UNIVERSITÀ DI PADOVA

FACOLTÀ DI INGEGNERIA

TESI DI LAUREA

CaRo 2.0: UN MODELLO PER L'ESECUZIONE ESPRESSIVA DELLA MUSICA BASATO SUL PERCEPTUAL PARAMETRIC SPACE

Laureando: Davide Ganeo

Relatore: Prof. Sergio Canazza

Correlatori: Maestro Davide Tiso, Prof. Antonio Rodà

Corso di Laurea Magistrale in Ingegneria Informatica

Anno Accademico 2010-2011

Ringraziamenti

Ringraziare in poche righe tutti coloro che mi hanno permesso di raggiungere questo risultato non è per niente facile, ma farò del mio meglio. Innanzitutto un grazie speciale va alla mia famiglia, in particolare ai miei genitori che mi hanno dato la possibilità di affrontare in modo sereno la carriera universitaria, sostenendomi nei momenti difficili e dandomi tutto l'aiuto necessario per arrivare al traguardo della laurea. Ovviamente ringrazio anche Fabio, mio fratello e mio primo creditore. Grazie a tutti i colleghi ingegneri Massimiliano, Michele, Luca, Federico e Lucio indispensabili compagni di studio e di pausa caffè. Grazie a tutti gli amici (Stefano P., Marco, Erik, Matteo, Silvia, Alessandro, Stefano S., Nicola, Stefano B., Alice, Marco, Annalisa, Erika, Andrea), fonte essenziale di svago e divertimento nonché ottimi consiglieri e compagni di bevute. Un ringraziamento particolare anche ad Eliana e Maristella, la parte femminile del gruppo studio (e non solo) estivo. E grazie anche ai prof. Canazza e Rodà, per l'aiuto e i preziosi consigli che mi hanno dato per questo lavoro di tesi. Per ultima (ma non in ordine di importanza) ringrazio una persona speciale a cui devo tanto sotto molti punti di vista. Lei è mia amica, confidente, segretaria, psicologa, compagna di vita e molto altro. È una parte fondamentale della mia vita. Grazie di cuore, Marta.

Sommario

I musicisti di rado suonano la musica esattamente come compare nello spartito: la interpretano aggiungendo proprie *intenzioni espressive*. Questa Tesi di Laurea intende dare un contributo innovativo nella definizione di un modello computazionale delle variazioni espressive inserite dagli interpreti umani in un'esecuzione musicale, basato su uno spazio di controllo bidimensionale correlato alle categorie fisiche *energia* e *cinetica* (Cap. 1). Utilizzando questo modello è possibile simulare l'espressione musicale di un performer, né più né meno di come il viso di un cartone animato (magari disegnato con l'aiuto di un computer) è in grado di far trasparire emozioni quali gioia o tristezza.

Nel Cap. 2 viene tratteggiato lo stato dell'arte relativo ai sistemi informatici per l'esecuzione musicale espressiva realizzati dai maggiori laboratori di *Sound and Music Computing* del mondo.

Il Cap. 3 presenta il modello realizzato al Centro di Sonologia Computazionale dell'Università di Padova, al cui sviluppo ha contribuito questo lavoro di Tesi. Il sistema interattivo CaRo 2.0, basato su questo modello, è stato presentato dall'Università di Padova al RenCon (*Rendering Contest*, v. Cap. 4) 2011, una delle maggiori competizioni a livello mondiale nel campo dell'informatica musicale. Si tratta di un *contest* finalizzato a selezionare il miglior sistema computazionale per l'esecuzione musicale espressiva automatica: ovvero calcolatori in grado, non solo di suonare, ma di farlo con una propria *intenzione espressiva*. Nel 2011 la fase finale della manifestazione è stata ospitata dal Dipartimento di Ingegneria dell'Informazione dell'Università di Padova e ha visto come vincitore il software CaRo 2.0.

Nel Capitolo 5 vengono discusse alcune considerazioni relative alle finalità scientifiche e commerciali della disciplina internazionalmente nota come *Expressive Information Processing*.

Indice

Capitolo 1

Introduzione

Le persone interagiscono con la musica in svariati e complessi modi. In base al contesto, la musica può assumere differenti funzioni: può essere il risultato di un processo di creazione, una partitura da interpretare, un suono strutturato da ascoltare o un oggetto da studiare da una prospettiva storica e culturale. Tutto ciò rende difficile progettare sistemi di elaborazione musicale in grado di coprire tutti gli aspetti dell'interazione fra l'uomo e la musica. Negli ultimi anni sono stati effettuati diversi studi per raggiungere questo obiettivo, focalizzati di volta in volta su specifici contesti applicativi.

Alcuni studi si sono concentrati sulla rappresentazione del segnale audio, altri sulla rappresentazione simbolica della musica, altri ancora sulla definizione di un framework XML (Haus et.al, 2002) per la rappresentazione strutturata dei differenti aspetti musicali.

Canazza (in Press.) afferma che la formalizzazione degli aspetti correlati a una performance musicale possano estendere la possibilità di interazione tecnologica con contenuti musicali. Informazioni sulla performance musicale, strutturate come metadati, possono consentire lo sviluppo di nuove applicazioni (ad esempio il software CaRo 2.0 descritto nel Capitolo 3) e offrire un contributo per migliorare sistemi già esistenti.

Diversi problemi sono legati alla definizione di una struttura dati per rappresentare un'esecuzione musicale. Innanzitutto, le caratteristiche di una performance musicale sono pesantemente influenzate dagli aspetti culturali, quindi è difficile trovare un modo generale per descriverle. Inoltre, l'espressività in ambito musicale ancora non è un concetto completamente definito e non esiste un'opinione comune su cosa la musica esprima (Imberty, 1986).

La comunicazione di un contenuto espressivo attraverso la musica può essere studiata a tre diversi livelli considerando:

- il messaggio del compositore;
- l'intenzione espressiva del musicista;
- l'esperienza dell'ascoltatore.

Il termine *intenzione espressiva* (Gabrielsson, 1996) evidenzia l'esplicita intenzione di un esecutore di comunicare un sentimento o un'emozione. Può essere descritta per mezzo di aggettivi sensoriali o legati alle emozioni.

Molti studi sul rendering espressivo di un'esecuzione musicale affiancano la presenza sistematica di deviazioni dalla partitura originale con l'intento del musicista di comunicare qualcosa all'ascoltatore. Le deviazioni introdotte da costrizioni tecniche o da imperfezione dell'esecutore non vengono comprese nella comunicazione espressiva e quindi sono spesso considerate rumore. L'analisi di queste deviazioni ha portato alla formulazione di diversi metodi con l'obiettivo di descrivere *come, quando* e *perché* un musicista modifica, a volte inconsciamente, la partitura originale. È necessario sottolineare che, anche se alcune deviazioni sono solo la parte superficiale di qualcosa di più profondo che spesso non è direttamente accessibile, queste sono facilmente misurabili e quindi sono ampiamente utilizzate per sviluppare modelli computazionali.

1.1 Elaborazione dell'informazione espressiva: gli spazi espressivi

Il meccanismo di comprensione di una performance musicale può essere descritto attraverso differenti livelli seguendo un approccio bottom up (Camurri et al., 2005):

- Livello 1: segnali fisici: segnale audio campionato.
- Livello 2: caratteristiche di basso livello e parametri statistici: estrazione di misure da una collezione di segnali audio che vengono elaborati attraverso metodi statistici.

- Livello 3: caratteristiche di medio livello e mappe: “L’obiettivo è rappresentare l’espressività, modellando le caratteristiche di basso livello in termini di eventi, forme, traiettorie nello spazio o mappe” (Camurri et.al, 2005). Una performance è suddivisa in *gesti musicali* ognuno dei quali è rappresentato dalle misure estratte al livello precedente.
- Livello 4: concetti e strutture: lo sono, ad esempio, i contenuti emozionali e KANSEI[1]. Queste informazioni ad alto livello sono elaborate a partire dalle caratteristiche di medio e basso livello attraverso varie tecniche di analisi.

Un ulteriore passo per la comprensione dell’espressività, potrebbe essere l’assegnazione di etichette che identificano le intenzioni espressive. Tuttavia queste etichette sono una descrizione troppo riduttiva. Inoltre, le persone utilizzano un numero troppo elevato di espressioni per descrivere le proprie intenzioni. Ad esempio, nelle sue ricerche sulle emozioni, Plutchik (1994) trovò centinaia di descrittori. È difficile immaginare un sistema artificiale in grado di discriminare ad un livello così dettagliato. Per questo motivo è stato preferito un approccio dimensionale nella descrizione dell’espressività, che include rappresentazioni sottoforma di mappe o spazi a poche dimensioni. In letteratura sono state proposte differenti soluzioni per rappresentare questi spazi. Plutchik (1980) offrì una formulazione dell*’emotion wheel* (Figura 1.1) come riferimento di posizione in uno spazio di *valence-activity* (Russel, 1980). Questa è una rappresentazione semplice e in grado di descrivere un’ampia gamma di caratteristiche dell’emotività. Si basa su due termini chiave:

- Valenza, il più comune elemento di uno stato emozionale che rappresenta quanto i sentimenti di una persona sono influenzati dalle valutazioni positive o negative di persone, cose o eventi (Cowie et.al, 2001).
- Livello di attivazione, cioè quanto una persona è disposta a compiere delle azioni anziché niente (Cowie et.al, 2001).

Gli assi dello spazio *valence-activity* descrivono queste variabili. In particolare l’asse verticale rappresenta il livello di attivazione mentre quello orizzontale rappresenta il livello di valutazione.

[1] KANSEI Information Processing: approccio sviluppato in Giappone per la comprensione di comunicazioni con contenuto espressivo.

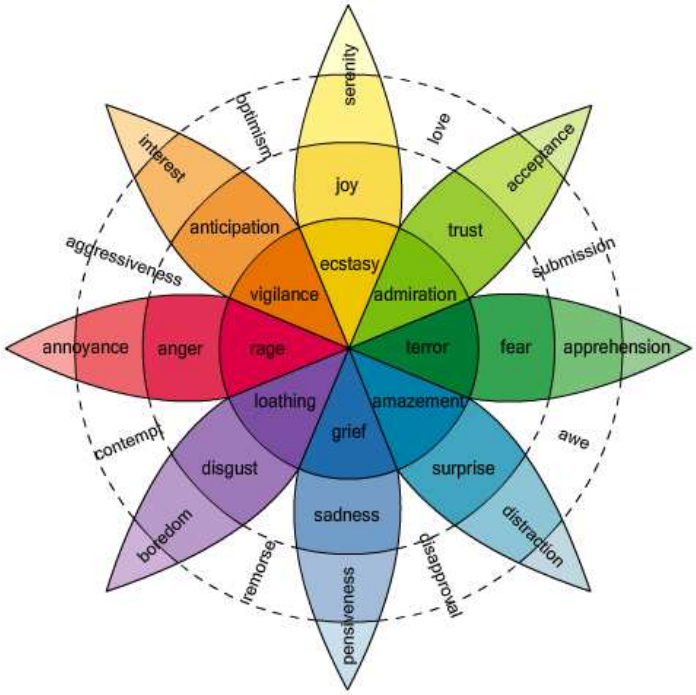


Figura 1.1: Emotion Wheel (Plutchik, 1980).

Juslin (2001) ha presentato uno studio sull'espressione emozionale in una performance musicale. Utilizzando cinque emozioni di base, *happiness, sadness, anger, fear* e *love/tenderness* ha proposto uno spazio bidimensionale (*valence* vs. *activity*) combinando l'approccio categorico (ad esempio *happiness*) e dimensionale (livello di attivazione) all'espressione emotiva (Figura 1.2). Ogni emozione è stata collocata in un punto approssimativo dello spazio bidimensionale costituito da *valence* e *activity level.* Il posizionamento di ogni emozione è stato fatto analizzando i risultati di una ricerca in cui era stato chiesto ai partecipanti di valutare la valenza e il livello di attivazione di 400 termini emotivi.

Come si può osservare in Figura 1.2, le caratteristiche acustiche includono tempo, intensità, intonazione, articolazione, timbro, vibrato, attacco e rilascio di un tono e pause. Sia il livello medio che la varianza lungo tutta performance di questi parametri influiscono sul processo comunicativo. Ad esempio, l'espressione *sadness* è associata ad un tempo lento, un basso livello di intensità e un timbro morbido mentre *happiness* è associata ad un tempo veloce, un alto livello di intensità e un timbro brillante.

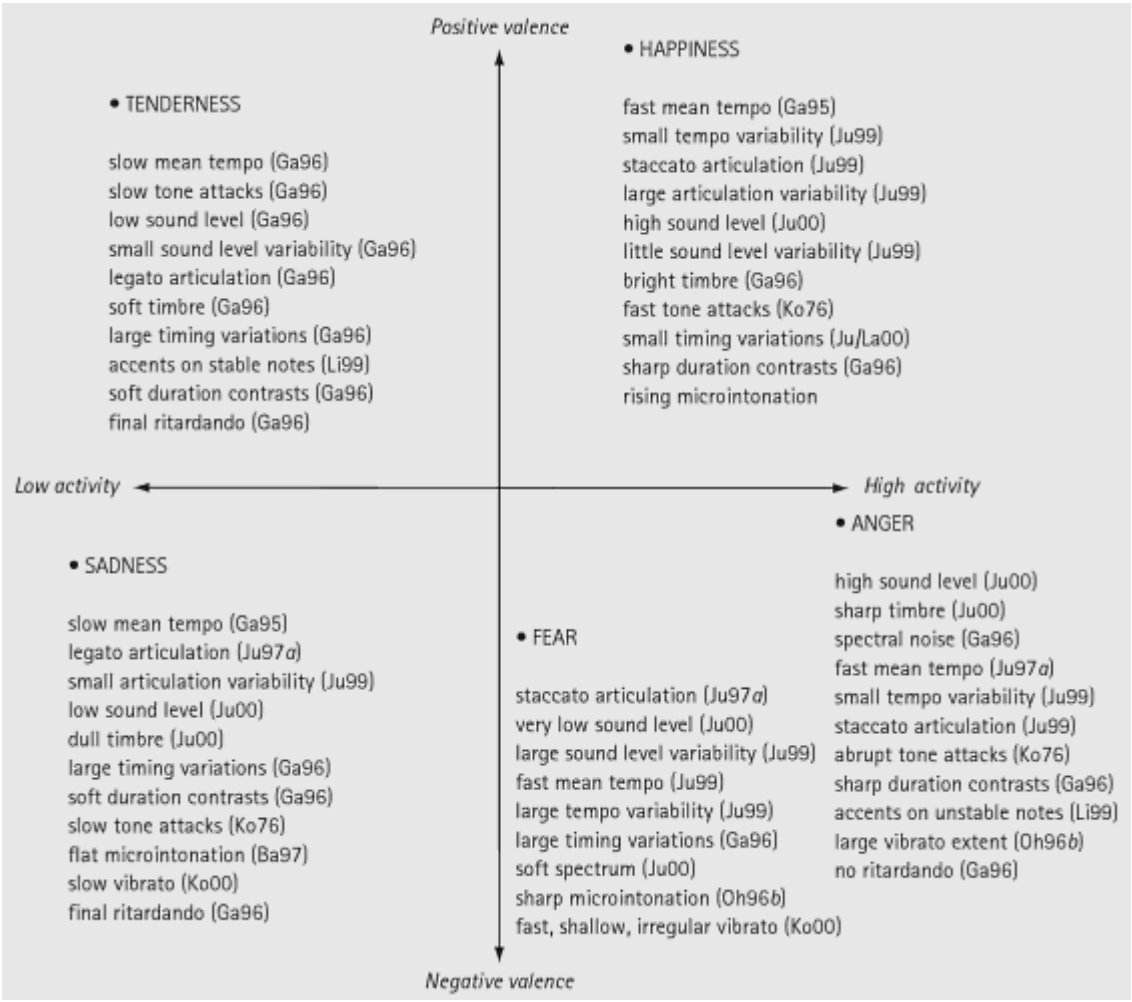


Figura 1.2: Spazio bidimensionale *valence* vs. *activity* di Juslin (2001).

1.1.1 Kinematics-Energy Space

Il CSC (Centro Sonologia Computazionale dell'Università di Padova) ha esplorato un differente approccio alla comprensione dimensionale su piccoli ma significativi sottoinsiemi di intensioni espressive nel dominio sensoriale. Il modello proposto da Canazza (2003) vede la definizione di uno spazio bidimensionale in relazione ad aggettivi di tipo sensoriale.

La metodologia utilizzata è riassunta in Figura 1.3. Inizialmente sono state valutate, attraverso esperimenti di tipo percettivo, una serie di esecuzioni basate su diverse intenzioni espressive. Dalla fase di comprensione è stata derivata una struttura a bassa dimensionalità attraverso metodi di analisi multivariata dei dati. Infine, ogni

esecuzione è stata analizzata con l'obiettivo di estrarre i parametri acustici che l'esecutore ha modificato per comunicare le diverse intenzioni espressive.

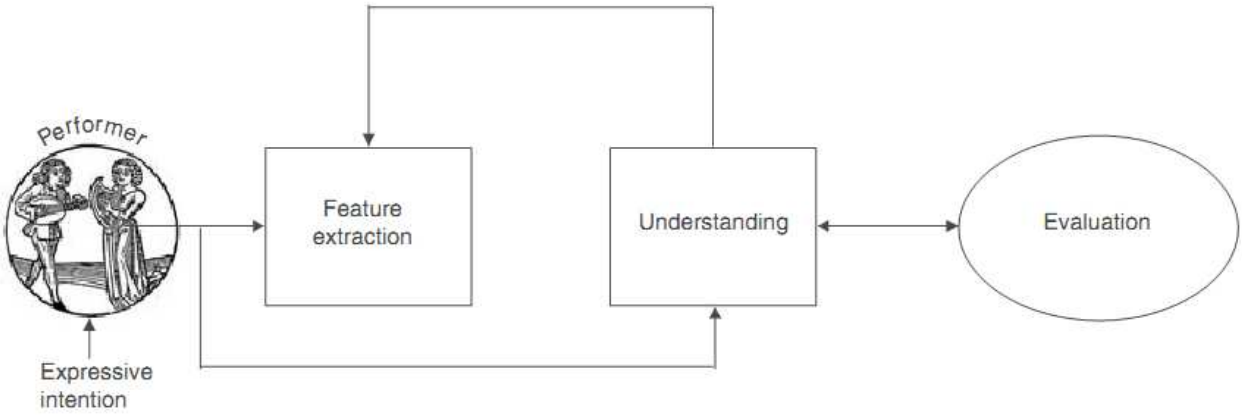


Figura 1.3: Analisi delle intenzioni espressive nel dominio sensoriale (Canazza et al., 2003).

Per lo sviluppo di questo modello è stato fondamentale disporre di diverse interpretazioni espressive dello stesso brano eseguite da più musicisti professionisti. Una perfomance musicale espressiva non ha sempre connotazioni emotive forti, ma esistono alcuni generi musicali che sfruttano la possibilità di interpretazione per aumentare il coinvolgimento dell'ascoltatore. Un esempio di questo fenomeno lo si trova nella musica jazz. Nello specifico, per l'esperimento condotto da Canazza, è stato cruciale avere a disposizione diverse interpretazioni espressive del medesimo brano nella fase di valutazione delle stesse.

I brani selezionati per l'esperimento sono stati eseguiti da musicisti professionisti secondo intenzioni espressive diverse seguendo gli aggettivi: *light, heavy, soft, hard, bright,* e *dark.* Inoltre, per ogni brano è stata effettuata una registrazione neutra.

Per ogni aggettivo, i partecipanti hanno espresso una valutazione quantitativa in ogni performance musicale presentata. Ai risultati prodotti da questo esperimento percettivo è stata applicata una doppia analisi fattoriale usando come variabili le esecuzioni e gli aggettivi valutati. Tramite un'analisi multivariata di tipo MDS [2] è stato stimato quanto le esecuzioni fossero distinguibili tra loro e in che modo possano mantenere la distanza in uno spazio a dimensionalità ridotta rispetto alle variabili di

[2] MDS (MultiDimensional Scaling): tecnica esplorativa dei dati che permette di ottenere una rappresentazione di 'n' oggetti in uno spazio a 'k' dimensioni derivate da informazioni relative alla similarità o dissimilarità tra ciascuna coppia di oggetti.

partenza. Infine, è stata condotta una *cluster analysis*, per valutare se i partecipanti all'esperimento avessero giudicato in modo simile alcuni aggettivi, provocando così la vicinanza di questi nello spazio dimensionale.

I risultati ottenuti hanno evidenziato una dimensionalità ottimale per quanto riguarda le tre dimensioni (Figura 1.4). Lo spazio ottenuto rappresenta un modello per l'espressività, nel quale i partecipanti hanno organizzato le loro percezioni dei brani. L'analisi acustica delle esecuzioni (Canazza et al., 1997b) ha mostrato come il Fattore 1 sia correlato con il tempo ed interpretato come fattore cinematico; il Fattore 3 invece, risulta correlato con il tempo di attacco, al legato/staccato e all'intensità ed è quindi interpretato come fattore energetico. Da qui la definizione dello spazio *Kinematics-Energy.*

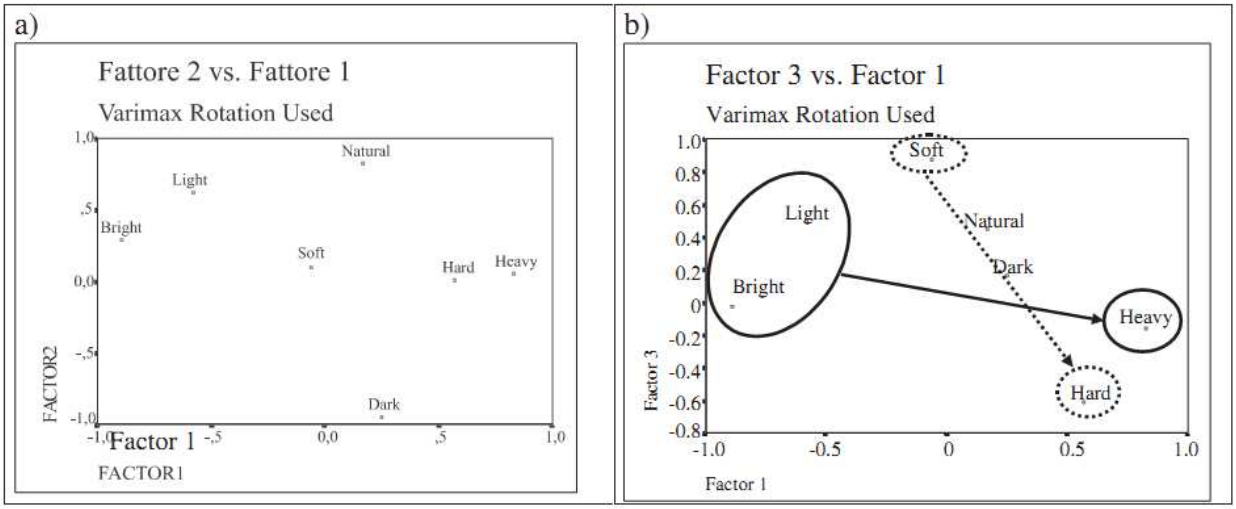


Figura 1.4: Risultato dell'analisi fattoriale (Canazza et al., 2003).

1.2 Modelli per la descrizione di performance musicali

La caratteristica principale di un sistema automatico per la performance musicale è la capacità di convertire una partitura musicale in una performance musicale espressiva includendo deviazioni di tempo, suono e timbro a partire da una riproduzione neutra della partitura. Generalmente, le principali strategie per progettare tale sistema sono il metodo di analisi per misura e il metodo di analisi per sintesi. Di recente si è

cominciato ad utilizzare anche altre tecniche derivanti dall'intelligenza artificiale (*machine learning e case-based reasoning*).

1.2.1 Metodo di analisi per misura

La strategia di analisi per misura è basata sull'analisi delle deviazioni che caratterizzano le performance umane. Lo scopo è quello di riconoscere delle regolarità all'interno di esecuzioni espressive e descriverle attraverso modelli matematici. Il metodo è strutturato in cinque diverse fasi:

1. Selezione delle performance. È fondamentale la scelta di una buona e/o tipica esecuzione musicale da studiare. Generalmente si preferisce analizzare piccoli insiemi di performance accuratamente selezionate. A seconda del tipo di esperimento che si vuole eseguire, all'esecutore può essere lasciata la libertà di riprodurre un brano secondo i propri gusti oppure di suonare ispirato da specifiche emozioni.
2. Misura delle proprietà fisiche di ogni singola nota. Le variazioni delle proprietà fisiche di una performance sono molteplici: durata, intensità, frequenza, inviluppo, vibrato. Decidere come e quale variabile studiare dipende dallo scopo dell'esperimento, dalle ipotesi iniziali che sono state formulate, dalle possibilità tecniche dello strumento e dallo strumento utilizzato.
3. Controllo dell'affidabilità e classificazione delle performance. È necessario valutare il grado di accuratezza e consistenza dei dati ottenuti dalle misurazioni delle variabili fisiche, classificando le esecuzioni in differenti categorie a seconda dei dati raccolti.
4. Selezione ed analisi delle variabili più significative. Questa fase dipende dalle due precedenti e conclude la parte analitica dello schema per dare spazio al giudizio degli ascoltatori nelle fasi successive.
5. Analisi statistica e sviluppo di un modello matematico. L'analisi delle variabili selezionate è spesso condotta con differenti scale di rappresentazione.

Gli approcci maggiormente utilizzati sono i modelli statistici e quelli matematici. A volte viene eseguita un'analisi multidimensionale sulle performance con l'obiettivo di identificare dei pattern indipendenti.

Diverse metodologie di approssimazione di un'esecuzione umana sono state sviluppate con questo metodo. Ad esempio l'approccio con logica fuzzy (descritto nel paragrafo 2.2), che utilizza reti neurali e la teoria degli spazi vettoriali lineari basata su algoritmi di regressione multipla, consente la generazione di un modello parametrico stimato a partire da un insieme dato di performance (Bresin 1998, Friberg 2004).

1.2.2 Metodo di analisi per sintesi

Il paradigma di analisi per sintesi si focalizza sulla percezione della performance ed è il passo successivo del processo iniziato con l'analisi per misura (fasi 1-5 del paragrafo precedente). In particolare si aggiungono i seguenti step:

6. Sintesi di performance con deviazioni sistematiche. In questa fase si analizzano diverse versioni del brano in cui le variabili fisiche oggetto di studio (durata, intensità ecc.) variano sistematicamente.
7. Giudizio sulla versione sintetizzata del brano, prestando particolare attenzione ai diversi aspetti sperimentali selezionati. È richiesta la conoscenza delle principali variabili sperimentali e della loro scala di valutazione.
8. Controllo dell'affidabilità del giudizio e classificazione degli ascoltatori. È fondamentale un metodo adeguato di controllo degli ascoltatori con lo scopo di verificare la consistenza dei loro giudizi, possibilmente classificandoli in diverse classi.
9. Studio della relazione fra le variabili della performance e quelle sperimentali. A questo punto si studiano le relazioni fra le esecuzioni con i parametri fisici modificati e le variabili fisiche selezionate per l'esperimento.
10. Ripetizione della procedura dalla fase 3 alla fase 9 finché i risultati convergono. In base ai risultati delle fasi 3-9 il processo può continuare in modo iterativo fino a quando le relazioni delle variabili selezionate convergono alle variabili sperimentali.

Mediante l'uso di questa strategia si derivano modelli che possono essere descritti attraverso delle regole. Il modello più importante è il sistema di regole (Friberg, 1991)

sviluppato al *Royal Institute of Technology* di Stoccolma (KTH) e descritto nel paragrafo 2.1.

1.2.3 Machine Learning

Generalmente, nello sviluppo dei modelli per l'espressività musicale, i ricercatori fanno alcune ipotesi sugli aspetti delle performance che vogliono modellare e successivamente cercano di stabilire la validità empirica del modello testandolo su dati reali. Un approccio differente è quello di Widmer (1996) il quale cerca di estrapolare regolarità nuove e potenzialmente interessanti da performance di esempio attraverso l'apprendimento automatico e algoritmi di data mining. Lo scopo di questi modelli è quello di scoprire complesse dipendenze su insiemi molto grandi di dati, senza effettuare nessuna ipotesi preliminare. Il vantaggio principale consiste nella possibilità di ottenere nuove informazioni evitando qualsiasi assunzione di carattere musicale.

1.2.4 Case-Based Reasoning

Un approccio alternativo, molto vicino al processo *osservo-imito-sperimento* comune fra gli essere umani, consiste nell'uso diretto della conoscenza implicitamente trasmessa da una performance umana. Sotto l'assunzione che un problema simile ha una soluzione simile, l'approccio *Case-Based Reasoning* (CBR) (Arcos, 2001) tenta di risolvere un problema utilizzando la stessa (o adattata) soluzione valida per un problema precedentemente risolto. Sono utilizzati due meccanismi di base:

- Ricerca di problemi risolti (cause) secondo particolari criteri,
- Adattamento delle soluzioni utilizzate nei casi precedenti per l'attuale problema.

Il paradigma CBR può essere descritto attraverso una scomposizione in sotto-attività:

- Recupero.
- Riutilizzo.
- Revisione.
- Mantenimento.

La prima attività, il recupero, ha lo scopo di scovare un insieme di problemi precedentemente risolti che presentano somiglianze con il problema corrente. Questa fase, a sua volta, è divisa in tre parti: identificazione, ricerca e selezione. L'identificazione determina un insieme di aspetti rilevanti nel problema corrente utilizzando la conoscenza sul dominio, la ricerca utilizza gli aspetti rilevanti precedentemente ottenuti per recuperare un insieme di problemi affini già risolti, infine nella fase di selezione si classificano i risultati della ricerca. L'insieme ordinato dei casi risolti, generato nella fase di identificazione, è il punto di partenza per la costruzione della soluzione del problema corrente (riutilizzo). Se la soluzione generata non è corretta si dà al sistema la possibilità di "imparare". Questo è il compito della fase di revisione nella quale vengono identificati gli errori della soluzione proposta e, tramite l'applicazione di tecniche di riparazione, viene formulata una soluzione alternativa. I risultati ottenuti vengono successivamente applicati al mondo reale. Infine, nella fase di mantenimento, i nuovi problemi risolti vengono integrati nel sistema per facilitare la risoluzione di futuri nuovi problemi. Il paradigma CBR è molto utile quando sono disponibili molti esempi di problemi risolti e quando la maggior parte della conoscenza utilizzata per risolvere i problemi è implicita e difficile da analizzare e generalizzare. Tuttavia, disporre di un grande archivio di problemi già risolti ed efficientemente organizzati non è facile e ciò, sfortunatamente, è il punto centrale del successo dell'approccio CBR.

Capitolo 2

Modelli espressivi per la performance musicale

In questo capitolo vengono illustrati alcuni fra i modelli proposti in letteratura per il rendering espressivo di performance musicali.

2.1 Modello basato su regole

Uno dei principali metodi utilizzato nel campo dei sistemi automatici per la performance musicale è un modello basato su regole sviluppato al KTH a Stoccolma. Esso consiste in una grammatica generativa per la performance musicale che include approssimativamente trenta regole. Queste regole, ottenute principalmente con il metodo di analisi per sintesi, sono state implementate nel *Director Musices (DM) Program* (Friberg, 1995; Friberg et.al, 2000) e possono essere combinate in modo tale da produrre deviazioni sulla durata e sull'intensità della nota, sul tempo e intensità globale e anche sul timbro dello strumento. Ogni nota può essere elaborata tramite diverse regole e la deviazione espressiva prodotta viene solitamente aggiunta al set delle regole.

2.1.1 Sviluppo delle regole

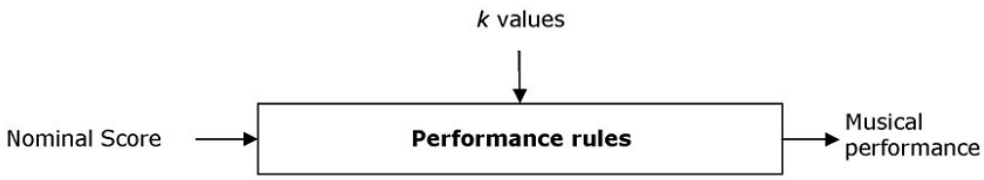


Figura 2.1: Schema per la generazione delle regole (Friberg et al., 2006).

Come illustrato in Figura 2.1, il modello è molto semplice. L'input consiste nella partitura originale che, attraverso l'applicazione delle regole, produce la performance musicale. Le regole agiscono su diversi parametri della performance come, ad

esempio, il tempo e il volume. Il parametro *k*, invece, determina l'incidenza delle regole sulla partitura: piccoli valori di *k* sono utilizzati quando si desidera apportare piccoli cambiamenti, mentre elevati valori producono marcati cambiamenti espressivi. A seconda delle regole, differenti combinazioni dei valori del parametro *k* possono essere utilizzate per modellare differenti stili e intenzioni espressive. Per questo motivo non esiste un settaggio ottimale dei valori di *k* che possono essere adatti per ogni tipo di musica. Il risultato è una rappresentazione simbolica della performance musicale che può essere utilizzata per controllare un sintetizzatore.

La maggior parte delle regole è costituita da due parti: una *context part* che descrive quando attivare la regola e una *execution part* che descrive come questa regola incide sulla situazione musicale. L'obiettivo di un sistema basato su regole è quello di trovare dei principi generali di performance musicale. Ciò significa che specifiche regole devono essere applicate per determinate situazioni musicali e queste devono essere indipendenti dallo stile e dal strumento musicale. Per questo motivo, per la generazione degli intenti espressivi si agisce sulle caratteristiche melodiche (pitch, durata) piuttosto che sulle caratteristiche metriche.

Il tempo deve essere considerato nella definizione della *context part* in quanto è importante che tutte le regole abbiano il medesimo effetto percettivo indipendentemente dal tempo. Per questa ragione, le deviazioni espressive relative al tempo sono il punto di partenza dell'implementazione della maggior parte delle regole.

Bisogna inoltre considerare il ruolo dei *performance mark* presenti nella partitura. Spesso, questi marcatori sono trattati come delle linee guida per l'interpretazione della partitura anziché come dei veri obblighi per la riproduzione del brano. Tuttavia si è scelto di non consentirne l'inserimento preferendo modellare la performance a partire da una partitura grezza.

2.1.2 Tipologie di regole

In questo paragrafo vengono descritte le principali regole implementate all'interno del *Director Musices (DM) Program*, descrivendo per ognuna di esse l'effetto sull'esecuzione del brano.

2.1.2.1 Phrasing

Una frase musicale è rappresentata da una forma ad arco applicata al tempo e alle dinamiche. La frase è tipicamente lenta/morbida all'inizio, veloce/ad alta intensità nel mezzo per poi tonare lenta nel finale. Questo andamento modella un pattern *crescendo/accelerando, decrescendo/rallentando.* Todd (1985, 1989) propose un primo prototipo per grandi frasi basandosi sui dati ottenuti da un piano al quale erano stati aggiunti dei sensori per poter registrare la pressione di ogni tasto. La regola *Phrase arch* estende il lavoro di Todd, includendo ulteriori parametri per descrivere le variazioni riscontrate nelle performance reali. Queste variazioni riguardano il livello strutturale delle frase, la posizione di massimo dell'arco, la durata dell'ultima nota della frase e la forma della curva. Solitamente, diverse istanze della regola *Phase arch* vengono applicate simultaneamente agendo su diversi livelli della struttura del brano.

Altri esempi possono essere le regole *Final ritardando* che fornisce un fraseggio alternativo per la fine di un brano e *High loud* che incrementa l'intensità del suono in modo proporzionale al pitch.

2.1.2.2 Micro-level timing

Un'attenta analisi dell'*inter-onset-interval* (IOI) misurato su performance reali, rivela come ogni nota sia riprodotta con piccole variazioni rispetto alla durata nominale data dalla partitura. Questo fenomeno è in parte dovuto ad imprecisioni durante l'esecuzione, tuttavia si possono identificare diverse variazioni sistematiche. Widmer (2002) ad esempio, ha identificato 17 regole per descrivere variazioni di tempo locali nell'applicazione di metodi derivanti dal machine learning su performance al piano.

La regola *Duration contrast* incrementa la differenza in termini di IOI fra note differenti, in modo che le note lunghe sono ulteriormente allungate mentre le note corte vengo accorciate. L'incremento o la diminuzione della durata (misurato in ms) è regolato da un parametro della regola e dipende dal valore originale dell'IOI.

La regola *Faster uphill* completa la *Duration contrast* in termini di valore di pitch, accorciando le note il cui andamento è ascendente.

2.1.2.3 Pattern metrici

Una musica con un ritmo regolare induce una percezione più stabile di tutti i parametri metrci. Questi pattern ritmici si traducono in variazioni dell'IOI o dell'intensità.

Un pattern ritmico molto comune, presente soprattutto con le metriche 3/4 e 6/8, consiste in una mezza nota seguita da un quarto di nota. Questo pattern è molto spesso riprodotto con una riduzione del contrasto di durata (Gabrielsson, 1987). La regola *Duoble duration* descrive questo comportamento riducendo il contrasto di durata mantenendo inalterata la durata totale dei due toni.

Un altro pattern molto comune è l'alternanza di pattern lungo-corto presente in molti stili musicali come folk o jazz. La regola *Inégales* ne è l'implementazione.

2.1.2.4 Articulation

Il termine articulation è utilizzato per descrivere l'ammontare di *legato/staccato* con cui viene riprodotta ogni nota. È definito come il rapporto fra la durata della nota e l'IOI; perciò un valore vicino ad 1 rappresenta legato mentre un valore attorno a 0,5 indica staccato. Questo è un parametro molto importante per cambiare il carattere complessivo di un brano includendo aspetti emozionali (De Poli et al., 1998).

La regola *Punctuation* identifica piccoli segmenti di melodia e li allunga aggiungendo alla fine dell'ultima nota una micro pausa. La fase di analisi utilizza un sottoinsieme di 13 regole per determinare i limiti dei segmenti della melodia.

Le regole *Score legato* e *Score staccato* vengono applicate in corrispondenza dei relativi marcatori nella partitura originale. Queste regole modificano il tempo a seconda delle indicazioni metriche e delle intenzioni espressive dell'artista.

La regola *Repetition articulation* aggiunge una micro pausa fra note ripetute. La sua durata varia a seconda delle intenzioni espressive del musicista.

Infine, la regola *Overall articulation* può essere utilizzata per cambiare l'articulation di tutte le note. Ciò è utile per controlli in tempo reali o per elaborare espressioni emozionali.

2.1.2.5 Tonal tension

Il concetto di tensione tonale è stato utilizzato per molto tempo nella teoria della musica e della psicologia per spiegare, ad esempio, il pattern tensione-rilassamento percepito in una cadenza armonica.

La regola *Melodic Charge* produce variazioni sulla durata, l'intensità e l'ampiezza del vibrato. Gli incrementi sono in relazione con un valore, attribuito a ciascuna nota della scala, partendo da una fondamentale e seguendo il circolo delle quinte[3].

In modo simile, la regola *Harmonic Change* enfatizza ad un livello più alto quelle note della melodia che si presentano nel momento in cui l'accordo cambia. In questo modo la regola agisce su spazi temporali più lunghi tenendo conto del percorso armonico contenuto in un intero periodo musicale.

Infine, la regola *Chromatic charge* sostituisce le due regole precedenti per la musica atonale.

2.1.2.6 Intonation

La regola *Melodic Intonation* determina l'intonazione di ogni nota in base al contesto melodico e all'accordo corrente. Questa regola (elaborata specificatamente per melodie monofoniche) è un complemento della regola *Melodic Charge,* infatti accorda le note della scala modificando l'intonazione in linea con il carico melodico.

La regola *Harmonic Intonation* è stata elaborata per minimizzare il tempo di battuta negli accordi attraverso un'intonazione simultanea di tutte le note, utilizzando l'accordo corrente come riferimento.

Le due regole precedenti sono state combinate dando vita alla regola *Mixed Intonation.* Ogni nota è inizialmente intonata applicando la regola *Melodic Intonation;* successivamente, applicando la regola *Harmonic Intonation,* l'intonazione è lentamente modificata in linea con l'intonazione dell'accordo corrente.

[3] Circolo delle quinte: grafico utilizzato nella teoria musicale per mostrare le relazioni tra le dodici note che compongono la scala cromatica.

Infine, la regola *High sharp,* attraverso la modifica della frequenza delle note, adegua l'esatta intonazione alle capacità percettive dell'udito.

2.1.2.7 Esemble timing

Dal momento che molte delle regole fin qui descritte introducono piccole e indipendenti variazioni di tempo ad ogni parte del brano, una performance polifonica può risultare non sincronizzata. L'applicazione delle regola *Melodic Sync* risolve qualsiasi dei suddetti problemi di sincronizzazione. Innanzitutto, tutte le voci del brano vengono processate dalle regole che non introducono variazioni di tempo. A questo punto la regola costruisce una nuova voce fittizia, prendendo tutte le note di minor valore presenti momento per momento nella partitura e vi applica le deviazioni di tempo.

2.1.2.8 Performance noise

Variazioni casuali (ad esempio rumore) sono sempre presenti in qualsiasi esecuzione di un musicista. Risulta difficile modellare queste variazioni in quanto il rumore è difficile da separare dalle variazioni intenzionali volute dall'esecutore. Per fare ciò, è stata creata la regola *Noise.*

2.1.2.9 Combinazione di regole

Nella fase di rendering di una performance musicale, possono essere applicare diverse regole alla stessa partitura. È possibile ottenere una specifica esecuzione selezionando delle regole e i corrispettivi parametri da un set predefinito. Ad esempio, per un fraseggio globale si utilizza sia la regola *Phrase arch* sia la *Final ritardando* e queste si sovrapporranno in prossimità della fine del brano. Quando vengono combinate diverse regole, l'effetto di ognuna di esse è sommato o moltiplicato alla altre e il risultato è applicato ad ogni parametro di esecuzione. Nel caso in cui diverse regole operino sulla stessa nota e sullo stesso parametro, si possono verificare degli effetti collaterali indesiderati. Ad esempio, quando diverse regole operanti sulla durata di una nota vengono combinate, il risultato può essere una nota troppo lunga. Tuttavia, i principali conflitti fra regole sono stati risolti nel

contesto di definizione delle regole. In Figura 2.2 è illustrato l'effetto della variazione dell'IOI dopo l'applicazione di sei regole alla stessa melodia.

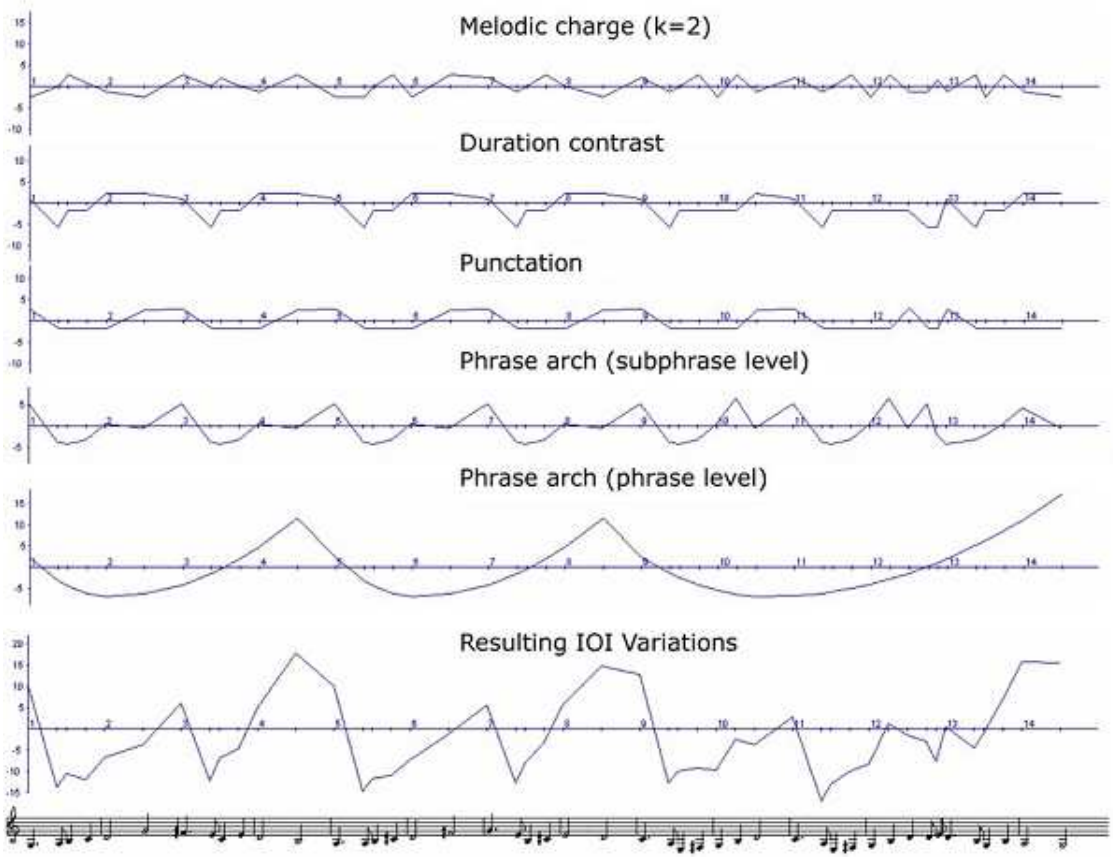


Figura 2.2: Variazione dell'IOI in seguito all'applicazione delle regole *Phrase arch, Duration contrast, Melodic charhe* e *Punctuation* (Friberg et al., 2006).

2.2 Modello basato su reti neurali

Roberto Bresin (1998) ha proposto l'idea di combinare sistemi basati su regole (*rule-based DM system*) con reti neurali artificiali (ANN, *Artificial Neural Network*), proponendo un sistema ibrido per la performance musicale in tempo reale basato sull'interazione di regole simboliche e sub-simboliche (Figura 2.3). L'idea principale è di sviluppare un sistema real-time per la simulazione dello stile di un pianista professionista. Per questa ragione il sistema deve essere basato su informazioni locali e quindi opera direttamente sulla struttura musicale.

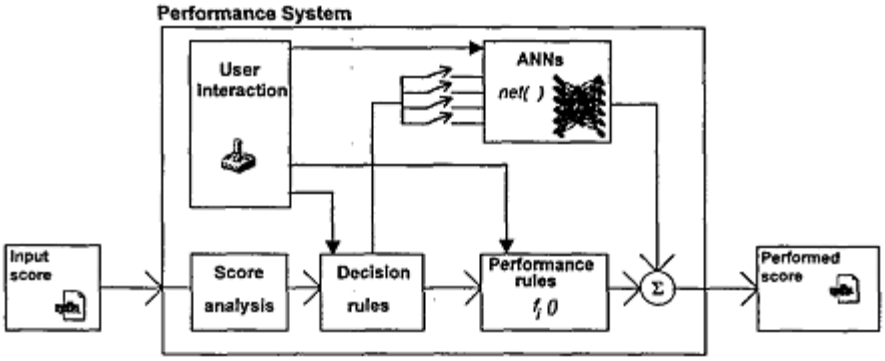


Figura 2.3: Schema a blocchi del sistema (Bresin, 1998).

La rete neurale è basata sul modello *feed-forward* e allenata con l'algoritmo *back-propagation error*. Le regole *DM* giocano un ruolo importante nel design della rete neurale in quanto influenzano la scelta e la rappresentazione dei parametri di input e output che costituisco un punto cruciale nella creazione della rete.

Per modellare la rete neurale sono state scelte sette diverse regole considerate rilevanti per lo studio dell'espressività in una performance col piano (Figura 2.4).

Rule Name	Input parameters	Output parameters
High loud	N	ΔL
Leap articulation	DR, ΔN, LA	ΔDRO
Leap tone duration	ΔN, LP	ΔDR
Articulation of repetition	AR	ΔDRO
Durational contrast	DR	ΔDR, ΔL
Melodic charge	Cmel	ΔDR, ΔL
Phrase	P	ΔDR, ΔDRO

Figura 2.4: Regole utilizzate per l'allenamento della rete neurale. Parametri in input: DR=Onset-Relase duration, C_{mel}=Melodic charge, L=livello di intensità, N=pitch, LP=presenza di un cambiamento fra la nota corrente e quella successiva, ΔN=numero di semitoni del salto, P=presenza dei limiti della frase, AR=indica se la nota corrente è una ripetizione della precedente e corrisponde alla regola *Articulation of Repetition*, LA=indica se il cambiamento fra la nota corrente e la successiva corrisponde al caso descritto dalla regola *Leap Articulation.* Parametri in output: ΔDR=variazione della durata nominale, ΔDRO=variazione della durata di off time, ΔL=variazione dell'intensità nominale.

I parametri utilizzati in queste regole sono stati codificati e assegnati a diversi nodi di input della rete, come raffigurato in Figura 2.5. Infine la rete è stata allenata perché apprendesse le sette regole menzionate prima. Durante la fase di allenamento, i nodi di output sono stati allenati con le deviazioni di tempo e di intensità prodotte dalle

sette regole. Qui si manifesta la relazione fra il modello basato sulle reti neurali e il sistema basato su regole sviluppato al KTH.

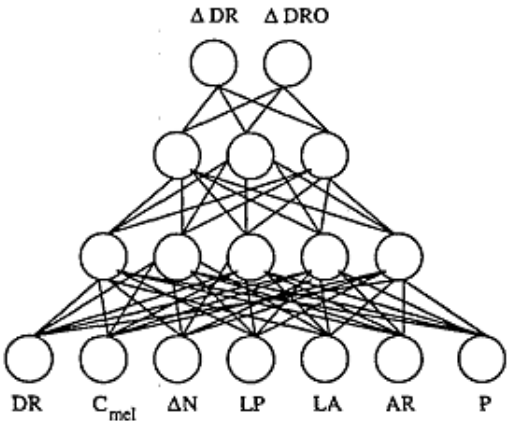


Figura 2.5: Rete neurale per le deviazioni di tempo. Nodi di input: DR=Onset-Relase duration, C_{mel}=Melodic charge, N=pitch, LP=presenza di un cambiamento fra la nota corrente e quella successiva, ΔN=numero di semitoni del salto, P=presenza dei limiti della frase, AR=indica se la nota corrente è una ripetizione della precedente e corrisponde alla regola *Articulation of Repetition*, LA=indica se il cambiamento fra la nota corrente e la successiva corrisponde al caso descritto dalla regola *Leap Articulation.* Nodi di output: ΔDR=variazione della durata nominale, ΔDRO=variazione della durata di off time (Bresin, 1998).

Il passo successivo è stato quello di costruire una rete più complessa in grado di apprendere lo stile di un pianista professionista. Durante la fase di training, i nodi di output sono stati allenati con le deviazioni di tempo e di intensità di una performance espressiva prodotta da un pianista professionista tramite un sintetizzatore connesso ad un personal computer. Sono state testate differenti architetture di rete in vari esperimenti che hanno portato alla definizione di due modelli di reti neurali complesse: *ecological ANN* e *ecological-predictive ANN*. Per ogni nota della partitura, il primo modello ha prodotto variazioni di intensità mentre il seconda ha generato deviazioni nella durata e nell' *Inter-onset-interval* (IOI).

Da notare che *DM* include un sottoinsieme di N regole, relativamente molto semplici che non richiedono nessun particolare trattamento della partitura. Pertanto, si è ipotizzato che le deviazioni prodotte da una rete neurale (scelta in base alle regole di decisione e al tipo di interazione dell'utente) potessero essere combinate con le deviazioni prodotte da queste N regole.

Il sistema ibrido risultante da questa combinazione può essere formalizzato dalla seguente espressione:

$$Y_n = \sum_{i=1}^{N} k_i \cdot f_i(\bar{x}_n) + net(\bar{k}, \bar{x}'_n) \quad (1)$$

Il primo termine dell'equazione prende in considerazione le regole *DM* incluse nel sistema, N è il numero delle regole, $\bar{x}_n$ rappresenta il vettore dei parametri delle regole associati con l'ennesima nota, la funzione $f_i()$ identifica le varie regole e k_i è una costante utilizzata per enfatizzare la deviazione generata da ogni regola.

Il secondo termine dell'equazione, *net(),* rappresenta l'insieme delle possibile reti neurali applicabili al sistema. Il vettore $\bar{k}$ corrisponde alla selezione di una particolare rete neurale o $f_i()$ e , $\bar{x}'_n$ è un vettore rappresentante i pattern di input alla rete per l'ennesima nota.

Una limitazione al modello per la performance musicale basato sulle reti neurali consiste nella difficoltà di scegliere la struttura della rete e l'allenamento della rete stessa, cioè nella scelta della codifica dei pattern di input e output. Una critica comune a questo modello consiste nella difficoltà d'interpretazione del comportamento di una rete neurale. Tuttavia un'attenta analisi della deviazione prodotta tramite l'utilizzo di una rete neurale può aiutare nell'identificazione di importanti metodi di set-up per la creazione di regole simboliche e può quindi fornire una spiegazione deterministica degli intenti espressivi consci e subconsci del musicista.

2.3 SaxEx

SaxEx (Arcos et al., 1998) è un sistema automatico per generare performance espressive sviluppato sul modello CBR. Il modello CBR risulta estremamente efficace nello sviluppo di questo sistema in quanto i numerosi esempi necessari per la sua applicazione, sono facilmente reperibili tramite registrazioni di performance umane. SaxEx, nel dettaglio:

- È implementato in Noos, un framework integrato per il *problem solving* e l'apprendimento.

- Utilizza la tecnica SMS (*Spectral Model Synthesis*) per estrarre le informazioni di base relative ai diversi parametri espressivi come dinamiche, rubato e vibrato consentendo al sistema la generazione di nuove interpretazioni espressive (nuovi file audio).
- Il modello computazionale su cui si basa SaxEx è sviluppato a partire dal *Narmour's implication/realization model* (Narmour, 1990) e dal *Lerdahl and Jackendoff's generative theory of tonal music* (Lerdahl et al., 1993).

2.3.1 SMS

Lo *Spectral Model Synthesis* comprende un insieme di tecniche per l'analisi, la trasformazione e la sintesi di suoni. L'obiettivo è quello di avere a disposizione una rappresentazione del suono (basata sull'analisi spettrale) generale e musicalmente significativa attraverso cui sia possibile manipolare i parametri musicali e allo stesso tempo mantenere inalterata l'identità percettiva del suono originale quando non siano state effettuate trasformazioni. Tramite questa tecnica, basata sull'analisi dello spettro, è possibile decomporre il suono in sinusoidi più un residuo spettrale da cui si possono estrarre parametri come il tempo di attacco e rilascio, l'ampiezza o il pitch. Una volta estratti i parametri necessari, si possono modificare e reinserire nella rappresentazione spettrale senza perdita nella qualità del suono.

2.3.2 Noos

Noos è un linguaggio multipiattaforma utilizzato per modellare la conoscenza nel problem solving e nei problemi di apprendimento implementato in Lisp. Per specificare un problema utilizzando Noos è necessario suddividere la conoscenza in tre tipi: *domain knowledge*, *problem solving knowledge* e *metalevel knowledge*. *Domain knowledge* specifica un insieme di concetti e relazioni fra concetti, rilevanti per l'applicazione. Nel caso di SaxEx questi concetti consistono, ad esempio, nelle note, negli accordi e nei parametri espressivi. *Problem solving knowledge*, comprende un insieme di compiti che devono essere risolti in un'applicazione. Il compito principale di SaxEx, è quello di derivare una serie di trasformazioni espressive per un data sequenza musicale. Per un dato compito esistono diversi metodi per risolverlo; questi metodi definiscono in che modo scomporre il problema

da risolvere in sottoproblemi e come combinare i risultati ottenuti per il raggiungimento della soluzione finale. Infine, *Metalevel knowledge* raggruppa la conoscenza relativa ai precedenti due tipi: sostanzialmente fornisce dei criteri per la scelta di un determinato metodo per la soluzione di un compito. Una volta che un problema è stato risolto, questo viene memorizzato e indicizzato, formando un insieme di problemi risolti chiamato *Episodic Memory* di Noos. Queste soluzioni sono accessibili e recuperabili e costituiscono la base per l'applicazione del modello CBR.

2.3.3 Il sistema SaxEx

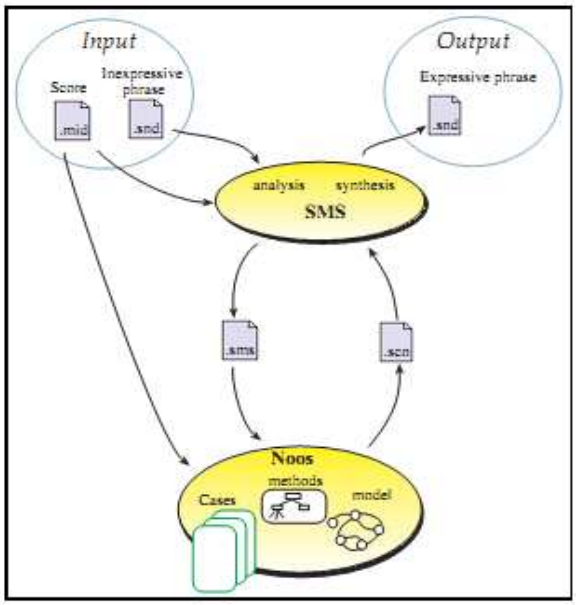


Figura 2.6: SaxEx: decomposizione in blocchi (Arcos et al., 1998).

L'input consiste nella partitura musicale (file Midi) e un file audio. La partitura contiene informazioni relative alla melodia e alle armoniche della sequenza musicale, mentre il file audio contiene la registrazione di un'interpretazione inespressiva della partitura, eseguita da un musicista. L'output del sistema è un nuovo file audio ottenuto dalla trasformazione del file originale e contenente una performance espressiva della partitura. La soluzione di un problema in SaxEx si suddivide in tre fasi: analisi, ragionamento, sintesi (Figura 2.6). L'analisi e la sintesi sono implementate utilizzando le tecniche definite nel modello SMS mentre la fase di ragionamento è realizzata secondo il modello CBR (*Case-Based Reasoning*) e implementata con Noos.

2.3.4 SaxEx Task

Data una performance musicale, SaxEx determina uno specifico set di trasformazioni espressive da applicare ad ogni nota della partitura utilizzando per ognuna di esse lo stesso metodo di *problem solving*. Per ogni nota si procede dunque alla decomposizione del problema in sotto attività (Figura 2.7) secondo le specifiche del modello CBR. In particolare, in SaxEx la decomposizione avviene nel seguente modo:

- Recupero: l'obiettivo è quello di scovare l'insieme delle note (casi) più simili alla nota corrente. Questa attività viene svolta procedendo all'esecuzione di tre sotto-attività:
  - Identificazione
  - Ricerca
  - Selezione
- Riutilizzo: in questa fase si determina l'insieme delle trasformazioni espressive da applicare al problema corrente scegliendo dall'insieme dei casi simili.
- Mantenimento: la nuova soluzione viene aggiunta all'*Episodic Memory* di Noos, contenente i problemi finora risolti.

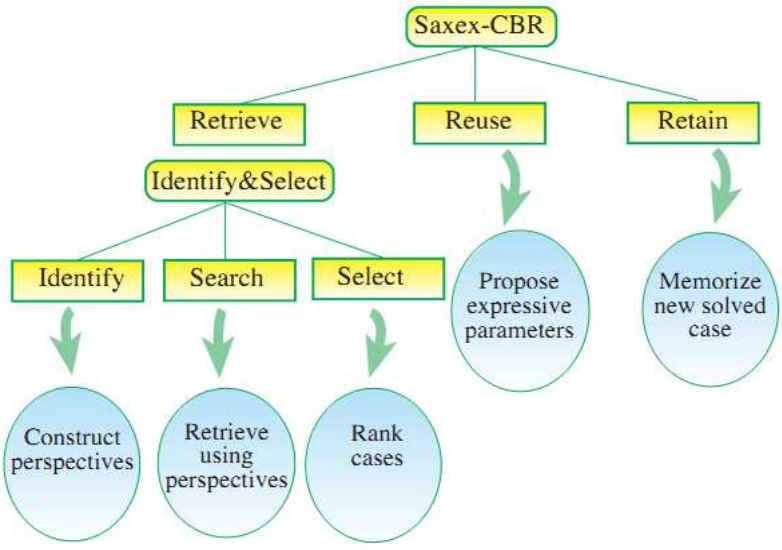


Figura 2.7: Decomposizione in task secondo le specifiche CBR (Arcos et al., 1998).

2.4 Air Worm

L'idea alla base di questo sistema è quella di trasformare l'ascolto passivo di una performance musicale in un coinvolgimento attivo con la musica attraverso un'interfaccia intuitiva e semplice da utilizzare. La maggior parte degli strumenti musicali offre un'interfaccia ricca di parametri (e quindi permette un elevato controllo del segnale musicale), ma questo livello di precisione richiede una specifica conoscenza dello strumento Inoltre, i parametri espressivi non sono facilmente esprimibili e quindi è difficile automatizzarne il settaggio. La soluzione proposta da Dixon (2005) si basa su una performance espressiva di un musicista esperto sulla quale è possibile intervenire, modificandone la struttura in un modo semplice e trasparente. L'attenzione è posta su tre aspetti: la scelta di un'interfaccia adatta per utenti non esperti, il mapping delle azioni dell'utente per la modifica della performance e gli aspetti implementativi per la modifica della performance. Il dispositivo di input del sistema è un theremin digitale che consente agli utenti il controllo di due parametri a seconda della posizione delle proprie mani rispetto a due antenne. Nella prima implementazione, chiamata Air Worm, il tempo e l'intensità sono rappresentati in uno spazio 2-D nel quale i movimenti dell'utente controllano i parametri espressivi della performance, consentendogli di specificare le complesse traiettorie espressive con un cenno della mano. La seconda implementazione del sistema, chiamata Air Tapper, utilizza un paradigma di conduzione più standard dove il tempo è definito dagli intervalli di battuta attraverso movimenti verticali della mano. Come alternativa al theremin è stata sviluppata un ulteriore versione di entrambi i sistemi che utilizza un mouse come input (Mouse Worm e Mouse Tapper).

2.4.1 Midi theremin

Il theremin (Figura 2.8) è uno strumento musicale sviluppato nei primi anni del Novecento da Leon Theremin. Nella sua forma più semplice, consiste in due oscillatori ad alta frequenza, uno impostato su una frequenza fissa mentre l'altro a frequenza variabile. L'oscillatore a frequenza variabile è controllato dal movimento della mano da e verso un'antenna verticale che modifica la frequenza di oscillazione. Un'antenna orizzontale, è utilizzata allo stesso modo di quella verticale per

controllare l'intensità del segnale. A differenza degli strumenti tradizionali, il musicista non ha contatti fisici con lo strumento. Ciò permette di fornire all'esecutore un elevato grado di libertà, ma a prezzo di non avere nessun feedback aptico. Il theremin fornisce un continuo controllo sul pitch e l'intensità in netto contrasto con la tastiera che ha invece un insieme fisso di pitch discreti e, solitamente, nessun controllo di intensità dopo l'attacco iniziale. Di contro, il protocollo MIDI è fortemente basato sull'idioma della tastiera in cui un tono musicale è rappresentato da un note-on e un note-off, dove l'intensità è determinata dal parametro velocità del messaggio note-on. Il MIDI theremin è una via di mezzo nel quale vengono combinate rappresentazioni continue e discrete del suono. In questo strumento i comandi note-on e note-off (tipici del protocollo midi) non vengono utilizzati; si utilizza invece un *pitch bend message* (con risoluzione di 14 bit) per rappresentare una variazione del pitch e un *controller change massage* (con risoluzione di 7 bit) per esprimere una variazione di intensità.

Figura 2.8: Theremin.

2.4.2 Il sistema Air Worm

Air Worm fornisce una visualizzazione dei due più importanti parametri di una performance espressiva, ovvero tempo e dinamiche, in un semplice spazio 2-D (Figura 2.9) dove l'evoluzione nel tempo dei due parametri è visibile come una traiettoria all'interno di questo spazio bidimensionale. Il sistema consente un controllo in tempo reale della traiettoria attraverso l'uso del MIDI theremin. Questo fornisce una stima lineare della posizione della mano che poi viene scalata in modo da poter essere interpretata come valori di tempo e dinamiche.

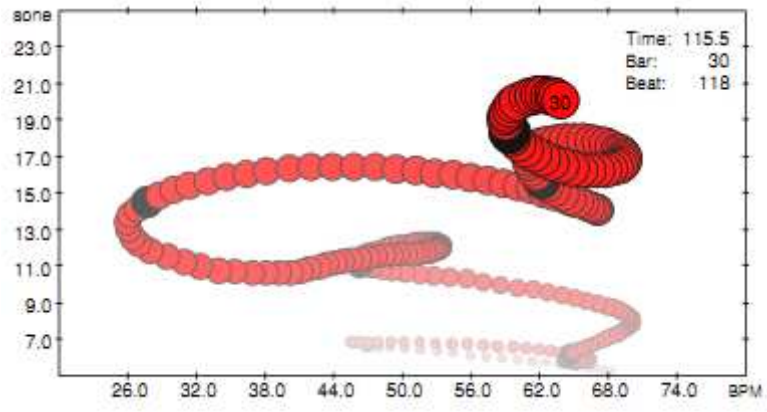


Figura 2.9: Performance Worm. L'asse orizzontale rappresenta il tempo corrente, l'asse verticale il livello di intensità Dixon et al. (2005).

La traiettoria disegnata con le mani viene mostrata sullo schermo di un computer dando la possibilità all'utente di avere un feedback visivo, elemento molto importante considerando che nel sistema non è presente un feedback tattile. Se la performance musicale è in formato MIDI, la modifica del tempo viene effettuata di un fattore F che scala tutti gli intervalli. Il valore di 14 bit in input (*pitch bend message*) viene mappato come una curva esponenziale secondo la formula:

$$F_{out} = k^{\frac{F_{in}}{8192}} \tag{2}$$

dove k è il valore massimo del fattore tempo. Ad esempio se k = 2, il tempo può variare fra la metà e il doppio del tempo originale.

Il valore a 7 bit (da 0 a 127) utilizzato come valore di input per il volume è scalato linearmente ed è utilizzato come volume principale:

$$V_{out} = \frac{V_{in}}{127} \tag{3}$$

Per i dati audio in input, esistono diversi metodi per la modifica della scala del tempo senza alterare il pitch o il timbro del suono. In questo caso viene utilizzato un metodo nel dominio del tempo che riduce le discontinuità di ampiezza e fase.

2.4.3 Mouse-worm

Un'estensione del sistema precedente utilizza come interfaccia un semplice mouse anziché il theremin. Il vantaggio principale di questo approccio è che non è richiesto

nessun particolare tipo di hardware per manipolare la performance. Inoltre, è possibile avere un controllo molto più accurato sui parametri, data la maggior precisione del mouse.

2.4.4 Air tapper

Un direttore d'orchestra comunica le proprie istruzioni ai musicisti attraverso movimenti di braccia e mani; questo è un naturale metodo di espressione in quanto non impone nessun tipo di costrizione al conduttore. Le informazioni relative al tempo sono comunicate attraverso la traiettoria della bacchetta e il tempo di battuta è dato da precisi punti di questa traiettoria. Il controllo dell'intensità non è esplicito, ma generalmente è correlato con un'estensione della traiettoria. Ovviamente questa è un'estrema semplificazione di un complicato protocollo di comunicazione, tuttavia è necessaria per descrivere un possibile protocollo utilizzando il theremin.

In particolare, se si volesse utilizzare un theremin come strumento di conduzione, si dovrebbe codificare le informazioni relative al tempo dai movimenti, trovare il corrispondente tempo di battuta nella musica e sincronizzare la riproduzione della musica con i gesti in tempo reale. Il tempo di battuta è estratto tracciando la distanza della mano dell'utente dall'antenna orizzontale e calcolando il minimo locale. Il tempo di ogni minimo viene considerato come il tempo di battuta e il tempo complessivo viene calcolato dagli intervalli di battuta. Le dinamiche sono controllate valutando la distanza della mano dall'antenna verticale. Poiché questa distanza non rimane costante durante la traiettoria, viene considerata la distanza media e, questa viene aggiornata per ogni battuta. Un metodo più semplice per la gestione delle dinamiche consiste nel condurre con una mano il tempo di battuta, posizionandosi all'estremità più lontana dell'antenna orizzontale, e utilizzando l'altra mano per le dinamiche. Per sincronizzare la musica alla conduzione è necessario conoscere il tempo di battuta del file musicale. Si è supposto che il tempo sia fornito come metadata in quanto deve essere allineato con il tempo calcolato dal theremin.

2.5 YQX

Il sistema proposto da Widmer (2011) per l'esecuzione espressiva di un brano utilizza un approccio modulare per trattare dinamica, articolazione e tempo globale.

In Figura 2.10 è riportato lo schema a blocchi dell'intero sistema. Da un insieme di esecuzioni musicali (*training data*) vengono estratte le caratteristiche della partitura (*score features*) e i parametri intensità, IOI e articolazione (*targets*) che vengono utilizzati per allenare le diverse componenti del sistema. Le *score features,* e le annotazioni di tempo e di dinamica vengono estratti a partire da un file *MusicXML*. Le prime vengono utilizzate per calcolare le predizioni di tempo e articolazione mentre le informazioni relative alla dinamica sono utilizzate per calcolare l'intensità della performance.

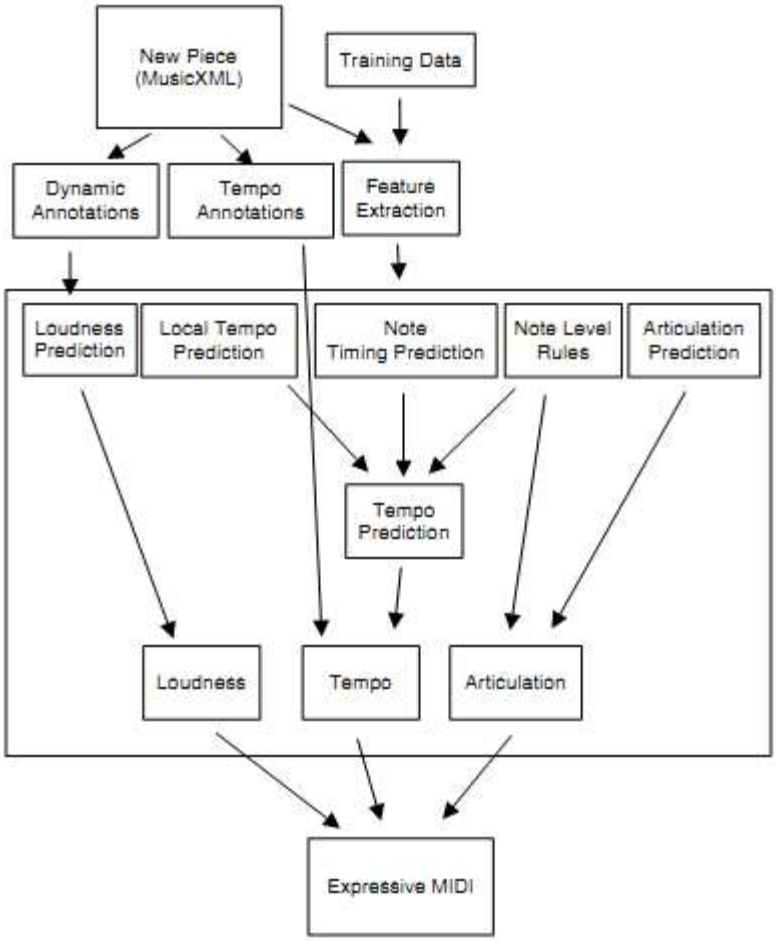


Figura 2.10: Schema a blocchi del sistema YQX (Widmer et al., 2011).

2.5.1 Rendering espressivo: Previsione del tempo e dell'articolazione

Il tempo e l'articolazione sono calcolate da una rete Bayesiana che elabora le relazioni fra la partitura e la performance come una distribuzione di probabilità condizionale. Le informazioni relative alla partitura (*score features*) comprendono semplici descrittori (ritmico, melodico e armonico) e caratteristiche di alto livello dal modello *Implication-Realization* (I-R) di Narmour (1990).

La predizione del tempo è costituita da tre componenti:

- *local tempo,* una predizione a lungo termine della variazione del tempo di una nota;
- *note timing,* deviazione dal tempo locale cola colato nota per nota;
- *global tempo,* tempo globale estratto dalla partitura (ad esempio, *andante*).

2.5.2 Note Level Rules

Le regole espressive utilizzate nel sistema sono:

- *Staccato rule:* in presenza di due note successive con lo stesso pitch, se la seconda è più lunga, la prima è suonata staccata.
- *Delay-next rule:* se due note della stessa lunghezza sono seguite da una nota più lunga, l'ultima è suonata con un leggero ritardo

I musicisti professionisti tendono a enfatizzare la melodia suonando le note melodiche leggermente in anticipo sul tempo, un fenomeno chiamato *melody leap.* Per simulare questo comportamento è stato applicato un anticipo di 13 ms a tutte le note della melodia.

2.5.3 Rendering espressivo: Previsione dell'intensità

L'algoritmo utilizzato per elaborare l'intensità è basato sulla regressione lineare. La curva d'intensità è composta associando delle funzioni di base rapportate a quanto appreso da performance musicali. Queste funzioni rappresentano caratteristiche musicali in relazione alle note nella partitura, permettendo predizioni separate per

ogni nota anziché una sola predizione in base alla posizione temporale. Il beneficio principale di questa scelta è la possibilità di predire differenti valori di intensità anche per note simultanee.

Le funzioni di base sono utilizzate per rappresentare le seguenti caratteristiche della partitura:

- Annotazioni sulla dinamica (*ff, crescendo* ecc.);
- Proprietà della nota: pitch, ruolo decorativo, enfasi;
- *Implication-Realization closure*.

2.5.4 La rete Bayesiana

La rete Bayesiana è un semplice modello condizionale Gaussiano in cui le *score features* sono suddivise in due insiemi: continue (X) e discrete (Q). Le continue sono modellate come distribuzioni gaussiane $p(x_i)$ mentre le discrete sono descritte da semplici distribuzioni di probabilità $P(q_i)$. La dipendenza fra le variabili *target* Y e le *score features* X e Q è modellata come una distribuzione condizionale $p(y_i|Q,X)$.

Il sistema è allenato stimando separatamente, per ogni variabile *target*, la distribuzione multinomiale che rappresenta la probabilità congiunta $p(yi,X)$. La dipendenza con le variabili *target* Q è modellata calcolando un singolo modello per ogni possibile combinazione di valori delle variabili discrete.

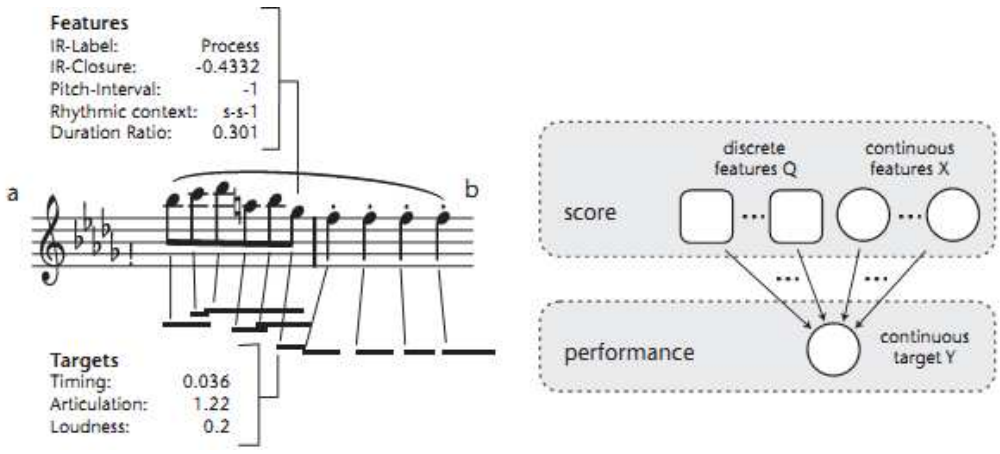


Figura 2.11: Esempi di Feature e Target estratti per una nota (a) e struttura del modello Bayesiano (b) (Widmer et al., 2009).

In Figura 2.11 viene rappresentato un esempio di *score features* e *target* calcolati a partire da una nota (a) e la struttura del modello Bayesiano (b).

2.5.5 Training data

Il sistema è allenato a partire da due insiemi di performance reali:

- 13 sonate di Mozart complete eseguite da R. Batik
- Il lavoro completo per piano di Chopin eseguito da N. Magaloff

Tutti i brani sono stati suonati su un *Bösendorfer grand piano* controllato via computer, per un totale di 400.000 note riprodotte.

L'allenamento di tutte le componenti della rete e la creazione dell'esecuzione espressiva è fatto in maniera autonoma dal sistema, senza nessun feedback umano. L'intero processo di rendering espressivo impiega un paio di minuti per essere portato a termine.

2.6 Modello genetico basato su regole per performance espressive con il sassofono jazz.

Generalmente, lo studio dell'espressività in ambito musicale utilizza approcci empirici basati su analisi statistiche, modelli matematici e analisi per sintesi. In tutti questi approcci la parte umana è fondamentale in quanto è responsabile dell'ideazione della teoria o del modello matematico utilizzato per catturare i diversi aspetti espressivi della performance musicale. Il modello sviluppato è quindi testato su performance reali per determinarne l'accuratezza.

L'approccio utilizzato in questo modello invece, utilizza un sistema computazionale innovativo (Ramirez et al., 2008). Invece di modellare manualmente la performance espressiva e poi testare il modello su dati reali, affida al computer il compito di analizzare automaticamente l'espressività attraverso l'esecuzione di un *sequential covering genetic algorithm* su performance reali composte da registrazioni audio jazz. Questo algoritmo combina *sequential covering* (Michalski 1969) e algoritmi genetici (Holland 1975). La componente *sequential covering* dell'algoritmo costruisce

incrementalmente un insieme di regole, rimuovendo gli esempi positivi coperti dall'ultima regola prima di tentare di imparare la regola successiva**.** La parte genetica dell'algoritmo invece, apprende nuove regole applicando un algoritmo genetico.

2.6.1 Algoritmo di estrazione delle informazioni

Inizialmente, viene eseguita un'analisi di una porzione di suono (*analysis frame*), la cui dimensione è un parametro dell'algoritmo. Questa analisi spettrale consiste nella moltiplicazione del frame audio con un'appropriata finestra di analisi e, successivamente, nel calcolo della trasformata discreta di Fourier (DFT) in modo da ottenere lo spettro di frequenza. Nell'algoritmo viene utilizzato un frame di larghezza pari a 46 ms, un fattore di sovrapposizione del 50% e una finestra Kaiser-Bessel di 25dB.

2.6.2 Calcolo dei descrittori di basso livello

I principali descrittori di basso livello utilizzati per caratterizzare una performance espressiva sono l'energia istantanea e la frequenza fondamentale. Il descrittore dell'energia è calcolato nel dominio della frequenza, utilizzando i valori spettrali di ampiezza di ogni *analysis frame*. Inoltre, l'energia è determinata a diverse bande di frequenza e questi valori sono utilizzati dall'algoritmo per la segmentazione in note.

Per la stima della frequenza fondamentale, viene utilizzato un modello di harmonic-matching, derivato dalla procedura *Two-Way Mismatch* (TWM) (Maher et al., 1994). Viene utilizzato uno schema pesato per rendere la procedura robusta alla presenza di rumore o all'assenza di certe parziali nello spettro.

Una volta analizzati tutti gli spettri dei frame audio, vengono individuati i principali picchi spettrali. Questi picchi sono definiti come il massimo locale nello spettro, la cui ampiezza è maggiore di una certa soglia. I picchi vengono confrontati con una serie di armoniche e viene calcolato il *TWM error* per ogni frequenza fondamentale candidata. La candidata con l'errore minimo è scelta come frequenza fondamentale.

2.6.3 Segmentazione in note

L'energia degli onset è inizialmente elaborata seguendo un algoritmo band-wise che utilizza una conoscenza psicoacustica (Klapuri, 1999). Successivamente, vengono calcolate le frequenze fondamentali di transizione. Questi risultati vengono combinati per ottenere le informazioni relative alla nota (informazione di onset e offset).

2.6.4 Elaborazione del descrittore della nota

Il descrittore della nota è ottenuto utilizzando i descrittori di basso livello e le informazioni sulle note ottenuti attraverso i processi descritti in precedenza. I descrittori di basso livello associati al segmento di nota sono ottenuti calcolando la media dei valori del frame all'interno del segmento. Sono utilizzati istogrammi di pitch per elaborare sia il pitch della nota sia la frequenza fondamentale che rappresenta questa nota.

2.6.5 Training data

Il training set utilizzato nel modello è composto da registrazioni monofoniche di quattro brani jazz (*Body and Soul, Once I Loved, Like Someone in Love e Up Jumped Spring*), eseguite da un musicista professionista con undici tempi diversi rispetto al tempo nominale. Per ogni brano, il tempo nominale è stato determinato dal musicista come il tempo più naturale e confortevole per eseguire il brano. Inoltre, il musicista ha identificato il tempo più veloce e più lento con cui il brano era ragionevolmente riproducibile. Le performance sono state registrate ad intervalli regolari vicini al tempo nominale (cinque più veloci e cinque più lente), senza includere il tempo massimo e il tempo minimo. Il data set è composto di 4360 note, ognuna delle quali è stata aggiunta al training set con le caratteristiche di esecuzione (durata, onset ed energia) e con un certo numero di attributi di partitura che descrivono la nota e alcuni aspetti del contesto in cui appare. Le informazioni relative alla nota includono la durata, la posizione metrica all'interno della battuta ed informazioni circa il contesto melodico che includono il tempo di esecuzione, informazioni sulle note vicine e la struttura Narmour (1990) nella quale si trova la nota.

2.6.6 Learning Task

Per lo sviluppo di questo modello, gli autori si sono concentrati sulle trasformazioni espressive a livello di nota, in particolare trasformazioni sulla durata della nota, onset ed energia. Inizialmente, per ogni trasformazione espressiva, si è affrontata la questione come un problema di classificazione: per la trasformazione della durata della nota, ad esempio, ogni nota è stata classificata come *lengthen, shorten* o *same.* Una volta elaborato un meccanismo di classificazione in grado di classificare tutte le note del training data, è stato applicato un algoritmo di regressione per produrre un valore numerico che rappresentasse il totale delle trasformazioni da applicare ad una determinata nota. L'algoritmo completo è descritto nel paragrafo 2.6.7.

Le classi di performance di interesse sono *lengthen, shorten* e *same* per la durata, *advance, delay* e *same* per le deviazioni dell'onset, *soft, loud* e *same* per l'energia, *ornamentation* e *none* per l'alterazione della nota. Una nota si considera appartenente alla classe *lengthen* se la sua durata di esecuzione è il 20% più lunga rispetto alla sua durata nominale. Altrimenti appartiene alla classe *shorthen*. Alla classe *advance* appartengo le note il cui onset d'esecuzione è il 5% più veloce in confronto con l'onset nominale. La classe *delay* è definita in maniera analoga. Infine, appartengono alla classe *loud* le note suonate più forte rispetto al suo predecessore e più forte rispetto il livello medio del pezzo. Analogamente è definita *soft.* Una nota (o un gruppo di note), è considerata appartenente alla classe *ornamentation* se queste note non esistono nella partitura originale, ma sono state aggiunte dal musicista per abbellire le note della melodia. Altrimenti appartengono alla classe *none.*

2.6.7 Algoritmo

È stato applicato al training data un algoritmo genetico a copertura sequenziale. Come già accennato in precedenza, l'algoritmo costruisce un insieme di regole, apprendendone di nuove e rimuovendo gli esempi positivi prima di tentare di apprendere una nuova regola. Viene costruito un insieme gerarchico di regole che vengono applicate nell'ordine di generazione in modo che almeno una regola venga sempre applicata.

Per ogni classe di interesse (ad esempio *lengthen, shorten* e *same*), vengono collezionate regole con il miglior fitness. Le regole per una particolare classe di interesse (*lengthen,* ad esempio) si ottengono considerando come esempi negativi gli esempi delle due classi complementari (*shorten* e *same*).

Una volta ottenuto l'insieme di regole che copre l'intero training data, a tutti gli esempi coperti da una determinata regola è applicata una regressione lineare, in modo da ottenere un'equazione lineare che predice un valore numerico. Ciò consente di avere un insieme di regole che genera delle previsioni numeriche e non solo una previsione nominale di classe. Nel caso di alterazioni di nota, non viene calcolato nessun valore numerico, ma viene semplicemente conservato l'insieme degli esempi coperti da quella regola.

Infine, viene applicato l'algoritmo *k-nearest-neighbor* per selezionare uno degli esempi coperti dalla regola e adattarlo al nuovo contesto melodico.

Questo algoritmo fornisce un modello generativo per performance musicali espressive in grado di generare esecuzioni automatiche con metrica ed energia espressiva simili a quelle che caratterizzano le performance musicali umane.

2.7 Kagurame System

Kagurame Phase-II e Kagurame Phase-III (Taizan et al., 2011) sono due sistemi per il rendering espressivo di una performance musicale sviluppati seguendo il paradigma di sviluppo *case-based reasoning* (CBR) descritto nel paragrafo 1.2.4.

Secondo questo modello, per acquisire la conoscenza espressiva presente in un'esecuzione musicale si utilizzano delle performance di esempio anziché regole espressive o algoritmi di apprendimento automatico. Sostanzialmente si recuperano frammenti musicali dalle esecuzioni di esempio, si identificano i pattern espressivi e si applicano quest'ultimi al brano in esame.

2.7.1 Approccio case-based per il rendering espressivo

Come detto precedentemente, questo paradigma utilizza un insieme di performance di esempio per estrapolare la conoscenza sull'espressività musicale. L'insieme

(chiamato *Performance Data Set*) è costituito da una collezione di esecuzioni umane ognuna delle quali contiene:

- espressività musicale;
- partitura;
- condizioni per l'esecuzione.

In ingresso (Input) sono richiesti:

- partitura del brano in esame;
- condizioni per l'esecuzione.

In Figura 2.12 viene riportato lo schema a blocchi dell'architettura *case-based* utilizzata.

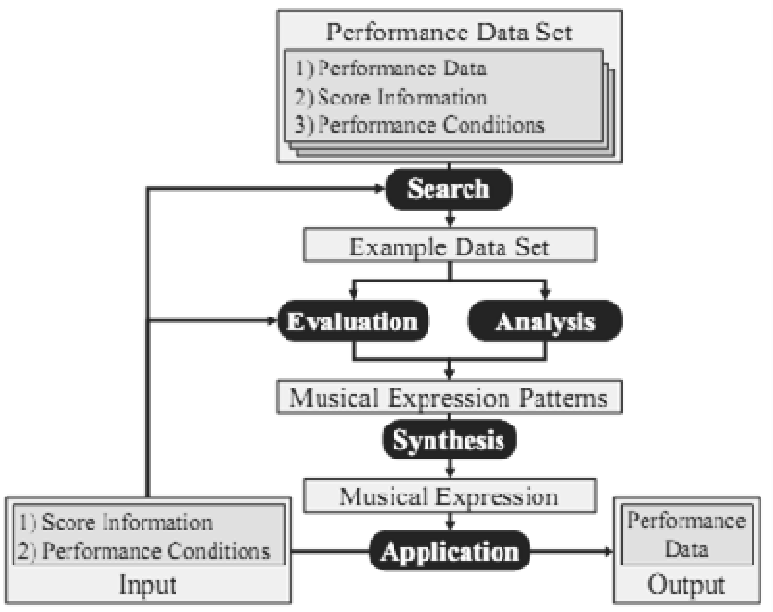


Figura 2.12: Architettura dell'approccio *case-based* (Taizan et al., 2011).

Una volta specificati i dati in input, viene applicata la seguente procedura per generare un'esecuzione espressiva utilizzando le performance di esempio:

- Ricerca: il brano viene suddiviso in una gerarchia di frammenti di varia lunghezza utilizzando le informazioni contenute nella partitura (note, sezioni, ecc.). Successivamente, il sistema ricerca fra le esecuzioni di esempio coppie di frammenti simili.

- Valutazione: il sistema valuta il grado di somiglianza delle coppie di frammenti selezionate nella fase precedente. Questa verifica è effettuata non solo confrontando la partitura, ma anche verificando se le condizioni di esecuzione sono compatibili. Quindi, anche se il brano in esame fosse identico ad uno contenuto nell'insieme degli esempi, le condizioni d'esecuzione discriminerebbero i due frammenti.
- Analisi: vengono estratti i pattern espressivi dai frammenti selezionati.
- Sintesi: le fasi di ricerca, valutazione e analisi vengono applicate a tutti frammenti del brano in esame. Il sistema accumula tutti i pattern espressivi estratti e sintetizza un unico pattern espressivo per il brano.
- Applicazione: il pattern espressivo elaborato viene applicato al brano in esame per generare un'esecuzione espressiva.

2.7.2 Il sistema Kagurame

Kagurame è il nome del sistema automatico per l'esecuzione espressiva di un brano che implementa l'architettura case-based descritta nel paragrafo precedente. Kagurame Phase-I è stata la prima versione del sistema che elaborava solamente brani monofonici-

Kagurame Phase-II è l'evoluzione della prima versione ed esegue il rendering espressivo anche di brani polifonici. L'architettura di base è la stessa con l'aggiunta di alcuni parametri espressivi relativi alla simultaneità.

Kagurame Phase-III invece è una variante della versione II che utilizza un algoritmo di similarità diverso per confrontare i frammenti musicali.

2.7.2.1 Valutazione della similarità

Il processo di valutazione della similarità è una fase molto importante del sistema Kagurame. Viene utilizzato nella fase di ricerca per estrapolare i frammenti musicali dalle performance di esempio e nella fase di valutazione per confrontare le coppie di frammenti selezionate. La valutazione della similarità è cruciale per la generazione di un'esecuzione espressiva, dato che il sistema estrae i pattern espressivi dai dati selezionati nella fase di ricerca ed esaminati nella fase di valutazione,

Kagurame Phase-II valuta la similarità utilizzando un vettore di caratteristiche che vengono estratte dalle partiture. Le caratteristiche estratte riguardano il ritmo, l'armonia e la melodia. Tuttavia, le dimensioni del vettore sono troppo ridotte e quindi alcune importanti caratteristiche non vengono estratte e questo limite si ripercuote inevitabilmente sul processo di valutazione della similarità. Per superare questo limite, Kagurame Phase-III introduce dati sottoforma di immagini. Dalla partitura vengono generate delle immagini simili ad un *piano roll* [4]che vengono confrontate utilizzando un algoritmo di confronto fra immagini. L'immagine preserva molte delle informazioni contenute nella partitura e quindi il processo di valutazione della similarità è molto più dettagliato.

2.7.2.2 Performance rendering

Nel sistema Kagurame Phase-II vengono utilizzati i seguenti brani per formare l'insieme degli esempi:

- Mozart:
    - *Piano Sonata K.545 1st Mov.*
    - *Piano Sonata K.545 2nd Mov.*
    - *Piano Sonata K.331 1st Mov.*
    - *Piano Sonata K.331 2nd Mov*.
- Chopin:
    - *Etude No. 3*
    - *Nocturne No. 1, Op. 32-2*
    - *Prelude Op. 28 No. 4*
    - *Prelude Op. 28 No. 7*
    - *Prelude Op. 28 No. 15*
    - *Prelude Op. 28 No. 20*
- Beethoven:
    - *Beethoven Piano Sonata No. 8, Op. 13*

[4] *Piano roll*: supporto di memorizzazione per pianoforte costituito da un rotolo di carta con delle perforazioni. La posizione e la lunghezza della perforazione determina la nota suonata al pianoforte.

Per Kagurame Phase-III invece viene utilizzato un insieme ridotto di esempi in quanto la versione corrente del sistema richiede un ammontare considerevole di memoria e CPU.

- Mozart:
    - *Piano Sonata K.545 2nd Mov.*
- Chopin:
    - *Prelude Op. 28 No. 20*

2.8 VirtualPhilarmony

VirtualPhilarmony (V.P.) è un sistema di conduzione che trasmette la sensazione di condurre un'orchestra (Takashi et al., 2011). Ci sono due problemi nel simulare questa interazione: il primo consiste nel progettare uno scheduler predittivo adeguato, il secondo è quello di creare un modello di performance dinamico che descriva l'espressività. Per risolvere questi problemi, sono state analizzate diverse esecuzioni reali e sono stati consultati diversi direttori d'orchestra per impostare in maniera corretta i parametri del sistema. La funzione che simula il processo di interazione fra il conduttore è l'orchestra è stata chiamata *Concertmaster Function.*

2.8.1 Il sistema VirtualPhilarmony

In Figura 2.13 è riportato lo schema del sistema. VirtualPhilarmony estrae il minimo e il massimo locale utilizzando un sensore che analizza il movimento del conduttore e, attraverso i dati raccolti, identifica il tempo di battuta, volume e *fermata*. La funzione *Concertmaster* predice il tempo della battuta successiva, modifica dinamicamente il modello della performance espressiva ottenuto dalle esecuzioni reali e schedula il modello usando il tempo definito dall'utilizzatore del sistema. VirtualPhilarmony controlla la velocità di ogni nota schedulata in base ai volumi che vengono rilevati dal sensore. Infine il sistema riproduce la performance e visualizza la partitura.

Come sensore di misura è stato utilizzato un theremin (vedi paragrafo 2.4.1).

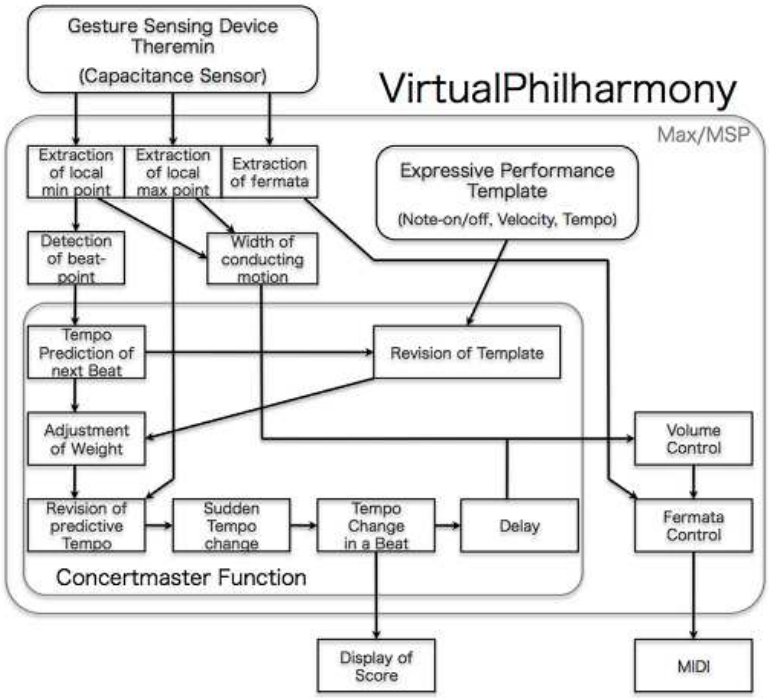


Figura 2.13: Schema a blocchi del sistema VirtualPhilarmony (Takashi et al., 2011).

La funzione *Concertmaster* è basata sull'euristica di conduzione di un'orchestra e consiste di sette sottofunzioni:

1. Predizione del tempo della battuta successiva;
2. Modifica del modello espressivo della performance;
3. Aggiustamento dei pesi;
4. Modifica della predizione del tempo utilizzando un punto di massimo locale nel movimento del conduttore;
5. Supporto per cambiamenti di tempo improvvisi;
6. Supporto per cambiamenti di tempo nella battuta;
7. Aggiustamento del ritardo fra il movimento di conduzione e la performance musicale.

Riguardo al primo punto, per ogni brano e battuta sono stati calcolati i parametri predittivi migliori utilizzando le registrazioni di esecuzioni reali. Al punto 2 il sistema modifica il modello con il tempo definito dal conduttore. Ad esempio, il rapporto di durata di tre battute nel valzer viennese nel modello ($Duration_W$) sono modificate in relazione al tempo definito dal conduttore secondo l'equazione (4). Oppure il rapporto di durata delle note puntate nel modello ($Duration_D$) sono modificate

secondo l'equazione (5). I coefficienti di queste equazioni sono ottenuti dall'analisi di performance reali.

$$Duration_W = a_W x_W + b_W \quad (4)$$

$$Duration_D = a_D x_D + b_D \quad (5)$$

Dove x_W e x_D rappresentano il tempo definito dal conduttore e *a* e *b* i coefficienti.

2.8.2 Performance Rendering

Il processo di rendering espressivo è composto dalle seguenti fasi (Figura 2.14):

1. Elaborazione manuale di una performance espressiva;
2. Conduzione con VirtualPhilarmony.

La performance espressiva è ottenuta con il processo illustrato nello Step 1 di Figura 2.14. L'espressività è aggiunta ad un brano in formato MIDI (SMF) utilizzando un'utility basata su regole che esplicitano i simboli espressivi e i rapporti di volume fra la mano destra e la mano sinistra. Il file prodotto viene poi processato dal sistema (Step 2 in Figura 2.14). Le modifiche di intonazione e volume vengono aggiunte al modello attraverso la conduzione effettuata dall'utilizzatore del sistema generando la performance espressiva desiderata.

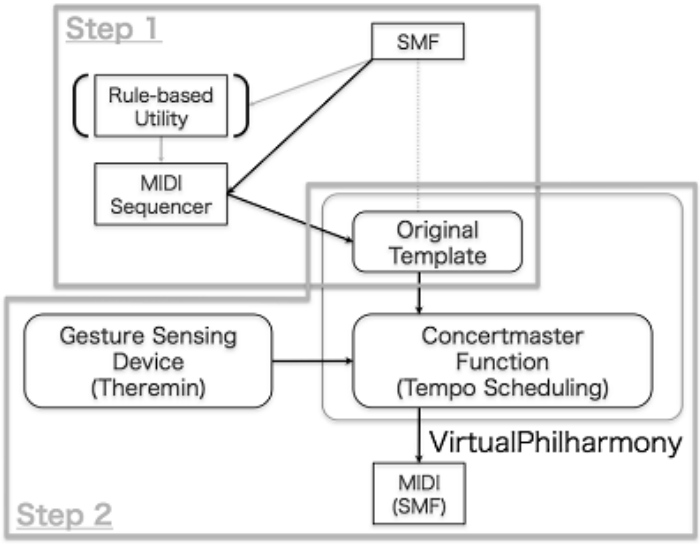


Figura 2.14: Processo di rendering (Takashi et al., 2011).

Capitolo 3

CaRo 2.0: Modellazione e controllo dell'espressività nell'esecuzione musicale

Un'interpretazione musicale è il risultato della combinazione di intenzioni espressive e competenze tecniche del musicista. La maggior parte delle scelte espressive di un'artista sono influenzate dagli aspetti culturali propri dell'esecutore; tuttavia, dalla partitura è possibile estrapolare relazioni significative attraverso alcuni aspetti del linguaggio musicale e una classe di deviazioni sistematiche dei parametri musicali. Per l'analisi di un'esecuzione musicale è necessario introdurre due sorgenti di *espressività*: la prima è relativa agli aspetti tecnici della partitura musicale come il fraseggio e la struttura armonica del brano, la seconda si riferisce alle intenzioni espressive che comunicano stati d'animo e sentimenti. Per enfatizzare alcuni elementi della struttura musicale (ad esempio frasi ed accenti) il musicista modifica la sua performance aggiungendo elementi espressivi come *crescendo*, *decrescendo*, *sforzando*, *rallentando*, ecc.; in caso contrario l'esecuzione non sarebbe una performance musicale.

Si definisce performance neutra (Canazza et al., 2004), un'esecuzione umana senza una precisa intenzione espressiva, riprodotta in modo scolastico e senza nessun scopo artistico. Il modello è costruito agendo solamente sui gradi di libertà disponibili senza intaccare la relazione fra la struttura musicale e i pattern espressivi. Già nella performance neutra, il musicista introduce un fraseggio che traduce in variazioni di tempo e intensità la struttura del brano, ma con lo sviluppo di questo modello si mira a controllare in modo automatico il contenuto espressivo di una performance neutra pre-registrata.

L’input del sistema è composto da una descrizione della performance neutra e da un controllo che specifica l’intenzione espressiva desiderata dall’utente. Il modello opera a livello simbolico, elaborando le deviazioni di tutti i parametri musicali coinvolti nella trasformazione. In particolare, il rendering è fatto attraverso un sintetizzatore MIDI, mentre un motore di elaborazione audio esegue le trasformazioni sull’audio pre-registrato calcolate dal modello. Il controllo sull’intero sistema viene effettuato attraverso un’interfaccia grafica in cui l’utente può variare, a piacimento, l’ espressivo di una performance musicale.

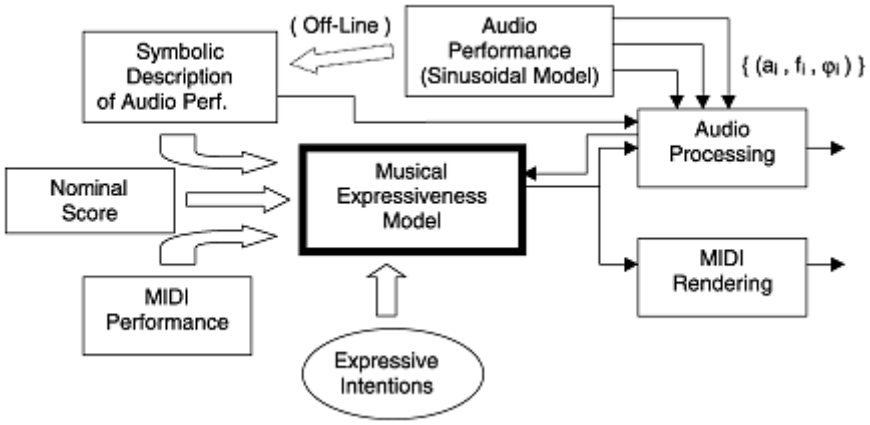


Figura 3.1: Schema del modello (Canazza et al., 2004).

3.1 Rappresentazione multilivello

Per descrivere il processo di rendering espressivo di una performance musicale, è stata proposta una rappresentazione multilivello dei dati musicali (Figura 3.2).

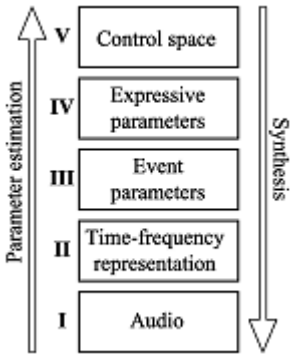


Figura 3.2: Rappresentazione Multilivello (Canazza et al., 2004).

Il primo livello rappresenta il segnale digitale audio. Il secondo livello è la rappresentazione in tempo e frequenza (TF) del segnale necessaria per l'analisi e la trasformazione della performance musicale. Questo tipo di rappresentazione è molto apprezzata nel campo dell'elaborazione di segnali musicali in quanto fornisce un ricco e robusto set di strumenti di trasformazione. Nel modello viene utilizzata una TF basata su una rappresentazione sinusoidale del segnale (Pielemeier et al., 1996), già utilizzata nel campo dell'analisi musicale. L'algoritmo di analisi opera su porzioni di segnale (frames) e produce una rappresentazione di quest'ultimo in funzione di una somma di sinusoidi (parziali) le cui frequenze, ampiezze e fasi variano molto lentamente nel tempo. Perciò, (1) rappresenta l' i-esimo frame del modello sinusoidale composto da un insieme di triple di frequenza (*f*), ampiezza (*a*) e fase (ϕ) che descrivono ogni parziale (*h*) .

$$\{(f_h(i), a_h(i), \phi_h(i))\}_{h=1}^{H} \quad (1)$$

Il terzo livello rappresenta la conoscenza sulla performance musicale sottoforma di eventi e corrisponde allo stesso livello di astrazione della rappresentazione MIDI. Nel dettaglio, una performance musicale può essere considerata una sequenza di note in cui l'n-esima nota è descritta da:

- *FR(n)*: valore di pitch (parametro relativo al tempo);
- *O(n)*: Onset Time (parametro relativo al tempo);
- *DR(n)*: Duration (parametro relativo al tempo);
- *I(n)*: Intensity (parametro relativo al timbro);
- *BR(n)*: Brightness (parametro relativo al timbro);
- *AD(n)*: Attack Duration (parametro relativo all'energia del segnale);
- *EC(n)*: Envelope Centroid (parametro relativo all'energia del segnale).

Infine, dai parametri relativi al tempo si possono calcolare:

- *IOI(n) = O(n+1) - O(n)*: Inter Onset Interval;
- *L(n) = DR(n) / IOI(n)*: Legato.

I parametri *P(n)* (P-parametri) modificati dal modello sono *L(n)*, *IOI(n)* e i parametri relativi al timbro per i MIDI oppure *I(n)*, *BR(n)*, *AD(n)* e *EC(n)* per performance audio.

FR(n)	pitch value
O(n)	onset time
DR(n)	duration
IOI(n)	inter onset interval
L(n)	legato
I(n)	intensity
BR(n)	brightness
AD(n)	attack duration
EC(n)	envelope centroid

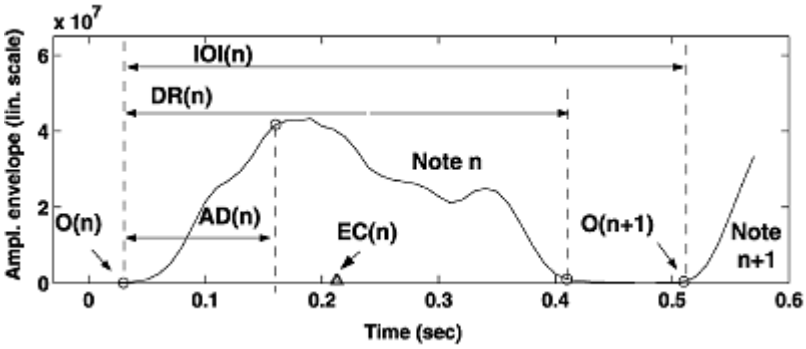


Figura 3.3: Parametri musicali coinvolti nel controllo dell'espressività (Canazza et al., 2004).

Il quarto livello rappresenta i parametri interni del modello espressivo. In particolare, viene utilizzata la coppia di valori E = {*k*, *m*} per descrivere ogni P-parametro. Infine, l'ultimo livello è lo spazio di controllo (*control space*) che gestisce, a livello astratto, il contenuto espressivo e l'interazione fra l'utente e la performance audio.

3.2 Modello per il rendering espressivo

Il modello si fonda sull'ipotesi che differenti intenzioni espressive possano essere ottenuti attraverso specifiche modifiche della performance neutra. Le trasformazioni realizzate soddisfano le seguenti condizioni:

1. mantengono inalterate le relazioni fra la struttura musicale e i pattern espressivi
2. mantengono il modello semplice, introducendo il minor numero possibile di parametri.

Per garantire il rispetto delle condizioni appena citate, vengono utilizzate solamente due tipi di trasformazioni: shift e compressione/decompressione. I migliori risultati sono stati ottenuti attraverso una mappa lineare che, per ogni P-parametro e per ogni intenzione espressivo *e,* è formalmente rappresentata dall'equazione:

$$P_e(n) = k_e\overline{P_0} + m_e(P_0(n) - \overline{P_0}) \quad (2)$$

dove $P_e(n)$ rappresenta il profilo stimato per la performance collegata all'intenzione espressiva *e,* $P_0(n)$ è il valore del P-parametro dell'n-esima nota della performance neutra, $\bar{P}_0$ è la media del profilo $P_0(n)$ calcolato sull'intero vettore, k_e e m_e sono rispettivamente i coefficienti di shift e compressione/decompressione associati

all'intenzione espressiva *e* (Figura 3.4). Perciò, (2) può essere generalizzata in modo da ottenere, per ogni P-parametro, una mappa per ogni differente intenzione espressiva. Si ottiene quindi:

$$P(n) = k(x,y)\overline{P_0} + m(x,y)(P_0(n) - \overline{P_0}) \quad (3)$$

Questa equazione lega ogni P-parametro con una generica intenzione espressiva rappresentata dai parametri *k* e *m;* questi parametri costituisco il quarto livello della rappresentazione multilivello e possono essere messi in relazione con le coordinate (*x,y*) dello spazio di controllo.

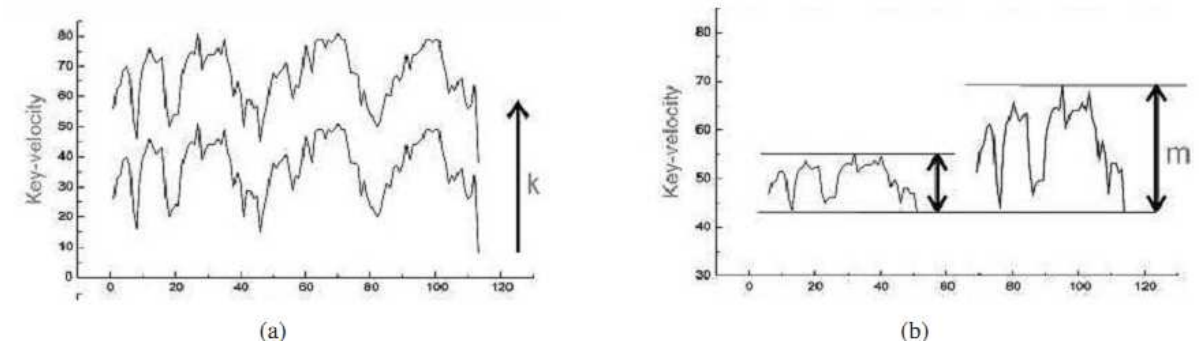


Figura 3.4: Interpretazione dei parametri espressivi *k* e *m* (Canazza et al., 2004).

3.3 Spazio di controllo (control space)

Lo spazio di controllo gestisce il contenuto espressivo e l'interazione fra l'utente e la performance audio finale. Per ottenere una trasformazione della performance sulla base di differenti intenzioni espressive, è stato sviluppato uno spazio di controllo astratto chiamato *perceptual parametric space* (PPS). Il PPS consiste in uno spazio bidimensionale derivato da un'analisi multidimensionale sui risultati di test percettivi effettuati su performance musicali professionali (Canazza et al., 1997a, 1999). Questo spazio rispecchia come la performance musicale è organizzata nella mente di un ascoltatore: è stato provato che gli assi del PPS sono correlati ai valori acustici e musicali percepiti dagli ascoltatori stessi (Canazza et al., 1997a). Per formalizzare il quinto livello della rappresentazione multilivello è stata fatta l'ipotesi che esista una relazione lineare fra gli assi del PPS e ogni coppia dei parametri espressivi {*k,m*}.

In particolare:

$$\begin{cases} k(x,y) = a_{k,0} + a_{k,1}x + a_{k,2}y \\ m(x,y) = a_{m,0} + a_{m,1}x + a_{m,2}y \end{cases} \quad (4)$$

Dove *x* e *y* sono le coordinate del PPS.

Come alternativa al PPS può essere utilizzato il *synthetic expressive space* (Canazza et al., 2000a), nel quale è possibile customizzare il proprio spazio di controllo, definendo i punti espressivi e posizionandoli a proprio piacimento.

3.4 Stima dei parametri

Gli ultimi tre livelli della rappresentazione multilivello (eventi, espressività e spazio di controllo) sono legati dalle espressioni (2) e (4). Nelle sezione corrente verrà descritto il processo di stima dei parametri utilizzati nel modello (Figura 3.5).

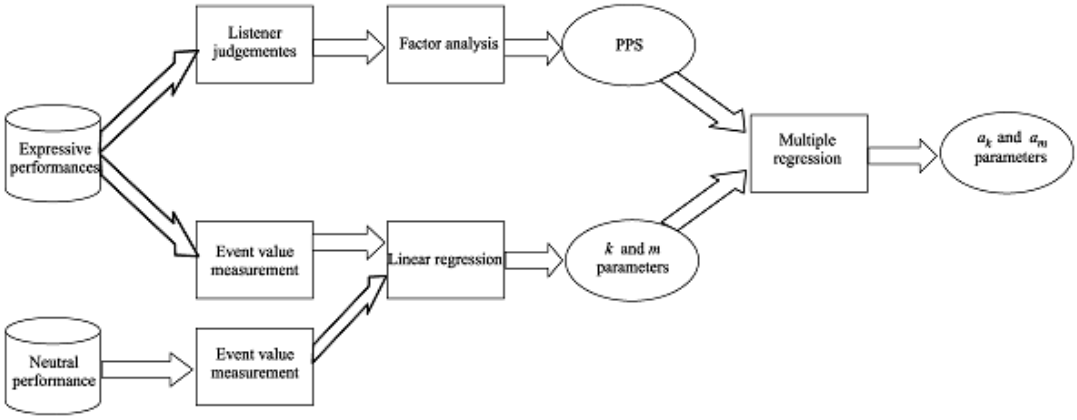


Figura 3.5: Processo per la stima dei parametri del modello (Canazza et al., 2004).

La stima è basata su un set di performance musicali ognuna delle quali è caratterizzata da una differente intenzione espressiva. Queste registrazioni sono state effettuate chiedendo a un musicista professionista di suonare più volte lo stesso brano, ogni volta ispirato da un differente intenzione espressiva. Inoltre è stata effettuata una registrazione della versione neutra del brano. Le registrazioni sono state successivamente valutate da un gruppo di ascoltatori i quali hanno assegnato un diverso punteggio alle varie performance compilando una tabella nella quale sono state riportate le intenzioni selezionate. I risultati sono stati infine

elaborati attraverso un'analisi fattoriale. Dall'analisi è emerso che due assi principali descrivono oltre il 75% della varianza totale. Ogni performance può, quindi, essere proiettata sullo spazio bidimensionale usando i due fattori sopra citati come coordinate *x* e *y*. Si considerino (x_e,y_e) come le coordinate della performance *e* nel PPS; nella Figura 3.6 sono riportati i valori dei fattori ottenuti dall'analisi fattoriale ed utilizzati come coordinate delle performance espressive nel PPS.

Per misurare il profilo di deviazione dei P-parametri, viene eseguita un'analisi acustica sulla performance espressiva. Per ogni intenzione espressiva, i profili vengono utilizzati per effettuare una regressione lineare che, rispettando i profili della performance neutra, calcola i parametri k_e ed m_e utilizzati nell'espressione (2). Il risultato di questo processo è un insieme di parametri espressivi E per ogni intenzione espressiva e per ogni P-parametro. Per ogni P-parametro, dati x_e, y_e e k_e, m_e è possibile calcolare il corrispettivo $a_{k,i}$ e $a_{m,i}$ (i = 0, 1, 2) dell'espressione (4) attraverso una regressione multipla sulle intenzioni espressive. In questo modo, partendo da alcune performance di esempio, sono stati elaborati i parametri del modello. Pertanto è possibile modificare l'espressività della performance neutra selezionando un punto arbitrario del PPS che causa la modifica dei parametri acustici della performance. Siano x_p e y_p le coordinate di un possibile punto (anche tempo variante) nel PPS, utilizzando l'equazione (3) si ottengono i valori dei parametri al livello III (livello degli eventi); questi valori vengono utilizzati come input, dal sintetizzatore MIDI e dai primi due livelli della rappresentazione (*Audio e Time Frequency Representation)*.

	Factor 1	Factor 2
Bright	0.8	0.1
Dark	-0.8	0.28
Hard	-0.4	0.6
Soft	-0.35	-0.7
Heavy	-0.75	0.5
Light	0.6	-0.5

Figura 3.6: Tabella dei fattori considerati come coordinate nel PPS.

3.5 Real-time rendering

Il rendering espressivo delle registrazioni audio è stato realizzato attraverso un motore audio basato sulla rappresentazione sinusoidale. Il modello descritto nelle sezioni precedenti è stato adattato per produrre un controllo tempo variante del motore sonoro. Nel dettaglio, lo sviluppo si è focalizzato su un'ampia classe di segnali chiamati suoni monofonici e quasi-armonici come, ad esempio, strumenti a fiato e a corda. Tutti i principali effetti sonori sono stati realizzati controllando i parametri della rappresentazione sinusoidale. In particolare sono stati ottenuti:

- *Time stretching:* ottenuto mediante il cambiamento del frame rate e attraverso l'interpolazione fra i parametri di due frame;
- *Pitch shift:* realizzato scalando le frequenze delle armoniche e preservando i formati con un'interpolazione spettrale d'inviluppo;
- *Intensity and brightness:* il controllo di questi parametri è ottenuto scalando l'ampiezza delle parziali e mantenendo la caratteristica spettrale naturale del suono.

Per modificare queste caratteristiche spettrali è stato realizzato un metodo originale di elaborazione dello spettro. Ciò permette la riproduzione del comportamento spettrale esibito da un insieme discreto di suoni di esempio, la cui intensità o brillantezza varia in un intervallo desiderato in funzione dell'intenzione espressiva della performance.

È stato introdotto un insieme di fattori moltiplicativi, γ_{IOI}, γ_{AD},γ_{DR},γ_{L},γ_{I},γ_{EC}, γ_{BR}, che rappresentano le variazioni dei parametri musicali sotto il controllo del motore audio. I primi tre fattori rappresentano, rispettivamente, il *time stretching* dell'intervallo di IOI, della durata dell'attacco e della durata dell'intera nota. Il fattore di variazione del *Legato* è correlato alla variazione della durata della nota e dell'IOI e può essere espressa come $\gamma_L = \gamma_{DR} / \gamma_{IOI}$. Il fattore d'intensità γ_I specifica il cambiamento uniforme dell'inviluppo dell'intera nota, mentre il fattore γ_{BR}, indica la variazione temporale nella posizione del centroide del profilo relativo alla dinamica del brano ed è legato a uno spostamento non uniforme lungo tutta la nota. Infine, il fattore γ_{BR} determina la modifica del centro dello spettro dell'intera nota. Pertanto il rendering

delle deviazioni compiute dal modello può imporre l'uso di uno o più degli effetti sonori visti sopra o la combinazione di questi, con la conseguente generazione delle seguenti nuove regole:

- *Local Tempo:* il *Time stretching* è applicato a ogni nota. È noto che negli strumenti a fiato e a corda la durata dell'attacco è essenziale per caratterizzare l'intenzione espressiva desiderata. Per questa ragione è calcolato uno specifico fattore di *time-streching* correlato ai parametri γ_{AD}, γ_{IOI} e al γ_{DR} (a sua volta influenzato dal controllo del *Legato* spigato al prossimo punto);
- *Legato:* questa caratteristica musicale è importante per la caratterizzazione espressiva delle performance con strumenti a fiato ed a corda. Il calcolo del *Legato* è un processo critico in quanto richiede la ricostruzione dell'attacco e del rilascio di una nota, se queste erano unite da un *Legato,* o la ricostruzione del transitorio se le note erano originariamente separate da una micro pausa. L'approccio utilizzato nel modello è quello di approssimare la ricostruzione del transitorio attraverso interpolazione delle ampiezze e delle frequenze delle tracce;
- *Envelope shape:* il centro di massa dell'energia di inviluppo è correlata all'accento musicale della nota che generalmente è presente nell'attacco per le intenzioni espressive *Leggero* e *Pesante,* mentre è vicina alla fine della nota per le intenzioni *Morbido* o *Cupo.* Per cambiare la posizione del centro di massa viene applicata una funzione triangolare all'energia di inviluppo, facendo corrispondere l'apice del triangolo alla nuova posizione dell'accento;
- *Intesity and Brightness Control:* l'intensità e la brillantezza del suono sono controllate da un processo di modellazione dello spettro. Viene utilizzata una rappresentazione pesata dell'inviluppo spettrale in modo da evidenziare le differenze rilevanti negli inviluppi. Si ottiene un modello parametrico per rappresentare le modifiche spettrali tramite il quale si calcolano le deviazioni dell'intensità e della brillantezza per il controllo dell'espressività.

3.6 Applicazioni e risultati

Il modello descritto è stato applicato a un'ampia varietà di registrazioni monofoniche, dalla musica classica alla musica popolare. Musicisti professionisti hanno suonato diverse partiture musicali ispirati dai seguenti aggettivi: *leggero (light)*, *pesante (heavy)*, *morbido (soft)*, *duro (hard)*, *brillante (bright)* e *cupo (dark)*. È stata aggiunta anche la performance neutra per poterla utilizzare come riferimento nell'analisi acustica delle varie interpretazioni. Sono stati deliberatamente utilizzati degli aggettivi non codificati nel campo musicale in modo da garantire al musicista la maggiore libertà di espressione possibile. A questo punto, il musicista ha scelto la performance che nella sua opinione corrispondeva maggiormente agli aggettivi proposti, in modo da limitare l'influenza che l'ordine di esecuzione avrebbe potuto esercitare sull'esecutore. Tutte le registrazioni sono state effettuate al Centro Sonologia Computazionale (CSC) dell'Università di Padova nel formato monofonico digitale a 16 bit e 44.1 kHz. In totale sono stati considerati 12 pezzi suonati con differenti strumenti (violino, clarinetto, piano, flauto, voce e sassofono) e da vari artisti (circa 5 per ogni melodia). Sono state considerate solamente brevi melodie (dai 10 ai 20 secondi) in modo da garantire che i parametri musicali ad alto livello (ad esempio, armonia e metronomo) rimangano costanti. È stata eseguita un'analisi acustica semiautomatica dei parametri relativi al tempo ed al timbro per stimare il valore di IOI, L, AD, I, EC e BR. Nella tabella di Figura 3.7, sono riportati i valori di *k* e *m* ottenuti attraverso l'analisi dei parametri descritta precedentemente. Ad esempio, si può notare come il *k* relativo al Legato (L) sia un parametro importante per distinguere l'intenzione espressiva *duro* (*k* = 0,92 che significa *abbastanza staccato*) da *morbido* (*k* = 1,43 che significa *molto legato*). Se consideriamo il parametro *Intensity* (I) si nota come *pesante* e *brillante* abbiano un valore di *k* molto simile ma un valore *m* differente; ciò è dovuto al fatto che nel *pesante* ogni nota è suonata con elevata intensità (*m* = 0,7), al contrario del *brillante* dove ogni nota è suonata con elevata variazione di intensità (*m* = 1,06).

	IOI		*L*		*DRA*		*I*		*EC*		*BR*	
	k	*m*	*k*	*m*	*k*	*m*	*k*	*m*	*k*	*m*	*k*	*m*
Bright	0.87	0.98	0.68	0.95	0.76	0.96	1.07	1.06	0.90	0.79	1.13	0.80
Dark	1.05	1.01	1.09	1.02	0.93	1.12	0.87	1.05	1.12	1.06	0.67	0.72
Hard	0.95	0.86	0.92	1.06	0.73	0.84	1.06	0.76	0.98	1.04	1.17	0.96
Soft	1.03	1.08	1.43	0.89	1.06	1.02	0.92	1.03	1.18	1.11	0.74	1.05
Heavy	1.16	0.91	1.35	0.98	0.97	1.05	1.06	0.70	0.98	1.06	1.10	0.99
Light	0.90	0.96	0.79	1.12	1.13	1.10	0.97	1.12	0.84	0.84	0.82	1.03

Figura 3.7: Parametri espressivi stimati per la Sonata K545 di Mozart.

I fattori ottenuti dall'analisi sui risultati dei test percettivi sono riportati in Figura 3.6; questi fattori vengono utilizzati come coordinate della performance espressiva nel PPS. Si nota come il fattore 1 distingua *brillante* (0,8) da *cupo* (-0,8) e *pesante* (-0,75), mentre il fattore 2 differenzi *duro* (0,6) e *pesante* (0,5) da *morbido* (-0,7) e *leggero* (-0,5). A partire dai dati di *k* e *m* riportati nella tabella di Figura 3.8 e dalla posizione sullo spazio di controllo si riesce quindi a calcolare il valore dei parametri dell'equazione (3); si può quindi utilizzare il modello per modificare iterativamente l'espressività della performance neutra, muovendosi nello spazio di controllo 2-D. In questo modo all'utente è data la possibilità di disegnare qualsiasi traiettoria che rispecchi la variazione espressiva che vuole infondere alla performance neutra (Figura 3.8).

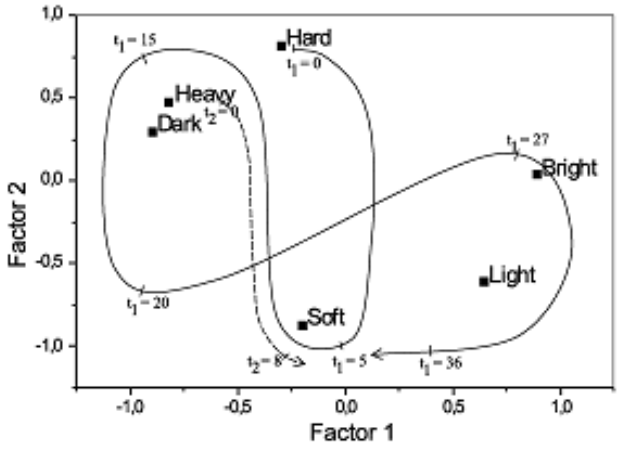
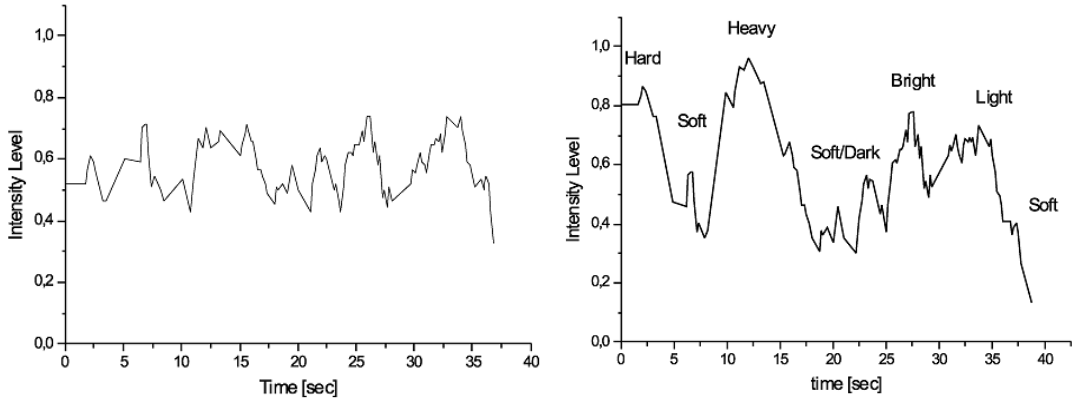


Figura 3.8: La traiettoria rappresenta la variazione nel tempo dell'intenzione espressiva dell'utente. Linea continua: traiettoria per il pezzo di Mozart; linea tratteggiata: traiettoria per il pezzo di Corelli (Canazza et al., 2004).

In Figura 3.9-B viene riportato l'effetto che la traiettoria (linea continua), disegnata in Figura 3.8, ha sul livello di intensità I rispetto al livello di intensità della performance neutra (Figura 3.9-A). Si nota come l'intensità vari in accordo con la traiettoria; ad esempio, le intenzioni espressive *duro* e *pesante* sono suonati con intensità maggiore rispetto a *morbido.* Questo fatto è evidente nella Figura 3.7, dove il valore di *k* è 1,06 sia per *duro* che per *pesante* mentre è di 0,92 per *morbido.* Di contro, si può osservare un range di variazione molto ampio per la performance leggera (m = 1,12) rispetto alla performance pesante (m = 0,70). La nuova curva di intensità è infine utilizzata per controllare il motore audio nel passo finale di rendering.



(A) (B)

Figura 3.9: Livello di intensità I della performance neutra (A) e della performance espressiva (B) (Canazza et al., 2004).

Come ultimo esempio viene considerato un estratto della sonata di Corelli op. V descritta in Figura 3.10.

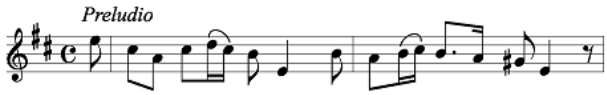


Figura 3.10: Partitura dell'estratto della sonata di Corelli op. V.

Le Figg. 3.11-3.13 raffigurano l'energia di inviluppo e il profilo del pitch della performance neutra, pesante e morbida (solo violino). Il modello è utilizzato per ottenere una transizione sfumata dal pesante al morbido (Figura 3.8, linea tratteggiata) dopo aver applicato le dovute trasformazioni sulla rappresentazione sinusoidale della performance neutra. Il risultato di queste trasformazioni è riportato

in Figura 3.14. Si nota come l'energia dell'inviluppo cambi da valori alti a valori bassi in accordo con le performance originali (pesante e morbido). Il profilo di *pitch* mostra il differente comportamento del parametro IOI: la performance morbida (k = 1,03) è suonata più velocemente rispetto la performance pesante (k = 1,16). Questo comportamento è preservato anche nell'esempio di sintesi effettuato.

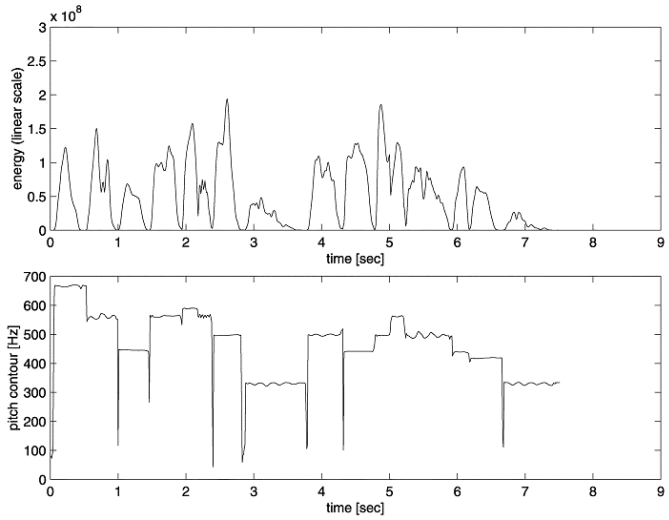


Figura 3.11: Energia dell'inviluppo e profilo di pitch della performance neutra della sonata di Corelli op.V (Canazza et al., 2004).

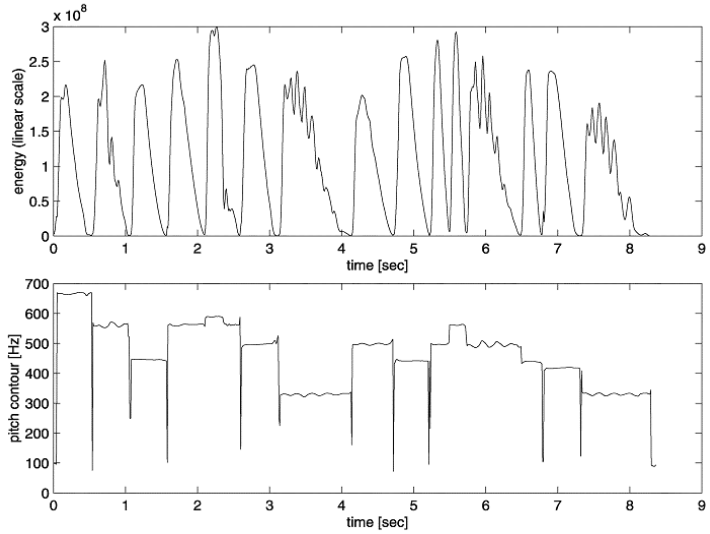


Figura 3.12: Energia dell'inviluppo e profilo di pitch della performance *pesante* della sonata di Corelli op.V (Canazza et al., 2004).

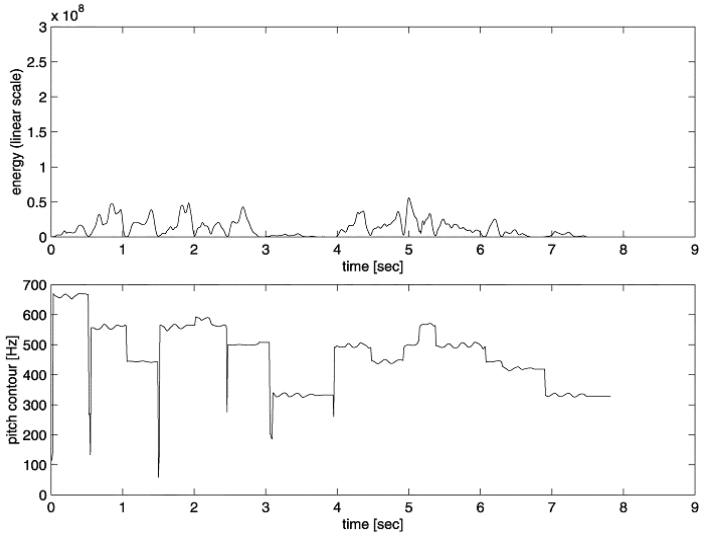


Figura 3.13: Energia dell'inviluppo e profilo di pitch della performance *morbida* della sonata di Corelli op.V (Canazza et al., 2004).

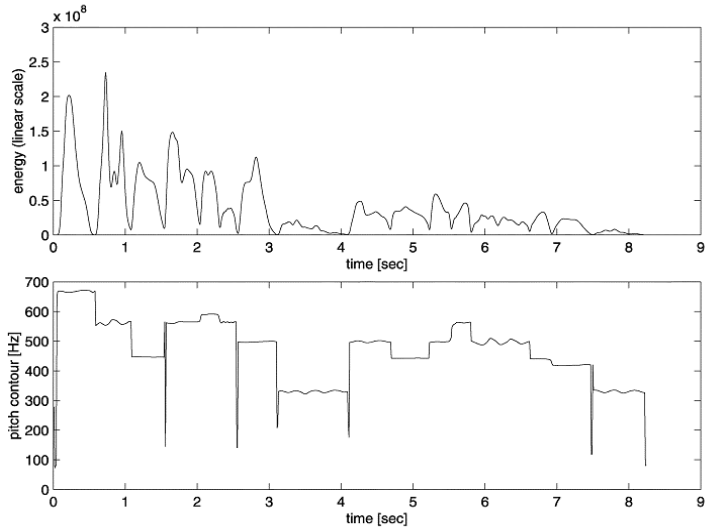


Figura 3.14: Energia dell'inviluppo e profilo di pitch della modifica espressiva della sonata di Corelli op.V. L'intenzione espressiva cambia gradualmente da pesante a morbido (Canazza et al., 2004).

3.7 Valutazioni

Per validare il sistema è stato effettuato un test percettivo volto a verificare se le performance sintetizzate corrispondevano effettivamente agli aggettivi utilizzati negli esperimenti. L'obiettivo principale del test era verificare se una determinata intenzione veniva identificata da un ascoltatore e se il sistema era in grado di convertire correttamente una particolare intenzione espressiva.

Data una partitura e una performance neutra sono state ottenute le cinque diverse interpretazioni (*brillante, duro, leggero, morbido* e *pesante*) dallo spazio di controllo. L'intenzione espressiva *cupo* non è stata considerata in quanto si è notato che poteva essere facilmente confusa con *pesante* (Figura 3.8).

Era importante testare il sistema con differenti partiture per capire quanto alta fosse la correlazione fra la struttura del pezzo e il riconoscimento dell'espressività; per questo motivo, nei test sono stati scelti tre pezzi per pianoforte con caratteristiche acustiche diverse. Nel dettaglio, sono stati scelti: *Sonatina in sol* di L. van Beethoven, *Valzer no. 7 op. 64* di F. Chopin e K545 di W. A. Mozart.

Il gruppo di ascolto era composto da 20 soggetti: 15 esperti (musicisti e/o diplomati al conservatorio) e 15 non esperti (persone senza particolare conoscenza musicale).

I soggetti hanno ascoltato lo stimolo attraverso delle cuffie ad un'intensità di volume adeguata. Agli ascoltatori era concesso di ascoltare lo stimolo tutte le volte che desideravano e successivamente era richiesto di valutare il grado di *brillantezza, durezza, leggerezza, morbidezza* e *pesantezza* di tutte le performance, riportando il valore (da 0 a 100) su una scala graduata.

I risultati sono stati analizzati attraverso procedure statistiche per valutare se le intenzioni espressive fossero state correttamente identificate.

Nella Figura 3.15 sono riportati i risultati ottenuti dal test.

	Sonatina						Valzer						K545					
	B	Hr	L	S	Hv	N	B	Hr	L	S	Hv	N	B	Hr	L	S	Hv	N
B	**73.2**	59.3	51.7	50.7	43.2	**55.3**	**69.8**	50.1	36.6	42.2	35.5	44.7	**71.6**	67.3	35.2	41.7	46.0	33.8
Hr	59.1	**79.2**	28.1	32.6	61.7	33.4	49.8	**74.7**	18.0	25.6	65.2	37.9	56.8	**68.4**	22.5	15.7	80.3	17.1
L	41.9	17.7	**66.6**	57.4	17.6	54.2	42.0	21.1	66.1	57.1	17.1	47.5	34.2	27.6	**75.0**	75.5	12.9	69.3
S	24.1	13.1	59.9	**64.6**	24.7	55.1	29.8	22.7	**72.5**	**66.0**	22.5	**53.0**	22.6	27.7	65.3	**77.2**	13.9	**72.7**
Hv	43.4	69.8	18.2	25.1	**69.8**	25.2	39.0	65.7	22.9	24.7	**78.1**	37.7	37.3	52.8	15.2	15.0	**82.8**	17.8

Figura 3.15. Media dei voti (da 0 a 100) assegnati nel test. Le righe rappresentano le etichette di valutazione mentre le colonne mostrano i differenti stimoli. Legenda: B=brillante, Hr=duro, L=leggero, S=morbido, Hv=pesante, N=neutra.

Il test ANOVA sulle risposte date dagli ascoltatori riporta un indice p minore di 0,001. Dall'analisi dei dati si nota che, data un'interpretazione del pezzo, l'intenzione

espressiva corretta era riconosciuta dalla maggior parte dei soggetti. L'unica eccezione è il Valzer dove l'intenzione *leggero* è stata interpretata come *morbido.*

È interessante evidenziare anche come gli ascoltatori hanno valutato la performance neutra; dall'analisi è emersa una predominanza delle intenzioni *leggero* e *morbido.*

Si nota infine, un'alta correlazione fra *duro* e *pesante* e fra *leggero* e *morbido.* L'intenzione espressiva *brillante* invece, sembra essere più complicata da individuare.

3.8 CaRo 2.0: l'implementazione del modello

Il modello descritto nei paragrafi precedenti è stato implementato e ha portato alla creazione di un software per l'esecuzione espressiva di una performance musicale chiamato CaRo (dal nome degli autori CAnazza e ROdà). Il progetto è stato implementato utilizzando il compilatore Builder C++ della Borland e si basa sulla libreria MidiShare[5] per la comunicazione e la gestione dei file MIDI.

Figura 3.16: Interfaccia del software CaRo 2.0.

[5] http://http://midishare.sourceforge.net/

All’avvio, l’applicazione si presenta con una finestra di nome CaRo (Figura 3.16) che costituisce l’interfaccia fra l’utente ed il programma. In alto si trova il menù per la gestione dell’applicativo mentre nella parte centrale è presente un’immagine bitmap di 400x400 pixel. Questa rappresenta lo spazio di controllo astratto chiamato P*erceptual Parametric Space* (PPS, vedi paragrafo 3.3) ed è lo strumento tramite cui è possibile infondere espressività ad un’esecuzione musicale.

3.8.1 CaRo 2.0: composizione del progetto

Il progetto per lo sviluppo di CaRo 2.0 si compone dai seguenti file:

- *XML.h:* contiene le variabili e i prototipi della classe XML utilizzata per la codifica delle informazioni espressive (tenuti, respiri ecc.) che si possono aggiungere al brano selezionato.
- *Parabola.h*: contiene le variabili e i prototipi della classe Parabola utilizzata per la gestione delle legature espressive.
- *Option.cpp*: contiene il codice per il funzionamento del form Opzioni. Attraverso questo form è possibile impostare i parametri di funzionamento del software (Figura 3.17).
- *Options.h*: header file del form Opzioni.
- *Modello.cpp*: file principale del progetto che contiene la maggior parte del codice dell’applicazione. Al suo interno vengono implementate le classi XML e Parabola e il codice relativo al form principale dell’applicazione CaRo.
- *Modello.h*: header file contenente le variabili e le funzioni implementate in Modello.cpp.
- *Oggettone.cpp*: file contenente il codice sorgente del progetto sviluppato con l’editor Builder C++ della Borland.

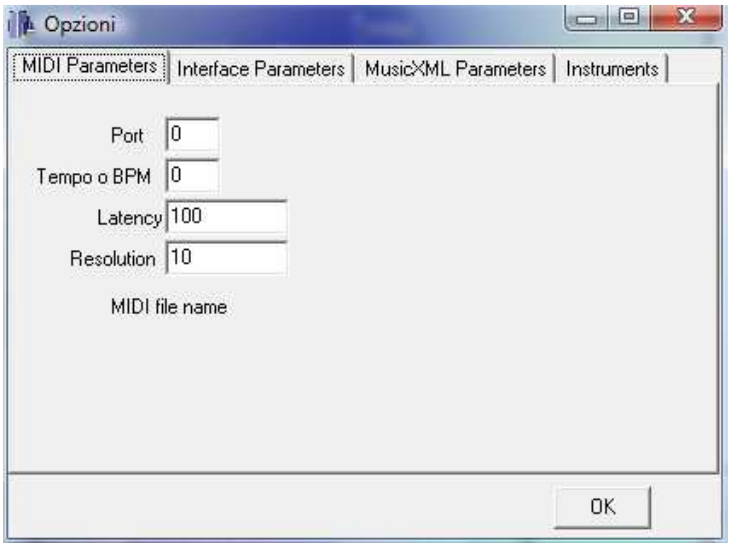


Figura 3.16: Form Opzioni.

3.8.2 CaRo 2.0: modalità di riproduzione di un brano

Le principali operazioni che si possono eseguire utilizzando il software CaRo sono:

- Esecuzione meccanica: il brano viene riprodotto in maniera tradizionale senza l'aggiunta di espressività.
- Esecuzione neutra: si simula un'esecuzione musicale umana senza particolari intenzioni espressive, ma con le deviazioni dei parametri tipiche di una performance reale. La specifica dei pattern espressivi avviene con l'ausilio di un file *MusicXML* che contiene la codifica della partitura in formato XML a cui sono stati aggiunti manualmente dei segni espressivi. Attraverso la scheda *MusicXML Parameters* è possibile specificare quanto i segni espressivi aggiunti incidano sui parametri nominali del brano (ad esempio velocità o tempo).Nell'ultima versione del software (versione 2.0) il sistema consente l'aggiunta dei seguenti segni espressivi:
    - Respiro
    - Accento
    - Tenuto
    - Staccato
    - Legature espressive
    - Dinamiche (*pp, p ,mp, mf, f, ff)*
    - Gestione del pedale di risonanza

- Esecuzione espressiva di un brano: il brano viene riprodotto inserendo l'intenzione espressiva desiderata dall'utente. Attraverso il movimento del mouse sul PPS l'utente può infondere al brano le seguenti cinque intenzioni espressive: *morbido, duro, pesante, leggero* e *brillante.* Il sistema consente di espressivizzare sia brani originali sia brani a cui sono stati aggiunti i segni espressivi descritti al punto precedente. Come nel caso di un'esecuzione neutra anche per l'esecuzione espressiva è possibile impostare i valori dei parametri musicali in funzione delle intenzioni espressive. Ciò è possibile attraverso l'utilizzo della scheda *Interface Parameters* del form Opzioni.

Capitolo 4

Performance Rendering Contest (Rencon)

Rencon[6] (*Performance Rendering Contest*) è una competizione a livello mondiale nel campo dell'informatica musicale. Sostanzialmente è una sfida fra sistemi automatici in grado di generare performance musicali espressive in modo autonomo o in modo interattivo. L'obiettivo è selezionare il miglior sistema in grado non solo di riprodurre un brano musicale, ma di farlo con "personalità".

Lo slogan di Rencon è che il sistema computazionale che vincerà il contest nel 2050 sarà in grado di vincere lo *Chopin contest*[7]. E poi, nel 2100 un esecutore umano, addestrato da un sistema automatico, tornerà a vincere lo *Chopin contest*.

Concretamente, il contest si pone l'obiettivo di progettare sistemi utili alla didattica musicale considerando che questa disciplina è ancora basata sull'apprendimento per imitazione.

4.1 Regolamento

L'obiettivo della sfida è molto semplice: un programma appositamente sviluppato da ogni partecipante dovrà eseguire due brani per pianoforte in maniera espressiva, modificandone perciò il tempo, la dinamica e l'articolazione in modo tale da rendere l'esecuzione simile ad una performance musicale umana. La partitura da riprodurre contiene note e alcuni segni espressivi (*f, p crescendo, andante,* legature ecc.) ma nessuna indicazione sulla struttura delle frasi e sugli accordi. L'interpretazione espressiva generata dal programma verrà salvata in formato MIDI e successivamente verrà riprodotta su un pianoforte controllato da un computer. Ai

[6] http://renconmusic.org/

[7] International Chopin Piano Competition: è una competizione al pianoforte che si tiene a Varsavia in Polonia in onore di Frédéric Chopin in cui partecipano i migliori pianisti a livello mondiale.

partecipanti sono dati 60 minuti per sistemare il proprio software e generare attraverso il proprio sistema l'esecuzione espressiva richiesta. Non sono permesse modifiche manuali durante il processo di rendering espressivo sul pianoforte e nemmeno l'ascolto del MIDI prodotto prima che questo venga pubblicamente riprodotto. Le esecuzioni verranno ascoltate e valutate da un gruppo di esperti e non, che dovranno considerare la naturalezza e l'espressività del brano riprodotto.

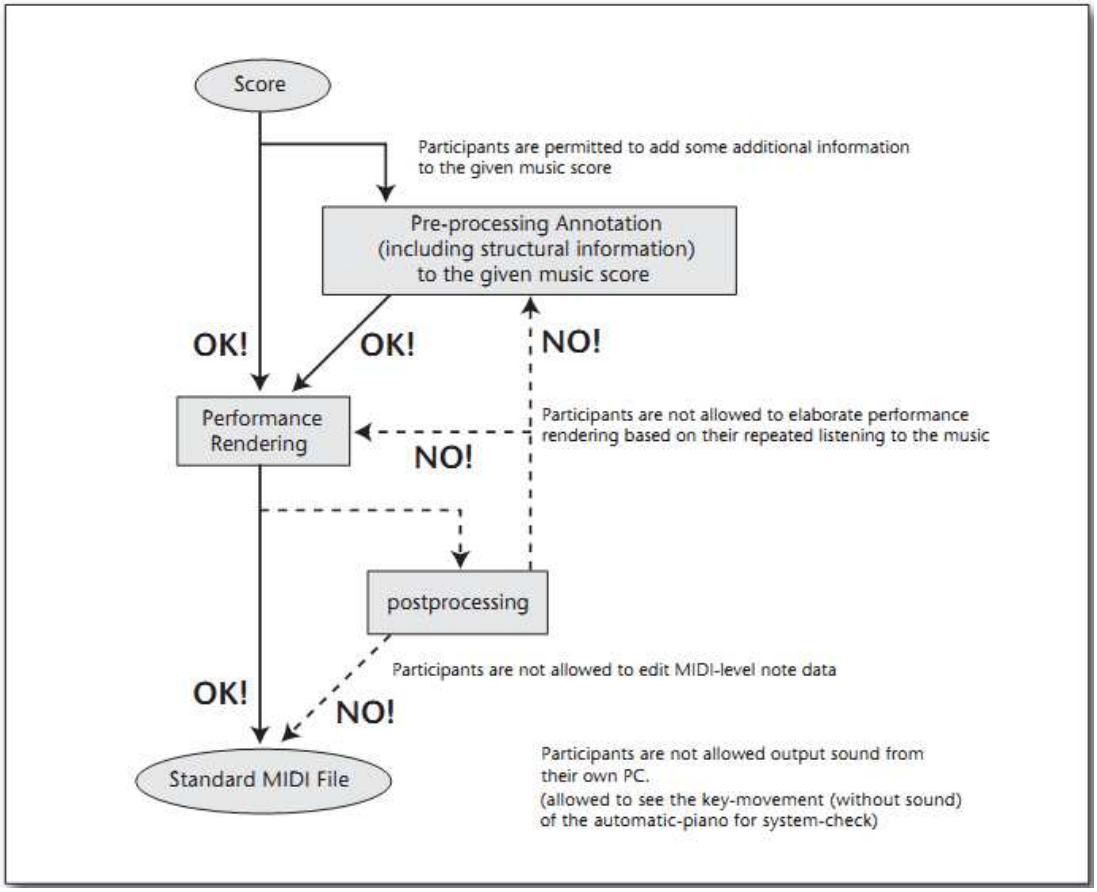


Figura 4.1: Sommario grafico delle istruzioni date ai partecipanti di Rencon.

4.2 SMC - Rencon 2011

La fase finale di Rencon 2011 (6 luglio 2011) si è tenuta per la prima volta in Italia, il presso il Dipartimento dell'Ingegneria dell'Informazione (DEI) dell'Università di Padova, all'interno dell'ottava edizione del *Sound and Music Computing Conference* (SMC). Il contest si è svolto in due differenti stage:

- Stage I: le performance musicali espressive generate dai sistemi in gara e i sistemi stessi, sono stati valutati da una commissione di esperti. Sono stati ammessi alla fase successiva solamente 8 sistemi:
    - *Director Musices* di Erica Bisesi (KTH), Anders Freiberg (KTH) e Richard Parncutt (Univerity of Graz).
    - *Usapi* di Keiko Teramura (Kyoto University) e Shin-ichi Maeda (Kyoto University).
    - *Kagurame Phase-II* di Taizan Suzuki (Picolab Co., LTD), Tatsuya Hino (Shibaura Institute of Technology), Masahiro hibasaki (Shibaura Institute of Technology) e Yukio Tokunaga (Shibaura Institute of Technology).
    - *Kagurame Phase-III* di Takashi Baba (Kwansei Gakuin University), Mitsuyo Hashida (Kwansei Gakuin University) e Haruhiro Katayose (Kwansei Gakuin University).
    - *Virtual Philharmony* di Taizan Suzuki (Picolab Co., LTD), Tatsuya Hino (Shibaura Institute of Technology), Masahiro Hibasaki (Shibaura Institute of Technology) e Yukio Tokunaga (Shibaura Institute of Technology).
    - *YQX* di Sebastian Flossmann (Johannes Kepler University), Maarten Grachten (Johannes Kepler University) e Gerhard Widmer (Johannes Kepler University).
    - *Shunji System* di Shunji Tanaka (Kwansei Gakuin University), Mitsuyo Hashida (Kwansei Gakuin University) e Haruhiro Katayose (Kwansei Gakuin University). Performer: Mami Yamaguchi.
    - *CaRo 2.0* di Sergio Canazza (Università di Padova), Giovanni De Poli (Università di Padova), Antonio Rodà (Università di Padova), Massimiliano Barichello (Università di Padova) e Davide Ganeo (Università di Padova). Performer: Davide Tiso.
- Stage II: ogni sistema ha generato un'esecuzione espressiva del brano selezionato che è stata valutata dai partecipanti dell'SMC e dagli utenti online che seguivano la manifestazione. Fra i partecipanti dell'SMC era presente anche una commissione scientifica che comprendeva i maggiori esperti nel settore dell'informatica musicale.

Per il rendering espressivo è stato selezionato il brano *Sonata No. 8 Op. 13 III. Allegro di Beethoven* da una lista di 20 brani che comprendeva:

1. **J. S. Bach**: *Wohltemperierte Klavier I-1 BWV846 Prelude*.
2. **J. S. Bach**: *Invention No. 15 BWV786*.
3. **J. S. Bach**: *The Little Notebook for Anna Magdalena Bach, No. 1 Menuette in G major (no.24).*
4. **T. Badarzewska**: *A Maiden's Prayer*.
5. **L.v. Beethoven**: *Piano Sonata No. 8, 3rd Mov*.
6. **L.v. Beethoven**: *Bagatelle No. 25 in A minor "For Elise"*.
7. **J. Brahms**: *Hungarian Dances No. 5 in F-sharp minor*.
8. **F. Chopin**: *Nocturne No. 2 in E Flat Major, Op.9-2*.
9. **F. Chopin**: *Etude No. 3 in E major, Op.10 Nr.3 "Chanson de L'adieu"*.
10. **F. Chopin**: *Waltz No.9 in A flat major, Op.69 Nr.1 "L'adieu"*.
11. **E. Elgar**: *Salut d'amour*.
12. **G. Faure**: *Sicilienne in G minor, Op. 78*.
13. **G.F. Händel**: *The Harmonious Blacksmith*.
14. **F. Liszt**: *2 Konzertetüden, S.145, No. 1 "imparare"*.
15. **F. Mendelssohn**: *Songs Without Words Op. 30 No. 6, fis-moll "Venezianisches Gondellied".*
16. **W. A. Mozart**: *Piano Sonata K. 545, 1st Mov*.
17. **W. A. Mozart**: *Piano Sonata K. 331, 3rd Mov. "Turkish march"*.
18. **D. Scarlatti**: *Sonata in C major K. 159, Allegro*.
19. **R. Schumann:** *Abegg Variations Op.1: 1. Theme*.
20. **P. I. Tchaikovsky**: *The Seasons: 7. July*.

I brani sono stati elaborati utilizzando il programma *Finale 2010*. Ad ogni partecipante è stato consegnato il brano in formato *MusicXML* (1.0/2.0) e *Standard MIDI File* (SMF format 1). Inoltre è stata fornita la partitura del brano in forma cartacea.

4.2.1 Disklavier

Per l'ascolto e la valutazione delle performance espressive prodotte dai sistemi in gara è stato utilizzato un pianoforte Yamaha Disklavier.

I Disklavier sono essenzialmente dei moderni pianoforti elettromeccanici che, tramite l'utilizzo di sensori ottici collegati a LED, permettono di riprodurre note e usare i pedali indipendentemente da qualsiasi operatore umano. La maggior parte dei modelli sono basati su pianoforti acustici reali e sono progettati in modo che i sensori e gli elementi elettromeccanici non interferiscano sulla riproduzione classica di un brano.

Questi pianoforti consentono la memorizzazione di dati, comprese le esecuzioni reali di un pianista umano e quindi tali dati possono essere utilizzati per riascoltare qualsiasi tipo di performance riprodotta sullo strumento. I Disklavier sono dotati di ingressi per i dati MIDI e anche da altri dispositivi di memorizzazione come CD-ROM, cavi seriali e USB.

Figura 4.2: Pianoforte Yamaha Disklavier.

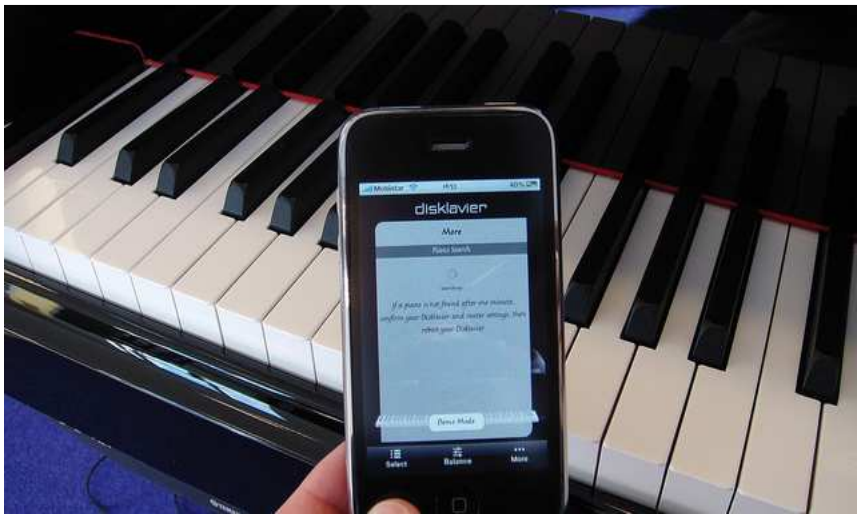


Figura 4.3: Controller del pianoforte. Tramite una specifica App e possibile controllare il Disklavier utilizzando un iPhone.

4.2.2 SMC Rencon 2011 – Risultati

Tutte le esecuzioni espressive generate dai sistemi automatici in gara sono state riprodotte sul Disklavier. I sistemi in gara sono stati suddivisi in due categorie:

- Sistemi automatici: le performance generate sono state riprodotte sul DiskLavier in modo autonomo senza l'intervento umano. A questa categoria appartengono i sistemi *Director Musices, Usapi, Kagurame Phase-II, Kagurame Phase-III, YQX* e *Shunji System.*
- Sistemi interattivi: il rendering espressivo è avvenuto in modo interattivo grazie all'azione di un esecutore. Appartengono a questa categoria i sistemi *CaRo 2.0* e *VirtualPhilarmony.*

Partendo dal brano selezionato (*Sonata No. 8 Op. 13 III. Allegro di Beethoven*) ogni sistema ha generato due differenti performance espressive (Performance A e Performance B) ognuna delle quali è stata valutata singolarmente. La somma dei punteggi ha decretato il vincitore della seconda fase di Rencon 2011. I sistemi Kagurame Phase II/III non hanno partecipato alla fase finale per problemi tecnici.

Grazie all'abilità del maestro Davide Tiso, che ha saputo esprimere al meglio le potenzialità del software CaRo 2.0, il sistema per la generazione di performance espressive dell'Università di Padova ha vinto il secondo stage dell'edizione 2011 del

Rendering Contest. Di seguito è riportata la classifica dello Stage II (Figura 4.4) e la classifica finale del contest (Figura 4.5).

Entrant System	Performance A			Performance B			Total Score
	Internet	Paper	Total	Internet	Paper	Total	
1st: CaRo 2.0	56	464	520	61	484	545	1065
2nd: YQX	64	484	548	65	432	497	1045
3rd: VirtualPhilarmony	46	452	498	32	428	460	958
4th: Director Musices	33	339	372	13	371	384	756
5th: Shunji System	24	277	301	20	277	297	598

Figura 4.4: Classifica della fase finale SMC Rencon 2011 (Stage II). La colonna Internet indica i voti espressi online, la colonna Paper rappresenta i voti dati degli ascoltatori presenti al workshop e la colonna Total è la somma dei due punteggi precedenti

System	Rank of Stage I	Rank of Stage II	Score of I + II
1st: YQX	1	2	3
2nd: VirtualPhilarmony	2	3	5
3rd: Director Musices	3	4	7

.Figura 4.5: Classifica complessiva di Rencon 2011 (Stage I e Stage II).

Capitolo 5

Conclusioni

Grazie all'evolversi della tecnologia informatica, le modalità di accesso alla musica si sono moltiplicate e questo ha portato alla necessità di avere nuovi paradigmi di interazione basati su una migliore comprensione dell'esperienza musicale. La definizione di modelli informatici per rappresentare l'espressività di un'esecuzione musicale è sicuramente un primo passo per colmare il divario esistente fra il semplice segnale audio fisico e la cognizione di ciò che si vuole esprimere attraverso una precisa performance musicale.

Lo studio delle connessioni esistenti fra le emozioni e le macchine può sembrare un ossimoro, ma, in realtà, approfondire e analizzare le modalità con cui un computer può comunicare contenuti espressivi utilizzando la musica, oggigiorno ricopre una notevole importanza. Per rendersi conto di questo é sufficiente pensare a contesti come quelli dell'insegnamento e dell'educazione. Migliorare i meccanismi di interazione uomo-macchina può fornire strumenti per l'insegnamento meno frustranti e più facili da assimilare per utenti senza particolari competenze informatiche (bambini, insegnanti o musicisti) o per utenti diversamente abili. Inoltre, lo sviluppo di sistemi automatici per l'esecuzione musicale espressiva risulta utile alla comprensione dei modelli esecutivi e, in futuro, potrà assumere un ruolo fondamentale nella didattica musicale dove l'apprendimento avviene ancora per imitazione.

La creazione di sistemi per esecuzioni musicali espressive offre, inoltre, nuove opportunità di business legate alla diffusione e alla vendita di prodotti musicali sintetizzati al computer. Un esempio che può sembrare fuori dal comune ma che in Giappone è divenuto realtà è *Hatsune Miku* (che in giapponese significa *voce del futuro*), un avatar che si esibisce "dal vivo" sottoforma di un ologramma 3D. Il suo aspetto è stato creato attraverso speciali programmi di grafica e la sua voce è stata interamente creata grazie ad uno speciale sintetizzatore della Yamaha chiamato

Vocaloid. L’espressività ha giocato un ruolo fondamentale nella fase di creazione della voce, in quanto ha permesso di generare delle performance paragonabili a quelle di un artista reale.

Gli esempi appena citati e i sistemi descritti all’interno di questo lavoro di tesi sono solo alcuni dei risultati prodotti dagli studi inerenti l’espressività musicale. La ricerca è ancora in una fase iniziale e certamente c’è ancora molto su cui lavorare. Tuttavia, i primi risultati sono molto incoraggianti e occasioni di confronto come Rencon sono appuntamenti essenziali per lo sviluppo di questa disciplina.

Bibliografia

Arcos, J. L., Mantaras, R. L., Serra, X. (1998). SaxEx: a case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27, 194-210.

Arcos, J. L., & Lopez de Mantaras, R. (2001). An interactive case-based reasoning approach for generating expressive music. *Applied Intelligence*, 14(1), 115-119.

Bresin, R. (1998). Artificial neural networks based models for automatic performance of musical scores. *Journal of New Music Research*, 27(3), 239-270.

Bresin, R. (2000). Virtual Virtuosity. Studies in Automatic Music Performance. Doctoral Dissertation, Speech Music and Hearing.

Camurri, A., De Poli, G., Leman, M., & Volpe, G. (2005). Toward Communicating Expressiveness and Affect in Multimodal Interactive Systems for Performing Arts and Cultural Applications. *IEEE Multimedia*, 12(1), 43-53.

Canazza, S., De Poli, G., & Vidolin, A. (1997a). Perceptual analysis of the musical expressive intention in a clarinet performance. In Leman, M., editor, *Music, Gestalt, and Computing*, pages 441-550. Berlin, Germany: Springer-Verlag.

Canazza, S., De Poli, G., Rinaldin, S., & Vidolin, A. (1997b). Sonological analysis of clarinet expressivity. In Leman, M., editor, *Music, Gestalt, and Computing - Studies in Cognitive and Systematic Musicology*, 431-440. Berlin, Heidelberg: Springer-Verlag.

Canazza, S. & Orio, N. (1999). The communication of emotions in jazz music: a study on piano and saxophone performances. *Gen. Psychol.(Special Issue on Musical Behavior and Cognition)*, 3/4, 261-276.

Canazza, S., De Poli, G., Drioli, C., Rodà, A., & Vidolin, A. (2000a). Audio morphing different expressive intentions for multimedia systems. *IEEE Multimedia*, 7(3):79-83.

Canazza, S., De Poli, G., Drioli, C., Rodà, A., & Vidolin, A. (2000b). Modeling and control of expressiveness in music performance. *Proceedings of IEEE*, 92(4):686-701.

Canazza, S., De Poli, G., Rodà, A., & Vidolin, A. (2003). An abstract control space for communication of sensory expressive intentions in music performance. *Journal of New Music Research*, 32(3), 281-294.

Canazza, S., De Poli, G., Rodà, A., & Vidolin, A. (in Press.). Expressiveness in music performance: analysis, models, mapping, encoding. In Jacques Steyn editor, *Structuring Music through Markup Language: Designs and Architectures*.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias ,S., Fellenz, W., & Taylor, J.G. (2001). Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 32-80.

De Poli, G., Rodà, A. & Vidolin, A. (1998). Note-by-Note Analysis of the influence of Expressive Intentions and Musical Structure in Violin performance. *Journal of New Music Research*, 27, 293-321.

Dixon, S., Goebl W., & Widmer, G. (2005). The “air worm”: an interface for real-time manipulation of expressive music performance. *Proceedings of the 2005 International Computer Music Conference (ICMC2005*), Barcelona, Spain, 614-617.

Friberg, A. (1991). Generative rules for music performance. *Computer Music Journal,* 15(2), 56-71.

Friberg, A. (1995). A Quantitative Rule System for Musical Expression. Doctoral dissertation, Royal Institute of Technology, Sweden.

Friberg, A., Colombo, V., Frydén, L., & Sundberg, J. (2000). Generating musical performances with director musices. *Computer Music Journal*, 24(3), 23-29.

Friberg, A. (2004). A fuzzy analyzer of emotional expression in music performance and body motion. In J. Sundberg & B. Brunson editors. *Proceedings of Music and Music Science*, Stockholm, October 28- 30, 2004.

Friberg, A., Bresin, R., & Sundberg, J. (2006). Overview of the KTH rule system for musical performance. Advances in Experimental Psychology, special issue on Music Performance.

Gabrielsson, A. (1987). Once again: the theme from Mozart's piano sonata in A major (K. 331). A comparison of five performances. In A. Gabrielsson editor, *Action and Perception in Rhythm and Music*, Stockholm: Royal Swedish Academy of Music, Publication No. 55, 81-103.

Gabrielsson, A. & Juslin, P. N. (1996). Emotional expression in music performance: between the performer's intention and the listener's experience. *Psychology of music*, 24, 68-91.

Haus, G., & Longari, M. (2002). Towards a Symbolic/Time-Based Music language based on XML. *Proc. of MAX 2002 International Conference,* Milan, September 19-20.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan: University of Michigan Press.

Imberty, M. (1986). *Suoni, emozioni, significati. Per una semantica psicologica della musica.* Bologna: CLUEB.

Juslin, P.N. (2000). Cue utilization in communication of emotion in music performance: relating performance to perception. *Journal of Experimental Psychology: Human perception and performance*, 26(6), 1797-1813.

Juslin, P.N., & Sloboda, J.A. (2001). *Music and emotion. Theory and research*. Oxford University Press.

Klapuri, A. (1999). Sound Onset Detection by Applying Psychoacoustic Knowledge. *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 3089-3092.

Lerdahl, F., & Jackendoff, R. (1993). An overview of hierarchical structure in music. *Music Perception: An Interdisciplinary Journal,* 1(2):229-252.

Maher, R. C., & Beauchamp, J. W. (1994). Fundamental Frequency Estimation of Musical Signals Using a Two-Way Mismatch Procedure. *Journal of the Acoustical Society of America,* 95(4):2254-2263.

Michalski, R. S. (1969). On the Quasi-Minimal Solution of the General Covering Problem. *Proceedings of the First International Symposium on Information Processing,* Bled, Yugoslavia: N.P., 125-128.

Narmour, E. (1990). *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press.

Pielemeier, W. J., Wakefield, G. H. & Simoni, M. H. (1996). Time-frequency analysis of musical signals. *Proc. IEEE*, 84, 1216-1230.

Plutchik, R. (1980). *Emotion: A Psychoevolutionary Synthesis*. New York: Harper & Row.

Plutchik, R. (1994). *The psychology and biology of emotions*. New York: Harper-Collins.

Ramirez, R., Hazan A., Maestre E., & Serra X. (2008). A Genetic Rule-based Expressive Performance Model for Jazz Saxophone. *Computer Music Journal*, 32, 38-50.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39, 1161-1178.

Taizan, S., Tatsuya, H., Masahiro S., & Yukio, T. (2011). Kagurame Phase-II and Kagurame Phase-III. In *Proc. of Rencon.*

Takashi, B., Mitsuyo, H., & Haruhiro, K. (2011). “VirtualPhilharmony”: A conducting system focused on a sensation of conducting an orchestra. In *Proc. of Rencon.*

Todd, N. P. McA. (1985). A model of expressive timing in tonal music. *Music Perception*, 3, 33-58.

Todd, N. P. McA. (1989). A computational model of rubato. *Contemporary Music Review*, 3, 69-88.

Widmer, G. (1996). Learning expressive performance: The structure-level approach. *Journal of New Music Research*, 25(2), 179-205.

Widmer, G. (2002). Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research*, 31, 37-50.

Widmer, G., Flossmann, S., and Grachten, M. (2009). YQX Plays Chopin. *AI Magazine*, 30(3), 35-48.

Widmer, G., Flossmann, & S., Grachten, M., (2011). Expressive Performance with Bayesian Networks and Linear Basis Models. In *Proc. of Rencon.*