo da arricchire l'interazione di nuove sfumature. Inoltre, la modularità del progetto implementato consente un futuro sviluppo di moduli cooperanti per il riconoscimento delle emozioni a partire da altri canali di comunicazione non verbale (intonazione del discorso, gestualità, ecc.), così da determinare in modo più robusto lo stato emotivo dell'utente, che potrà poi essere fornito direttamente al modulo di interazione.

Ringraziamenti

Raggiungere un traguardo, di qualsiasi natura sia, richiede impegno personale, forza di volontà e numerosi sacrifici. Tuttavia, questi ingredienti non bastano, da soli, ad assicurare il successo. Oltre ad una dose non indifferente di buona sorte, fondamentali sono l'aiuto e il sostegno, anche psicologico, di chi ci sta a fianco nel quotidiano. Se il mio lavoro è ora giunto felicemente al termine, buona parte del merito va a chi, anche inconsapevolmente, ha saputo darmi il suo appoggio in questi mesi di gestazione.

Ringrazio in primo luogo il mio relatore, Prof. Borghese, per avermi concesso l'opportunità di esplorare un campo stimolante dell'informatica, quello dell'*affective computing*, e per aver riposto nelle mie capacità una fiducia che, tuttora, non credo di meritare. Ringrazio i miei correlatori per tutti i consigli che da loro ho ricevuto e per la loro pronta disponibilità, sempre. Un grande grazie va a tutti i compagni di laboratorio (Gilberto – il nostro guru degli AIBO –, Mirko, Pino, Giovanni, e tutti gli altri), per i momenti di spensieratezza e le parole di conforto. Non posso dimenticare tutti coloro che hanno prestato il loro volto ai miei bizzarri esperimenti, regalandomi un po' del loro tempo e mostrandomi che, anche quando mi sembra di essere sola, c'è sempre qualcuno su cui posso contare.

Ringrazio i miei genitori, perché senza di loro non sarei qui (non solo letteralmente!) e Daniele, che ogni giorno dei nostri 6 anni insieme mi ha sostenuto, amato e sopportato senza sosta. Siete la mia famiglia, non occorre aggiungere altro.

A tutti coloro che hanno creduto in me, grazie.

Milano, 4 aprile 2006 Isabella Cattinelli

Capitolo 1

Introduzione

I do not feel pleasure. I am only an android.

Tratto da *Star Trek: The Next Generation*

Negli ultimi anni stiamo assistendo ad una progressiva e sempre più capillare diffusione dei robot nella nostra società. Mentre negli anni '60-'70 i robot erano comuni solo nelle industrie, dove fungevano da operai ideali, per via della loro forza fisica ed alta affidabilità, oggi queste figure dal sapore fantascientifico conquistano sempre nuovi spazi, dall'esplorazione spaziale, all'assistenza domestica, all'intrattenimento di grandi e piccoli. Il cane robot di Sony, **AIBO** [AIBO, 1999], creato esclusivamente a scopo ludico, testimonia proprio l'estensione della robotica alla sfera della vita comune e il desiderio di fare del robot non solo un utile strumento, ma anche un compagno di vita.

Affinché una macchina possa essere considerata una compagna di vita, tuttavia, occorre che essa manifesti una certa dose di *umanità*. Se in futuro avremo robot dedicati all'assistenza di anziani e malati, per esempio, dovremo assicurarci che la presenza di tali figure sia gradita e accettata dai pazienti: non solo dotando la macchina di un aspetto antropomorfo (si parla, in questo caso, di *androidi*), ma anche

degli strumenti adatti per *interagire* con le persone. Non basterà che il robot sia in grado di rispondere correttamente agli ordini che gli vengono impartiti, ma dovrà essere in grado di agire con gentilezza, di sorridere ad una battuta di spirito, di interpretare il linguaggio non verbale (pensiamo ad una smorfia di dolore, a un gemito, oppure a un sorriso) e reagire di conseguenza. In sintesi, il robot del futuro dovrà essere in grado di interagire anche emotivamente con gli esseri umani.

Sebbene i robot prodotti negli ultimi anni (e.g. [ASIMO, 1986], [QRIO, 2003]) mostrino sempre maggiori capacità (di calcolo, di movimento, ecc.), l'obiettivo di un'interazione emotiva convincente è ancora lontano. Ciò che per noi è immediato e istintivo, come riconoscere un'espressione di sorpresa o di rabbia nel nostro interlocutore, per una macchina è estremamente complesso, e richiede che siano risolti molti ordini di problemi. Ciononostante, diversi tentativi sono stati operati al fine di ottenere delle *macchine sociali*: ricordiamo, a titolo d'esempio, Kismet [Kismet, 1998], una testa robotica dalle fattezze antropomorfe, in grado di interagire con le persone, per esempio avvicinandosi all'interlocutore quando questi sia troppo distante o, al contrario, ritraendosi quando il suo spazio personale viene violato. A Kismet seguì Leonardo [Leonardo, 2001], un robot ispirato ad una creatura di fantasia e dotato di una vasta gamma di movimenti espressivi. Questi comportamenti danno la sensazione di trovarsi al cospetto di un essere senziente, con cui è realmente possibile comunicare (anche a livello emotivo), e agevolano l'integrazione dei dispositivi robotici nella nostra società. Sebbene l'idea di dotare un robot di una coscienza, e di reali sentimenti, sia ancora molto lontana, possiamo lavorare affinché esso *simuli* tali sentimenti, e sia in grado di recepirli negli altri (proprio come il comandante Data, della serie *Star Trek: The Next Generation* che, pur essendo un androide dalle abilità – anche sociali

– straordinarie, rimane pur sempre "solo un androide", senza la capacità di provare amore, odio o emozioni di alcun tipo).

Pur se a scala molto più ridotta, l'obiettivo di questa tesi si inserisce nel quadro presentato: dotare un robot, nella fattispecie AIBO, della capacità di interagire emotivamente con il proprio padrone umano. Più precisamente, ci si propone di permettere ad AIBO di riconoscere, tramite l'analisi dell'espressione del viso, l'emozione mostrata dall'essere umano che ha di fronte e di reagire ad essa esibendo il proprio *stato emotivo*. Più propriamente, il nostro progetto mira al riconoscimento di specifiche espressioni facciali intese come possibile veicolo di emozioni, piuttosto che al rilevamento delle emozioni stesse. Naturalmente, le emozioni umane sono fenomeni estremamente complessi e difficilmente penetrabili con gli strumenti a disposizione al giorno d'oggi; pertanto, dobbiamo accontentarci di analizzarne le sole manifestazioni esteriori (in questo caso, l'espressione del volto) e basare, su quest'unica informazione, la nostra inferenza dello stato emotivo sperimentato dall'utente. Ciò rappresenta, evidentemente, una forte semplificazione: un'espressione può essere pilotata per mostrare un'emozione diversa da quella effettivamente provata; inoltre, i vincoli imposti dal modulo di riconoscimento soffocano il carattere di spontaneità tipico di un'emozione. Tuttavia, allo stato attuale (e con la tecnologia a nostra disposizione) non è possibile effettuare un'analisi più approfondita dello stato emotivo dell'utente; per i nostri scopi, riteniamo sufficiente procedere al riconoscimento di fissate espressioni facciali cui associare specifiche emozioni. Nel seguito, quando si faccia riferimento al riconoscimento di emozioni, si intenderà sempre il riconoscimento di espressioni facciali ragionevolmente riconducibili ad emozioni.

Come detto, anche il robot disporrà di suoi propri stati emotivi, innescati dal

comportamento dell'utente nei suoi confronti; ciò consente all'utente di interagire con il robot non solo tramite comandi standard, ma anche tramite le proprie emozioni. Da un lato, quindi, AIBO è in grado di leggere e interpretare, con i limiti sopra illustrati, l'emozione provata dall'uomo, dall'altro manifesta le *proprie* emozioni, in risposta a quelle dell'utente: in questo modo, la coppia AIBO–uomo dà vita ad un'**interazione emotiva**, nel senso di uno scambio reciproco di emozioni. Il primo di questi aspetti, il *riconoscimento di espressioni facciali*, è già stato affrontato in un precedente lavoro [D'Angelo, 2005], di cui il presente intende proporsi come estensione; ma è il secondo aspetto, l'*interazione emotiva* vera e propria, il vero nucleo di questa tesi. Esso richiede una seria riflessione sulla natura delle emozioni, sul ruolo della personalità individuale e dell'atteggiamento nei confronti dell'interlocutore durante l'interazione, nonché la ricerca di un modello adeguato per la rappresentazione formale di tali concetti.

L'obiettivo è ambizioso e il compito è complesso, e ciò richiede l'introduzione di vincoli sulle condizioni di utilizzo del software (per quanto riguarda la fase di elaborazione delle espressioni facciali), di assunzioni e di semplificazioni (relativamente alla complessità degli aspetti psicologici). Ma crediamo che questo possa essere un piccolo passo verso la progettazione di automi sempre più completi, in grado non solo di semplificarci la vita tramite nuove tecnologie, ma anche di comunicare con noi in modo naturale, divertente e ricco di sfumature.

La presente tesi descrive il lavoro compiuto in questa direzione. Il capitolo 2 introduce le caratteristiche principali, sia hardware sia software, di AIBO, mentre il capitolo 3 affronta i concetti di emozione e di comunicazione non verbale. Il capitolo 4 descrive le tecniche utilizzate in questo lavoro di tesi per l'individuazione del volto

dell'interlocutore nell'immagine registrata dal robot e, successivamente, delle regioni a maggior contenuto espressivo: bocca, occhi e sopracciglia; questo passo preliminare guida il successivo processo di riconoscimento delle espressioni, che viene descritto nel capitolo 5. Una volta determinata l'emozione espressa dall'utente, AIBO deve agire di conseguenza: l'interazione emotiva vera e propria viene affrontata nel capitolo 6. Infine, il capitolo 7 riporta i risultati ottenuti, una panoramica sul lavoro svolto e possibili sviluppi ed estensioni del progetto.

Capitolo 2

Sony AIBO: panoramica del sistema

La piattaforma su cui è stato sviluppato il progetto oggetto della presenti tesi è AIBO [AIBO, 1999] (vd. Fig. 2.1), un robot prodotto da Sony, destinato all'intrattenimento ma impiegato sempre più frequentemente anche nell'ambito universitario come mezzo per la ricerca nel campo della robotica e dell'intelligenza artificiale. In questo capitolo descriveremo brevemente le caratteristiche hardware del sistema e le peculiarità del suo ambiente di programmazione [D'Angelo et al., 2004] [Serra e Baillie, 2003] [D'Angelo e Colombo, 2004].

Figura 2.1: AIBO ERS-7.

2.1 L'hardware di AIBO

AIBO ERS-7 è un sistema piuttosto sofisticato, dotato di numerosi sensori ed attuatori, oltre che di un processore RISC a 64 bit, con clock a 576 Mhz e 64 MB di RAM. AIBO dispone di sensori a infrarossi per il calcolo delle distanze, di sensori di accelerazione e vibrazione e di sensori a pressione, posizionati sulla testa, sulla schiena, sotto il mento e sotto le zampe del cane robot; tali sensori costituiscono uno dei mezzi di interazione con il mondo esterno e con l'utente.

AIBO riceve informazioni dall'esterno tramite la propria camera CMOS a 350.000 pixel, al ritmo di 30 frame al secondo, e tramite microfoni stereo, posizionati sulle orecchie; a sua volta, può comunicare con l'emissione di suoni (è dotato di speaker) o di luci: ha infatti ben 28 LED distribuiti su tutto il corpo. Inoltre, AIBO possiede una scheda di rete WiFi (IEEE 802.11b), particolarmente utile al programmatore per il monitoraggio dello stato d'esecuzione dei programmi sul robot.

Naturalmente, AIBO può muoversi nell'ambiente e interagire con oggetti di varia natura: in totale, il suo corpo robotico ha 20 gradi di libertà, tra testa, zampe, orecchie e coda. La batteria in dotazione fornisce circa 90 minuti di autonomia al sistema, prima che una nuova ricarica si renda necessaria.

AIBO è un sistema *programmabile*: sebbene venga venduto equipaggiato di un software, chiamato AIBO MIND (attualmente alla versione 3), che gestisce le funzioni standard del robot (dalla locomozione al riconoscimento degli oggetti in dotazione, dai balletti allo scatto di istantanee), è possibile scrivere programmi ad hoc e caricarli su dispositivi di memorizzazione chiamati *Sony AIBO Programmable Memory Stick*. È quindi possibile programmare nuove funzioni e nuovi comportamenti per il robot, ottenendo un sistema personalizzabile, versatile ed espandibile.

2.2 OPEN-R SDK

AIBO monta un sistema operativo dedicato, chiamato *Aperios*. Per la programmazione del sistema, Sony ha reso disponibile un ambiente di sviluppo, basato sul linguaggio C++, dal nome **OPEN-R SDK**[1].

Un programma OPEN-R è, in realtà, una collezione di *oggetti OPEN-R* concorrenti, in grado di comunicare tra loro tramite il passaggio di messaggi; si parla perciò di programmi *modulari*. In una comunicazione possiamo distinguere un *soggetto*, cioè l'oggetto che invia il messaggio, e un *osservatore*, il destinatario; l'osservatore deve annunciare al soggetto di essere pronto a ricevere un nuovo messaggio mediante l'invio di un segnale speciale, l'*ASSERT_READY*, usualmente spedito al termine dell'elaborazione del messaggio precedente. I vari oggetti comunicano lungo canali unidirezionali, dove vi sono un unico soggetto e un unico osservatore, i cui ruoli sono fissati; questo significa che, per consentire a due oggetti di comunicare in maniera bidirezionale, dovranno essere creati due distinti canali. Inoltre, ogni canale può trasportare solo una tipologia di messaggio; perciò, se un soggetto desiderasse inviare allo stesso osservatore due tipi di messaggi (e.g., un intero e un array), dovrebbero essere istituiti due diversi canali. Nella pratica, la comunicazione infra-oggetti richiede la compilazione di due file di configurazione: `stub.cfg`, specifico per ciascun oggetto, e `connect.cfg`. Il primo elenca i *servizi* dell'oggetto (cioè i suoi soggetti e osservatori), mentre il secondo interconnette gli oggetti assegnando a ciascun soggetto il proprio osservatore e viceversa.

Un oggetto OPEN-R, generalmente, si trova, in ogni istante, in uno (e in uno soltanto) specifico stato, scelto nell'insieme degli stati definiti dal programmatore per

[1] Il sito web di riferimento per OPEN-R è *http://openr.aibo.com/*.

quell'oggetto; la transizione da uno stato ad un altro è attivata dai messaggi provenienti dagli altri oggetti e controllata da condizioni; infatti, l'arrivo di un messaggio può innescare la transizione a più stati successivi, e la scelta tra questi dello stato in cui entrare viene effettuata in base al soddisfacimento (esclusivo) della condizione che regola quella specifica transizione.

Ogni oggetto OPEN-R deriva dalla compilazione di una classe C++, che deve ereditare dalla classe base, *OObject*. Ciascuna classe deve inoltre implementare quattro funzioni standard:

- `OStatus DoInit(const OSystemEvent& event)`: questa funzione viene invocata al caricamento dell'oggetto e si occupa di istituire i canali di comunicazione di quest'ultimo con i vari soggetti e osservatori.

- `OStatus DoStart(const OSystemEvent& event)`: al termine dell'esecuzione di `DoInit()` in ogni oggetto, viene eseguita la `DoStart()`, che generalmente si occupa di inviare un messaggio di *ASSERT_READY* a tutti gli osservatori.

- `OStatus DoStop(const OSystemEvent& event)`: allo spegnimento del sistema, questa funzione chiude i canali di comunicazione ed invia a tutti gli osservatori un *DEASSERT_READY*, col quale annuncia l'impossibilità dell'oggetto a ricevere ulteriori messaggi.

- `OStatus DoDestroy(const OSystemEvent& event)`: è l'ultima funzione ad essere invocata, quando tutti gli oggetti abbiano eseguito la propria `DoStop()`.

L'architettura OPEN-R distingue il livello di sistema, contenente gli oggetti dedicati al controllo dell'hardware, caricati da Aperios ad ogni avvio di AIBO, e il livello delle applicazioni, che include gli oggetti definiti dal programmatore. L'interfaccia

tra i due livelli è costituita da un insieme di oggetti, tra cui particolare importanza rivestono *OVirtualRobotComm* (per il controllo di giunture, sensori e telecamera) e *OVirtualRobotAudioComm* (per la gestione dei dispositivi audio). Tramite il passaggio di messaggi con questi oggetti, è quindi possibile, per esempio, recuperare l'immagine acquisita dalla camera o provocare il movimento della coda di AIBO. Più dettagliatamente, l'oggetto *OVirtualRobotComm* dispone di due soggetti: `Sensor`, che fornisce informazioni sui sensori, e `FbkImageSensor`, che restituisce l'immagine acquisita dalla camera. Queste informazioni vengono inviate in messaggi e memorizzate in strutture dati dedicate, rispettivamente *OSensorFrameVectorData* e *OFbkImageVectorData*. Inoltre, *OVirtualRobotComm* ha un osservatore, `Effector`, a cui vengono inviati, nell'apposita struttura *OCommandVectorData*, i comandi relativi a giunture e LED. Infine, *OVirtualRobotAudioComm* dispone dell'osservatore `Speaker`, che riceve le informazioni audio codificate nella struttura dati *OSoundVectorData*. Pertanto, con un opportuno scambio di messaggi con questi servizi base, è possibile gestire agevolmente ogni input e output del sistema.

2.2.1 La gestione delle immagini

Di particolare interesse, ai fini di questo lavoro di tesi, sono le caratteristiche del sistema di acquisizione e trattamento delle immagini su AIBO. Le immagini a colori non sono rappresentate nel diffuso spazio colore RGB, ma nel formato **YCbCr**: cioè, ciascuna immagine è codificata sulle bande Y, che rappresenta la luminanza, Cb, che visualizza la componente cromatica rossa, e Cr, che registra la componente cromatica blu. Pertanto, ogni immagine a colori è in realtà l'insieme di tre immagini monocromatiche, accessibili separatamente.

Ciascuna immagine può essere elaborata su quattro distinti *layer*:

- `ofbkimageLAYER_L` rappresenta l'immagine alla risoluzione più bassa, 52×40 pixel;
- `ofbkimageLAYER_M` rappresenta l'immagine alla risoluzione media, 104×80;
- `ofbkimageLAYER_H` rappresenta l'immagine alla risoluzione alta, 208×160;
- `ofbkimageLAYER_C` è il *color detection layer*, di risoluzione 104×80.

AIBO possiede, infatti, un algoritmo, codificato via hardware, per la rilevazione di specifici colori nell'immagine, fino a un massimo di 8. Ciascun colore che siamo interessati a individuare sarà descritto da un canale, programmabile dall'utente mediante l'inserimento di intervalli di valori per i piani Cb e Cr. Più precisamente, i valori sul piano Y vengono suddivisi in 32 intervalli; per ciascuno, il programmatore deve specificare un valore minimo e uno massimo sia per il piano Cb sia per il piano Cr. L'algoritmo confronterà i valori di ciascun pixel con quelli definiti dal programmatore e verificherà pertanto l'appartenenza del pixel al colore di interesse. Il risultato di questo processo viene memorizzato nel *color detection layer* e può essere visualizzato, per ogni canale colore definito, sotto forma di un'immagine binaria in cui sono mostrati i soli pixel appartenenti al colore cercato. L'algoritmo descritto, quindi, rappresenta uno strumento rapido ed efficace per la segmentazione dell'immagine in regioni a diversa colorazione.

Oltre ai tre livelli di risoluzione sopraccitati, è possibile, con una decompressione basata sulla *trasformata di Haar*, ricostruire l'immagine corrente, in bianco e nero,

ad una risoluzione di 416×320. Questo consente di ottenere, nell'applicazione di tecniche di elaborazione delle immagini, risultati più precisi di quanto sarebbe possibile lavorando alle risoluzioni più basse fornite direttamente da AIBO.

Capitolo 3

Cosa sono le emozioni?

Dal momento che il presente lavoro di tesi si propone di realizzare un'*interazione emotiva* tra il robot AIBO ed un essere umano, occorre, in primo luogo, definire cosa si intenda con il termine *emozione*. In questo capitolo daremo quindi le nozioni base di psicologia necessarie per comprendere la natura delle emozioni e i canali mediante i quali vengono espresse.

3.1 Definire le emozioni

Sebbene il termine "emozione" sia utilizzato abitualmente nel linguaggio comune, fornirne una definizione completa che possa essere condivisa universalmente appare molto complesso; allo stesso modo, sottili sono le distinzioni tra il concetto di emozione e concetti affini, come sentimento, atteggiamento, ecc. Secondo il vocabolario Treccani[1], un'emozione è

> *impressione viva, turbamento, eccitazione. In psicologia, il termine indica genericamente una reazione complessa di cui entrano a far parte variazioni*

[1]Sito web: *www.treccani.it.*

> *fisiologiche a partire da uno stato omeostatico di base ed esperienze soggettive variamente definibili (sentimenti), solitamente accompagnata da comportamenti mimici.*

La natura complessa del fenomeno emozione è riconosciuta da Scherer [Scherer, 2005], che la definisce come

> *un episodio di cambiamenti correlati e sincronizzati fra loro negli stati di tutti o della maggior parte dei cinque sottosistemi dell'organismo, in risposta alla valutazione di un evento di stimolo interno o esterno come rilevante per i principali interessi dell'organismo.*

Scherer definisce *componenti* di un'emozione gli stati di cinque sottosistemi: componente cognitiva, componente neurofisiologica, componente motivazionale, componente di espressione motoria e componente di sentimento soggettivo. Secondo questa definizione, pertanto, emozione e sentimento si distinguono in quanto il secondo termine rappresenta una sola componente del processo multi-modale che il primo termine denota. Ulteriori distinzioni possono essere affrontate. Per esempio, Scherer definisce

- *atteggiamento* una predisposizione relativamente persistente verso persone od oggetti specifici;
- *umore* uno stato affettivo diffuso, generalmente di bassa intensità e durata estesa, che può emergere anche senza una causa apparente che possa essere collegata ad uno specifico evento;

- *disposizione affettiva* la tendenza di una persona a sperimentare più frequentemente certi stati d'animo o a reagire con determinati tipi di emozione. La disposizione rappresenta il nucleo affettivo di molti tratti stabili della personalità (per esempio, è una disposizione affettiva essere irritabile o ostile);

- *posizione interpersonale* uno stile affettivo che si sviluppa spontaneamente o viene utilizzato in maniera strategica nell'interazione con una persona (per esempio, rientra in questa definizione l'essere educato o freddo).

Nel linguaggio comune, i termini illustrati sono spesso impiegati come sinonimi e, anche tra i ricercatori, non sempre c'è accordo sulla semantica associata a questi concetti. Analogamente, è controverso determinare il numero delle possibili emozioni. Ekman [Ekman, 1992] ha proposto l'esistenza di 6 emozioni di base, o *universali*: felicità, tristezza, sorpresa, rabbia, paura e disgusto. Ekman utilizza l'aggettivo *basic* (di base) investendolo di un duplice significato: da una parte, le emozioni individuate sono di base in quanto differiscono tra loro non solo per quanto riguarda la loro espressione, ma anche per altri importanti aspetti (fisiologia, eventi antecedenti, ecc.); dall'altra, *di base* vuole indicare che tali emozioni sono il risultato di un'evoluzione mirata a prepararci ad affrontare le attività fondamentali della vita (le emozioni di base hanno cioè un valore adattativo). Le 6 categorie indicate non sono singoli stati affettivi, ma piuttosto *famiglie* di stati correlati; ciascun membro di una famiglia ha in comune con gli altri certe caratteristiche (per esempio, similarità d'espressione), che differiscono per ogni gruppo di emozioni. Pertanto, una famiglia di emozioni è costituita da un *tema*, cioè l'insieme delle caratteristiche fondamentali e specifiche della famiglia, e da *variazioni*, dovute a varie influenze (per esempio, variazioni d'intensità). Altri autori hanno suggerito un diverso numero ed una diversa varietà di

emozioni di base. Anche in questo caso, non vi è convergenza unanime su un'unica teoria, sebbene la classificazione di Ekman sia probabilmente la più diffusa; per questo motivo, e per l'essenzialità delle categorie emotive che individua, abbiamo scelto di adottarla nel presente progetto.

Anche la misurazione delle emozioni è resa problematica dalla mancanza di una terminologia condivisa. Naturalmente, non esistono metodi oggettivi per misurare l'esperienza soggettiva di una persona durante un episodio emotivo: l'unica via per conoscere tale esperienza consiste nell'interrogare il soggetto stesso. In molti esperimenti al soggetto viene fornita una lista precompilata di etichette affettive tra cui scegliere per descrivere il proprio stato emotivo; questo approccio, sebbene consenta di ottenere dati omogenei e standardizzati, rischia di suggerire al soggetto risposte che, altrimenti, non avrebbe considerato, e gli impedisce di rispondere facendo uso di una categoria non disponibile tra le opzioni date ma che, a suo avviso, può rivelarsi più appropriata al suo corrente stato. Questo può verificarsi anche nel caso in cui l'etichetta corrispondente all'emozione che il partecipante al test sperimenta sia effettivamente presente nella lista fornita, ma il termine usato non sia familiare al soggetto. L'alternativa consiste nell'impiegare un formato libero per le risposte che, però, può mettere in difficoltà i partecipanti meno abili a trovare etichette appropriate per le proprie emozioni e poco si presta ad un'analisi quantitativa, a causa dell'alto numero di termini che possono essere utilizzati e della bassa frequenza di ciascuno di essi. Per superare questo inconveniente, i ricercatori raggruppano le risposte libere in categorie più generali, basandosi, per esempio, sulle sinonimie; tuttavia, non esiste una procedura universalmente adottata per operare una tale classificazione. In [Scherer, 2005] è proposta una categorizzazione in 36 gruppi basata sul riconoscimento di specifici

termini in un file di testo contenente le etichette emerse durante un esperimento; per esempio, la categoria *surprise* sarà costituita da quelle etichette aventi radice *amaz**, *astonish**, *wonder**, ecc. Sulla base degli spunti presenti in [Scherer, 2005] sono stati scelti i termini per identificare gli stati emotivi del modello di interazione (vd. Sezione 6.2.1); la scelta è, in qualche modo, arbitraria, poiché, come visto, manca un dizionario di termini emotivi che sia diffusamente adottato.

3.2 La comunicazione non verbale

L'espressione delle emozioni è una delle funzioni essenziali della *comunicazione non verbale* (CNV – anche detta *linguaggio del corpo*) [Argyle, 1975]. Essa riveste un ruolo centrale nel comportamento sociale dell'uomo ed è alla base dei rituali, della presentazione di sé, della comunicazione di atteggiamenti interpersonali. Sebbene l'essere umano disponga di un articolato linguaggio verbale per esprimersi al meglio, la CNV mantiene un compito comunicativo di primaria importanza, per una serie di ragioni. Per esempio:

- i segnali non verbali sono generalmente più potenti, specie nell'ambito degli atteggiamenti interpersonali, e meno controllati, il che li rende più autentici (al contrario, le parole non sempre comunicano la verità);
- la CNV rappresenta un canale di supporto al linguaggio, fornendo in maniera immediata segnali di sincronizzazione e di feedback durante un dialogo, senza richiedere interruzioni nel discorso;

- in alcune aree si registra una mancanza di codificazione verbale (per esempio, esistono relativamente poche parole per le forme), a cui sopperisce l'uso della gestualità.

Possiamo distinguere tra numerosi *canali* di comunicazione: espressione facciale, sguardo (e dilatazione delle pupille), gesti ed altri movimenti del corpo, postura, contatto fisico, comportamento spaziale, abbigliamento e altri componenti dell'aspetto esteriore, vocalizzazioni non verbali, odore. I segnali emessi tramite i canali della CNV possono essere intenzionali (nel qual caso si parla propriamente di *comunicazione*) oppure non intenzionali (*comportamento* non verbale); spesso, i segnali non verbali comprendono sia una componente intenzionale sia una non intenzionale: per esempio, le espressioni facciali dell'emozione sono in parte costituite dall'espressione spontanea e in parte da tentativi di controllarla o di nasconderla. La CNV ha luogo ogni volta che una persona ne influenza un'altra attraverso uno qualsiasi dei canali elencati. In generale, uno schema di comunicazione non verbale prevede un emittente che codifica il proprio stato (per esempio, il suo stato emotivo) mediante uno specifico segnale, che il ricevente provvede a decodificare (vd. Figura 3.1); naturalmente, la correttezza della decodifica non è assicurata (a causa dell'inefficacia dell'emittente, del ricevente, o di entrambi, oppure perché il messaggio inviato è ingannevole, ecc.).

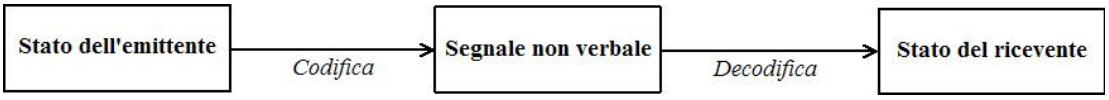


Figura 3.1: Schema di una comunicazione non verbale.

In particolare, siamo interessati alla CNV come veicolo per l'espressione di emozioni. A questo scopo, i canali più informativi sono il volto (in particolare bocca,

sopracciglia e movimento facciale), gli occhi (apertura oculare, dilatazione della pupilla), i gesti, la postura e il tono di voce. Tra questi, è predominante il ruolo del viso. Lo studio delle espressioni facciali richiede un sistema di codifica che individui gli elementi di base, la cui composizione produrrà le diverse espressioni: il sistema più diffuso in questo campo è il *Facial Action Coding System*.

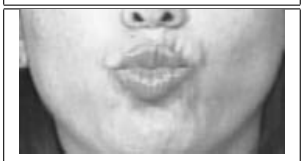
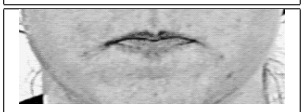
3.2.1 Facial Action Coding System

Il **Facial Action Coding System** (FACS) [Ekman e Friesen, 1978] è uno schema, sviluppato da Ekman e Friesen, per la descrizione delle espressioni facciali sulla base dei movimenti muscolari che concorrono a comporla. Gli elementi base per la costruzione di un'espressione sono detti *action unit* (AU) e rappresentano azioni facciali minime, cioè non ulteriormente separabili in azioni più semplici. Azioni muscolari e AU non coincidono: infatti, una AU può corrispondere all'azione di uno o più muscoli e uno stesso muscolo può essere associato a diverse AU. Un'AU rappresenta, in poche parole, una modifica elementare nell'apparenza del volto, causata dall'attivazione di uno o più muscoli facciali. Di seguito riportiamo una tabella riassuntiva[2] delle principali AU definite nel FACS.

AU	**Descrizione**	**Muscolo**	**Immagine**
1	Inner Brow Raiser	*Frontalis, pars medialis*	
2	Outer Brow Raiser	*Frontalis, pars lateralis*	

[2]La tabella è stata tratta dal sito web *http://www.cs.cmu.edu/afs/cs/project/face/www/facs.htm*.

AU	**Descrizione**	**Muscolo**	**Immagine**
4	Brow Lowerer	*Corrugator supercilii, Depressor supercilii*	
5	Upper Lid Raiser	*Levator palpebrae superioris*	
6	Cheek Raiser	*Orbicularis oculi, pars orbitalis*	
7	Lid Tightener	*Orbicularis oculi, pars palpebralis*	
9	Nose Wrinkler	*Levator labii superioris alaquae nasi*	
10	Upper Lip Raiser	*Levator labii superioris*	
11	Nasolabial Deepener	*Zygomaticus minor*	
12	Lip Corner Puller	*Zygomaticus major*	
13	Cheek Puffer	*Levator anguli oris (a.k.a. Caninus)*	
14	Dimpler	*Buccinator*	

AU	Descrizione	Muscolo	Immagine
15	Lip Corner Depressor	*Depressor anguli oris (a.k.a. Triangularis)*	
16	Lower Lip Depressor	*Depressor labii inferioris*	
17	Chin Raiser	*Mentalis*	
18	Lip Puckerer	*Incisivii labii superioris and Incisivii labii inferioris*	
20	Lip stretcher	*Risorius w/ platysma*	
22	Lip Funneler	*Orbicularis oris*	
23	Lip Tightener	*Orbicularis oris*	
24	Lip Pressor	*Orbicularis oris*	
25	Lips part	*Depressor labii inf. or relaxation of Mentalis, or Orbicularis oris*	

AU	**Descrizione**	**Muscolo**	**Immagine**
26	Jaw Drop	*Masseter, relaxed Temporalis and internal Pterygoid*	
27	Mouth Stretch	*Pterygoids, Digastric*	
28	Lip Suck	*Orbicularis oris*	
41	Lid droop	*Relaxation of Levator palpebrae superioris*	
42	Slit	*Orbicularis oculi*	
43	Eyes Closed	*Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis*	
44	Squint	*Orbicularis oculi, pars palpebralis*	
45	Blink	*Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis*	Non disponibile.
46	Wink	*Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis*	Non disponibile.

Il manuale FACS fornisce un'accurata descrizione, comprensiva di fotografie e filmati, per ogni AU, e spiega come riconoscerle nel quadro di un'espressione facciale. Un decodificatore FACS, quindi, è in grado di decomporre l'espressione che osserva nell'insieme delle AU che la costituiscono, registrando anche attributi aggiuntivi come durata, intensità, asimmetria. Il FACS è un sistema esclusivamente descrittivo e non si propone di fornire indicazioni circa il significato da attribuire all'espressione rilevata. Tuttavia, è piuttosto naturale associare gruppi di AU ad espressioni di specifiche emozioni. Per esempio [Parke e Waters, 1996]:

- Nella *felicità*, le palpebre vengono compresse dalle guance, che si sollevano (AU6), mentre gli angoli della bocca si alzano (AU12).
- Nella *tristezza*, le parti interne delle sopracciglia si sollevano e si avvicinano (AU1 + AU2 + AU4) e gli angoli della bocca sono rivolti verso il basso (AU15).
- Nella *sorpresa*, le sopracciglia sono curvate e sollevate (AU1 + AU2) e la mascella scende, causando l'apertura della bocca (AU25).
- Nella *rabbia*, le sopracciglia sono abbassate (AU4), mentre la bocca può essere serrata (AU24) o, al contrario, aperta per mostrare i denti (AU25).
- Nella *paura*, le sopracciglia sono sollevate e raddrizzate (AU1 + AU2 + AU4) e gli occhi spalancati (AU5); gli angoli della bocca possono tirare verso i lati, oppure la bocca può essere aperta per l'abbassamento del labbro inferiore.
- Nel *disgusto*, le sopracciglia si abbassano (AU4) e il labbro superiore si alza (AU10).

Le corrispondenze qui elencate, con modifiche minori, sono quelle adottate nel presente progetto.

Capitolo 4

Localizzazione del volto e delle regioni ad alta espressività

In questo capitolo descriveremo le tecniche utilizzate per l'individuazione, nell'immagine acquisita da AIBO, del volto dell'interlocutore e delle regioni che di esso riteniamo rilevanti per i nostri scopi: bocca, occhi e sopracciglia. Abbiamo optato, in generale, per tecniche base di elaborazione delle immagini, al fine di ottenere un codice leggero e veloce e, di conseguenza, un'applicazione che possa essere eseguita su AIBO in *tempo reale*. Combinando tali tecniche, otteniamo in output un insieme di regioni del viso che, approssimativamente, coincidono con le *feature* di cui intendiamo monitorare il movimento (per dedurne il tipo di espressione facciale eseguita). L'estrazione di queste regioni, permettendo di circoscrivere l'area di ricerca dei dettagli di interesse, consente di limitare i tempi di calcolo nel passo successivo e di renderlo il più possibile mirato. Di contro, tali tecniche sono molto sensibili alle condizioni di illuminazione e di posizionamento del soggetto. Pertanto, per ottenere dei risultati soddisfacenti, occorre soddisfare alcune condizioni:

- l'illuminazione del soggetto deve essere ottenuta tramite luce naturale, diffusa, frontale, con minima presenza di ombre; non viene comunque richiesto un

controllo stretto sul tipo di illuminazione;

- il viso del soggetto deve essere totalmente incluso nell'immagine acquisita da AIBO; inoltre, deve essere in posizione approssimativamente frontale (non sono ammesse rotazioni significative) e assumere un'espressione neutra, con occhi aperti;

- il viso del soggetto non deve essere occluso: ad esempio, vanno evitati occhiali e ciuffi di capelli che coprano occhi o sopracciglia;

- è bene evitare di indossare sciarpe, maglie, ecc. con colorazione affine a quella della pelle (e.g. rosa, salmone).

Se tali vincoli non vengono rispettati, non è possibile garantire il corretto funzionamento della procedura di riconoscimento del volto e delle regioni di interesse.

4.1 Estrazione della *skin map*

Una volta caricato il software sulla memory stick e avviato il robot, è possibile dare il via all'applicazione con un semplice tocco dei sensori posti sulla schiena di AIBO: così facendo, il sistema visivo di AIBO acquisirà un'immagine della scena di fronte a lui. Il primo passo di elaborazione consiste nell'estrarre da questa immagine la regione corrispondente al viso dell'interlocutore. A questo scopo sfruttiamo l'algoritmo di rilevamento dei colori reso disponibile dall'hardware di AIBO e già descritto in 2.2.1; questo algoritmo ha il pregio di essere molto rapido, e quindi ben si adatta alle nostre esigenze. Impostando intervalli di valori per le componenti Y, Cb e Cr che,

sperimentalmente, permettano di isolare con buona precisione i pixel di pelle[1], otteniamo in output un'immagine binaria, che prende il nome di **skin map**, in cui solo i pixel dell'immagine il cui colore appartiene agli intervalli definiti vengono mostrati in bianco. In Figura 4.1 è riportato un esempio di skin map prodotta dall'algoritmo di *color detection* di AIBO; la risoluzione delle immagini così ottenute è di 104×80.

Figura 4.1: Esempio di skin map restituita da AIBO.

La skin map così ottenuta viene ulteriormente elaborata per eliminare eventuali regioni che, pur soddisfacendo i requisiti di colore, non fanno parte del viso dell'utente. In primo luogo, viene applicato l'operatore morfologico di *chiusura*, che ha l'effetto di chiudere eventuali buchi nella skin map (dovuti, per esempio, a zone in ombra o, al contrario, eccessivamente illuminate). Viene poi invocato un semplice algoritmo di *etichettatura delle componenti connesse* per estrarre le diversi regioni presenti nell'immagine binaria: è ragionevole assumere che la regione di area maggiore sia quella corrispondente al viso, mentre le restanti possono essere attribuite a particolari dello sfondo e, pertanto, ignorate. Per maggior robustezza, la componente connessa così individuata viene riconosciuta come viso umano solo se la sua area è sufficientemente grande (almeno pari all'8% dell'area totale dell'immagine): in questo modo, l'algoritmo si arresterà sia nel caso in cui nessun volto sia effettivamente presente nell'immagine, sia nel caso in cui, a causa di cattive condizioni di illuminazione,

[1]Nel dettaglio, abbiamo considerato l'intervallo $(126, 173)$ per la componente Cr, $(77, 130)$ per Cb e $[80, 255]$ per Y.

la skin map estratta sia tanto rumorosa da pregiudicare una buona riuscita dei passi di elaborazione successivi.

4.2 Individuazione della regione della bocca

Anche il processo di estrazione della regione della bocca sfrutta le informazioni di colore presenti nell'immagine: il colore delle labbra è, infatti, ben caratterizzato nello spazio YCbCr. Al fine di evidenziare i pixel appartenenti alla bocca attuiamo una trasformazione sul piano colore che produce in output un'immagine a livelli di grigio, dove il valore di ciascun pixel è indicativo del grado di somiglianza tra il suo colore e il colore caratteristico delle labbra.

La trasformazione utilizzata, tratta da [Lanzarotti, 2003] e successivamente rielaborata dall'autrice stessa, è la seguente:

$$Transformed\_GrayLevel = [(255 - (Cr\_value - Cb\_value^2)) \cdot Cr\_value^2]$$

Per tagliare i tempi di calcolo ed escludere eventuali regioni dello sfondo aventi colore simile a quello delle labbra, la trasformazione descritta viene applicata solamente alla sottoimmagine definita dai confini della skin map.

A questo punto, sogliamo l'immagine ottenuta mostrando in bianco solo una percentuale (definita nel file di configurazione del progetto e fissata, sperimentalmente, al 3% dell'area totale dell'immagine considerata) dei pixel aventi valori più alti: ciò che otteniamo è la **mouth map**, un'immagine binaria che mostra le regioni che, con alta probabilità, appartengono alla bocca. In Figura 4.2 è riportato un esempio di output della procedura descritta.

Figura 4.2: Esempio di mouth map.

In maniera analoga a quanto fatto per la skin map, andiamo ad etichettare le componenti connesse della mouth map ed estraiamo la regione di dimensioni maggiori che, intuitivamente, dovrà corrispondere alla bocca del soggetto. Facendo esclusivamente uso di informazioni di carattere cromatico, questa procedura è estremamente sensibile alla presenza nell'immagine di altri particolari di colore simile a quello delle labbra: pertanto, è opportuno evitare di indossare capi di abbigliamento di tal colore. L'algoritmo descritto consente di individuare la posizione approssimativa della bocca nell'immagine: a questo stadio, infatti, non è essenziale estrarre tutti i dettagli, ma solo localizzare ad alto livello le aree di interesse per poi circoscrivere l'applicazione dei passi successivi, più costosi computazionalmente.

4.3 Individuazione delle regioni degli occhi

La regione della bocca così individuata ci consente, inoltre, di limitare l'area di ricerca per le regioni degli occhi: combinando le informazioni circa la posizione della bocca e la dimensione del volto nell'immagine, possiamo considerare esclusivamente una fascia del viso come candidata a contenere gli occhi, evitando di esplorare l'intera immagine. Per la precisione, a partire dalla porzione d'immagine, di altezza h, compresa tra il limite superiore della regione della bocca e quello della skin map, ricaviamo la fascia d'interesse sottraendo, dall'alto, la sezione approssimativamente

corrispondente alla fronte (la cui altezza è stata fissata sperimentalmente al 20% di h) e, dal basso, quella corrispondente al naso (la cui altezza è posta al 30% di h). La coordinata x del centroide della bocca può essere utilizzata per separare l'area di ricerca per l'occhio destro da quella per l'occhio sinistro.

L'elaborazione avviene, separatamente per ciascun occhio, sulla skin map prodotta da AIBO: infatti, se le condizioni di illuminazione sono ottimali, la skin map presenterà, in corrispondenza degli occhi, due buchi – come accade in Fig. 4.1 –, facilmente individuabili sfruttando informazioni di dimensione e posizione. Pertanto, il primo passo di calcolo richiede l'estrazione dei buchi all'interno della skin map, naturalmente limitati alla fascia di ricerca precedentemente individuata: un esempio di output è mostrato in Fig. 4.3.

Figura 4.3: Sottoimmagine per la ricerca dell'occhio.

Viene poi invocato l'algoritmo di etichettatura delle componenti connesse: se solo una componente è individuata, possiamo concludere che essa corrisponda all'occhio che stiamo cercando. Altrimenti, per guidare ulteriormente la ricerca, vengono calcolate le *proiezioni orizzontali* dell'immagine: in corrispondenza dell'occhio, tali proiezioni avranno un picco. Per evitare interferenze dovute, ad esempio, alle sopracciglia o a ciuffi di capelli, il picco viene selezionato partendo dalla parte bassa della fascia di ricerca; in Fig. 4.4 è mostrato un esempio di grafico delle proiezioni orizzontali, in cui possono essere distinti due picchi: uno corrispondente all'occhio, l'altro al sopracciglio.

Quanto descritto ci consente di scartare, tra le componenti connesse individuate,

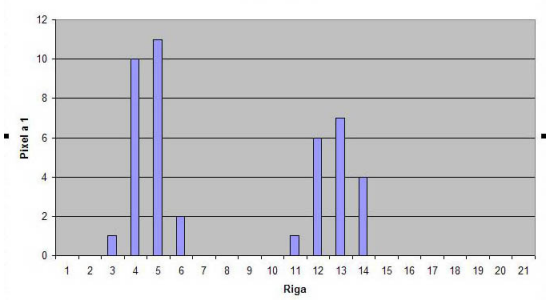


Figura 4.4: Esempio di proiezioni orizzontali nell'area dell'occhio.

quelle il cui centroide non sia allineato al picco. Se, nuovamente, più di una componente connessa rimane candidata al ruolo di occhio, selezioniamo come vincente la regione avente centroide più vicino al limite inferiore della fascia di ricerca (come già detto, questo permette di escludere componenti corrispondenti a sopracciglia – vd. Fig. 4.3 – o capelli).

Eseguendo la procedura descritta per entrambi gli occhi, è possibile estrarre le due regioni di interesse.

4.4 Individuazione delle regioni delle sopracciglia

Infine, le ultime regioni ad alta espressività che restano da estrarre sono quelle corrispondenti alle sopracciglia. Poiché non sempre le sopracciglia producono buchi nella skin map (per esempio, se esse sono molto chiare – vd. Fig. 4.1), occorre adottare, in questo caso, una tattica diversa. Dal momento che le sopracciglia sono generalmente caratterizzate da bordi orizzontali ben evidenti, la scelta più naturale consiste nell'applicare un filtro per l'estrazione dei bordi. Finora, abbiamo lavorato alla risoluzione di 104×80; affinché il filtro derivativo possa dare risultati accurati,

tuttavia, è preferibile applicarlo ad un'immagine a risoluzione maggiore. Quindi, passiamo a considerare l'immagine a livelli di grigio alla risoluzione massima disponibile (416×320).

Anche in questo caso, le informazioni raccolte nelle fasi precedenti ci consentono di limitare l'area di ricerca per il sopracciglio, che dovrà naturalmente trovarsi nella fascia di viso al di sopra dell'occhio; per ogni sopracciglio ci limitiamo, quindi, ad analizzare la metà inferiore della porzione di skin map soprastante la regione dell'occhio. Sulla sottoimmagine così definita viene applicato il *filtro di Sobel* per l'estrazione dei bordi orizzontali; l'immagine a livelli di grigio restituita dal filtro viene poi sogliata in modo da mostrare in bianco solo una percentuale (l'8% dell'area totale dell'immagine dei bordi) dei pixel aventi valori più alti. Il risultato è mostrato in Fig. 4.5.

Figura 4.5: Immagine binaria dei bordi per la ricerca del sopracciglio.

Otterremo quindi un'immagine binaria in cui potremo distinguere, utilizzando l'algoritmo per le componenti connesse già citato, una o più regioni. In primo luogo, vengono scartate quelle componenti aventi dimensioni non corrette per un sopracciglio (una componente candidata dovrà avere area almeno pari al 2% dell'area totale considerata). Se più regioni sopravvivono a questa selezione, escludiamo quelle aventi centroide troppo vicino al limite inferiore dell'area di ricerca: questo ci consente di ignorare l'eventuale presenza del bordo superiore dell'occhio (visibile, per esempio, in Fig. 4.5), che potrebbe essere scambiato, erroneamente, per il sopracciglio. Se nemmeno questo ulteriore criterio è sufficiente ad individuare univocamente il sopracciglio,

selezioniamo come regione corretta quella maggiormente allineata all'occhio (avente cioè coordinata x del centroide più vicina alla coordinata x del centroide dell'occhio).

Invocata la procedura descritta per entrambe le sopracciglia, e combinando i risultati ottenuti ai passi precedenti, possiamo ottenere un output grafico (mostrato in Fig. 4.6) che evidenzia tutte le regioni estratte, che costituiranno l'input alla fase successiva: il riconoscimento delle espressioni emotive.

Figura 4.6: Esempio di output della prima fase.

4.5 Risultati

La procedura descritta per l'estrazione delle regioni ad alta espressività è, come abbiamo accennato, sensibile alle condizioni di illuminazione, che influenzano direttamente le prestazioni dell'algoritmo di *color detection* di AIBO. Per esempio, sull'immagine mostrata in Figura 4.7 il nostro algoritmo fallisce: il volto risulta eccessivamente illuminato e, in conseguenza di ciò, gli intervalli fissati per la segmentazione nei piani Y, Cb e Cr non consentono l'estrazione di una skin map sufficientemente completa da permettere l'individuazione delle regioni degli occhi. Nel caso riportato

in Figura 4.8, invece, si presenta il problema opposto: l'illuminazione è scarsa e la skin map ottenuta risulta molto rumorosa, tanto da impedire un corretto riconoscimento dell'occhio sinistro.

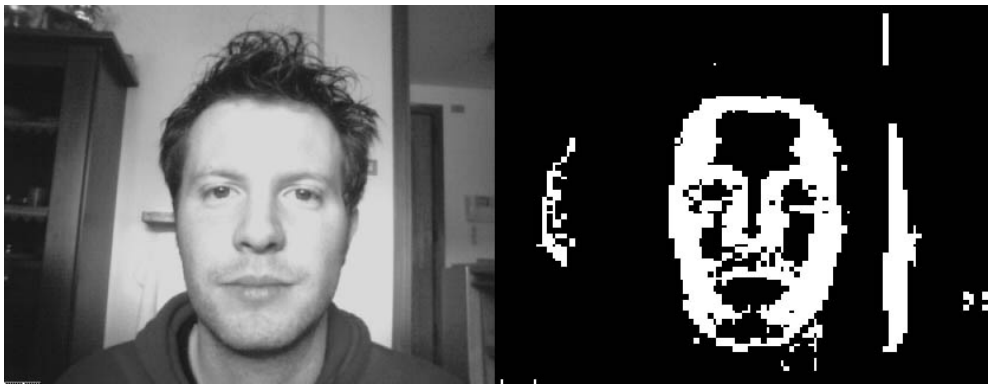


Figura 4.7: Cattiva illuminazione: l'algoritmo di estrazione delle regioni di interesse fallisce – 1.

In presenza di una buona illuminazione, invece, l'algoritmo è in grado di estrarre le regioni d'interesse con buoni risultati (vd. Figura 4.9). In alcuni casi tali regioni possono risultare sottodimensionate (e. g. le sopracciglia nel soggetto in basso a sinistra) oppure, al contrario, eccessivamente estese (per esempio, l'occhio destro nel caso dell'immagine in alto a sinistra); tuttavia ricordiamo che lo scopo di questo passo d'elaborazione è l'individuazione *approssimativa* delle aree ad alta espressività, al fine di semplificare e guidare i calcoli successivi, e pertanto è tollerabile un certo margine di dissimilarità tra le feature di interesse e le regioni individuate dall'algoritmo.

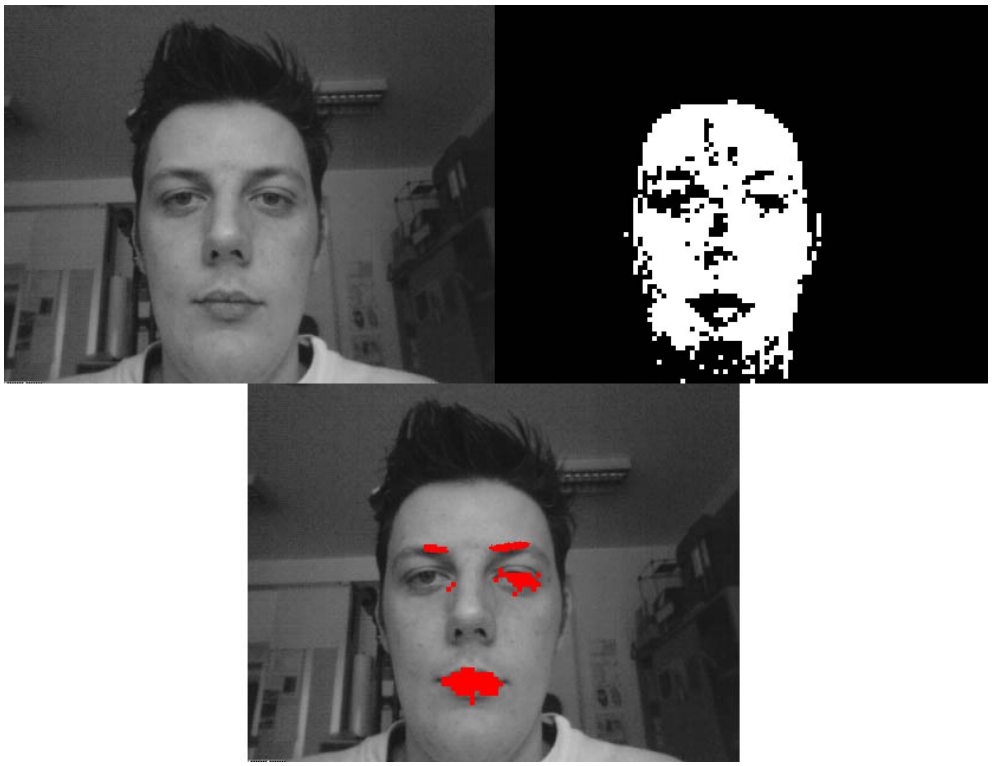


Figura 4.8: Cattiva illuminazione: l'algoritmo di estrazione delle regioni di interesse fallisce – 2.

Figura 4.9: Estrazione delle aree ad alta espressività.

Capitolo 5

Riconoscimento delle espressioni emotive

Come precedentemente accennato, in questo lavoro si è scelto di classificare le espressioni facciali mediante il sistema FACS (vd. Sezione 3.2.1): pertanto, un'espressione sarà descritta da un insieme di movimenti muscolari elementari, denominati *Action Unit* (AU). Se assegniamo a gruppi significativi di AU un'interpretazione psicologica, traducendoli quindi in *emozioni* specifiche, il riconoscimento dell'emozione espressa da un viso umano potrà essere ricondotta al rilevamento delle AU esibite. A sua volta, ciascuna AU può essere codificata come insieme di movimenti specifici di una o più *feature* del volto, che si manifestano durante l'esecuzione dell'espressione. Il riconoscimento delle AU e, quindi, delle espressioni emotive si basa perciò su informazioni di movimento delle feature di interesse, dall'immagine con espressione neutra alla successiva immagine con volto espressivo.

Il movimento di un punto attraverso due o più immagini può essere determinato tramite tecniche di **flusso ottico** [Barron et al., 1994] [Beauchemin e Barron, 1995]:

la differenza tra le posizioni assunte da uno stesso punto nei diversi fotogrammi considerati fornisce informazioni sul movimento cui quel punto è stato soggetto nell'intervallo di tempo coperto dalle immagini. Una delle tecniche di flusso ottico più semplici prende il nome di **block matching** e consiste nel misurare lo spostamento, in immagini successive, di *regioni* dell'immagine (*blocchi).* Il block matching è la tecnica adottata in questo progetto per l'estrazione dell'informazione di movimento delle feature di interesse del volto nella composizione dell'espressione facciale.

5.1 Il Block Matching

Intuitivamente, la tecnica del block matching richiede di definire una o più regioni di interesse all'interno dell'immagine di partenza, di rintracciarle poi nell'immagine d'arrivo e di calcolare, infine, la differenza tra le posizioni occupate da ogni coppia di blocchi corrispondenti: tali differenze rappresentano lo spostamento nello spazio di ciascun blocco.

Il *matching* tra blocchi corrispondenti richiede l'ottimizzazione di una *misura di similarità*: il blocco (che funge da template) viene, idealmente, sovrapposto ad ogni sottoregione dell'area di ricerca, per ciascuna delle quali viene calcolato il grado di similarità con il template; la sottoregione avente similarità massima viene riconosciuta come la candidata più probabile al ruolo di blocco corrispondente al template.

Più formalmente, sia $T(x, y)$ la funzione immagine descrivente il template (con $0 \leq x \leq N$, $0 \leq y \leq M$) e indichiamo con $I(x, y)$ l'area di ricerca, cioè l'immagine (o la sottoimmagine) in cui vogliamo rintracciare il template; chiamiamo, infine, $s(i, j)$ una generica misura di similarità. Se $s(a, b) = \max s(i, j)$, la regione nell'area di ricerca corrispondente al template può allora essere definita come l'area di dimensione

$N \times M$ avente origine in $I(x+a, y+b)$. Sono disponibili molteplici misure di similarità (o di dissimilarità, nel qual caso si procede ad una minimizzazione della misura), per esempio:

- La somma delle differenze assolute:

$$s(i,j) = \sum_{x=0}^{N} \sum_{y=0}^{M} |T(x,y) - I(x+i, y+j)|$$

- La somma delle differenze al quadrato:

$$s(i,j) = \sum_{x=0}^{N} \sum_{y=0}^{M} (T(x,y) - I(x+i, y+j))^2$$

- La cross-correlazione:

$$s(i,j) = \sum_{x=0}^{N} \sum_{y=0}^{M} T(x,y) \cdot I(x+i, y+j)$$

- La cross-correlazione normalizzata:

$$s(i,j) = \sum_{x=0}^{N} \sum_{y=0}^{M} \frac{(T(x,y) - \mu_T) \cdot \left(I(x+i, y+j) - \mu_{I(i,j)}\right)}{\sqrt{\sum_{x,y} (T(x,y) - \mu_T)^2 \cdot \sum_{x,y} \left(I(x+i, y+j) - \mu_{I(i,j)}\right)^2}}$$

  dove μ_T è il valor medio del template e $\mu_{I(i,j)}$ è il valor medio della sottoimmagine avente origine in $(x+i, y+j)$.

Il principale svantaggio di questa tecnica risiede nella sua complessità computazionale: per un template di dimensione $N \times M$ e un'area di ricerca di dimensione $R \times S$, la misura di similarità dovrà essere calcolata per $(R - N + 1) \times (S - M + 1)$ sottoaree, rendendo questa strategia inefficiente per R, S grandi. Pertanto, qualora si adotti la tecnica del block matching è essenziale essere in grado di circoscrivere il più

possibile le aree di ricerca, per evitare tempi di calcolo eccessivi. Inoltre, il block matching è sensibile alle eventuali deformazioni che i punti nel template possono subire a causa del movimento: tali deformazioni possono abbassare il punteggio di similarità e quindi provocare il fallimento della procedura di corrispondenza.

In questo progetto abbiamo scelto di adottare, come misura di similarità, la *cross-correlazione normalizzata* che, rispetto alle altre misure, risulta maggiormente robusta, proprio a causa della normalizzazione applicata. La scelta dell'area corrispondente a ciascun blocco avviene quindi per massimizzazione del coefficiente di correlazione; tuttavia, per ottenere corrispondenze più affidabili, meno sensibili al rumore, non viene automaticamente selezionata l'area con coefficiente massimo, ma vengono considerate tutte le posizioni aventi similarità superiore ad una soglia fissata. Di tali posizioni viene calcolata una media pesata, al fine di determinare la posizione finale del template. Più precisamente, identifichiamo cluster di posizioni adiacenti ad alta correlazione e, per ciascuno, determiniamo l'area e il centroide, pesato con i livelli di grigio dei pixel che lo compongono. In questo modo, associamo a ciascun cluster la posizione più probabile al suo interno. Successivamente, sui centroidi così calcolati viene effettuata una media pesata, in cui ciascun peso rappresenta l'area del cluster cui il centroide appartiene. In questo modo vengono avvantaggiate posizioni appartenenti ad un vicinato ad alta correlazione, rispetto a singoli punti isolati, che verosimilmente non identificheranno la regione di appartenenza del template, ma saranno piuttosto effetto di rumore.

Avendo a disposizione la posizione (x, y) del template nell'immagine di partenza e determinata tramite i passi descritti la sua nuova posizione (x', y') nell'immagine finale, possiamo automaticamente derivare il vettore spostamento $\vec{v} = [x' - x, y' - y]$

e, quindi, estrarre l'entità e la direzione del movimento a cui la regione di interesse è stata sottoposta nel passaggio dalla prima alla seconda immagine.

5.2 Il riconoscimento delle Action Unit

Come accennato in apertura di capitolo, il riconoscimento delle Action Unit eseguite durante l'espressione dell'emozione si basa su informazioni di movimento, estratte mediante la tecnica del Block Matching appena descritta. Il primo passo richiede, naturalmente, il posizionamento, sull'immagine con espressione neutra, dei blocchi di cui siamo interessati a registrare il movimento. Le sottoimmagini corrispondenti ai blocchi devono essere significative, cioè includere una feature del volto il cui movimento sia riconducibile ad una AU. Abbiamo pertanto deciso di impiegare un totale di 12 blocchi, così distribuiti:

- 6 blocchi per la bocca, e precisamente un blocco per ciascun angolo, due per il labbro superiore e due per il labbro inferiore;
- 3 blocchi per ciascun sopracciglio: uno per la parte interna, uno per la parte esterna ed uno per il tratto intermedio.

Il posizionamento dei blocchi è guidato dalle informazioni note circa la localizzazione di bocca e sopracciglia, ottenute al passo precedente. Applicando il filtro di Sobel sulla sottoimmagine contenente la regione della bocca e sogliando opportunamente l'immagine dei bordi (conservando il 10% dei pixel aventi livelli di grigio più alti, in modo da evidenziare il taglio della bocca), possiamo estrarre infatti gli angoli della bocca, che fungono da principale punto di riferimento per la collocazione dei 6 blocchi

destinati a questa regione. In maniera analoga vengono posizionati i blocchi delle sopracciglia (con la differenza che, in questo caso, non è necessario filtrare tramite Sobel perchè i bordi delle sopracciglia sono già stati estratti nel passo d'elaborazione precedente). La Figura 5.1 mostra la posizione dei blocchi sull'immagine con espressione neutra.

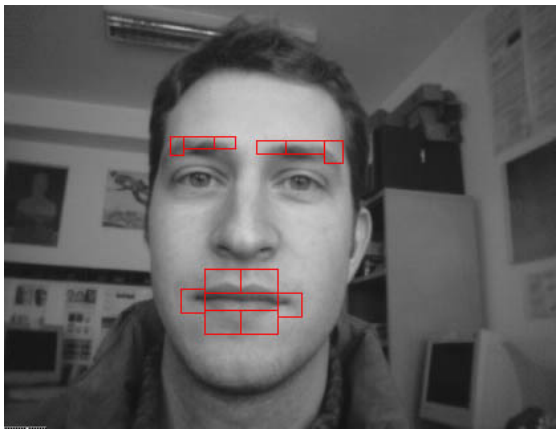


Figura 5.1: Posizionamento dei blocchi sull'immagine con espressione neutra.

Come descritto nella sezione precedente, le sottoimmagini definite dai blocchi vengono poi ricercate nell'immagine con l'espressione emozionale. In primo luogo, analogamente a quanto visto per l'immagine con espressione neutra, viene estratto il volto umano sfruttando la clusterizzazione sullo spazio colore YCbCr. Confrontando le informazioni di posizione del volto nella prima e nella seconda immagine, è possibile calcolare l'eventuale spostamento dell'utente tra un'istantanea e la successiva: lo spostamento del volto dalla posizione iniziale determina necessariamente uno spostamento solidale di tutti i blocchi nella stessa direzione, compromettendo, quindi, il calcolo del movimento delle feature di interesse. Predeterminare la differenza di posizione del volto consente quindi di annullare il suo effetto sul movimento dei blocchi.

È però doveroso rimarcare che il sistema è in grado di correggere solo piccoli spostamenti laterali: se lo spostamento è eccessivo, o interviene una rotazione, l'algoritmo non sarà in grado di calcolare correttamente il movimento delle feature.

Per limitare il più possibile il costo computazionale del block matching, per ciascun blocco viene definita un'area di ricerca limitata. Le feature del volto avranno movimenti relativamente piccoli ed è pertanto ragionevole limitare la ricerca di un blocco alle immediate vicinanze della posizione che occupava inizialmente; questo consente di ottenere tempi di calcolo ragionevoli e risultati più accurati, poiché non vengono considerate nella procedura di block matching aree dell'immagine che, pur corrispondendo ad altri dettagli del volto, potrebbero ottenere un alto punteggio di correlazione e influenzare il risultato. Una volta determinata la posizione finale per ciascuno dei blocchi definiti (come illustrato nella sezione precedente – nella Figura 5.2 è mostrato un esempio di risultato), viene calcolato il vettore spostamento e ne sono memorizzati il modulo e la direzione (espressa in termini di una delle 8 punte della rosa dei venti – S, SE, SO, E, ecc.). Lo spostamento dei blocchi può essere visualizzato tramite un insieme di frecce sovrapposte all'immagine di partenza (vd. Figura 5.3).

La procedura descritta non viene invece eseguita per la regione degli occhi. L'estrazione dei bordi (tramite il filtro di Sobel) in tale regione, necessaria per un corretto posizionamento dei blocchi sulla palpebra superiore ed inferiore, risulta infatti poco robusta e la successiva operazione di template matching risente molto della comparsa di rughe di espressione e ombre. Abbiamo pertanto stabilito di utilizzare una tecnica diversa per analizzare i mutamenti d'aspetto nella regione degli occhi. L'idea consiste nello sfruttare, nuovamente, le informazioni di colore. L'occhio (iride e sclera) ha

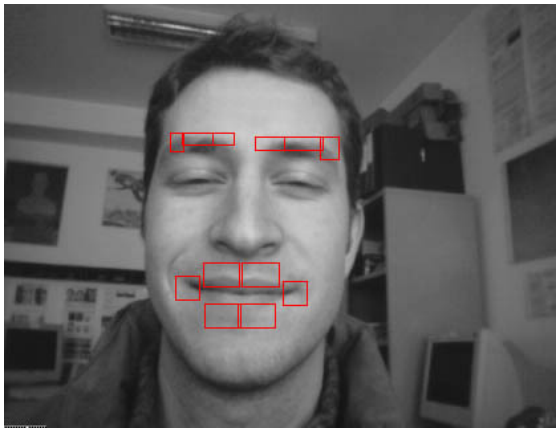


Figura 5.2: Risultato della procedura di Block Matching applicata ai blocchi definiti in Fig. 5.1.

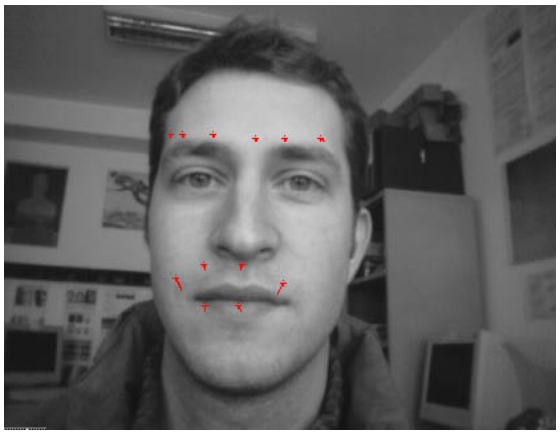


Figura 5.3: Movimento dei blocchi.

generalmente componente cromatica rossa bassa; estraendo dalla regione dell'occhio, precedentemente individuata, i pixel con valori più bassi sul piano Cr possiamo avere un'idea del grado di apertura dell'occhio. Pertanto, confrontando l'area della regione così estratta per l'immagine neutra e per l'immagine con espressione possiamo dedurre se, nel comporre l'espressione, l'occhio abbia avuto un movimento d'apertura o di chiusura. In sostanza, abbiamo ricondotto il movimento delle palpebre, codificato dalle AU 5 e 6, al suo effetto sul grado d'apertura degli occhi. In Figura 5.4 sono

mostrate le due immagini binarie rappresentanti il grado d'apertura dell'occhio nell'immagine neutra e nell'immagine con espressione, rispettivamente; in questo caso, si è avuto un movimento di apertura dell'occhio (l'area dell'occhio è infatti aumentata nel passaggio dalla prima alla seconda immagine).

Figura 5.4: Grado di apertura dell'occhio nell'immagine neutra e nell'immagine con espressione.

Disponiamo ora di tutte le informazioni necessarie per poter determinare quali AU siano state eseguite e, di conseguenza, quale sia l'espressione emotiva mostrata dall'utente. Per determinare ciò, occorre, in primo luogo, codificare opportunamente le AU che intendiamo rilevare. Tra tutte le AU definite dal FACS (vd. Sezione 3.2.1), abbiamo scelto di limitare il riconoscimento a 13 di esse, selezionate tra le più espressive e rilevanti. Nel dettaglio:

- per la zona delle sopracciglia, vengono considerate le AU 1, 2 e 4;
- per gli occhi, abbiamo selezionato le AU 5 e 6;
- per l'area della bocca, abbiamo scelto le AU 10, 12, 15, 17, 20, 23, 24 e 25.

Ciascuna di queste AU viene descritta, in un opportuno file di configurazione, in funzione dei movimenti fondamentali dei blocchi di interesse (nel caso di AU relative a bocca o sopracciglia) o dello stato degli occhi (in apertura/in chiusura). Un'AU relativa alle sopracciglia, per esempio, sarà descritta da un elenco di blocchi interessati da quella specifica AU, per ciascuno dei quali vengono indicati la direzione del movimento ed un punteggio, che misura la rilevanza relativa di quel movimento nella formazione della AU. Per esempio:

```
#AU_NAME::1 Inner_Brow_Raiser Frontalis,Pars_Medialis 1
Block 7::1 0.5
Block 10::1 0.5
Block 7::5 0.5
Block 10::7 0.5
Block 7::7 0.3
Block 10::5 0.3
```

L'AU 1, di cui sono riportati il nome, il muscolo del volto interessato e la parte del volto di pertinenza (1 è il codice relativo alle AU per le sopracciglia), prevede il movimento di due soli blocchi, corrispondenti alla parte interna di ciascun sopracciglio. Entrambi i blocchi possono spostarsi in direzione Nord (codice 1), Nord-Est (codice 5) o Nord-Ovest (codice 7); tra questi movimenti ammissibili, alcuni hanno un peso maggiore degli altri, ad indicare la maggior rilevanza dello spostamento corrispondente al fine del riconoscimento della AU in questione. Il seguente testo

```
#AU_NAME::5 Upper_Lid_Raiser Levator_Palpebrae_Superioris 3
Eyes_Status::2
```

descrive invece un'AU relativa agli occhi; in questo caso, ci limitiamo a dichiarare che gli occhi devono subire un movimento di apertura (codice 2) affinché l'AU 5 possa essere rilevata.

Il file di configurazione così composto viene caricato in memoria all'avvio di AIBO ed un'opportuna struttura dati viene inizializzata con i dati in esso contenuti. Le informazioni di movimento dei blocchi e il grado di apertura degli occhi vengono poi utilizzati per calcolare il punteggio di riconoscimento per ciascuna delle AU considerate. Il calcolo del punteggio di un'AU relativa agli occhi è immediato: se lo stato d'apertura degli occhi corrisponde con quanto richiesto, nel file di configurazione, per l'AU, essa è dichiarata riconosciuta. Per quanto riguarda le sopracciglia, se uno dei

movimenti componenti un'AU si è verificato, andremo a sommare al punteggio complessivo di riconoscimento per quell'AU il valore riportato per quel movimento nel file di configurazione. Analogamente si procede per le AU della bocca, in questo caso però tenendo conto anche dell'entità (modulo) dei movimenti registrati. Il punteggio per un singolo movimento viene infatti pesato tramite il suo modulo (opportunamente normalizzato); in questo modo, le AU corrispondenti ai movimenti più marcati otterranno un punteggio maggiore. Questo ci consente di distinguere con maggior efficacia tra AU diverse: per esempio, l'AU 12 descrive il sollevamento degli angoli della bocca ma, generalmente, ha come effetto collaterale anche un lieve spostamento verso l'alto delle labbra; questo determinerebbe pertanto il riconoscimento non solo dell'AU 12, ma anche di altre AU che, in realtà, non sono state eseguite. Pesando maggiormente i movimenti più rilevanti e scegliendo, tra le AU della bocca, solo quella con punteggio massimo, è possibile ottenere risultati più precisi. Una volta determinato il punteggio di riconoscimento per ogni AU, tutte quelle i cui punteggi superino la soglia di accettazione vengono dichiarate riconosciute.

5.3 Dalle Action Unit all'espressione emotiva

Resta ora da ricondurre l'insieme delle AU rilevate ad una delle emozioni che questo progetto si propone di riconoscere[1]. Abbiamo scelto di considerare le 6 emozioni universali secondo Ekman [Ekman, 1992]: **felicità**, **tristezza**, **sorpresa**, **rabbia**,

[1]Come spiegato nel Capitolo 1, non è possibile allo stato attuale determinare lo stato emotivo sperimentato dall'utente, ma solo analizzare l'espressione facciale da egli prodotta e, da questa sola informazione, dedurre l'emozione provata. Pertanto, parlando di riconoscimento di un'emozione, intendiamo riferirci, più propriamente, al riconoscimento di specifiche espressioni facciali che possono essere ragionevolmente ritenute manifestazioni di quell'emozione.

paura e **disgusto**. In maniera analoga a quanto fatto per le AU, le sei emozioni prescelte sono state codificate in uno specifico file di configurazione; ciascuna emozione è descritta da un insieme di punteggi associati alle AU considerate. Tali punteggi possono essere positivi (l'AU corrispondente si può verificare nell'espressione dell'emozione in questione), pari a 0 (l'AU non è indicativa dell'emozione corrente), oppure negativi (la presenza dell'AU rende meno verosimile che l'emozione espressa sia quella in esame). Per esempio, le seguenti righe:

```
#EMOTION_CODE 1::JOY
AU 1::0.10
AU 2::0.20
AU 4::-0.10
AU 5::0.0
AU 6::0.20
AU 10::0.0
AU 12::0.50
AU 15::-0.50
AU 17::-0.30
AU 20::0.0
AU 23::0.0
AU 24::-0.10
AU 25::0.20
```

descrivono l'espressione di felicità: nella sua composizione, alta rilevanza ha, per esempio, l'AU 12 (corrispondente ad un sorriso), mentre il verificarsi dell'AU 15 (che ha l'effetto di incurvare gli angoli della bocca verso il basso) porta un contributo negativo al riconoscimento di questa espressione. La corrispondenza tra AU ed espressioni emotive è stata determinata empiricamente, privilegiando espressioni molto marcate e tra loro ben differenziate.

Caricate in memoria, all'avvio di AIBO, le descrizioni delle espressioni emotive, si procede alla fase di riconoscimento: ogni AU rilevata al passo precedente contribuirà a comporre il punteggio di riconoscimento per le diverse emozioni, secondo i valori stabiliti nel file di configurazione. Poiché siamo interessati a rilevare, per ogni espressione, una sola emozione, solo quella che consegue punteggio massimo viene considerata. Se il suo punteggio supera la soglia di riconoscimento, l'emozione in questione viene restituita come output di questa fase d'elaborazione; altrimenti, è possibile concludere che nessuna emozione sia stata espressa (o, quantomeno, che nessuna emozione sia stata espressa in maniera sufficientemente intensa da poter essere rilevata con affidabilità) e, in questo caso, diremo che l'utente è rimasto *neutrale*.

Con la restituzione in output dell'emozione riconosciuta nell'utente termina la fase di *inferenza dello stato emotivo* ed inizia, invece, l'interazione emotiva vera e propria: l'emozione rilevata nell'essere umano diventa l'input che determina, congiuntamente alla *personalità* di AIBO, lo stato emotivo simulato del robot e, quindi, la sua risposta comportamentale all'atteggiamento dell'uomo.

5.4 Risultati

L'algoritmo descritto per il riconoscimento delle espressioni emotive è stato testato su 10 soggetti diversi, a ciascuno dei quali è stato richiesto di riprodurre tramite l'espressione del volto le 6 emozioni che il nostro progetto si propone di rilevare. Sono state quindi acquisite, tramite la camera di AIBO, 60 coppie di immagini[2], su cui sono stati poi eseguiti, su PC, i passi di elaborazione illustrati in questo capitolo.

[2]La prima istantanea di ogni coppia è sempre un'immagine del volto del soggetto con espressione neutra.

L'elaborazione offline su PC e non, direttamente, sulla piattaforma robotica è stata preferita, in sede di test, per motivi di praticità. L'algoritmo ha avuto esito positivo (ha cioè correttamente riconosciuto l'emozione mostrata) su 43 delle 60 istanze considerate, per una percentuale di successo del $71,6\%$. In chiusura di capitolo sono riportati alcuni esempi di espressioni correttamente identificate, con relativi punteggi di riconoscimento di AU ed espressioni emotive.

I principali motivi di fallimento del nostro algoritmo possono essere così riassunti:

- *Cattive condizioni di illuminazione.* Come abbiamo visto nel Capitolo 4, una buona illuminazione è indispensabile per la corretta estrazione delle aree di interesse del volto e, di conseguenza, dei successivi passi d'elaborazione. Se le regioni non vengono individuate con sufficiente precisione, i blocchi su cui opera l'algoritmo verranno collocati in posizione scorretta e non sarà possibile riconoscere correttamente le AU. Nell'esempio mostrato in Figura 5.5 la regione della bocca estratta risulta eccessivamente estesa, pregiudicando una buona collocazione dei blocchi in quell'area.

Figura 5.5: Esempio di fallimento: i blocchi della bocca non sono correttamente posizionati.

- *Spostamento del soggetto.* È richiesto che il soggetto si mantenga il più possibile fermo durante la cattura delle due immagini: movimenti eccessivi possono impedire il corretto calcolo del movimento dei blocchi e portare una o più feature del volto al di fuori delle aree di ricerca definite per ciascun blocco, determinando necessariamente risultati inattendibili da parte dell'algoritmo di block matching. In Figura 5.6 è illustrato l'effetto sulla procedura di block matching di un movimento di inclinazione in direzione della camera.

Figura 5.6: Esempio di fallimento: il soggetto si sposta durante il test.

- *Difficoltà di riproduzione di specifiche espressioni.* Il dizionario di movimenti facciali utilizzato, costituito dalle 13 AU da noi selezionate, è necessariamente limitato e rigido; la riproduzione di un'espressione richiede l'esecuzione di specifiche AU, il più possibile marcate per facilitarne il riconoscimento. Per molti dei partecipanti ai test è risultato difficoltoso, quando non impossibile, ripetere con sufficiente precisione specifiche espressioni; in questi casi, è inevitabile che l'algoritmo fallisca. In particolare, mentre le espressioni di felicità, tristezza e sorpresa si sono dimostrate più facilmente riproducibili, l'esecuzione delle restanti (in particolare paura e disgusto) è stata generalmente più problematica.

- *Inesatto rintracciamento dei blocchi da parte dell'algoritmo di block matching.* Oltre alle difficoltà introdotte dall'eventuale spostamento del soggetto, già descritte, anche la comparsa di ombre o di linee d'espressione sul volto o, in generale, di mutamenti nell'aspetto delle feature dovuti proprio alla contrazione muscolare prodotta dall'espressione possono influenzare l'esito dell'algoritmo di block matching, sebbene la simmetria delle feature d'interesse possa compensare il problema (se ristretto a pochi blocchi). Nell'esempio di Figura 5.7 i blocchi sul sopracciglio destro non sono stati rintracciati con precisione, ma il buon esito del block matching sul sopracciglio sinistro ha consentito comunque il riconoscimento dell'espressione di sorpresa.

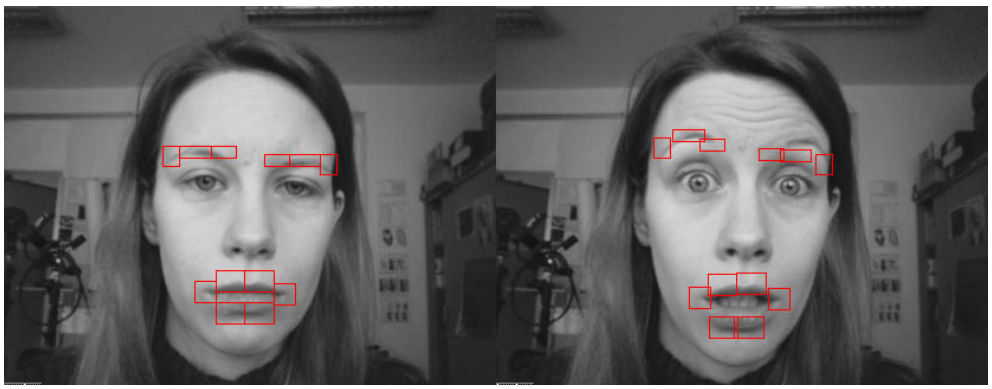
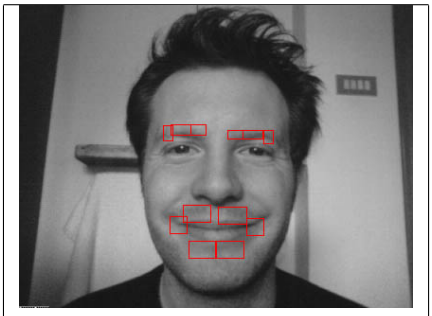
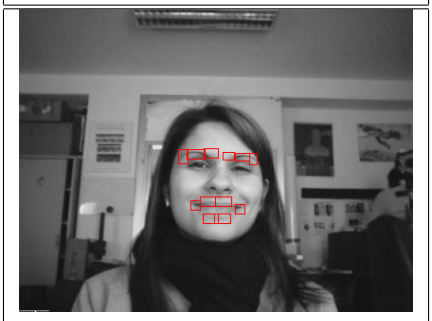
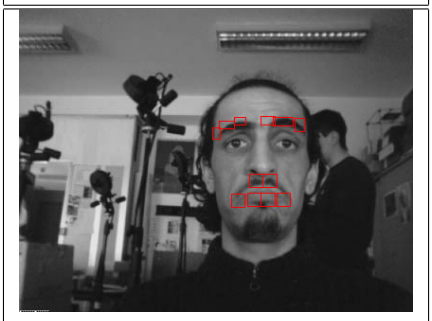
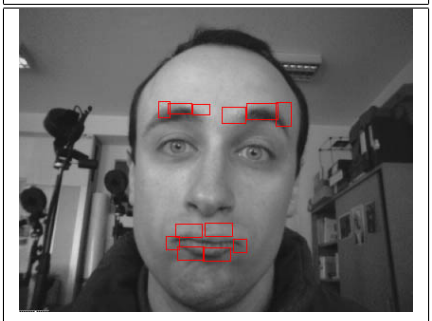
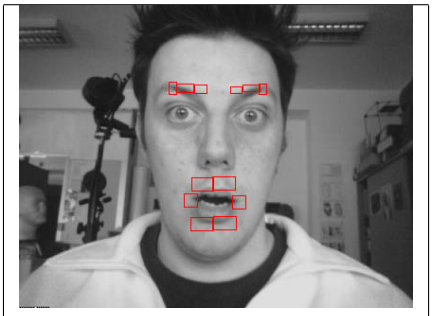
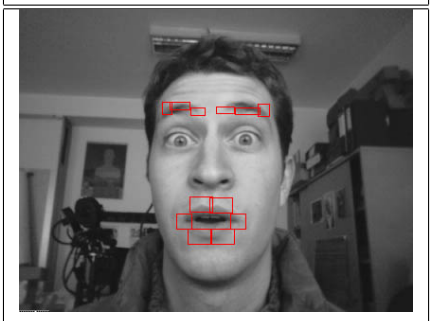
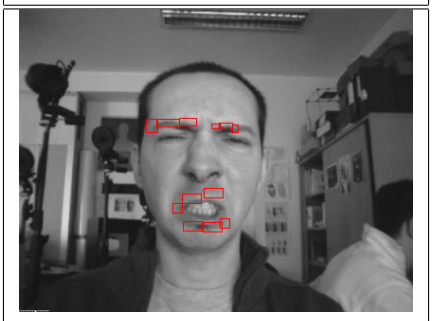
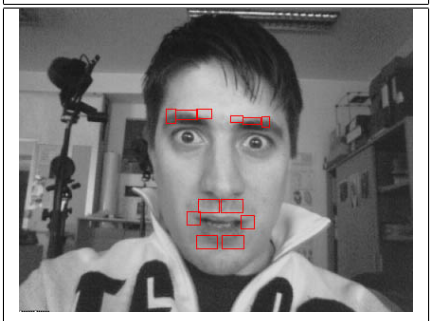


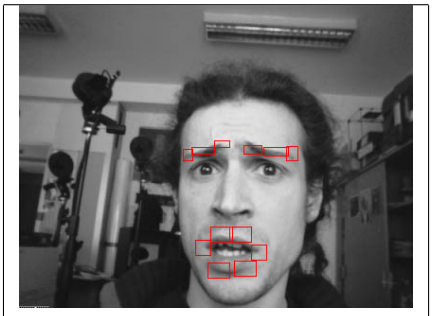
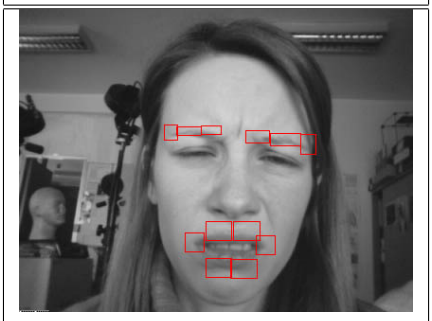
Figura 5.7: Esempio di block matching inesatto: i blocchi sul sopracciglio destro non sono correttamente rintracciati.

Sebbene certamente migliorabile, riteniamo che l'algoritmo da noi implementato possa costituire uno strumento valido, pur con tutte le limitazioni descritte, per la rilevazione di espressioni facciali. L'uso di tecniche base di elaborazione delle immagini, in luogo di metodi più sofisticati e robusti ma computazionalmente più onerosi, permette di disporre di un applicativo eseguibile in tempo reale, condizione essenziale

in un contesto di interazione con un essere umano. L'intero processo di riconoscimento dell'espressione emotiva, che comprende i passi descritti nel Capitolo 4 e nel presente, richiede su PC un tempo di calcolo di approssimativamente 0.2 secondi. Dal momento che le risorse computazionali su robot sono maggiormente limitate, i tempi registrati su AIBO risultano anche significativamente maggiori dei corrispondenti su PC, ma pur sempre inseriti in un quadro di esecuzione *real-time*: nel dettaglio, l'estrazione delle aree ad alta espressività e il posizionamento dei blocchi sull'immagine con espressione neutra richiedono circa 0.07 secondi, mentre l'elaborazione sull'immagine con espressione emotiva registra circa 0.6 secondi, per un totale di 0.7 secondi circa.

Espressione	AU riconosciute	Emozione
	• AU4: 0.600 • AU6: 1.000 • AU12: 0.875	**Felicità**: 0.600
	• AU1: 1.000 • AU2: 1.000 • AU6: 1.000 • AU12: 1.000	**Felicità**: 1.000
	• AU1: 0.800 • AU2: 0.600 • AU17: 1.000	**Tristezza**: 0.800
	• AU4: 0.600 • AU15: 0.786	**Tristezza**: 0.700

Espressione	AU riconosciute	Emozione
	• AU1: 0.800 • AU2: 1.000 • AU5: 1.000 • AU25: 0.775	**Sorpresa**: 1.000
	• AU1: 1.000 • AU2: 1.000 • AU5: 1.000 • AU10: 0.916	**Sorpresa**: 0.600
	• AU4: 0.800 • AU6: 1.000	**Rabbia**: 0.600
	• AU4: 0.900 • AU5: 1.000 • AU25: 0.800	**Paura**: 0.800

Espressione	AU riconosciute	Emozione
	• AU1: 0.500 • AU4: 0.800 • AU5: 1.000 • AU25: 0.777	**Paura**: 1.000
	• AU4: 1.000 • AU6: 1.000 • AU10: 1.000	**Disgusto**: 1.000

Capitolo 6

Interazione emotiva

Mentre i partecipanti ad un dialogo si scambiano parole, gli attori di un'interazione emotiva si scambiano *emozioni*. Nel presente lavoro, l'essere umano comunica ad AIBO l'emozione che sta sperimentando tramite un'espressione del viso; abbiamo visto nei capitoli precedenti i passi di elaborazione necessari per estrapolare questa informazione. Essa diventa l'input che contribuisce a determinare lo stato emotivo (più propriamente, la sua simulazione) di AIBO, che a sua volta lo renderà manifesto all'uomo con un comportamento appropriato.

La dinamica degli stati emotivi di AIBO viene modellata da un *automa a stati finiti probabilistico*. Prima di descrivere nel dettaglio il modello utilizzato, richiamiamo alcune nozioni di teoria degli automi.

6.1 Automi a stati finiti

Un **automa a stati finiti** è un modello di calcolo adatto a descrivere sistemi dotati di un numero finito di stati, la cui dinamica sia regolata da un insieme di input e da una funzione di transizione. Gli automi a stati finiti sono stati impiegati per modellare una grande varietà di problemi: circuiti sequenziali, sistemi economici,

reti di neuroni, problemi di trasporto, ecc. In questo senso, rappresentano un potente strumento di modellizzazione di sistemi. Gli automi a stati finiti possono anche essere utilizzati come *accettori* di linguaggi, cioè dispositivi in grado di discriminare tra parole definite su uno stesso alfabeto e, quindi, di riconoscere (e perciò descrivere) uno specifico linguaggio. In questa sezione introdurremo la teoria degli automi secondo i due punti di vista presentati: automi come *sistemi dinamici* [Rinaldi, 1977] ed automi come *accettori di linguaggi* [Hopcroft e Ullman, 1979].

6.1.1 Sistemi dinamici

Con il termine "sistema" vogliamo riferirci ad un generico ente fisico su cui, tramite un ingresso u, venga esercitata un'azione, e da cui, in risposta a tale ingresso, venga fornita un'uscita y. Il nostro generico sistema può essere rappresentato, quindi, come una *scatola nera* che riceve un input e restituisce un output (vd. Figura 6.1).

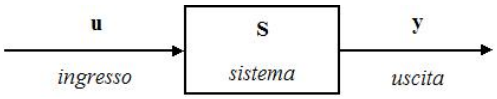


Figura 6.1: Rappresentazione di un sistema.

Diremo che un sistema è *dinamico* se evolve nel tempo. In questo caso, al sistema sarà associato un insieme T (l'insieme dei tempi) tale che per ogni $t \in T$ siano definiti un ingresso $u(t) \in U$ (insieme di ingresso) e un'uscita $y(t) \in Y$ (insieme di uscita). L'uscita ad un dato istante t, in generale, non sarà funzione del solo ingresso allo stesso istante, ma sarà in qualche modo dipendente dalla storia del sistema; tale storia è sintetizzata nel concetto di *stato* del sistema. Conoscere lo stato assunto dal sistema al tempo t è sufficiente per conoscerne l'uscita allo stesso istante; noti, invece, lo stato al tempo t_1, $x(t_1)$, e l'ingresso limitatamente all'intervallo $[t_1, t_2)$,

$u_{[t_1,t_2)}$, possiamo calcolare lo stato (e, per quanto appena detto, l'uscita) al tempo t_2 ($t_1 < t_2$). Più formalmente:

Definizione 6.1.1. *Un* ***sistema dinamico*** *è un ente caratterizzato da:*

- *un* ***insieme ordinato dei tempi*** T*;*
- *un* ***insieme di ingresso*** U*;*
- *un* ***insieme di funzioni di ingresso ammissibili*** Ω*;*
- *un* ***insieme di stato*** X*;*
- *un* ***insieme di uscita*** Y*;*
- *un* ***insieme di funzioni di uscita*** Γ*;*
- *una* ***funzione di transizione*** $\varphi() : T \times T \times X \times \Omega \to X$*, che restituisce lo stato ottenuto al tempo $t \in T$ partendo dallo stato iniziale ($x \in X$) all'istante $\tau \in T$ e applicando la funzione di ingresso $u() \in \Omega$: $x(t) = \varphi(t, \tau, x, u())$;*
- *una funzione $\eta()$, detta* ***trasformazione di uscita****, che definisce l'uscita: $y(t) = \eta(t, x(t))$.*

La funzione di transizione gode delle seguenti proprietà:

1. *consistenza*: $\varphi(t, t, x, u()) = x \ \forall (t, x, u()) \in T \times X \times \Omega$;
2. *irreversibilità*: φ è definita $\forall t \geq \tau$, $t \in T$;
3. *composizione*: $\varphi(t_3, t_1, x, u()) = \varphi(t_3, t_2, \varphi(t_2, t_1, x, u()), u()) \ \forall (x, u()) \in X \times \Omega$ e per ogni $t_1 < t_2 < t_3$;

4. *causalità*: $u'_{[\tau,t)}() = u''_{[\tau,t)}()$ implica $\varphi(t,\tau,x,u'()) = \varphi(t,\tau,x,u''()) \ \forall (t,\tau,x) \in T \times T \times X$.

Un'importante classe di sistemi è quella dei cosiddetti **sistemi invarianti**: si tratta di sistemi che non variano nel tempo le proprie caratteristiche. Più formalmente:

Definizione 6.1.2. *Un sistema si dice* ***invariante*** *se*

- T *è un semigruppo additivo;*
- *per ogni* $u() \in \Omega$ *e per ogni* $s \in T$ *la funzione* $u^s()$ *ottenuta da* $u()$ *per traslazione* $(u(t) = u^s(t+s),\ t \in T)$ *appartiene a* Ω*, cioè* $u^s() \in \Omega$*;*
- *la funzione di transizione gode della proprietà*

$$\varphi(t,\tau,x,u()) = \varphi(t+s,\tau+s,x,u^s()) \ \forall s \in T$$

- *la trasformazione d'uscita è indipendente da* t*, cioè* $y(t) = \eta(x(t))$.

In particolare, le prime due condizioni assicurano che siano applicabili al sistema tutte le funzioni di ingresso ottenute per traslazione nel tempo da funzioni in Ω; le ultime due affermano invece che partendo dallo stesso stato iniziale in due istanti diversi si otterrà lo stesso comportamento del sistema, a parità di ingressi (cioè applicando $u^s()$ invece di $u()$).

Definizione 6.1.3. *Un sistema si dice* ***discreto*** *se* $T = \mathbb{N}$*; un sistema si dice* ***continuo*** *se* $T = \mathbb{R}$.

Alla luce delle definizioni date è allora possibile comprendere cosa si intenda, in teoria dei sistemi, per *automa*.

Definizione 6.1.4. *Un **automa** è un sistema dinamico discreto ed invariante in cui gli insiemi di ingresso e uscita sono finiti. Un **automa a stati finiti** è un automa con insieme di stato finito.*

Vale la pena ricordare che gli automi finora descritti vengono anche chiamati *macchine di Moore*: in questo modello, l'uscita dipende esclusivamente dallo stato del sistema (in alcuni casi, uscita e stato possono anche coincidere). Nelle *macchine di Mealy* (anche detti automi impropri), invece, il valore dell'uscita al tempo t dipende anche dallo stato assunto nello stesso istante; quindi, $y(t) = \eta(x(t), u(t))$.

Gli automi fin qui descritti sono di tipo *deterministico*: la funzione di transizione, infatti, associa ad ogni quadrupla $(t, \tau, x, u()) \in T \times T \times X \times \Omega$ uno e un solo stato $x(t) \in X$. Qualora più stati possano essere assunti, indifferentemente, allo stesso istante t (cioè $\varphi(t, \tau, x, u()) = \{x_1, \ldots, x_n\}$, con $x_1, \ldots, x_n \in X$), si parla invece di automi *non deterministici*.

Di particolare interesse per il nostro lavoro sono gli **automi probabilistici**: in questi modelli, la funzione di transizione associa ad ogni quintupla $(t, \tau, x, x', u()) \in T \times T \times X \times X \times \Omega$ la *probabilità* che, partendo al tempo τ dallo stato x e applicando la funzione di ingresso $u()$, al tempo t lo stato assunto dal sistema sia x'. È evidente come gli automi probabilistici costituiscano una generalizzazione del caso deterministico; infatti, un automa deterministico può essere reso probabilistico assegnando probabilità 1 alla quintupla $(t, \tau, x, x', u())$ se $\varphi(t, \tau, x, u()) = x'$, 0 altrimenti.

6.1.2 Accettori di linguaggi

Come precedentemente accennato, gli automi a stati finiti possono essere impiegati in qualità di accettori di linguaggi. In questo caso, oltre ad un insieme di stati, ad

un insieme di ingressi e alla funzione di transizione, è definito un insieme di stati privilegiati, detti *stati finali*, in base ai quali avviene la selezione delle sequenze di input fornite all'automa.

Anche in questo caso, è la natura della funzione di transizione a determinare il tipo di automa: distinguiamo, nuovamente, tra *automi deterministici*, *non deterministici* e *probabilistici*.

Sia Σ un insieme finito e non vuoto, che prende il nome di *alfabeto*; i suoi elementi sono chiamati *simboli*, o *lettere*. Con Σ^* denotiamo l'insieme di tutte le sequenze finite di simboli in Σ, le *parole* sul nostro alfabeto. Se $x = \sigma_1\sigma_2 \dots \sigma_k$ è una parola, indichiamo con $l(x) = k$ la sua lunghezza; la parola di lunghezza 0 viene chiamata *parola vuota* e denotata con Λ. Se x e y sono due parole, la loro concatenazione sarà indicata come xy.

Definizione 6.1.5. *Un* ***automa a stati finiti deterministico*** *sull'alfabeto* Σ *è una quadrupla* $\mathcal{A} = \langle Q, \delta, q_0, F \rangle$*, dove*

- Q *è un insieme finito di* ***stati****;*
- $q_0 \in Q$ *è detto* ***stato iniziale****;*
- $\delta : Q \times \Sigma \to Q$ *è la* ***funzione di transizione****;*
- $F \subseteq Q$ *è l'insieme degli* ***stati finali****.*

In altre parole, se l'automa si trova nello stato q e riceve come input il simbolo σ, lo stato prossimo sarà *univocamente determinato* da $\delta(q, \sigma) = q'$. δ può essere estesa ad una funzione da $Q \times \Sigma^*$ in Q, definendo:

- $\delta(q, \Lambda) = q$

- $\delta(q, x\sigma) = \delta(\delta(q, x), \sigma)$

dove $q \in Q$, $x \in \Sigma^*, \sigma \in \Sigma$. Quindi, $\delta(q, x)$ indica lo stato in cui l'automa si troverà dopo aver ricevuto la parola x in input, partendo dallo stato q.

Definizione 6.1.6. *Una parola x è **accettata** (o riconosciuta) da $\mathcal{A}$ se e solo se $\delta(q_0, x) \in F$. L'insieme di tutte le parole accettate da $\mathcal{A}$, $L(\mathcal{A})$, è il **linguaggio riconosciuto** da $\mathcal{A}$. Un generico linguaggio $U \subseteq \Sigma^*$ è un **linguaggio regolare** se esiste un automa a stati finiti deterministico $\mathcal{U}$ tale che $U = L(\mathcal{U})$.*

Un automa a stati finiti può, quindi, essere interpretato come un *accettore* di linguaggi (regolari).

Esempio 1. *Consideriamo il seguente automa deterministico $\mathcal{A} = \langle Q, \delta, q_0, F \rangle$ sull'alfabeto $\Sigma = \{a, b, c\}$, dove:*

- $Q = \{q_0, q_1, q_2\}$
- $\delta(q_0, a) = \delta(q_0, b) = q_1$,
  $\delta(q_0, c) = q_2$,
  $\delta(q_1, a) = \delta(q_1, b) = \delta(q_1, c) = q_1$,
  $\delta(q_2, a) = \delta(q_2, b) = \delta(q_2, c) = q_2$;
- $F = \{q_1\}$.

L'automa è rappresentato sotto forma di grafo diretto in Fig. 6.2.

È immediato verificare che l'automa descritto accetta il linguaggio delle parole che iniziano per a o per b, cioè $L(\mathcal{A}) = (a + b)(a + b + c)^*$.

Un *automa a stati finiti non deterministico* viene definito in maniera del tutto analoga a quanto appena visto, ma la funzione di transizione è, in questo modello,

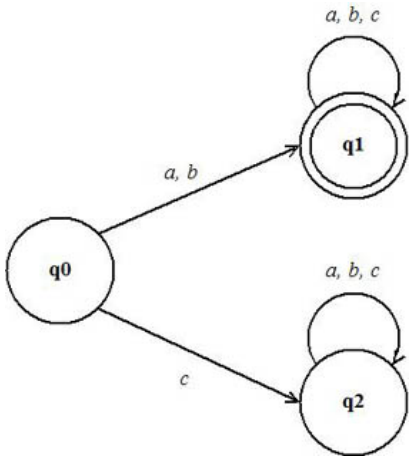


Figura 6.2: Automa a stati finiti deterministico per l'Esempio 1

non deterministica: $\delta : Q \times \Sigma \to \mathcal{P}(Q)$[1]. Ciò significa che, dato lo stato corrente e l'input ricevuto, non è determinato l'unico stato prossimo, ma un *insieme* di possibili stati prossimi. Diremo che una parola $x \in \Sigma^*$ è riconosciuta da un automa non deterministico $\mathcal{N} = \langle Q, \delta, q_0, F \rangle$ se esiste almeno uno stato $q \in \delta(q_0, x)$ che appartenga ad F. Un automa deterministico è perciò un caso particolare di automa non deterministico, in cui ogni insieme di stati prossimi definiti dalla funzione di transizione è, in realtà, costituito da un unico elemento. È inoltre possibile dimostrare che per ogni linguaggio L riconosciuto da un automa non deterministico esiste un automa deterministico che lo riconosce. Ne consegue che automi deterministici e non deterministici sono modelli di calcolo equivalenti, in quanto riconoscono la stessa classe di linguaggi (i linguaggi regolari).

Esempio 2. *In Figura 6.3 è riportato un automa a stati finiti non deterministico* $\mathcal{N} = \langle Q, \delta, q_0, F \rangle$ *sull'alfabeto* $\Sigma = \{a, b\}$*, dove:*

- $Q = \{q_0, q_1, q_2\}$

[1]Con $\mathcal{P}(Q)$ indichiamo l'*insieme delle parti* di Q, cioè l'insieme di tutti i possibili sottoinsiemi di Q.

- $\delta(q_0, a) = \{q_0, q_1\}$,
  $\delta(q_0, b) = \{q_0\}$,
  $\delta(q_1, a) = \{q_2\}$,
  $\delta(q_2, a) = \delta(q_2, b) = \{q_2\}$;
- $F = \{q_2\}$.

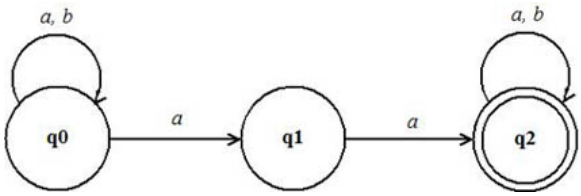


Figura 6.3: Automa a stati finiti non deterministico per l'Esempio 2

Il linguaggio riconosciuto dall'automa $\mathcal{N}$ coincide con l'insieme delle parole su Σ contenenti almeno due a consecutive: $L(\mathcal{N}) = (b + ab)^*aa(a + b)^*$. In Figura 6.4 è rappresentato un automa deterministico che riconosce lo stesso linguaggio.

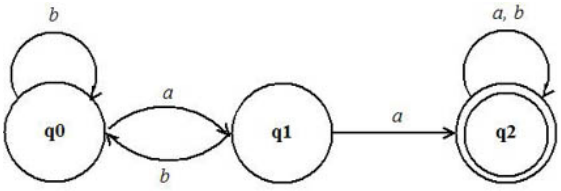


Figura 6.4: Automa a stati finiti deterministico per l'Esempio 2

Gli **automi a stati finiti probabilistici** [Rabin, 1963] [Paz, 1971] costituiscono una generalizzazione del modello deterministico.

Definizione 6.1.7. *Un **automa probabilistico** sull'alfabeto Σ è una quadrupla $\mathcal{P} = \langle Q, \delta, q_0, F \rangle$, dove:*

- *$Q = \{q_0, \ldots, q_n\}$ è l'insieme finito degli **stati**;*

- $\delta : Q \times \Sigma \to [0,1]^{n+1}$ *è la* ***funzione probabilistica di transizione*** *tale che per* $(q, \sigma) \in Q \times \Sigma$

$$\delta(q,\sigma) = (p_0(q,\sigma), \ldots, p_n(q,\sigma)) \text{ con } p_i(q,\sigma) \geq 0, \sum_i p_i(q,\sigma) = 1$$

- $q_0 \in Q$ *è lo* ***stato iniziale****;*
- $F \subseteq Q$ *è l'insieme degli* ***stati finali****.*

Un automa probabilistico può essere descritto da un insieme di ***matrici stocastiche*** *così definite:*

$$A(\sigma) = [p_j(q_i,\sigma)]_{0 \leq i,j \leq n} \text{ per ogni } \sigma \in \Sigma$$

Quando il sistema si trova nello stato q_i e riceve in input il simbolo σ, può entrare in ciascuno stato $q_j \in Q$ con diverse probabilità; la probabilità di ingresso nello stato q_j sarà data dall'elemento in posizione (i, j) della matrice $A(\sigma)$. Si assume che le probabilità di transizione rimangano fisse e siano indipendenti dal tempo e dai precedenti input. In questo modo, è possibile calcolare la probabilità di transizione dallo stato q_i allo stato q_j avendo in ingresso un'intera parola $x = \sigma_1\sigma_2 \ldots \sigma_m \in \Sigma^*$; tale probabilità sarà data dall'elemento in posizione (i, j) della matrice così definita:

$$A(x) = A(\sigma_1)A(\sigma_2)\ldots A(\sigma_m) = [p_j(q_i,x)]_{0 \leq i,j \leq n}$$

Definizione 6.1.8. *Sia* $\mathcal{P} = \langle Q, \delta, q_0, F\rangle$*,* $F = \{q_{i_0}, \ldots, q_{i_r}\}$ *e* $I = \{i_0, \ldots, i_r\}$*. Allora*

$$p(x) = \sum_{i \in I} p_i(q_0, x)$$

è la probabilità che $\mathcal{P}$*, partendo dallo stato* q_0*, entri in uno stato finale ricevendo in input la stringa* x*.*

Per definire l'insieme delle parole riconosciute da un automa probabilistico è centrale la nozione di *cut-point* (punto di taglio).

Definizione 6.1.9. *Sia $\mathcal{P}$ un automa probabilistico e λ un numero reale tale che $0 \leq \lambda < 1$. Il linguaggio riconosciuto da $\mathcal{P}$ con punto di taglio λ è*

$$L(\mathcal{P}, \lambda) = \{x | x \in \Sigma^*, p(x) > \lambda\}$$

È facile vedere che gli automi deterministici sono un caso speciale di automi probabilistici. Dato un automa deterministico $\mathcal{A}$, possiamo costruire un automa probabilistico $\mathcal{P}$ ad esso equivalente ponendo

$$p_j(q_i, \sigma) = \begin{cases} 1 & \text{se } \delta_{\mathcal{A}}(q_i, \sigma) = q_j \\ 0 & \text{altrimenti} \end{cases}$$

In questo caso, $p(x) = 1$, per $x \in \Sigma^*$, se e solo se $x \in L(\mathcal{A})$. Per la definizione di punto di taglio, abbiamo allora che per ogni λ $(0 \leq \lambda < 1)$, $L(\mathcal{A}) = L(\mathcal{P}, \lambda)$.

Ogni linguaggio riconosciuto da un automa deterministico è, quindi, riconoscibile da un automa probabilistico; non vale il viceversa: esistono quindi linguaggi, che prendono il nome di *linguaggi stocastici*, riconoscibili solo da automi probabilistici.

Esiste tuttavia una nozione più ristretta di punto di taglio: il *punto di taglio isolato*.

Definizione 6.1.10. *Un punto di taglio λ è detto **isolato** rispetto a $\mathcal{P}$ se esiste un $\varepsilon > 0$ tale che*

$$|p(x) - \lambda| \geq \varepsilon \text{ per ogni } x \in \Sigma^*$$

Vale il seguente importante Teorema:

Teorema 6.1.1 (Rabin). *Sia $\mathcal{P}$ un automa probabilistico e sia λ un punto di taglio isolato. Allora esiste un automa deterministico $\mathcal{A}$ tale che $L(\mathcal{P}, \lambda) = L(\mathcal{A})$.*

Pertanto, la classe dei linguaggi riconoscibili da un automa probabilistico con punto di taglio isolato coincide con la classe dei linguaggi regolari.

Esempio 3. *In Figura 6.5 è raffigurato un automa probabilistico* $\mathcal{P} = \langle Q, \delta, q_0, F \rangle$ *su alfabeto* $\Sigma = \{a, b\}$*, dove:*

- $Q = \{q_0, q_1\}$*;*
- $\delta(q_0, a) = (0.3\ 0.7)$
  $\delta(q_0, b) = (0.8\ 0.2)$
  $\delta(q_1, a) = (0.1\ 0.9)$
  $\delta(q_1, b) = (0.6\ 0.4)$*;*
- $F = \{q_1\}$.

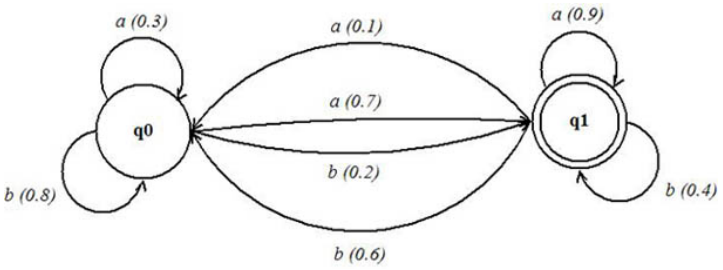


Figura 6.5: Automa a stati finiti probabilistico per l'Esempio 3

Le matrici stocastiche descriventi il sistema sono pertanto:

$$A(a) = \begin{bmatrix} 0.3 & 0.7 \\ 0.1 & 0.9 \end{bmatrix} \qquad A(b) = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix}$$

Moltiplicando queste matrici possiamo ottenere le probabilità di transizione per ogni $x \in \Sigma^*$*. Per esempio:*

$$A(ab) = \begin{bmatrix} 0.66 & 0.34 \\ 0.62 & 0.38 \end{bmatrix} \qquad A(aa) = \begin{bmatrix} 0.16 & 0.84 \\ 0.12 & 0.88 \end{bmatrix}$$

Per le definizioni date, avremo che $p(ab) = 0.34$ *e* $p(aa) = 0.84$*. Se fissiamo il punto di taglio* $\lambda = 0.6$*, il linguaggio* $L(\mathcal{P}, \lambda)$ *riconosciuto dall'automa conterrà la parola aa ma non la parola ab.*

6.1.3 Confronto tra le interpretazioni

Abbiamo visto come gli automi a stati finiti possano essere impiegati per distinte finalità: da una parte, come modelli per sistemi dinamici, dall'altra, come validatori di stringhe. I due diversi punti di vista si rispecchiano nella forma delle definizioni di automa date in 6.1.1 e 6.1.2. In sintesi, un automa, interpretato come sistema dinamico, è un'ottupla

$$S = \langle T, U, \Omega, X, Y, \Gamma, \varphi, \eta \rangle$$

dove $T = \mathbb{N}$ e vale la proprietà di invarianza. Secondo la definizione di accettori, invece, un automa su alfabeto Σ è una quadrupla

$$\mathcal{A} = \langle Q, \delta, q_0, F \rangle$$

Vediamo analogie e differenze nelle due definizioni.

- L'insieme dei tempi T in S non è esplicitato nella definizione di $\mathcal{A}$: mentre in un sistema dinamico la componente temporale è fondamentale, l'accettazione di una stringa ha proprietà di atemporalità. L'ordinamento temporale introdotto in S diventa piuttosto, in $\mathcal{A}$, ordinamento spaziale dei simboli che compongono la parola fornita in input all'accettore.

- L'insieme degli ingressi U di S coincide con l'alfabeto Σ su cui $\mathcal{A}$ è definito.

- L'insieme delle funzioni di ingresso Ω di S, come anche l'insieme delle funzioni di uscita Γ, non ha corrispondenza in $\mathcal{A}$.

- L'insieme di stato X in S è l'equivalente di Q in $\mathcal{A}$.
- L'insieme delle uscite Y di S non è previsto nella definizione data di automa accettore; analogamente, in $\mathcal{A}$ non è prevista alcuna trasformazione di uscita η.
- La funzione di transizione φ in S corrisponde a δ in $\mathcal{A}$.
- Lo stato iniziale q_0 di $\mathcal{A}$ non ha esplicita controparte in S; sebbene anche in S possa essere specificato uno stato iniziale, l'evoluzione del sistema può essere osservata a prescindere da esso, mentre l'insieme delle stringhe riconosciute da $\mathcal{A}$ dipende direttamente da q_0.
- Infine, l'insieme degli stati finali F di $\mathcal{A}$ non ha corrispondenza in S: infatti, S ha lo scopo di modellare un sistema in evoluzione, all'interno del quale tutti gli stati hanno medesimo valore, mentre $\mathcal{A}$ ha l'obiettivo di distinguere tra input "buoni" (quelli che terminano in uno stato di F) e input "cattivi".

In conclusione, in entrambi i casi un automa è un modello caratterizzato da uno stato interno, che evolve in base agli input ricevuti secondo una legge assegnata (la funzione di transizione); ma mentre in un'interpretazione l'accento è posto sull'osservazione della dinamica del sistema, istante dopo istante, nell'altra è centrale il perseguimento di un *goal* (valutare la qualità di una stringa in input, raggiungere uno stato favorevole, ecc.).

6.2 Il modello di interazione

Come accennato, abbiamo scelto di modellare l'interazione emotiva mediante un automa a stati finiti probabilistico, ispirandoci al *modello di personalità* descritto in

[Gmytrasiewicz e Lisetti, 2002] [Kopecek, 2003]. Il modello di personalità è lì definito come una quadrupla $P = \langle Q, U, \lambda, s_0 \rangle$ dove

- Q è l'insieme, finito e non vuoto, degli stati emotivi;
- U è l'insieme, finito e non vuoto, dei simboli di input e output;
- $\lambda : Q \times U \times U \times Q \rightarrow [0, 1]$ è la funzione di transizione probabilistica, che definisce la probabilità di passare da uno stato ad un altro ricevendo in ingresso un simbolo di input e restituendo in uscita un simbolo di output;
- s_0 è lo stato emotivo iniziale.

Nel modello da noi adottato, invece, non sono previsti output (o, più precisamente, l'output coincide con lo stato corrente); inoltre, il modello è *non omogeneo* nel senso che, durante l'interazione, le probabilità di transizione evolvono in accordo alla storia dell'interazione stessa.

Il nostro modello di interazione è quindi una quadrupla $I = \langle Q, U, P, q_0 \rangle$, dove:

- Q è l'insieme degli stati emotivi di AIBO:

$$Q = \{\text{AIBO\_NEUTRAL, AIBO\_JOYFUL, AIBO\_SAD, AIBO\_ANGRY}\}$$

- U è l'insieme degli input emozionali forniti dall'utente:

$$U = \{\text{USER\_NEUTRAL, USER\_JOYFUL, USER\_SAD, USER\_SURPRISED,}$$
$$\text{USER\_ANGRY, USER\_FEARFUL, USER\_DISGUSTED}\}$$

- $P = \{P_0, P_1, \dots\}$ è un insieme di funzioni di transizione probabilistiche $P_i : Q \times U \times Q \rightarrow [0, 1]$, dove P_i è la funzione di transizione al tempo i. In particolare, P_0 prende il nome, nel nostro modello, di *personalità*.

- q_0 =AIBO_NEUTRAL.

L'impiego di transizioni probabilistiche rende più interessante e ricca l'interazione, in quanto consente, a parità di input e stato corrente, una molteplicità di risposte alternative, ciascuna pesata con un valore di probabilità diverso. Osserviamo che l'automa probabilistico non è, in questo contesto, utilizzato nella sua interpretazione di accettore di linguaggi, ma piuttosto come modello per un sistema dinamico quale è, certamente, un'interazione.

I due concetti fondamentali alla base del modello impiegato sono quelli di **personalità** e **comportamento**. Come accennato, la personalità coincide con le probabilità di transizione all'inizio dell'interazione; dato lo stato corrente q e l'input u[2], la probabilità di entrare nello stato q' sarà quindi, inizialmente, $P_0(q, u, q')$. Come determinare tali probabilità? Non esiste, infatti, una regola universale che stabilisca che la probabilità di passare dallo stato q, con input u, allo stato q' debba essere maggiore della probabilità, a pari condizioni, di entrare nello stato q''. Verosimilmente, ciascun individuo avrà una specifica propensione a sperimentare alcuni stati emotivi a discapito di altri; ogni individuo, infatti, ha un diverso carattere, una diversa *personalità*. Le transizioni probabilistiche che regolano la risposta emotiva di AIBO, dunque, sono un riflesso della personalità che vogliamo che esso assuma. Abbiamo pertanto definito un insieme di *file di personalità*, ciascuno contenente i valori di una specifica funzione P_0. Tali valori sono stati pensati per descrivere possibili personalità da assegnare ad AIBO: amichevole, scontrosa, malinconica, casuale (tutte le probabilità di transizione assumono ugual valore). Riportiamo, a titolo d'esempio, il file corrispondente alla personalità amichevole:

[2]Ricordiamo che l'input corrisponde all'emozione riscontrata nell'utente tramite l'analisi della sua espressione facciale.

```
#####PERSONALITY_CONFIGURATION::FRIENDLY
#####
#STATE::AIBO_NEUTRAL 0
P(0, 0) = 1.0
P(1, 0) = 0.1
P(1, 1) = 0.9
P(2, 0) = 0.2
P(2, 2) = 0.8
P(3, 0) = 0.2
P(3, 1) = 0.8
P(4, 0) = 0.1
P(4, 2) = 0.8
P(4, 3) = 0.1
P(5, 0) = 0.3
P(5, 2) = 0.7
P(6, 0) = 0.1
P(6, 2) = 0.8
P(6, 3) = 0.1
#####
#STATE::AIBO_JOYFUL 1
P(0, 0) = 0.3
P(0, 1) = 0.7
P(1, 1) = 1.0
P(2, 0) = 0.2
P(2, 2) = 0.8
P(3, 0) = 0.1
P(3, 1) = 0.9
P(4, 0) = 0.3
P(4, 2) = 0.7
P(5, 0) = 0.3
P(5, 1) = 0.1
P(5, 2) = 0.6
P(6, 0) = 0.1
P(6, 2) = 0.9
#####
```

```
#STATE::AIBO_SAD 2
P(0, 0) = 0.5
P(0, 2) = 0.5
P(1, 0) = 0.2
P(1, 1) = 0.7
P(1, 2) = 0.1
P(2, 2) = 1.0
P(3, 0) = 0.5
P(3, 1) = 0.1
P(3, 2) = 0.4
P(4, 2) = 0.4
P(4, 3) = 0.6
P(5, 2) = 1.0
P(6, 2) = 0.5
P(6, 3) = 0.5
#####
#STATE::AIBO_ANGRY 3
P(0, 0) = 0.1
P(0, 2) = 0.7
P(0, 3) = 0.2
P(1, 0) = 0.3
P(1, 1) = 0.7
P(2, 0) = 0.1
P(2, 2) = 0.9
P(3, 0) = 0.8
P(3, 1) = 0.2
P(4, 3) = 1.0
P(5, 2) = 0.8
P(5, 3) = 0.2
P(6, 2) = 0.1
P(6, 3) = 0.9
#####
#####END_OF_FILE#####
```

Per ciascuno stato corrente (indicato sulla riga `#STATE` tramite l'identificativo e il codice associato) vengono elencate le probabilità di transizione $P(u, q)$ per ogni input u e stato prossimo q; le probabilità pari a 0 sono omesse. Per esempio, se AIBO si trova in stato AIBO_SAD (codice 2) e riceve l'input USER_JOYFUL (codice 1) avrà probabilità 0.2 di entrare in AIBO_NEUTRAL (codice 0), 0.7 di passare allo stato AIBO_JOYFUL (codice 1), 0.1 di restare nello stato in cui si trova. I valori di probabilità riportati nel file di personalità corrispondono alle seguenti matrici stocastiche:

$$P(\text{USER\_NEUTRAL}) = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.7 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 \\ 0.1 & 0.0 & 0.7 & 0.2 \end{bmatrix}$$

$$P(\text{USER\_JOYFUL}) = \begin{bmatrix} 0.1 & 0.9 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.2 & 0.7 & 0.1 & 0.0 \\ 0.3 & 0.7 & 0.0 & 0.0 \end{bmatrix}$$

$$P(\text{USER\_SAD}) = \begin{bmatrix} 0.2 & 0.0 & 0.8 & 0.0 \\ 0.2 & 0.0 & 0.8 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.1 & 0.0 & 0.9 & 0.0 \end{bmatrix}$$

$$P(\text{USER\_SURPRISED}) = \begin{bmatrix} 0.2 & 0.8 & 0.0 & 0.0 \\ 0.1 & 0.9 & 0.0 & 0.0 \\ 0.5 & 0.1 & 0.4 & 0.0 \\ 0.8 & 0.2 & 0.0 & 0.0 \end{bmatrix}$$

$$P(\text{USER\_ANGRY}) = \begin{bmatrix} 0.1 & 0.0 & 0.8 & 0.1 \\ 0.3 & 0.0 & 0.7 & 0.0 \\ 0.0 & 0.0 & 0.4 & 0.6 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$P(\text{USER\_FEARFUL}) = \begin{bmatrix} 0.3 & 0.0 & 0.7 & 0.0 \\ 0.3 & 0.1 & 0.6 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.2 \end{bmatrix}$$

$$P(\text{USER\_DISGUSTED}) = \begin{bmatrix} 0.1 & 0.0 & 0.8 & 0.1 \\ 0.1 & 0.0 & 0.9 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.1 & 0.9 \end{bmatrix}$$

dove il valore in posizione (q, q') di $P(u)$ rappresenta la probabilità di passare dallo stato q allo stato q' su input u. In generale, la personalità amichevole è caratterizzata da alte probabilità di ingresso nello stato di gioia; al contrario, la personalità malinconica prevede valori di probabilità più sbilanciati a favore dello stato di tristezza.

Avendo a disposizione più file di personalità, l'utente può scegliere di interagire con una personalità specifica oppure lasciare che il sistema ne selezioni, in maniera casuale, una tra quelle fornite.

È ragionevole pensare che, nel corso di un'interazione, l'atteggiamento nei confronti dell'interlocutore cambi, anche più volte, influenzato da quanto avvenuto nelle fasi precedenti dell'incontro; la frequenza di certe risposte emotive aumenterà, come effetto dei passati scambi emotivi, a discapito di altre. Abbiamo tentato di catturare questa idea di *evoluzione* dell'interazione consentendo la modifica periodica delle probabilità di transizione in risposta all'andamento dell'interazione stessa. Il criterio che regola tale evoluzione prende il nome, nel nostro modello, di **comportamento**.

L'aggiornamento delle probabilità di transizione (che genera, a partire da P_i, la nuova funzione di transizione P_{i+1}) avviene in base alla *qualità* degli input ricevuti fino al momento corrente. Nel dettaglio, distinguiamo tra:

- *interazione positiva*: contribuiscono a tratteggiarla gli input USER_JOYFUL e USER_SURPRISED;
- *interazione negativa*: determinata dagli input USER_ANGRY e USER_DISGUSTED;
- *interazione triste*: caratterizzata dagli input USER_SAD e USER_FEARFUL.

Ad ogni passo, l'input ricevuto determina l'incremento di un contatore della qualità di interazione, secondo la corrispondenza sopra indicata; non appena viene raggiunto un numero sufficientemente alto di input della stessa qualità, scatta l'aggiornamento delle probabilità di transizione. Tale aggiornamento, come detto, viene operato in base al comportamento selezionato:

- comportamento *imitativo*: le probabilità vengono modificate in modo da uniformare l'atteggiamento di AIBO a quello mostrato dall'utente;
- comportamento *compensativo*: le probabilità vengono modificate in modo che l'atteggiamento di AIBO risulti opposto rispetto a quello esibito dall'utente.

Nel primo caso, una tendenza marcata all'ostilità da parte dell'essere umano produrrà un'accresciuta avversione, di riflesso, in AIBO; nel secondo caso, lo stesso atteggiamento determinerà in AIBO, per compensazione, una maggior propensione a manifestare gioia (possiamo pensare che AIBO tenti di ingraziarsi, o consolare, l'utente di cattivo umore). L'utente, quindi, può liberamente scegliere non solo la personalità di AIBO, ma anche il suo comportamento e, dosando opportunamente gli input, può guidare l'interazione verso specifici risultati (per esempio, ottenere frequentemente dal robot risposte emotive di gioia).

Dal punto di vista implementativo, l'aggiornamento avviene incrementando tutte le probabilità di ingresso negli stati determinati, congiuntamente, dalla qualità rilevata di interazione e dal comportamento adottato; naturalmente, poiché deve sempre verificarsi che $\sum_{s \in Q} P(q, u, s) = 1 \ \forall q \in Q, u \in U$, è altresì necessario decrementare opportunamente le probabilità di ingresso nei restanti stati. Per esempio, se è stata rilevata un'interazione positiva (quindi, il contatore corrispondente ha superato la soglia d'attivazione) e il comportamento in atto è imitativo, andremo ad incrementare le probabilità di ingresso nello stato AIBO_JOYFUL: $P(q, u, \text{AIBO\_JOYFUL}) += \Delta$ $\forall q \in Q, u \in U$, dove Δ indica l'entità dell'incremento. Al termine dell'aggiornamento il contatore relativo alla qualità di interazione rilevata viene azzerato; sarà necessario che un numero rilevante di input della stessa qualità vengano riproposti ad AIBO affinchè lo stesso tipo di aggiornamento possa avere luogo.

Una volta determinato lo stato corrente e proceduto all'eventuale passo d'aggiornamento, AIBO può infine comunicare all'utente il proprio stato emotivo con un comportamento appropriato, per esempio scodinzolando se prova gioia oppure ringhiando se è arrabbiato. Un nuovo input da parte dell'utente determinerà in AIBO una nuova transizione di stato ed una risposta appropriata; il processo di interazione, virtualmente infinito, viene terminato dall'utente con un tocco sul sensore posto sotto il mento del robot.

6.2.1 Simulazione di interazione

Il numero di input e di stati emotivi nel nostro modello di interazione è, necessariamente, limitato: da una parte, il sistema è in grado di riconoscere, allo stato

attuale, solo 6 emozioni, veicolate tramite espressioni facciali esasperate, senza possibilità di distinguere tra diversi livelli di intensità; dall'altra, un numero elevato di stati emotivi determinerebbe un'occupazione di memoria non gestibile su AIBO. Inoltre, le capacità espressive di AIBO sono ridotte: pertanto, anche se fosse possibile definire una gran varietà di stati emotivi per il robot, esso non sarebbe in grado di comunicarli all'esterno in maniera apprezzabile.

Per poter quindi estendere il modello presentato, abbiamo realizzato una simulazione di interazione su calcolatore: in tale simulazione, due agenti, ciascuno modellato da un automa probabilistico, interagiscono fra loro, in modo che lo stato corrente di uno diventi l'input per l'altro. Il modello è quindi costituito da due automi $I^1 = \langle Q, U, P^1, q_0^1 \rangle$ e $I^2 = \langle Q, U, P^2, q_0^2 \rangle$, dove:

- l'insieme Q degli stati emotivi è lo stesso per I^1 e I^2;
- l'insieme U degli input coincide con Q;
- i due insiemi P^1 e P^2 delle funzioni di transizione probabilistiche sono distinti, e ciascuna dipende dalla personalità e dal comportamento dell'agente corrispondente;
- i due stati iniziali q_0^1 e q_0^2 sono distinti e non predeterminati;
- l'output di ciascun automa coincide con il proprio stato.

In questo modello l'insieme Q degli stati (e quindi, per quanto detto, l'insieme U degli input) è stato esteso a comprendere 19 stati emotivi. In particolare, oltre allo stato neutrale, abbiamo considerato tre diversi livelli di intensità (lieve, media, elevata) per ciascuna delle 6 emozioni fondamentali, come mostrato in Tabella 6.1.

Intensità lieve	**Intensità media**	**Intensità elevata**
PLEASED	JOYFUL	EXCITED
MELANCHOLIC	SAD	IN_DESPAIR
WONDERING	SURPRISED	ASTONISHED
ANNOYED	ANGRY	FURIOUS
WORRIED	FEARFUL	TERRIFIED
DISLIKING	DISGUSTED	CONTEMPTUOUS

Tabella 6.1: Stati emotivi definiti, classificati per intensità.

Aumentare il dettaglio degli stati emotivi consente di rendere più verosimile e ricca l'interazione emotiva; inoltre, la suddivisione in livelli di intensità rende più accurata la procedura di aggiornamento: input a intensità bassa contribuiranno alla determinazione della qualità dell'interazione in maniera meno decisiva rispetto ad emozioni più intense. Questo implica che un aggiornamento delle probabilità di transizione sarà innescato solo da un numero assai rilevante di input a bassa intensità, ma saranno sufficienti (relativamente) poche emozioni intense per ottenere lo stesso effetto; è, infatti, ragionevole ipotizzare che scambi emotivi intensi abbiano sull'atteggiamento dell'interlocutore un impatto ed un'influenza maggiori.

L'interazione prende il via specificando lo stato iniziale, la personalità e il comportamento per ciascun agente, e indicando il numero di interazioni che si intende visualizzare. Manualmente sono stati compilati i file di personalità casuale ed amichevole. Poiché la definizione di una personalità, in questo modello esteso, può rivelarsi compito estremamente dispendioso, è stato preparato un semplice applicativo che si occupa della generazione automatica di file di personalità. Il programma prende in input un insieme di file di personalità preesistenti, per ciascuno dei quali deve essere

specificato un peso[3], e ne determina la combinazione lineare ottenuta tramite i pesi forniti; in output restituisce il nuovo file di personalità, debitamente formattato. Per esempio, combinando il file FRIENDLY.PER, descrivente la personalità amichevole, con un file compilato ad hoc, in cui abbiano probabilità diverse da 0 solo le transizioni in ingresso a stati di paura (WORRIED, FEARFUL, TERRIFIED) e pesando opportunamente i due contributi, è possibile ottenere automaticamente un nuovo file, corrispondente ad una personalità amichevole ma timorosa.

Per concludere, riportiamo due esempi di interazione ottenuti applicando il modello descritto. In Figura 6.6 è mostrato uno scambio ricco, con frequenti transizioni di stato e pochi auto-anelli; l'agente #1 ha personalità amichevole e comportamento imitativo, mentre l'agente #2 ha personalità casuale e comportamento compensativo. La Figura 6.7 illustra, invece, un'interazione più statica: i due agenti entrano nello stato SAD e vi rimangono, in ciclo, influenzandosi a vicenda; entrambi hanno personalità amichevole e comportamento imitativo. In entrambi gli esempi, $q_0^1 =$EXCITED e $q_0^2 =$ FURIOUS; sono mostrati 10 passi d'interazione. La numerazione utilizzata evidenzia come gli input dell'agente #1 coincidano con gli stati assunti dall'agente #2 e viceversa.

[3]I pesi forniti devono essere compresi tra 0 e 1 e la loro somma deve essere uguale a 1.

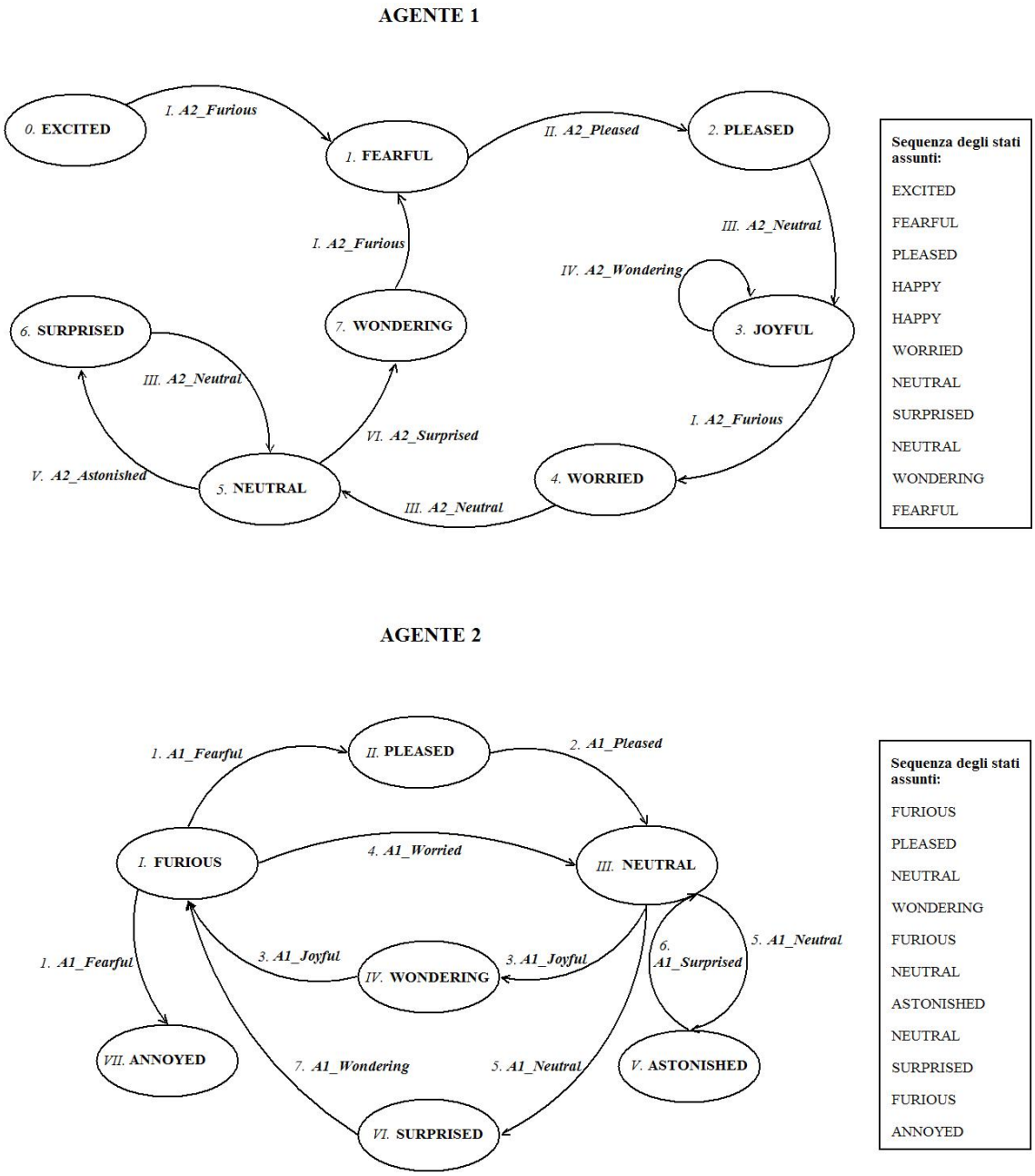


Figura 6.6: Esempio di interazione emotiva su modello esteso – 1.

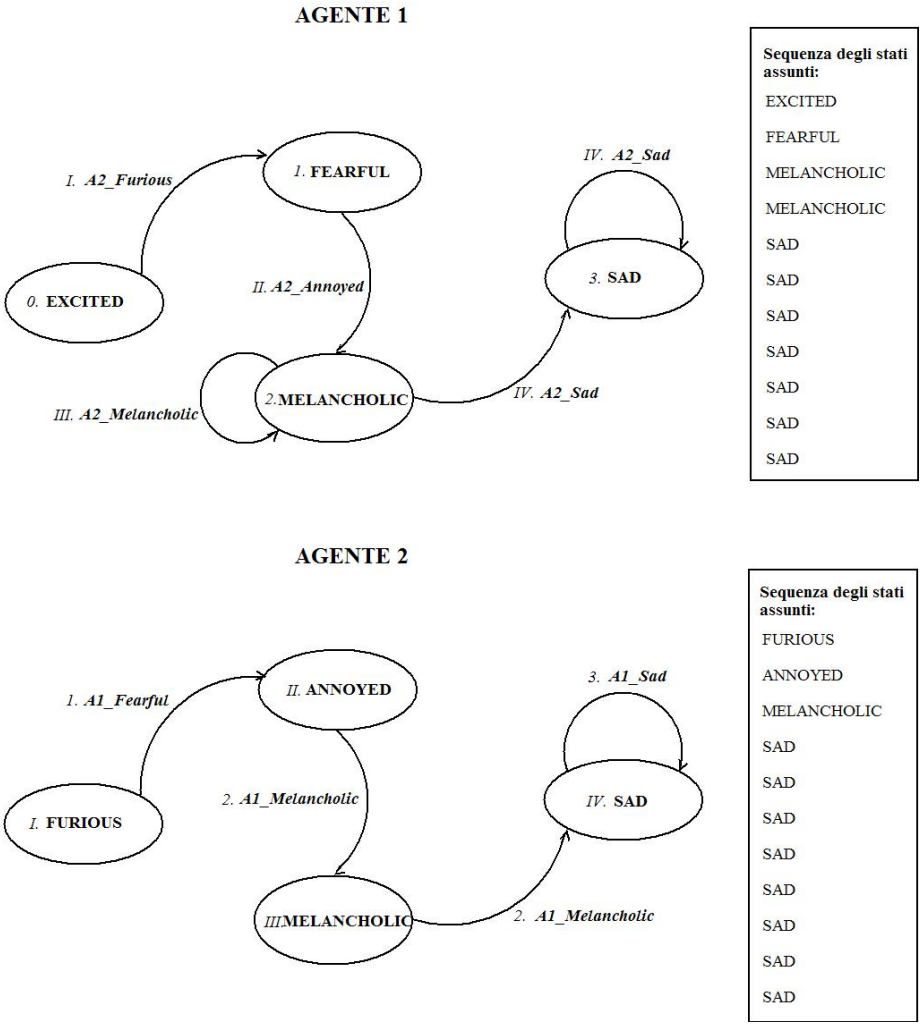


Figura 6.7: Esempio di interazione emotiva su modello esteso – 2.

6.2.2 Analisi di pattern comportamentali

Negli esempi riportati nella sezione precedente è possibile individuare alcuni cicli[4] di interazione; per esempio, in Figura 6.6 l'agente #2 parte dallo stato FURIOUS e in esso ritorna per ben due volte. L'analisi dei cicli di interazione possibili può fornire un quadro generale sull'andamento dell'interazione in esame, identificando pattern di comportamento che potranno ricorrere con alta probabilità.

A questo scopo, alteriamo il modello presentato supponendo, per semplicità, che l'automa #2 sia deterministico e concentriamo la nostra analisi sulla dinamica dell'automa #1, che, al contrario, mantiene la sua natura probabilistica. In questo quadro, l'automa #2 assume il ruolo di *ambiente* in cui l'automa #1 si trova e da cui riceve gli input emotivi che innescano le sue transizioni di stato. Formalizzando, il modello che assumiamo è costituito da:

- $I^1 = \langle Q, R, P, q_0 \rangle$, dove Q è l'insieme degli stati (e $q_0 \in Q$ è lo stato iniziale), R è l'insieme degli input e $P : Q \times R \times Q \to [0, 1]$ è la funzione di transizione probabilistica. Notiamo che, per ogni $r \in R$, la matrice $M^{(r)} \in [0, 1]^{Q \times Q}$ tale che $M^{(r)}_{qq'} = P(q, r, q')$ è una matrice stocastica.

- $I^2 = \langle R, Q, \delta, r_0 \rangle$, dove R è l'insieme degli stati (e $r_0 \in R$ è lo stato iniziale), Q è l'insieme degli input e $\delta : R \times Q \to R$ è la funzione di transizione deterministica.

Osserviamo che P, per semplificare l'analisi dei comportamenti, è mantenuta costante nel tempo (non è quindi soggetta ad aggiornamenti come, invece, accade nel modello finora descritto). Come in Sezione 6.2.1, $Q = R$ è l'insieme dei 19 stati emotivi da noi definiti; la notazione qui usata evidenzia come gli stati di un automa fungano da

[4] Qui e nel seguito intendiamo riferirci, con il termine *ciclo*, ad un percorso chiuso tra gli stati del modello, cioè una sequenza in cui lo stato di partenza e quello di arrivo coincidano.

input per l'altro automa e viceversa, secondo quanto previsto dal modello originale a due automi.

In questo contesto è possibile costruire l'*albero delle computazioni* (rappresentato in Figura 6.8) relativo a I^1: fissati due stati iniziali q_0 e r_0 e la lunghezza n di ciascuna computazione, possiamo infatti calcolare la probabilità di occorrenza di ciascuna possibile computazione. Tale valore sarà dato dal prodotto delle probabilità delle singole transizioni che compongono la computazione.

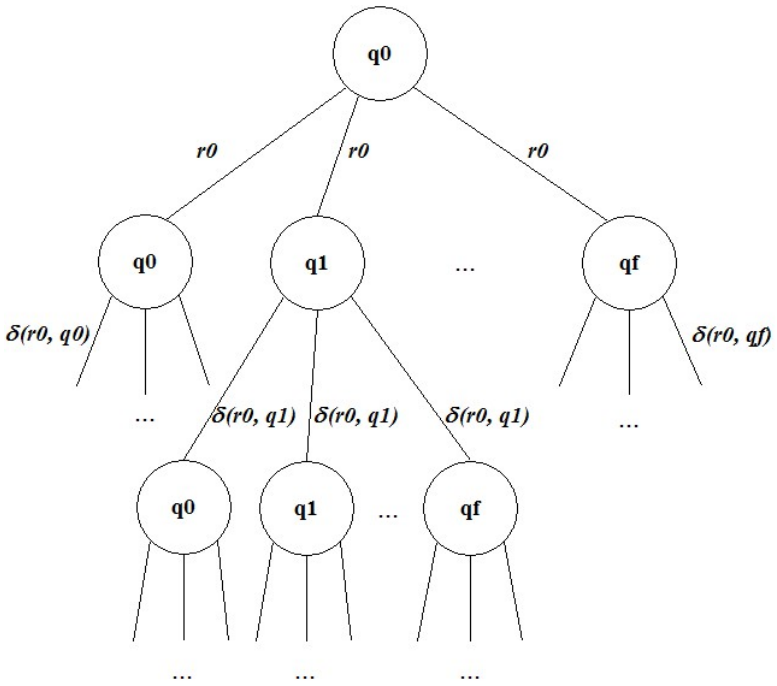


Figura 6.8: Albero di computazione dell'automa I^1.

Se chiamiamo **ciclo** una computazione che termini nello stato q_0 (ossia lo stato iniziale di I^1) e fissiamo un punto di taglio λ, possiamo individuare i cicli di interazione di lunghezza n aventi alta probabilità (cioè, maggiore di λ) di verificarsi visitando l'albero delle computazioni e scartando, passo dopo passo, quei rami il cui tratto iniziale

assuma un valore di probabilità minore di λ; infatti, scendendo in profondità nell'albero i valori di probabilità necessariamente decrescono, tendendo asintoticamente a zero. Questa osservazione consente un'opera di *pruning* sulle soluzioni candidate che permette di contenere i tempi di calcolo.

Alternativamente, è possibile adottare un formalismo diverso per descrivere il nostro problema. Sia

$$I = I^1 \otimes I^2 = \langle Q \times R, W, (q_0, r_0) \rangle$$

dove $W \in [0,1]^{(Q\times R)\times(Q\times R)}$ è una matrice stocastica in cui l'elemento di posizione $((q,r),(q',r'))$ fornisce la probabilità di passare da q a q' con input r e da r a r' con input q'. La seconda transizione è regolata da una funzione δ deterministica ($r' = \delta(r,q')$). Ciò equivale a scrivere che

$$W_{(q,r)(q',r')} = \begin{cases} P(q,r,q') & \text{se } r' = \delta(r,q') \\ 0 & \text{altrimenti} \end{cases}$$

Secondo questa impostazione, in I sono quindi riassunti e combinati I^1 e I^2, l'*agente* e l'*ambiente*.

In questo contesto una computazione che parta dallo stato iniziale (q_0, r_0) non è altro che una sequenza $S = ((q_0,r_0),(q_1,r_1),\ldots,(q_n,r_n))$, la cui probabilità è data da

$$Prob(S) = W_{(q_0,r_0)(q_1,r_1)} \cdot \ \ldots \ \cdot W_{(q_{n-1},r_{n-1})(q_n,r_n)}$$

Inoltre, per ogni coppia di stati $(q,r),(q',r') \in Q \times R$ e per ogni $n \in \mathbb{N}$,

$$W^n_{(q,r),(q',r')} = Prob\{S = ((q_0,r_0),(q_1,r_1),\ldots,(q_n,r_n)) \mid (q_0,r_0) = (q,r) \text{ e } (q_n,r_n) = (q',r')\}$$

$W^n_{(q,r),(q',r')}$ coincide quindi con la probabilità di passare da (q,r) a (q',r') in n passi.

Definiamo **ciclo** una computazione da (q_0, r_0) a (q_n, r_n) tale che $q_0 = q_n$ (l'uguaglianza $r_0 = r_n$ non è richiesta perché, come detto, intendiamo analizzare i cicli solo sull'automa I^1). Siamo interessati ad individuare i cicli aventi probabilità superiore al punto di taglio fissato.

Tali cicli coincideranno, infatti, con pattern di comportamento altamente probabili e, quindi, fortemente caratterizzanti l'interazione in esame. L'analisi di questi cicli può portare alla previsione di interazioni virtuose o, al contrario, negative e consente, in generale, di anticipare la qualità complessiva dell'interazione considerata. A questo scopo abbiamo progettato un semplice algoritmo che calcola i cicli di lunghezza data che abbiano probabilità maggiore o uguale a un valore λ prefissato. L'algoritmo è basato su una classica visita in profondità dell'albero delle computazioni del nostro modello di interazione che ha per radice uno stato della forma (q, r_0) con $q \in Q$, $r_0 \in R$.

Riportiamo una definizione formale del problema e una descrizione ad alto livello dell'algoritmo, impiegando, per compattezza, il formalismo appena introdotto.

Problema: matrice stocastica $W \in [0,1]^{(Q\times R)\times(Q\times R)}$.

Istanza: punto di taglio $\lambda \in (0,1)$, stato $r_0 \in R$, $n \in \mathbb{N}, n > 0$.

Soluzione: l'insieme delle computazioni $S = ((u_1, v_1), \ldots, (u_n, v_n))$ tali che

$$v_1 = r_0,\ u_n = u_1 \text{ e } Prob(S) \geq \lambda.$$

Algoritmo:

for $(q \in Q)$ do

$$\begin{cases} C[0] = (q, r_0) \\ \text{Visit}(q, r_0, 1.0, 1, C) \end{cases}$$

```
Visit(q, r, prob, length, seq):
    if (length = n)
        { (u_1, v_1) = seq[0]
        { (u_2, v_2) = seq[n − 1]
        { if (u_1 = u_2)
        {         Output seq
    else
        { for (q' ∈ Q) do
        {                 { r' = δ(r, q')
        {                 { p = prob · W_{(q,r)(q',r')}
        {                 { seq[length] = (q', r')
        {                 { if (p ≥ λ)
        {                 {     Visit(q', r', p, length + 1, seq)
```

L'algoritmo descritto ci permette, per esempio, di verificare che, per $n = 4$, se le due personalità interagenti sono amichevoli (e $r_0 =$ NEUTRAL), la probabilità del ciclo NEUTRAL–NEUTRAL–NEUTRAL–NEUTRAL è pari a 0.729, mentre il pattern EXCITED–JOYFUL–JOYFUL–EXCITED ha probabilità 0.2. Questi dati evidenziano come l'interazione tra personalità simili tenda a produrre scambi emotivi piuttosto statici, caratterizzati dalla permanenza ciclica nello stesso stato. Se invece alteriamo la personalità dell'agente #2 introducendo un aspetto randomico nelle sue risposte comportamentali, nella dinamica dell'agente #1 tendono ad emergere cicli di interazione più interessanti (per esempio, il ciclo WORRIED–NEUTRAL–MELANCHOLIC–WORRIED con probabilità 0.24). In Figura 6.9 è riportato il grafo di un'interazione in cui tale ciclo ha effettivamente avuto luogo.

Osserviamo che, se λ decresce esponenzialmente rispetto a n, il numero di cicli forniti in output dall'algoritmo presentato potrebbe essere esponenziale. Se, invece, λ è costante e in W non sono presenti cicli assorbenti[5], il numero di soluzioni restituite per tutti i possibili n è limitato da una costante opportuna. In tempo polinomiale rispetto ad n possono inoltre essere calcolati i valori $W^n_{(q_0,r_0)(q_0,r)}$ per ogni $r \in R$, cioè i valori di probabilità complessivi per tutti i cammini che, partendo da (q_0, r_0), giungono in n passi a (q_0, r).

Un altro problema "naturale" relativo ai cicli e risolubile in tempo polinomiale è il seguente:

> Problema: matrice stocastica $W \in [0,1]^{(Q\times R)\times(Q\times R)}$.
>
> Istanza: stato iniziale $(q_0, r_0) \in Q \times R$ e $n \in \mathbb{N}$.
>
> Soluzione: per ogni $r \in R$, un ciclo di lunghezza n da (q_0, r_0) a (q_0, r) di probabilità massima.

L'algoritmo per la risoluzione di questo problema (una semplificazione del noto *algoritmo di Viterbi*) richiede, secondo il paradigma della programmazione dinamica, la gestione di una matrice $|Q \times R| \times n$ in cui memorizzare i risultati parziali ottenuti durante l'elaborazione. Il generico elemento in posizione $((q, r), k)$ di tale matrice è una coppia di valori, $P_{(q,r),k}$ e $S_{(q,r),k}$, così definiti:

$$P_{(q,r),k} = \max\{P_{(q',r'),k-1} \cdot W_{(q',r')(q,r)} | (q', r') \in Q \times R\}$$

cioè $P_{(q,r),k}$ rappresenta la massima probabilità di un cammino di lunghezza k che termini in (q, r); $S_{(q,r),k}$ è lo stato che precede (q, r) in tale cammino. Dunque, fissato r, $P_{(q_0,r),n}$ fornisce la probabilità del massimo ciclo di lunghezza n da (q_0, r_0) a (q_0, r),

[5]Informalmente, un ciclo assorbente è un ciclo nel quale tutte le transizioni hanno probabilità 1.

che può essere ricostruito a partire dai valori S memorizzati nella matrice. Una descrizione ad alto livello dell'algoritmo è la seguente:

Algoritmo:

for $((q,r) \in Q \times R)$ do
- $P_{(q,r),1} = W_{(q_0,r_0)(q,r)}$
- $S_{(q,r),1} = (q_0, r_0)$

for $(k = 2, \ldots, n)$ do
- for $((q,r) \in Q \times R)$ do
  - $max = 0$
  - for $((q',r') \in Q \times R)$ do
    - $prob = P_{(q',r'),k-1} \cdot W_{(q',r')(q,r)}$
    - if $(prob \geq max)$
      - $max = prob$
      - $P_{(q,r),k} = max$
      - $S_{(q,r),k} = (q',r')$

for $(r \in R)$ do
- if $P_{(q_0,r),n} \neq 0$
  - Output $P_{(q_0,r),n}$
  - Ricostruisci il ciclo risalendo agli stati visitati partendo da $S_{(q_0,r),n}$

Il calcolo dei cicli di probabilità massima e l'analisi della matrice W^n, sopra accennati, rappresentano due possibili ulteriori approcci allo studio dei pattern di interazione; il loro impiego potrà essere approfonditamente investigato negli sviluppi futuri di questo lavoro.

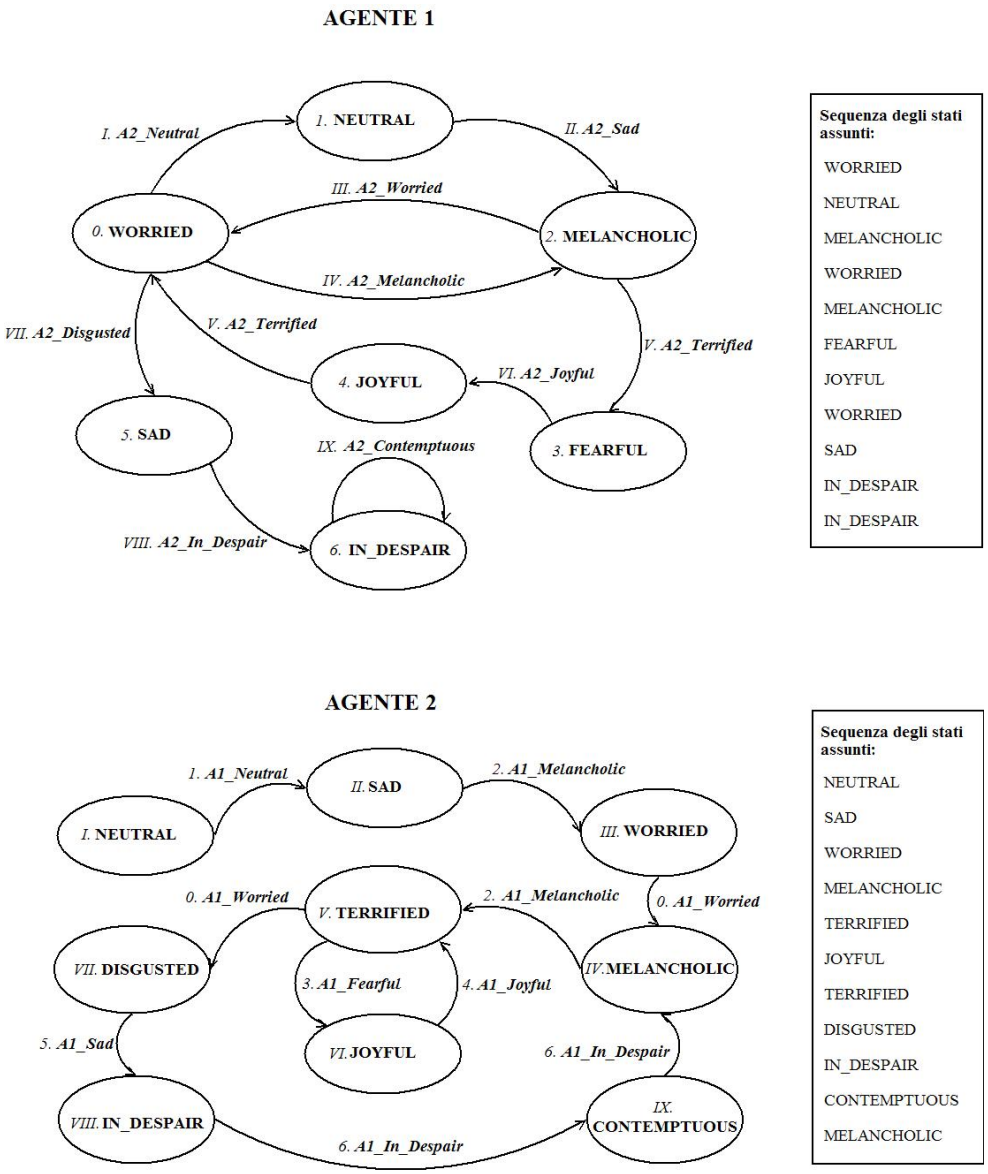


Figura 6.9: Esempio di interazione emotiva su modello esteso – 3.

Capitolo 7

Conclusioni e sviluppi futuri

In questo capitolo presentiamo una panoramica sul progetto, ricapitolando le fasi di elaborazione viste in dettaglio nei capitoli precedenti, e riassumiamo i risultati ottenuti, con particolare attenzione ai possibili sviluppi futuri.

7.1 Panoramica sul progetto

Il progetto *Emotional Interaction* è costituito da più moduli cooperanti o, più precisamente, da più *oggetti OPEN-R* (vd. Sezione 2.2); lo schema del progetto è riportato in Figura 7.1, dove ciascun riquadro corrisponde ad un distinto oggetto e le frecce rappresentano i canali di comunicazione.

L'oggetto FACEEXPR si occupa dell'analisi dell'espressione facciale, secondo le tecniche descritte nei Capitoli 4 e 5, operando sull'immagine ricevuta dall'oggetto di sistema *OVirtualRobotComm*. Quando pronto, AIBO emette un suono per invitare l'utente a posizionarsi di fronte ad esso ed assumere un'espressione neutra; a questo punto, un tocco dei sensori sulla schiena del robot provoca la cattura dell'immagine. Terminata l'elaborazione di questa prima istantanea, viene eseguito un nuovo suono

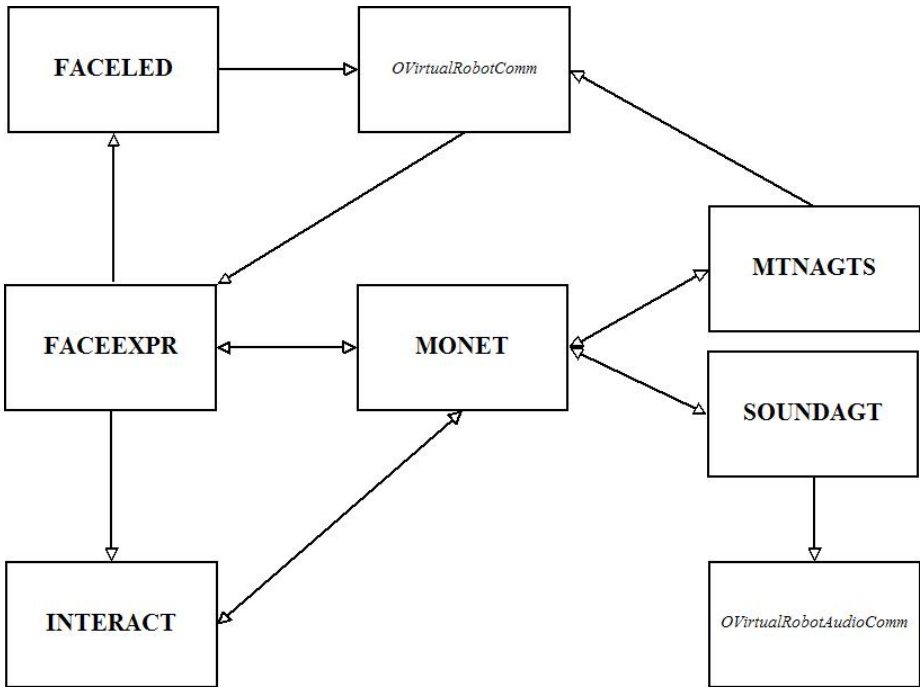


Figura 7.1: Oggetti OPEN-R utilizzati nel progetto.

e l'immagine con espressione viene acquisita, secondo le medesime modalità. L'utilizzo di suoni di avvertimento e il ricorso alla pressione dei sensori per registrare una nuova immagine consentono all'utente di preparare l'espressione che intende esibire. Se l'elaborazione sull'immagine neutra incontra difficoltà (per esempio, non è stato possibile estrarre correttamente le regioni degli occhi), AIBO richiederà all'utente di ripetere l'operazione.

Terminata l'analisi del volto, avremo determinato l'espressione emotiva esibita; questa informazione viene passata da FACEEXPR a FACELED, che si occupa di accendere i LED sul muso di AIBO secondo la configurazione corrispondente all'emozione ricevuta. In questo modo, è possibile ottenere un rapido feedback circa i risultati della procedura di riconoscimento delle espressioni emotive. L'attivazione dei LED richiede l'invio di un opportuno messaggio a *OVirtualRobotComm* che, come detto, è

responsabile della gestione degli attuatori e dei sensori (eccezion fatta per il sistema audio, a cui è dedicato *OvirtualRobotAudioComm*) di AIBO.

L'espressione riconosciuta viene comunicata, inoltre, a INTERACT, che si occupa di determinare il nuovo stato emotivo di AIBO e, eventualmente, di aggiornare le probabilità di transizione del modello di interazione. A seconda dello stato correntemente assunto dal robot, un diverso complesso di movimenti e suoni verrà eseguito, tramite l'invio di un messaggio a MONET. *MoNet* è uno dei programmi d'esempio disponibili insieme all'ambiente OPEN-R e si occupa dell'esecuzione, anche simultanea, di mosse e suoni definibili dal programmatore tramite un editor di movimenti (per esempio, MotionEditor di Sony); l'oggetto principale, MONET, dialoga con SOUNDAGT e MTNAGTS, che fungono da interfaccia verso i due oggetti di sistema *OVirtualRobotComm* e *OVirtualRobotAudioComm*. Anche l'emissione dei suoni di avvertimento da parte di FACEEXPR avviene tramite comunicazione con MONET. A sua volta, MONET invia messaggi, a scopo di coordinamento, a FACEEXPR e INTERACT, annunciando la fine dell'esecuzione di un movimento o di un file audio.

Una volta terminata l'esternazione del proprio stato emotivo, AIBO richiede una nuova immagine neutra dell'utente, dando vita ad un nuovo passo di interazione; il ciclo d'esecuzione (mostrato in Figura 7.2) si interrompe qualora l'utente prema il sensore posto sotto il mento del robot. AIBO si porterà allora in posizione sdraiata e rimarrà in attesa di spegnimento.

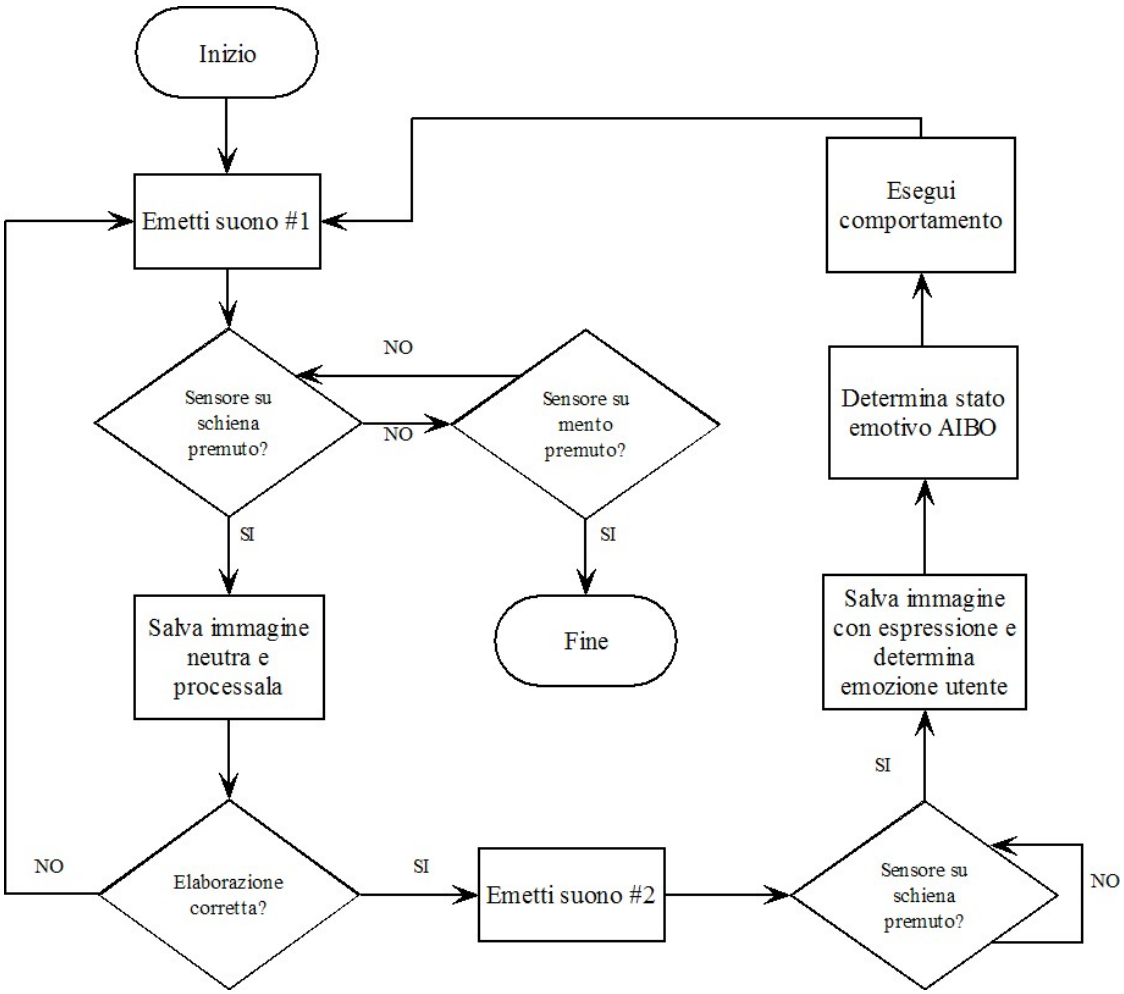


Figura 7.2: Diagramma di flusso del ciclo d'esecuzione.

7.2 Risultati e sviluppi futuri

L'intero processo di elaborazione avviene in tempo reale: una singola interazione (costituita dal passo di analisi dell'espressione facciale dell'utente e dal successivo calcolo dello stato emotivo di AIBO) richiede tempi inferiori al secondo. Come riportato in Sezione 5.4, infatti, il riconoscimento dell'espressione emotiva mostrata dall'utente (che rappresenta la fase di calcolo più onerosa in termini di tempo) richiede approssimativamente 7 decimi di secondo. La natura interattiva del progetto necessita di

tempi di calcolo il più possibile contenuti, il che esclude l'impiego di algoritmi avanzati ad alto consumo di risorse, a maggior ragione considerando la ridotta potenza di calcolo disponibile su AIBO. È stato quindi necessario adottare tecniche elementari per l'elaborazione delle immagini ed introdurre vincoli sull'utilizzo del software (relativi alle condizioni di illuminazione, alla posizione del soggetto, ecc.) per poter ottenere una procedura rapida ma, comunque, sufficientemente efficace. I test condotti hanno rilevato una percentuale di successo nel riconoscimento dell'espressione emotiva del 71.6%, risultato più che accettabile tenendo conto dell'intrinseca difficoltà del compito e delle limitazioni imposte (non ultima, la ridotta risoluzione delle immagini catturate da AIBO).

La procedura di riconoscimento delle espressioni facciali è, tuttavia, ampiamente perfezionabile: potendo disporre di maggiori risorse computazionali (e.g., spostando il calcolo da AIBO su PC) ed eventualmente rinunciando al requisito di elaborazione in tempo reale, si possono sostituire o integrare le tecniche illustrate con altre più avanzate. Per esempio, la ricerca delle aree ad alta espressività del volto può essere condotta tramite template deformabili, cioè generici modelli di forma in grado di adattarsi, con processo iterativo, alle feature cui vengono applicate, nonostante la variabilità individuale (gli occhi di due persone possono differire per dimensione, taglio, ecc.). Inoltre, l'insieme delle Action Unit riconoscibili può essere esteso e le corrispondenze AU-emozione raffinate.

Il risultato principale di questo lavoro consiste nello sforzo di modellizzazione dell'interazione emotiva fra uomo e robot. Il modello scelto, un automa a stati finiti probabilistico, consente di descrivere lo stato emotivo corrente del robot come risultato congiunto di eventi esterni e di dinamiche interne. Dall'esterno giungono input

emotivi, forniti dal partner umano, che innescano le transizioni di stato; internamente, tali mutamenti di stato sono influenzati dalla *personalità* dell'automa, che definisce i valori iniziali di probabilità delle transizioni. Analogamente, l'aggiornamento dei valori di probabilità avviene in risposta a stimoli esterni (l'andamento generale dell'interazione) e attributi interni (il *comportamento*, cioè la politica di adattamento agli input ricevuti). Il segnale inviato dall'utente non determina, pertanto, una reazione automatica ed univocamente determinata, ma viene filtrato dai processi interni all'automa, corrispondenti alle peculiarità caratteriali proprie di ogni essere umano. La natura probabilistica del modello consente di ottenere interazioni sempre diverse, a parità di input: in questo modo, cogliamo l'imprevedibilità che accompagna ogni interazione.

La semplicità del modello ne rende immediata l'estensione o la modifica (nel numero e nel tipo degli input, degli stati, delle personalità, dei comportamenti); in futuro potranno essere pensati meccanismi più sofisticati per l'aggiornamento delle probabilità e algoritmi intelligenti per la generazione automatica di personalità ricche e convincenti.

Il modello presentato può essere impiegato nella realizzazione di agenti "emotivi", per esempio nell'ambito dei videogiochi, oppure, come in questo progetto, della robotica di intrattenimento. L'introduzione di una componente emotiva nei robot può agevolare la loro integrazione nella società umana, integrazione resa complessa da una tradizionale diffidenza nei confronti delle macchine e da oggettive difficoltà nell'utilizzo, da parte del pubblico non specializzato, di questi sofisticati strumenti. La comunicazione non verbale (quindi, la decodifica e l'invio di segnali emotivi)

può semplificare e rendere più immediata l'interazione uomo-robot e, contemporaneamente, investire la macchina di tratti umani in grado di trasformarla, agli occhi dell'utente, da strumento inanimato a compagno di vita. Ciò è particolarmente rilevante se pensiamo ai progetti di assistenza robotica ad anziani o malati: è essenziale che il paziente accetti la presenza e l'aiuto del robot, non ne sia spaventato e collabori attivamente con esso per il proprio benessere; d'altro canto, il robot dovrebbe avere la capacità di comprendere le necessità del paziente senza che questi debba esplicitarle tramite un insieme rigido e necessariamente limitato di comandi standard. A questi scopi ben si adatta l'impiego della comunicazione non verbale.

Sebbene in questo lavoro si sia considerato un solo canale comunicativo, cioè il volto, nulla vieta di estendere il progetto implementando nuovi moduli per l'analisi di segnali emotivi diversi: il tono della voce, la gestualità, la postura, lo sguardo sono tutti canali tramite cui vengono espresse emozioni. Integrando siffatti moduli al progetto presentato, si potrebbe determinare in modo più robusto lo stato emotivo dell'utente e, senza modifiche sostanziali al codice, recapitarlo direttamente all'oggetto che gestisce l'interazione.

Il modello di interazione può anche costituire uno strumento per l'analisi di dinamiche comportamentali. Abbiamo visto, in sezione 6.2.2, come lo studio dei cicli sul modello proposto possa fornire indicazioni sull'andamento previsto di un'interazione; variando le personalità in gioco, è possibile osservare emergere pattern comportamentali specifici. L'analisi può essere estesa per stimare, per esempio, il numero di attese transizioni verso un determinato stato; più in generale, può essere eseguita un'analisi statistica per estrarre parametri di interesse.

Sebbene macchine ed emozioni appaiano, ancora oggi, mondi inconciliabili, crediamo che lo sforzo per avvicinarli non sia ozioso, ma possa aiutare, da una parte, a progettare strumenti sempre più efficaci e utili alla società e, dall'altra, a comprendere a fondo i meccanismi che regolano la sfera emotiva umana.

Bibliografia

[AIBO, 1999] AIBO (1999). Web site:
http://www.sony.net/Products/aibo.

[Argyle, 1975] M. Argyle (1975). *Il corpo e il suo linguaggio. Studio sulla comunicazione non verbale*. Zanichelli, seconda edizione.

[ASIMO, 1986] ASIMO (1986). Web site:
http://world.honda.com/ASIMO/.

[Barron et al., 1994] J. L. Barron, D. J. Fleet e S. S. Beauchemin (1994). *Performance of optical flow techniques*. International Journal of Computer Vision, 12(1):43–77.

[Beauchemin e Barron, 1995] S. S. Beauchemin e J. L. Barron (1995). *The computation of optical flow*. ACM Computing Survey (CSUR), 27(3):433–466.

[D'Angelo, 2005] L. D'Angelo (2005). *Riconoscimento di espressioni facciali da parte di un robot mobile*. Tesi di Laurea Triennale in Informatica, Università degli Studi di Milano.

[D'Angelo e Colombo, 2004] L. D'Angelo e I. Colombo (2004). *Aibo Sony – Introduzione alla gestione delle immagini*. Web Site:
http://ais-lab.dsi.unimi.it.

[D'Angelo et al., 2004] L. D'Angelo, I. Colombo, G. De Caro e A. Quattro (2004). *Aibo Sony – Introduzione all'uso*. Web Site: *http://ais-lab.dsi.unimi.it*.

[Ekman, 1992] P. Ekman (1992). *An argument for basic emotions*. Cognition and Emotion, 6(3-4):169–200.

[Ekman e Friesen, 1978] P. Ekman e W. V. Friesen (1978). *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Inc.

[Gmytrasiewicz e Lisetti, 2002] P. J. Gmytrasiewicz e C. L. Lisetti (2002). *Emotions and personality in agent design*. In AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems, pagg. 360–361.

[Hopcroft e Ullman, 1979] J. E. Hopcroft e J. D. Ullman (1979). *Introduction to automata theory, languages and computation*. Addison-Wesley.

[Kismet, 1998] Kismet (1998). Web site: *http://www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html*.

[Kopecek, 2003] I. Kopecek (2003). *Constructing Personality Model from Observed Communication*. In Proc. 9th International Conference on User Modeling; Assessing and Adapting to User Attitides and Effect: Why, When and How?, pagg. 28–30.

[Lanzarotti, 2003] R. Lanzarotti (2003). *Facial Feature Detection and Description*. Tesi di Dottorato di Ricerca, Università degli Studi di Milano.

[Leonardo, 2001] Leonardo (2001). Web site: *http://robotic.media.mit.edu/projects/Leonardo/Leo-intro.html*.

[Parke e Waters, 1996] F. I. Parke e K. Waters (1996). *Computer facial animation*. A K Peters.

[Paz, 1971] A. Paz (1971). *Introduction to probabilistic automata*. Academic Press, New York and London.

[QRIO, 2003] QRIO (2003). Web site: *http://www.sony.net/SonyInfo/QRIO/*.

[Rabin, 1963] M. O. Rabin (1963). *Probabilistic Automata*. Information and Control, 6(3):230–245.

[Rinaldi, 1977] S. Rinaldi (1977). *Teoria dei sistemi*. Clup – Milano.

[Scherer, 2005] K. L. Scherer (2005). *What are emotions? And how can they be measured?* Social Science Information, 44(4):695–729.

[Serra e Baillie, 2003] F. Serra e J. C. Baillie (2003). *Aibo programming using OPEN-R SDK*. Web Site: *http://www.ensta.fr/~baillie*.