Tese de Doutorado
Movimentos de Cabeça guiados pela Voz

Anderson Mayrink da Cunha
IMPA

09 de outubro de 2009

Sumário

Lista de Figuras

À Clarissa Luz Bittencourt

Resumo

O objetivo dessa tese é sintetizar movimentos de cabeça (crânio) guiados pelo som. Para realizar nosso intento, utilizamos modelos estatísticos que são treinados a partir de um banco de dados de movimentos reais de cabeça e de seus sons correspondentes. Redes Bayesianas e HMM (Hidden Markov Models) são os modelos estatísticos utilizados nesse trabalho. Construímos um modelo em dois níveis, onde no primeiro nível descobrimos independentemente os padrões básicos de movimentos de cabeça e de som e no segundo nível relacionamos os padrões básicos de movimento e de som obtidos no primeiro nível do modelo. Um novo movimento de cabeça pode ser sintetizado a partir do modelo treinado e de um som de entrada. Neste texto também apresentamos um resumo de modelos estatísticos e de artigos que serviram de inspiração para esse trabalho.

Agradecimentos

Qualquer trabalho necessita de muito apoio e ajuda. Listar todos que nos auxiliaram é difícil. Ficam meus especiais agradecimentos às seguintes pessoas, sem as quais a conclusão dessa tese seria impossível.

Ao professor Luiz Velho, meu orientador e fonte de inspiração nessa pesquisa. Esta tese não estaria pronta se não fossem sua paciência, idéias e estímulo.

Ao professor Paulo Cezar Carvalho, que ministrou o primeiro curso que assisti no IMPA, há mais de dez anos atrás, e que ao longo do tempo me deu a oportunidade de assistir a vários outros excelentes cursos.

A todos os professores que me auxiliaram ao longo dos anos. A eles devo tudo o que sei.

A todos os colegas e amigos do IMPA que me ajudaram nessa longa e muitas vezes difícil jornada do doutorado.

Ao IMPA e ao CNPq pelo suporte financeiro recebido ao longo desses anos.

Aos meus pais, que sempre me apoiaram, tanto emocionalmente quanto financeiramente, mesmo nos momentos mais difíceis.

Ao pequeno João, que me dá o estímulo para percorrer o meu caminho.

Capítulo 1

Introdução

Um sistema automático que permita a síntese de movimentos de cabeça guiados pela voz pode ser bastante útil em várias aplicações que explorem a interatividade entre o computador e o usuário. Uma outra aplicação é na área de animação, onde os animadores tradicionais em geral começam criando o movimento de cabeça e depois fazem outros detalhes como o sincronismo de lábios com a voz e outras expressões faciais [Chuang 05].

Como apontado por Busso [Busso 07], apesar do movimento de cabeça ser parte importante da linguagem corporal, ele tem recebido pouca atenção comparado à outros sinais não verbais. Alguns estudos mostram que o movimento de cabeça melhora a capacidade de reconhecer a voz e ajuda a diferenciar entre declarações afirmativas e interrogativas [Munhall 04], assim como ajuda no reconhecimento de sexo e identidade [Hill 01].

Existem estudos que indicam uma correlação significativa entre a frequência fundamental $F0$ da voz e o movimento de cabeça, havendo inclusive evidências anatômicas dessa relação [Yehia 98], [Yehia 02]. Apesar disso não há uma relação tão clara entre voz e movimento da cabeça como há entre fonemas e visemas (aparência visual da boca). Isso indica que a síntese do movimento da cabeça tem diferentes desafios que o da animação da face, que pode ser implementada por métodos estatísticos mais simples.

A prosódia acústica[1] e sua relação com a prosódia visual, que ainda não são bem compreendidas, são fatores importantes para se entender a relação entre a voz e o movimento de cabeça. Esse é um fator de dificuldade para que a síntese da cabeça

[1] A prosódia comumente é definida como o estudo do ritmo, entonação e demais atributos correlatos na fala. Ela descreve todas as propriedades acústicas da fala que não podem ser preditas pela transcrição ortográfica.

guiada pelo som seja de fato realista e efetiva. Outro fator que dificulta a nossa tarefa é a alta capacidade humana para discernir movimentos faciais e de cabeça, o que torna pequenos erros na síntese facilmente perceptíveis.

A emoção do locutor também é um fator importante para a naturalidade de uma animação. Experimentos indicam que movimentos de cabeça em estados emocionais de felicidade ou raiva têm o dobro da velocidade do que num estado neutro [Busso 07]. Isso indica a necessidade de modelos específicos para diferentes estados emocionais. Humanos também são particularmente bons em inferir estados emocionais de outras pessoas, o que é mais um desafio para a bem sucedida síntese de movimentos de cabeça.

Um modo para gerar movimentos de cabeça é a partir de regras baseadas em comportamento, como em [Cassell 94]. Um problema desse tipo de técnica é o excesso de especificações e parâmetros manuais que a tornam uma técnica complexa e "ad hoc".

Nossa proposta é gerar movimentos realistas de cabeça guiados pelo som de forma automática, com um sistema flexível, geral e com poucos parâmetros manuais. Desejamos ainda que seja um método que seja guiado por dados (data driven), isto é, desejamos treinar nosso modelo a partir de dados reais de voz e movimento de cabeça. Para isso vamos fazer uso de modelos estatísticos, que é um modo mais atual para atacar esse problema. Mais especificamente, vamos empregar métodos baseados em redes bayesianas, que é um modelo bem posto matematicamente e que já mostrou bons resultados em vários problemas de aprendizagem de máquina e inteligência artificial [Murphy 02], [Russell 02].

Contribuições da Tese

Esta tese tem duas principais contribuições ao estudo da síntese de movimentos de cabeça guiados pelo som ou, de maneira mais geral, ao estudo das relações entre sinais de diferentes origens, mas que guardam algum tipo de ligação entre eles.

- Um modo de relacionar sinais com pontos de segmentação diferentes a partir de um modelo probabilístico que modela uma relação entre os sinais com seus diferentes pontos de segmentação, possibilitando uma posterior síntese de um deles, conhecido o outro sinal.

- Um modelo flexível que permite vários tipos de expansões, não só acrescentando níveis de padrões (ou complexidade) ao modelo, como também permitindo o uso de diferentes sub-modelos e métodos para modelagem e síntese.

Organização da Tese

O texto da tese é dividido em três capítulos principais. O capítulo 2 trata de métodos estatísticos, particularmente de redes bayesianas e HMM. Esse capítulo é uma tentativa de ser uma introdução ao importante tópico de redes bayesianas e tem o objetivo de tornar a tese mais auto contida e também possibilitar que a descrição do método, no capítulo 4, fique mais enxuta.

No capítulo 3 descrevemos brevemente alguns artigos que foram inspirações para esse trabalho, mencionando suas principais contribuições. Por fim, indicamos quais os métodos e idéias desses artigos que foram aproveitados no nosso trabalho.

No capítulo 4 apresentamos o método proposto, o modelo em dois níveis utilizado e descrevemos o seu treinamento e síntese. Tecemos algumas considerações sobre a captura, análise e segmentação dos dados. No final do capítulo 4 apresentamos alguns resultados e aplicações do método.

O texto é fechado com as conclusões e possíveis trabalhos futuros que podem enriquecer o nosso trabalho.

Capítulo 2

Modelos Estatísticos

Vários campos de conhecimento têm aplicado métodos probabilísticos com muito sucesso. Nas áreas em que se tenta simular a percepção ou o comportamento humano, como por exemplo, reconhecimento de voz e escrita, métodos estatísticos são onipresentes. Pesquisas atuais em síntese de movimentos do corpo humano, em particular da cabeça, normalmente empregam métodos estatísticos. Na tentativa de tornar essa tese mais auto contida, vamos apresentar uma introdução a modelos estatísticos e em especial à redes bayesianas, que é a base para o nosso modelo.

Nesse capítulo vamos apresentar algumas propriedades básicas de probabilidade e estatística e também alguns modelos estatísticos úteis no nosso trabalho como Cadeias de Markov, Hidden Markov Models (HMM) e Redes Bayesianas, que é um modelo elegante e de grande utilidade prática e que é o caso geral de vários modelos importantes, entre eles a Cadeia de Markov e HMM. Apresentamos também algumas aplicações como reconhecimento de escrita e voz, que são muito importantes por si só e ajudam no entendimento de HMM, redes bayesianas e suas aplicações em síntese de movimentos.

2.1 Variável Aleatória e Processo Estocástico

Intuitivamente, uma **Variável Aleatória** (ou randômica) é uma variável cujos possíveis valores são os resultados numéricos de um experimento aleatório. Uma variável é um atributo ou uma medida que pode ser tomada entre várias possíveis saídas ou valores de um domínio especificado. Se tivermos crenças (probabilidades) sobre quais os possíveis valores para a variável, ela é chamada de variável aleatória.

Uma variável aleatória é uma função que mapeia eventos de um fenômeno aleatório em números, ou mais formalmente: Seja um espaço de probabilidade $(\Omega, \mathcal{A}, P)$ onde

Ω é o espaço amostral, $\mathcal{A}$ é uma σ-álgebra de subconjuntos de Ω e P uma medida que satisfaz aos axiomas de probabilidade [James 81]. Uma **Variável Aleatória** no espaço de probabilidade $(\Omega, \mathcal{A}, P)$ é uma função real definida em Ω e mensurável em $\mathcal{A}$. Maiores detalhes podem ser encontrados em [James 81], [Applebaum 96] ou [Russell 02].

A uma variável aleatória X está associada a uma distribuição de probabilidade $P(X)$, que mede o quão prováveis são os possíveis valores de X. Uma variável aleatória pode ser contínua ou discreta, de acordo com seu espaço amostral. Se o espaço amostral é discreto, podemos representar $P(X)$ de forma tabular. No caso de um espaço amostral contínuo, um exemplo importante é a distribuição Normal (gaussiana) com parâmetros μ (média) e U (matriz de covariância), cuja distribuição de probabilidade em $\mathbb{R}^d$ é definida por:

$$\mathcal{N}(X, \mu, U) = \frac{1}{(2\pi)^{d/2}|U|^{1/2}} \exp(-\frac{1}{2}(X-\mu)^T U^{-1}(X-\mu))$$

Uma distribuição mais flexível é a Mistura de gaussianas, que consiste na soma ponderada de gaussianas:

$$M(X, c_1, \ldots, c_M, \mu_1, \ldots, \mu_M, U_1, \ldots, U_M) = \sum_{m=1}^{M} c_m \mathcal{N}(X, \mu_m, U_m)$$

onde c_m é o coeficiente da m-ésima mistura, com $\sum c_m = 1$ e $\mathcal{N}$ é a função gaussiana (ou distribuição normal) multidimensional.

Um **Processo Estocástico** (ou randômico) é uma família de variáveis aleatórias $X(t)$ (ou X_t) indexadas por um parâmetro de tempo t. Os possíveis valores de X_t são chamados de estados do processo no tempo t. O conjunto de todos os estados possíveis para todo tempo t é chamado de **espaço de estados**.

Um processo estocástico é útil quando modelamos fenômenos que variam ao longo do tempo. Um dos casos mais simples e importantes de processos estocásticos são as **Séries Temporais**, que são sequências de variáveis aleatórias X_t, com $t = 1, 2, 3 \ldots$.

2.2 Probabilidade Condicional e Independência

A **Probabilidade Condicional** de um evento A dado que o evento B ocorreu é definida por:

$$P(A \mid B) = \frac{P(A.B)}{P(B)}, \text{ se } P(B) \neq 0 \text{ e onde } A.B = A \cap B$$

Segundo os estatísticos clássicos (ou frequentistas), a probabilidade (condicional) possui uma interpretação intuitiva em termos de frequências relativas:

$$P(A \mid B) = \frac{P(A.B)}{P(B)} = \frac{\lim \frac{1}{n}(\text{ocorrências de } A \cap B)}{\lim \frac{1}{n}(\text{ocorrências de } B)} = \lim \frac{\text{ocorrências de } A \cap B}{\text{ocorrências de } B}$$

Estatísticos bayesianos vêem o evento condicional $A|B$ como mais básico que o evento conjunto $A.B$ pois é mais compatível com o conhecimento humano. Nesse ponto de vista B indica o contexto e $A|B$ indica o evento A no contexto especificado por B. O conhecimento empírico vai sempre usar probabilidade condicional e a crença em eventos conjuntos $A.B$, se necessário, vai ser calculada a partir da probabilidade condicional.

Para estatísticos bayesianos (ou subjetivos) a probabilidade mede a crença em meu conhecimento do mundo. Não é importante se uma variável é ou não aleatória, mas sim que há incerteza no meu conhecimento. Segundo estatísticos bayesianos, a base para inferir probabilidades ou crenças é o famoso **Teorema de Bayes**:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Do Teorema de Bayes temos que a nossa crença em A dado a evidência B pode ser calculada a partir da nossa prévia crença na hipótese A ($P(A)$) e da probabilidade do evento B dado que a hipótese A é verdadeira ($P(B|A)$). $P(B)$ muitas vezes não é levada em consideração no cálculo pois é constante com respeito à variável desconhecida.

Uma fórmula útil quando não conhecemos a probabilidade de um evento B, mas conhecemos sua distribuição condicional em uma partição do espaço é o **Teorema da Probabilidade Total**:

$$P(B) = \sum_i P(A_i B) = \sum_i P(B \mid A_i)P(A_i)$$

onde A_i é uma partição do espaço amostral Ω (ie: $\cup A_i = \Omega$ e $A_i.A_j = A_i \cap A_j = \emptyset$).

Sejam A, B e C variáveis aleatórias. A é **independente** de B (denotamos $A \perp B$) se, por definição:

$$P(A.B) = P(A).P(B)$$

Uma definição equivalente de **Independência** é:

$$A \perp B \iff P(A|B) = P(A)$$

isto é, o conhecimento de B não altera a probabilidade de A.

A é **condicionalmente independente** de B dado C (denotamos $A \perp B|C$) se, por definição:

$$P(A.B|C) = P(A|C).P(B|C)$$

Uma definição equivalente de **Independência Condicional** é:

$$A \perp B|C \iff P(A|B.C) = P(A|C)$$

o que indica que conhecido C, B é irrelevante para o cálculo de A.

Em geral quando é dito que A só depende de B quer dizer que A e todas as outras variáveis aleatórias do problema são condicionalmente independentes, conhecido B.

2.3 Classificação e Treinamento

Seja X um conjunto de variáveis aleatórias e seja um modelo probabilístico paramétrico totalmente descrito pelo conjunto de parâmetros θ e onde podemos calcular a distribuição conjunta $P(X|\theta)$, a probabilidade de ocorrência de X nesse modelo.

Dois problemas básicos e assemelhados em modelos probabilísticos são a classificação de padrões e o treinamento. O problema de classificação consiste em decidir dentre os vários modelos θ_i qual gerou o dado de entrada, a observação X. Já no problema de treinamento devemos estimar os parametros θ a partir da observação X.

Há dois métodos principais de classificação ou treinamento: Máxima Verossimilhança e Máximo a Posteriori.

2.3.1 Máxima Verossimilhança (Maximum Likelihood)

Classificação

O problema de classificação de padrões consiste em decidir dentre os vários modelos θ_i qual tem maior probabilidade de ocorrência da observação X, isto é, escolhemos o modelo θ_i tal que:

$$\theta^* = \arg\max_i \quad P(\mathbf{X}|\theta_i)$$

Treinamento

Estimamos os parâmetros θ do modelo como aqueles que nos fornecem o máximo da função de verossimilhança $\mathcal{L}(\theta) = P(\mathbf{X}|\theta)$, isto é:

$$\theta^* = \arg\max_\theta \quad \mathcal{L}(\theta) = \arg\max_\theta \quad P(\mathbf{X}|\theta)$$

θ^* é chamado de estimador de máxima verossimilhança (maximum likelihood estimator) de θ.

2.3.2 Máximo a Posteriori

Nesta solução para o problema, sugerida por estatísticos bayesianos, devemos levar em conta a distribuição dos parâmetros $P(\theta)$. A forma do Teorema de Bayes para aprendizagem é:

$$P(modelo \mid dado) = \frac{P(dado \mid modelo) \cdot P(modelo)}{P(dado)}$$

onde comumente chamamos $P(modelo \mid dado)$ de distribuição à posteriori, $P(dado \mid modelo)$ de distribuição de verossimilhança, $P(modelo)$ de distribuição à priori e $P(dado)$ de evidência, que em geral não é levada em consideração no cálculo pois é constante com respeito à variável desconhecida.

Classificação

Devemos achar qual o modelo θ_i que tem maior probabilidade condicionada a X, isto é:

$$\theta^* = \arg\max_i \quad P(\theta_i|\mathbf{X})$$

aplicando o teorema de Bayes e ignorando o fator comum $P(\mathbf{X})$, temos que:

$$\theta^* = \arg\max_i \quad P(\mathbf{X}|\theta_i) \cdot P(\theta_i)$$

Essa equação também é conhecida como o classificador ótimo de Bayes. Note que se os $P(\theta_i)$ são iguais, o classificador de Bayes se reduz ao classificador de máxima verossimilhança.

Treinamento

No treinamento temos um cálculo parecido:

$$\theta^* = \arg\max_\theta \quad P(\theta|\mathbf{X}) = \arg\max_\theta \quad P(\mathbf{X}|\theta).P(\theta)$$

2.4 Cadeias de Markov

Em 1907, o matemático russo Markov investigou propriedades de processos que são hoje conhecidos como processos de Markov. A principal característica dos processos de Markov é que o modo que toda a história passada afeta o futuro está completamente resumida no presente, ou ainda, o futuro e o passado são condicionalmente independentes conhecido o presente.

Seja $t = \{1, 2, ..., T\}$ e X_t um processo estocástico com tempo discreto que assume valores de um conjunto finito $\mathcal{S}$, chamado espaço de estados. A cada tempo t é observado o estado particular $X_t = s_t \in \mathcal{S}$.

X_t é uma **Cadeia de Markov** se, por definição:

$$P(X_t = s_t \mid X_1 = s_1, X_2 = s_2, \ldots, X_{t-1} = s_{t-1}) = P(X_t = s_t \mid X_{t-1} = s_{t-1})$$

que é uma condição equivalente a X_t e $X_1, X_2, \ldots, X_{t-2}$ serem condicionalmente independentes dado X_{t-1}:

$$X_t \perp X_1, X_2, \ldots, X_{t-2} | X_{t-1}$$

Isso quer dizer que se observamos $X_{t-1} = s_{t-1}$, é indiferente para X_t se observamos ou não todo o passado $X_1 = s_1, \ldots, X_{t-2} = s_{t-2}$, isto é, X_t depende somente de observarmos $X_{t-1} = s_{t-1}$ e não de observarmos todo o passado $X_1 = s_1, X_2 = s_2, \ldots, X_{t-1} = s_{t-1}$.

Assumimos comumente que a cadeia de Markov é homogênea, isto é, a probabilidade condicional $P(X_t = s_j \mid X_{t-1} = s_i)$ é independente do tempo t. Nesse caso temos que $a_{ij} = P(X_t = s_j \mid X_{t-1} = s_i)$ é independente de t, logo podemos organizar os valores a_{ij} numa matriz de transição de probabilidades.

Uma cadeia de Markov homogênea fica totalmente determinada pelos seus parâmetros: a matriz de transição $A = \{a_{ij} = P(X_t = s_j \mid X_{t-1} = s_i)\}$ e o vetor de probabilidades iniciais $\pi = \{\pi_i = P(X_1 = s_i)\}$.

Exemplo 1 *Modelagem simples de chuva*

Uma modelagem simples da ocorrência de chuva pode ser feita com a cadeia de Markov. Supomos que $X_t = 1$ se choveu no dia t e $X_t = 0$, caso contrário. O espaço de estados é de tamanho 2. A matriz de transição A e o vetor de probabilidades iniciais π podem, por exemplo, ser da forma:

$$A = \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} \text{ e } \pi = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$$

Das matrizes A e π vemos que a probabilidade de chover amanhã, dado que choveu hoje é $a_{2,2} = 0.7$ e que a probabilidade de não chover no dia inicial é $\pi_1 = 0.7$.

Exemplo 2 *Modelagem de Linguagem*

Apesar da simplicidade na modelagem com cadeias de Markov, muitas vezes resultados surpreendentes são obtidos. Um exemplo clássico é o modelagem de linguagem do famoso artigo de Shannon [Shannon 48].

Neste exemplo temos 27 símbolos (26 letras e um espaço). Tomando um texto (grande) calculamos as frequências relativas da ocorrência dos símbolos, levando em conta, quando necessário, quais os símbolos anteriores[1]. Shannon construiu vários modelos para o idioma inglês em ordem crescente de complexidade e dá exemplos de síntese nesses modelos. Note que com o aumento da complexidade do modelo, maior a semelhança do exemplo sintetizado com o idioma inglês.

1. Símbolos independentes e equiprováveis

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD.

2. Unigrama - frequências dos símbolos é a do idioma inglês

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Cadeia de Markov (de primeira ordem), também chamado de bigrama, pois leva em conta a frequência dos símbolos e a frequência da ocorrência de dois símbolos consecutivos.

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Cadeia de Markov de segunda ordem[2], também chamado de trigrama, pois leva em conta a frequência de até três símbolos consecutivos.

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

5. Modelando não a partir das letras, mas a partir das palavras e levando em conta as frequências das palavras do idioma inglês.

[1] O treinamento da matriz de transição de uma cadeia de Markov pelo método de máxima verosimilhança é feito contando sua ocorrência (frequência relativa) no texto de treinamento.

[2] Uma cadeia de Markov de segunda ordem é uma cadeia de Markov (de primeira ordem), bastando para isso criar mega-estados consistindo de dois estados consecutivos.

REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TOOF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Cadeia de Markov (de primeira ordem) com palavras do inglês

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

2.5 Hidden Markov Models (HMM)

Um Modelo Oculto de Markov (Hidden Markov Model - **HMM**) é composto de dois processos estocásticos. O primeiro é uma cadeia de Markov X_t, que é chamado de estado oculto, pois em geral X_t não é observado. O segundo processo estocástico Y_t é gerado somente a partir de X_t para cada t.

Em um HMM, a distribuição conjunta de $\{X_t, Y_t\}$, com $t = \{1, 2, \ldots, T\}$, pode ser calculada da seguinte forma:

$$P(\{X_t, Y_t\}) = P(X_1) \cdot P(Y_1|X_1) \cdot \prod_{t=2}^{T} P(X_t|X_{t-1}) \cdot P(Y_t|X_t)$$

Um HMM também pode ser definido em termos de independência condicional:

$$\begin{cases} X_t \text{ é uma cadeia de Markov, isto é, } X_t \perp X_1, X_2, \ldots, X_{t-2}|X_{t-1} \text{ e} \\ Y_t \text{ depende somente de } X_t, \text{ ie, } Y_t \perp X_i, Y_i|X_t,\ i = \{1, 2, \ldots, t-1, t+1, \ldots, T\} \end{cases}$$

Num HMM, o espaço de estados (valores possíveis para X) é sempre discreto e finito. Já o espaço das observações (valores possíveis para Y) pode ser contínuo ou discreto.

Em um HMM discreto (quando o espaço das observações é discreto) temos que o modelo do sensor $P(Y_t|X_t) = B$, onde B é uma matriz cujo elemento $b_{i,j}$ indica qual a probabilidade da emissão da observação Y_j para o estado oculto X_i. Já no caso de HMM contínuo, é comum que $P(Y_t|X_t)$ seja a distribuição normal ou uma mistura de gaussianas.

Um HMM fica totalmente determinado pelos parâmetros $\theta = \{A, B, \pi\}$, onde B é o modelo do sensor e A e π são, respectivamente, a matriz de transição e o vetor de probabilidades iniciais da cadeia de Markov em X.

HMM tem tido muito sucesso em reconhecimento de padrões, como por exemplo, para modelar as características naturais de um sinal de voz ou de escrita à mão.

Exemplo 3 *Clima e Guarda-Chuva*

Vamos descrever um exemplo simples de HMM. Considere que a chuva pode ser modelada por uma cadeia de Markov X_t, como no exemplo 1. Vamos supor que não sabemos se choveu, mas observamos se uma dada pessoa usa o guarda-chuva (processo Y_t). Supomos que a probabilidade da pessoa usar o guarda-chuva no dia t depende somente de observar-mos se está ou não chovendo no dia t. Vários questionamentos podem surgir desse problema. Por exemplo, se observamos Y_t para $t = 1, \ldots, T$, podemos perguntar:

1. Qual a probabilidade da ocorrência de Y neste HMM?
2. Qual a sequência X de estados ocultos mais prováveis?
3. Filtragem: Qual o valor mais provável para X_{t+1}?
4. Treinamento: Quais os parâmetros do HMM que maximizam a probabilidade de ocorrência de Y?

Na próxima seção vamos ver como fazer alguns desses cálculos.

2.5.1 Cálculos em um HMM

Algoritmos de HMM discreto podem ser descritos de maneira elegante em versão matricial [Russell 02]. Nesse texto vamos apresentar a versão mais conhecida dos algoritmos, conforme descrita no clássico [Rabiner 89] ou em [Nechyba 00], [Rabiner 93], [Dugad 96], [Cunha 02a]. Nesta seção vamos usar a notação de [Rabiner 89].

Vamos tratar aqui somente de HMM com número de estados e de unidades de tempo (tamanho da observação) finitos. O sinal O emitido em cada estado pode assumir os valores definidos pelo alfabeto, que também assumimos como sendo finito. Na última subseção apresentamos brevemente o caso do sinal O ser contínuo.

Um outro exemplo de HMM é uma partícula que para cada $t = 1, 2, \ldots, T$ muda para um local (ou estado) entre os N possíveis. Em cada um dos estados, a partícula emite um sinal (de um alfabeto de tamanho M). Temos que o estado futuro da partícula depende somente do estado atual, e não dos estados anteriores ou do tempo t. Não conhecemos os estados da partícula, mas somente os sinais emitidos por ela. A seguir temos algumas notações.

- N é o tamanho do espaço de estados do modelo, ou locais onde a partícula pode ir. Para simplificar a notação denominamos o conjunto de estados $\mathcal{S} = \{1, 2, \ldots, N\}$.

- M é o número total de símbolos distintos, o tamanho do alfabeto de sinais que a partícula emite. $V = \{v_1, v_2, \ldots, v_M\}$ é o alfabeto.

- T é a quantidade de unidades de tempo (tamanho da observação).

- $Q = \{q_1, q_2, \ldots, q_T\}$ onde q_t é o estado do modelo no tempo t.

- $O = \{O_1, O_2, \ldots, O_T\}$ onde O_t é símbolo observado no tempo t.

- $\pi = \{\pi_i\}, i = 1, \ldots, N.$ onde $\pi_i = P(q_1 = i)$ é a probabilidade de i ser o estado inicial do experimento.

- $A = \{a_{ij}\}$ é uma matriz $N \times N$, onde $a_{ij} = P(q_{t+1} = j \mid q_t = i)$ é a probabilidade da partícula ir do estado i para o estado j. Os a_{ij} são independentes do tempo t (matriz homogênea).

- $B = \{b_i(k)\}$ é uma matriz $N \times M$, onde $b_i(k)$ é a probabilidade do símbolo v_k ser observado no estado i.

- $\lambda = (A, B, \pi)$ é a notação compacta dos parâmetros de um HMM.

Nas próximas subseções apresentamos algoritmos para os três principais problemas de HMM que são calcular:

1. $P(O \mid \lambda)$, a probabilidade da ocorrência da observação O dado o modelo $\lambda = (A, B, \pi)$.

2. $\arg\max_{Q} P(Q \mid O, \lambda)$, a sequência de estados mais provável dado a observação O e o modelo λ.

3. $\arg\max_{\lambda} P(\lambda \mid O)$, quais os parâmetros ótimos do modelo λ, conhecido o dado de treinamento O.

Cálculo de $P(O \mid \lambda)$

- *Dado um modelo $\lambda = (A, B, \pi)$ como calcular $P(O \mid \lambda)$, a probabilidade da ocorrência da observação $O_1, O_2, \ldots, O_T$?*

O modo mais imediato de se calcular $P(O \mid \lambda)$ é achar $P(O \mid Q, \lambda)$ para um estado fixado $Q = q_1, q_2, \ldots, q_T$ e somar as probabilidades sobre todos os estados possíveis, isto é:

$$P(O \mid \lambda) = \sum_{Q} P(O, Q \mid \lambda) = \sum_{Q} P(O \mid Q, \lambda) P(Q, \lambda)$$

onde $P(O \mid Q, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \cdots b_{q_T}(O_T)$ e $P(Q, \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$.

Apesar do cálculo acima ser um procedimento simples, ele é muito caro computacionalmente, pois existem N^T sequências de estados possíveis, o que nos leva a um algoritmo com ordem de $T.N^T$ multiplicações para o cálculo de $P(O \mid \lambda)$. Isso é inviável computacionalmente pois, por exemplo, um modelo com 10 estados e 100 instantes de tempo (observações), isto é, $N = 10, T = 100$, temos ordem de 10^{102} multiplicações. Para executar esse cálculo mais rapidamente usamos o procedimento forward descrito a seguir.

Algoritmo Forward A variável forward $\alpha_t(i)$ é definida como a probabilidade da observação da sequência parcial $O_1, O_2, \ldots, O_t$ e que no tempo t tenhamos o estado i.

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, q_t = i \mid \lambda)$$

$\alpha_t(i)$ pode ser calculada recusirvamente da seguinte forma:

$$\begin{cases} \alpha_1(i) = \pi_i b_i(O_1) & i = 1, \ldots, N \\ \alpha_{t+1}(j) = b_j(O_{t+1}) \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) \quad t = 1, 2, \ldots, T-1 & , \quad j = 1, \ldots, N \end{cases}$$

A probabilidade $P(O \mid \lambda)$ pode ser calculada facilmente com a variável forward:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Esse procedimento para o cálculo de $P(O \mid \lambda)$ envolve ordem de $N^2 \times T$ multiplicações, o que é bem eficiente, sendo um algoritmo linear em t.

A dedução da fórmula $\alpha_{t+1}(j) = b_j(O_{t+1})\left(\sum_{i=1}^{N}\alpha_t(i)\,a_{ij}\right)$ não é difícil:

$\alpha_{t+1}(j) = P(O_1,\ldots,O_t,O_{t+1},q_{t+1}=j \mid \lambda) =$

$= \sum_{i=1}^{N} P(O_1,\ldots,O_t,O_{t+1},q_{t+1}=j \mid q_t=i,\lambda).P(q_t=i \mid \lambda) =$

$= \sum_{i=1}^{N} P(O_1,O_2,\ldots,O_t \mid q_t=i,\lambda).P(q_t=i \mid \lambda).P(O_{t+1},q_{t+1}=j \mid q_t=i,\lambda) =$

$= \sum_{i=1}^{N} P(O_1,\ldots,O_t,q_t=i \mid \lambda).P(O_{t+1} \mid q_{t+1}=j,q_t=i,\lambda).P(q_{t+1}=j \mid q_t=i,\lambda) =$

$= P(O_{t+1} \mid q_{t+1}=j,\lambda).\sum_{i=1}^{N} P(O_1,\ldots,O_t,q_t=i \mid \lambda).P(q_{t+1}=j \mid q_t=i,\lambda) =$

$= b_j(O_{t+1}).\left(\sum_{i=1}^{N}\alpha_t(i)\,a_{ij}\right)$

Algoritmo Forward Escalado O cálculo da variável forward com o algoritmo descrito na subseção anterior envolve o problema de precisão numérica (underflow) pois, por exemplo, se temos um alfabeto de tamanho 100 e 100 observações ($M = T = 100$), α_{t+1} é de ordem 100 vezes menor que α_t. Daí temos que $P(O \mid \lambda)$ é da ordem de 10^{-200}.

Para resolver este problema temos que escalar esse algoritmo. Esse escalamento consiste basicamente, a cada passo do algoritmo, em criar uma variável auxiliar que indica o valor de $\sum_i \alpha_t(i)$. Descrevemos a seguir o algoritmo forward escalado.

1. Inicialização:

$\widetilde{\alpha}_1(i) = \pi_i b_i(O_1), i = 1,\ldots,N$

$c_1 = 1/\left(\sum_{i=1}^{N}\widetilde{\alpha}_1(i)\right)$

$\widehat{\alpha}_1(i) = c_1\widetilde{\alpha}_1(i), i = 1,\ldots,N$ (escalado)

2. Indução

$\widetilde{\alpha}_{t+1}(j) = b_j(O_{t+1})\left(\sum_{i=1}^{N}\widehat{\alpha}_t(i)\,a_{ij}\right), t = 1,\ldots,T-1 \quad , \quad j = 1,\ldots,N$

$c_{t+1} = 1/\left(\sum_{i=1}^{N}\widetilde{\alpha}_{t+1}(i)\right), t = 1,\ldots,T-1$

$\widehat{\alpha}_{t+1}(j) = c_{t+1}\widetilde{\alpha}_{t+1}(i), t = 1,\ldots,T-1 \quad , \quad j = 1,\ldots,N$ (escalado)

3. Finalização

$$P(O \mid \lambda) = 1/\left(\prod_{t=1}^{T} c_t\right), \text{ pois } P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) = \underbrace{\sum_{i=1}^{N} \widehat{\alpha}_T(i)}_{=1} / \left(\prod_{t=1}^{T} c_t\right).$$

Como $P(O \mid \lambda)$ pode ser muito pequeno, calculamos o seu logaritmo:

$$\log P(O \mid \lambda) = -\sum_{t=1}^{T} \log c_t$$

Cálculo da Sequência Ótima de Estados

- *Dado um modelo $\lambda = (A, B, \pi)$ e a sequência de observação $O_1, O_2, \ldots, O_T$. Qual a sequência de estados $Q = (q_1, q_2, \ldots, q_T)$ para que $P(Q \mid O, \lambda)$ seja maximizada?, isto é,* $\arg\max_Q P(Q \mid O, \lambda) = ?$

Como $P(Q \mid O, \lambda) = \frac{P(Q,O|\lambda)}{P(O|\lambda)}$ e $P(O|\lambda)$ é constante em relação à seqüência de estados Q, basta calcular:

$$Q^* = \arg\max_Q P(Q, O \mid \lambda)$$

Para executar o cálculo acima de modo eficiente usamos o algoritmo de Viterbi, que é um algoritmo de programação dinâmica e é parecido com o algoritmo forward para o cálculo de $P(O \mid \lambda)$. Para o cálculo do algoritmo de Viterbi precisamos da variável auxiliar $\delta_t(i)$ definida por:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1, q_2, \ldots, q_t = i, O_1, \ldots, O_t \mid \lambda)$$

Podemos calcular $\delta_{t+1}(i)$ a partir da seguinte relação indutiva:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(O_{t+1})$$

onde $\delta_1(i) = P(q_1 = i, O_1|\lambda) = \pi_i b_i(O_1)$.

O valor de $\max_Q P(Q, O \mid \lambda)$ pode ser calculado com a relação:

$$P^* = \max_Q P(Q, O \mid \lambda) = \max_i [\delta_T(i)]$$

Temos ainda que traçar o caminho de volta para achar a seqüência de estados:

$$q_T^* = \arg\max_i [\delta_T(i)] \; e \; q_t^* = \arg\max_i [\delta_t(i) a_{iq_{t+1}^*}]$$

Como no caso do algoritmo forward, é necessário algumas modificações para o cálculo escalado do algoritmo de Viterbi, o que evita underflow em computadores de precisão finita. Para maiores detalhes consulte [Rabiner 89], [Nechyba 00] ou [Cunha 02a].

Treinamento de HMM

- *Quais os parâmetros* $\lambda = (A, B, \pi)$ *de um HMM para que* $P(O \mid \lambda)$ *seja maximizado?, isto é,* $\arg\max_\lambda P(O \mid \lambda) =$?

O treinamento de um modelo estatístico é de longe o mais difícil e a existência de um algoritmo eficiente para esse problema é condição fundamental para a aplicabilidade desse modelo estatístico em reconhecimento de padrões. Esse é o caso de HMM, pois existe o eficiente algoritmo de **Baum-Welch**, que é um caso particular do algoritmo **EM** (Expectation-Maximization).

O treinamento de HMM é um problema de otimização contínua de várias variáveis (os parâmetros $\lambda = (A, B, \pi)$) e pode ser resolvido aplicando métodos de otimização contínua não linear. Uma melhor opção é o algoritmo de Baum-Welch que consiste dos seguintes passos:

1. Inicialização: Obter parâmetros iniciais $\lambda_0 = (A_0, B_0, \pi_0)$ do modelo. Podemos fazer isso de modo aleatório ou usando algum tipo de estimativa para a inicialização do HMM.

2. Passo **E** (Expectation): Calcular as esperanças matemáticas das variáveis ocultas, dado λ. No caso de HMM as variáveis são: a ocupação dos estados (o número de vezes em cada estado), a transição dos estados e o número de vezes que o símbolo v_k foi emitido no estado j.

3. Passo **M** (Maximization): Calcular novos parâmetros $\hat{\lambda}$ para que $P(O \mid \hat{\lambda})$ seja maximizado, assumindo que os valores das variáveis ocultas são as esperanças calculadas no passo **E**. Se já conhecemos os valores dos estados ocultos (obtidos no passo **E**), o cálculo dos parâmetros λ do modelo para que $P(O \mid \lambda)$ seja máximo é simples, bastando calcular frequências relativas (análogo ao treinamento

de cadeias de Markov):

$$\begin{cases} a_{ij} = \frac{\text{número de transições do estado } i \text{ para o estado } j}{\text{número de vezes no estado } i} \\ b_j(k) = \frac{\text{número de vezes no estado } j \text{ com o símbolo } v_k}{\text{número de vezes no estado } j} \\ \pi_i = \text{número de vezes no estado } i \text{ no tempo } t = 1 \end{cases}$$

4. Iteração: Se $P(O \mid \hat{\lambda}) - P(O \mid \lambda) > \varepsilon$ (valor fixo), ir para o passo **E** (com os novos parâmetros $\hat{\lambda}$). Se não, terminar a execução e retornar os novos parâmetros $\hat{\lambda}$.

Pode-se provar que a cada passo $P(O \mid \hat{\lambda}) \geq P(O \mid \lambda)$ e que a sequência de modelos $\hat{\lambda}_i$ obtidos com o algoritmo EM converge para λ^*, um máximo local de $\arg\max_{\lambda} P(O \mid \lambda)$.

Apesar do algoritmo de Baum-Welch convergir localmente, mas não globalmente, para um máximo, o algoritmo de Baum-Welch é rápido, geralmente dá bons resultados na prática e é baseado em médias e conceitos probabilisticos, o que não é o caso de algoritmos gerais de otimização contínua não linear.

Para maiores detalhes e a dedução das fórmulas explícitas para os parâmetros do HMM, assim como o algoritmo de Baum-Welch escalado, consulte [Rabiner 89], [Nechyba 00] ou [Rabiner 93].

HMM para Observações Contínuas

Apesar de ser possível discretizar um sinal contínuo com um algoritmo de quantização vetorial, em geral uma melhor performance é obtida trabalhando diretamente com o sinal contínuo. Para que os cálculos em um HMM sejam eficientes, comumente definimos a função densidade de distribuição $b_i(O)$ como a distribuição normal (gaussiana) ou como uma soma (mistura) de gaussianas, o que é comum em reconhecimento de voz.

Os algoritmos para os 3 problemas de HMM contínuas são parecidos com o caso discreto. Para maiores detalhes, consulte [Rabiner 89], [Nechyba 00] ou [Rabiner 93].

Exemplo 4 *Reconhecimento de Dígitos*

Inspirado no sucesso de HMM na modelagem da percepção humana da voz [Rabiner 93], [Jelinek 97], [Cunha 03a], muitas aplicações utilizam HMM, principalmente na modelagem de séries temporais. HMM também tem atraído interesse em reconhecimento de escrita [Hu 96].

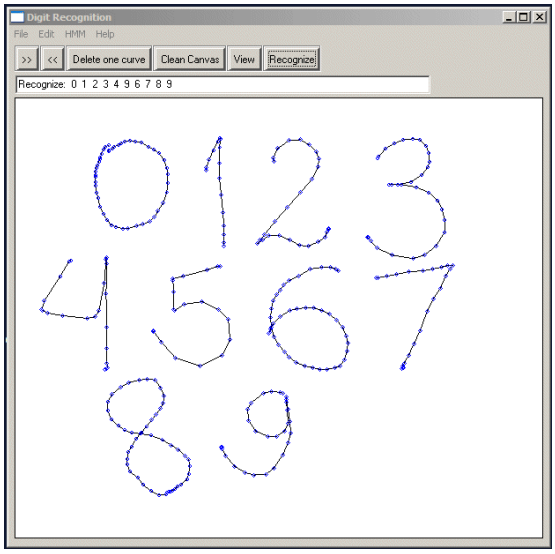

Figura 2.1: Programa de Reconhecimento de Dígitos

Vamos descrever um sistema bastante simples de reconhecimento de dígitos (de 0 a 9) baseado em HMM. Maiores detalhes podem ser encontrados em [Cunha 03b].

Exigimos que os dígitos sejam escritos em um único traço, de modo que sabemos exatamente onde começa e termina cada dígito. O nosso sinal de entrada para o reconhecimento de dígitos são sequências de pontos no plano (pontos assinalados em azul nos dígitos da figura 2.1) que foram gerados a partir do mouse ou da mesa digitalizadora (tablet), e que foram obtidas pela interface do sistema. Os valores das coordenadas x e y de cada ponto não são os valores mais relevantes para representar a escrita. O ângulo é um aspecto relevante (feature) comumente usado num sinal de escrita [Hu 96]. Com isso cada dígito vai ser representado por $O = O_1, O_2, \ldots, O_n$ onde O_i é o ângulo entre pontos consecutivos da curva.

O problema de reconhecimento de escrita consiste em descobrir de qual dígito (entre 0 e 9), a observação O é mais parecida. Um modo de resolver esse problema é usar um modelo probabilístico em que possamos:

1. Treinar o modelo, isto é, obter parâmetros específicos λ_i para cada dígito i. O treinamento deve ser feito a partir de dados reais. Por exemplo, o usuário (ou vários usuários) escreve várias vezes o dígito i e o algoritmo de treinamento se encarrega de obter parâmetros específicos λ_i para o modelo do dígito i.

2. Calcular o valor de $P(O|\lambda_i)$, que é a probabilidade de ocorrência da observação

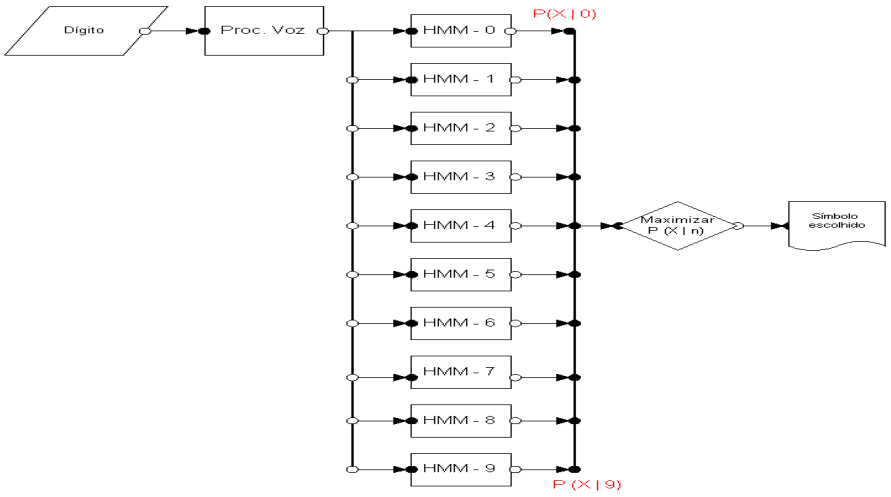

Figura 2.2: Sistema de Reconhecimento de Dígitos

O (sinal de entrada que queremos reconhecer) dado um modelo de parâmetros λ_i. Podemos decidir de qual dígito o dado de entrada O é mais parecido calculando o valor $P(O|\lambda_i)$ para todos símbolos i. O símbolo i^* escolhido é aquele com maior valor de $P(O|\lambda_{i^*})$, isto é:

$$i^* = \arg\max_i P(O|\lambda_i)$$

O modelo probabilístico utilizado é o HMM, cujos algoritmos para o cáculo de $P(O|\lambda_i)$ e para o treinamento já foram descritos. Na figura 2.2 temos um diagrama esquemático do reconhecimento de escrita (ou de voz com pequeno vocabulário). O sinal de entrada é processado (cálculo dos ângulos entre pontos consecutivos da curva), obtendo-se O. Calculamos $P(O|\lambda_i)$ para todos os dígitos i e selecionamos aquele com valor máximo.

Na implementação optamos por modelar cada dígito com um HMM de 5 estados e com observação O discreta, logo devemos efetuar uma quantização vetorial para obter um conjunto de ângulos discretos. Optamos pela quantização vetorial mais simples possível dividindo o espaço de ângulos em 64 partes iguais, obtendo um codebook de tamanho 64.

Para o treinamento é necessário informar ao sistema qual o dígito que escrevemos. O treinamento do HMM é feito com o algoritmo de Baum-Welch, que nos fornece os

parâmetros λ_i^* ótimos para o modelo do dígito i:

$$\lambda_i^* = \arg\max_{\lambda} P(L_i|\lambda)$$

onde L_i é a lista de curvas que informamos serem do dígito i. Cada curva é representada por $O = O_1, O_2, \ldots, O_n$ onde O_i é o ângulo entre pontos consecutivos.

Na implementação o modelo de cada dígito é treinado com um banco de dados de 30 dígitos de exemplo. A inicialização do algoritmo é feita de modo aleatório e arbitramos um modelo de 5 estados. Como o algoritmo de Baum-Welch só garante maximização local, fazemos 3 ou 5 tentativas com inicializações distintas e selecionamos aquela com maior valor de $P(L_i|\lambda)$.

Essa implementação tem várias limitações como, por exemplo, o pequeno banco de dados de treinamento de apenas 30 amostras para cada dígito, o uso de somente uma feature (ângulo), a ausência de suavização ou reparametrização do dado de entrada e o uso de um HMM discreto ao invés de um HMM contínuo. Apesar disso a performance do sistema não é ruim. Note que para os dígitos escritos na figura 2.1, o programa reconheceu erroneamente somente o dígito 5. O sistema teve 96% de acertos no banco de dados de treinamento. Tivemos 69.4% de acertos para o banco de dados não usado para o treinamento, mas que foi escrito pela mesma pessoa que escreveu o dado de treinamento (validação cruzada). Tivemos ainda, respectivamente, 52.6% e 58.9% de acertos em banco de dados escritos por outras pessoas.

Exemplo 5 *Reconhecimento de Voz*

Na década de 80 foi introduzido em reconhecimento de voz o método estatístico baseado em Hidden Markov Models (HMM) e desde então HMM é a técnica mais comum em reconhecimento de voz [Rabiner 93].

É comum em reconhecimento de sinais processar o sinal original para obter sinais mais convenientes ou relevantes para a tarefa de reconhecimento. No caso da voz é comum uma pré-ênfase, que é usada para amplificar as altas frequências do sinal e calcular as features do som usando LPC (Linear Predictive Coding) ou MFCC (Mel Frequency Cepstral Coefficients) [Bechetti 99]. Obtida as features da voz podemos passar para o reconhecimento propriamente dito.

O tipo mais simples de reconhecimento de voz é quando temos um pequeno vocabulário e as palavras são isoladas (há uma pequena pausa entre elas), sendo possível separá-las. Nesse caso o reconhecimento de voz é feito como no exemplo 4 de reconhecimento de dígitos, onde nossa unidade básica são as palavras. Na figura 2.2 temos o diagrama esquemático desse sistema.

Figura 2.3: HMM de estrutura left-right de cinco estados.

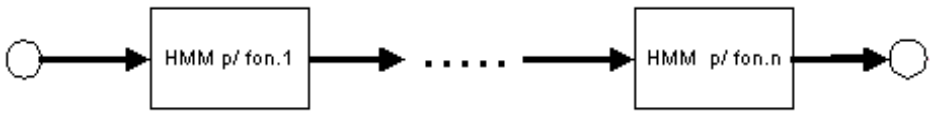

Figura 2.4: Modelo de uma palavra - concatenação de HMMs dos fonemas

O tipo mais geral e importante de reconhecimento de voz é quando temos um grande vocabulário e a fala é conectada, como na fala natural. Nesse caso o fenômeno da co-articulação (interferência na produção do final de um som com o início do som seguinte) altera bastante o sinal de voz e as fronteiras das palavras não são bem definidas, tornando difícil, ou mesmo impossível, especificar precisamente as fronteiras de uma palavra.

Quando o vocabulário é grande ($\geq$ 1000 palavras), se torna impraticável que a unidade básica sejam as palavras. Necessitamos de unidades menores como o fonema (existem em torno de 50 fonemas na língua inglesa) que é comumente usado. Para cada símbolo do alfabeto fonético é construído um HMM, em geral com estrutura left-right, onde as únicas transições permitidas são para o próprio estado e para o estado seguinte, isto é, sua matriz de transição de estados A tem o elemento $a_{ij} = 0$, se $j \neq i$ e $j \neq i + 1$. Essa matriz A pode ser representada graficamente como na figura 2.3. Comumente são usados de 3 a 5 estados para se modelar um fonema [Jelinek 97].

Escolhido o fonema como unidade básica da modelagem com HMM, o modelo das palavras ou de sentenças inteiras é criado simplesmente concatenando os HMM de fonemas, como vemos nas figuras 2.4 e 2.5.

Para o reconhecimento de voz devemos considerar o conjunto de variáveis aleatórias $\{A, W\}$ onde A é o dado de entrada de voz processado, a evidência acústica, as features da voz e W é uma sequência de palavras, que são as variáveis ocultas que descrevem eventos causados por A. Devemos estimar qual a sequência de palavras W mais provável, dado a evidência acústica A.

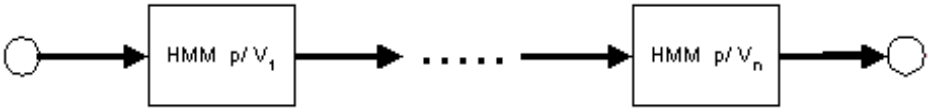

Figura 2.5: Modelo de uma sentença - concatenação de HMMs de palavras

Seja A é uma sequência de símbolos a_i (gerados no tempo):

$$A = a_1, a_2, \ldots, a_m$$

W é uma sequência de n palavras, onde cada uma pertence a um dicionário fixo (e finito) $\mathcal{W}$.

$$W = w_1 w_2 \cdots w_n, \text{ onde } w_i \in \mathcal{W}$$

Dado a evidência A, devemos escolher o conjunto de palavras mais provável W^*:

$$W^* = \arg\max_W P(W|A)$$

aplicando o Teorema de Bayes e ignorando o fator de escala $P(A)$, temos que:

$$W^* = \arg\max_W P(A|W) \cdot P(W)$$

Como o HMM concatenado de W é conhecido, o cálculo de $P(A|W)$ se resume ao cálculo da probabilidade da ocorrência da acústica A no HMM concatenado de W. Para o cálculo de $P(W)$ é necessário modelarmos a linguagem. É comum usar o modelo de trigramas [Jelinek 97], como no exemplo 1 de Modelagem de Linguagem.

Apesar dos algoritmos de HMM serem muito eficientes, devido ao grande número de estados possíveis nesse caso, é necessário um método de busca aproximada como, por exemplo, o algoritmo A^*, que é uma estratégia de busca em árvore muito usada em reconhecimento de voz [Jelinek 97].

2.6 Modelos Gráficos

Um modelo gráfico une duas áreas bem estudadas da matemática: Teoria da probabilidade e Teoria de grafos. Modelos gráficos são representados por um grafo onde os nós representam variáveis aleatórias de interesse, muitas vezes com significado físico e os arcos representam o tipo de dependência entre os nós. Modelos gráficos são modulares, isto é, um sistema complexo é construído de partes simples. Modelos

Gráficos são uma ferramenta útil para lidar com incerteza e complexidade e suas aplicações cobrem os mais variados ramos do conhecimento, entre eles aprendizagem de máquina e inteligência artificial.

Segundo [Pearl 00] grafos são aplicados em modelagem probabilística e estatística por três razões:

1. Provê uma maneira conveniente de expressar hipóteses relevantes.
2. Facilita a representação econômica da distribuição de probabilidade conjunta.
3. Facilita inferências eficientes das observações.

Antes de discorrer sobre modelos gráficos, e particularmente sobre redes bayesianas, vamos apresentar um breve resumo sobre grafos.

2.6.1 Grafos

Um **Grafo** consiste de um conjunto V de vértices (ou nós) e um conjunto A de arestas (ou arcos) que conectam alguns pares de nós. Os nós correspondem a variáveis e os arcos denotam o tipo de relação entre os nós conectados.

Os arcos podem ser **direcionados** (ou orientados, onde temos arcos com direção, com setas) ou **não-direcionados** (ou não orientados, onde temos arcos sem setas), como na figura 2.6. Se todos os arcos de um grafo são direcionados, ele é chamado de **grafo direcionado**, como por exemplo, na figura 2.7.

Em um grafo direcionado, um **caminho direcionado** (ou orientado) é um caminho entre dois nós em que respeitamos a direcionalidade dos arcos e um **caminho não-direcionado** é um caminho entre dois nós onde ignoramos a direcionalidade dos arcos. Um caminho direcionado em que retornamos ao nó origem é chamado de **ciclo direcionado** ou loop ou circuito de retro-alimentação (feedback). Um caminho não-direcionado em que retornamos ao nó origem é chamado de **ciclo não-direcionado**.

Se num grafo direcionado não há ciclos direcionados ele é chamado de **Grafo Acíclico Direcionado (Directed Acyclic Graph - DAG)**. Redes bayesianas, como veremos adiante, são redes probabilísticas construídas em DAG's.

Num DAG, se existe um arco de A para B ($A \longrightarrow B$) dizemos que A é **pai** de B (ou que B é **filho** de A). Se um nó não tem pais, ele é chamado de **nó fonte (ou raiz)**. Sempre existe pelo menos um nó fonte num DAG.

Um DAG sempre pode ser ordenado topologicamente[3], isto é, podemos ordenar todos os N vértives de um DAG na forma $X_1, X_2, \ldots, X_N$, onde se o vértice X_a é pai de X_b então $a < b$.

[3] Em geral essa ordenação não é única.

Uma **polytree** é um DAG onde há no máximo um caminho não-direcionado entre quaisquer dois nós, isto é, uma **polytree** (também chamada de **rede simplesmente conectada**) é um DAG sem nenhum ciclo não-direcionado.

A figura 2.7 mostra um DAG que não é uma polytree pois podemos ir do nó *Cloudy* para o nó *WetGrass* por dois caminhos diferentes. Já o DAG $A \rightarrow B \rightarrow C \leftarrow D \leftarrow E \rightarrow F$ é uma polytree. Como veremos adiante, algoritmos de inferência em polytrees são muito eficientes.

Um problema clássico em DAG é o problema de menor caminho, onde devemos achar o caminho de menor custo entre dois nós considerando que a cada arco há um custo associado. Algoritmos de programação dinâmica são eficientes para o problema de menor caminho, assim como para muitos outros problemas em DAG.

2.6.2 Tipos de Modelos Gráficos

Modelos Gráficos têm uma variedade de nomes na literatura e podem ser classificados em dois tipos principais:

- Não-Direcionados ou não-orientados: Construídos em grafos não-direcionados. Em geral são chamados de Markov Networks ou Markov Random Fields (MRF). Os arcos indicam uma relação espacial ou de vizinhança.

- Direcionados ou Orientados: Construídos em DAG. São chamados de Redes Bayesianas, Modelos Causais, Redes de Crença (Belief Networks) ou Modelos Generativos. Nesse trabalho vamos sempre nos referir a esse tipo de modelo como rede bayesiana. Nesse tipo de modelo os arcos indicam influência direta causal, causalidade[4] e não fluxo de informações.

2.7 Markov Random Fields

Os Modelos Não-Direcionados não são usados nesse trabalho, mas daremos aqui um breve resumo desta importante classe de modelos.

Um conjunto X de variáveis aleatórias (sobre um grafo $G = (V, A)$) é um $MRF \iff$

$$P(X_s|X_r, r \neq s) = P(X_s|\ \textit{Vizinhos de}\ X_s)$$

onde os vizinhos de X_s são as variáveis aleatórias conectadas por arcos a X_s.

[4] Não é correto dizer que os arcos de uma Rede Bayesiana indicam causalidade como veremos na seção de Redes Bayesianas, mas essa intuição é útil e nas aplicações geralmente é uma interpretação correta.

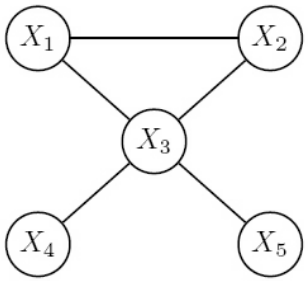

Figura 2.6: Exemplo de Markov Random Fields

No exemplo da figura 2.6 temos que X_1 só é vizinho de X_2 e X_3, como indicam os arcos, logo temos que $P(X_1|X_2, X_3, X_4, X_5) = P(X_1|X_2, X_3)$.

Uma importante propriedade (ou outra definição) de MRF é o teorema de Hammersley-Clifford que diz que a distribuição de probabilidade conjunta $P(X = x)$ pode ser calculada, excetuando a constante de normalização, a partir de uma função potencial definida no conjunto de pontos vizinhos entre si (clique). Infelizmente o cálculo exato da constante de normalização tem custo exponencial, o que torna necessário cálculos aproximados.

MRF é uma boa ferramenta para modelar relações espaciais ou de vizinhança. MRF são importantes em vários ramos do conhecimento, particularmente em visão computacional, e apareceram na literatura pela primeira vez na década de 20 como o Modelo de Ising para análise de magnetismo de estruturas $2D$.

2.8 Redes Bayesianas

Redes Bayesianas são construídas em DAG onde os nós representam as variáveis aleatórias e os arcos indicam uma relação de dependência (condicional) entre as variáveis aleatórias. Informalmente um arco do nó A para o nó B ($A \longrightarrow B$) indica que A causa B. Um exemplo de rede bayesiana é uma árvore genealógica, onde os nós representam as pessoas e os filhos recebem setas (arcos) de seus pais. Redes Bayesianas, em geral, são intuitivas e são representações diretas do mundo. Na figura 2.7 temos um exemplo simples de uma rede bayesiana em que vemos possíveis causas para que a grama fique molhada.

Seja um DAG onde cada nó i ($1 \leqslant i \leqslant N$) representa a variável aleatória X_i. Esse

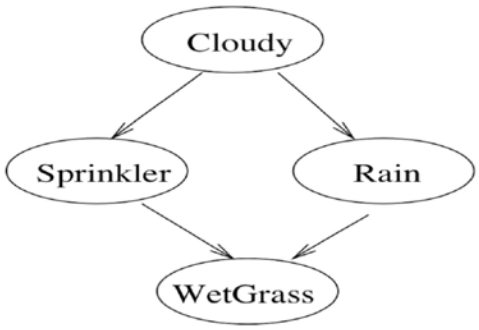

Figura 2.7: Exemplo de Rede Bayesiana

DAG é uma **Rede Bayesiana** se, por definição:

$$P(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} P(X_i|Pais(X_i))$$

se um nó i é fonte (não tem pais) tomamos $P(X_i|Pais(X_i)) = P(X_i)$ na fórmula acima. Note que a rede bayesiana fica completamente definida pelas distribuições condicionais de probabilidades $P(X_i|Pais(X_i))$.

A fórmula acima, também conhecida como propriedade de fatoração recursiva, mostra que uma rede bayesiana têm uma representação econômica da distribuição de probabilidade conjunta. Em um modelo geral, se $X_1, X_2, \ldots, X_N$ são variáveis lógicas, a distribuição de probabilidade conjunta fica totalmente determinada de forma tabular com 2^N parâmetros. Já numa rede bayesiana, se cada filho tem k pais, a distribuição de probabilidade conjunta fica determinada com ordem de $N.2^k$ parâmetros. A representação econômica da distribuição de probabilidade conjunta é de vital importância para a aplicabilidade de um modelo pois o cálculo eficiente da distribuição conjunta $P(X|\theta)$ é o cálculo básico em um modelo.

Uma rede bayesiana tem outras definições que podem ser provadas equivalentes:

- Propriedade de Markov Local: Um nó é condicionalmente independente de todos seus não descendentes dado seus pais. Esta propriedade é a base para inferências eficientes numa rede bayesiana. Outra forma, que pode ser provada equivalente, da propriedade de Markov local é: Um nó é condicionalmente independente de todos os outros nós da rebe bayesiana conhecidos seus pais,

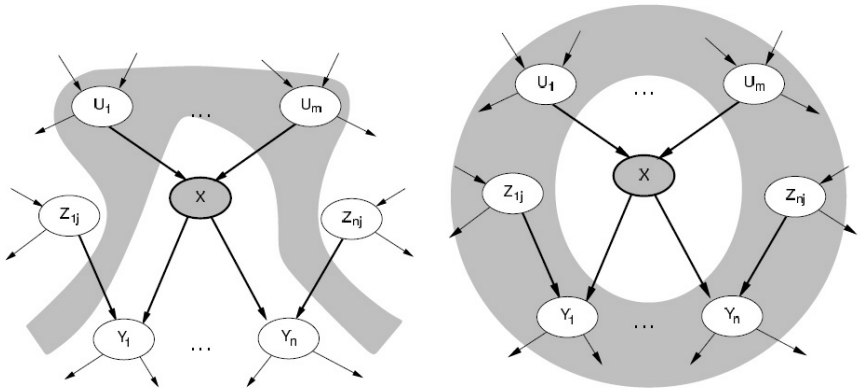

Figura 2.8: Propriedade de Markov Local em uma Rede Bayesiana, de [Murphy 02]

filhos e os outros pais de seus filhos. A figura 2.8 mostra graficamente essas duas versões da propriedade local de Markov.

- Separação Direcional (d-separação): É um critério puramente topológico num DAG que define quando Z d-separa X e Y, onde X, Y e Z são conjuntos disjuntos de nós do DAG. Pode ser provado que se numa rede bayesiana Z d-separa X e Y então $X \perp Y|Z$, também conhecida como propriedade de Markov global numa rede bayesiana. O conceito de d-separação não vai ser usado nesse trabalho, porém sua definição e maiores detalhes podem ser encontrados em [Pearl 88] ou [Pearl 00].

Em uma rede bayesiana os espaços amostrais dos nós podem ser discretos ou contínuos. Se o nó e seus pais são discretos, a distribuição de probabilidade condicional $P(filho|pais)$ pode ser expressa de forma tabular. Se um nó Y e seu pai X representam variáveis aleatórias contínuas, é comum definir a distribuição de probabilidade condicional como sendo a linear gaussiana:

$$P(Y|X = x) = \mathcal{N}(A \cdot x + \mu, \sigma)$$

o que é equivalente a:

$$Y = A \cdot x + \mathcal{N}(\mu, \sigma)$$

onde A é uma matriz e $\mathcal{N}$ é a distribuição normal de parâmetros μ e σ.

Se um nó Y é contínuo e seu pai X discreto, uma representação comum é:

$$P(Y|X = i) = \mathcal{N}(\mu_i, \sigma_i)$$

O uso de gaussianas para representar a distribuição de probabilidade condicional é útil não só para a simplificação dos cálculos, mas também pelo fato de que se todas as variáveis contínuas tem distribuição de probabilidade condicional gaussiana então a distribuição de probabilidade conjunta é uma distribuição gaussiana, isto é, a família de distribuições lineares gaussianas é fechada sobre operações padrões em redes bayesianas. Com exceção de alguns casos especiais, como a linear gaussiana, o número de parâmetros da distribuição conjunta em uma rede bayesiana cresce sem limite com respeito ao número de nós [Russell 02].

2.8.1 Cálculos em uma Rede Bayesiana

Uma característica importante e útil de uma rede bayesiana é que ela admite em geral algoritmos simples e eficientes, além de já existirem pacotes computacionais em várias linguagens.

O cálculo da distribuição conjunta como já vimos, pode ser feito de modo eficiente usando a propriedade (definição) de fatoração recursiva. Outros cálculos importantes são a inferência e o treinamento.

Inferência

Na inferência queremos estimar qual a probabilidade de um conjunto de nós conhecido o valor de um outro conjunto de nós, isto é, queremos estimar as variáveis ocultas das observadas, isto é, queremos calcular $P(X_q|X_o = x_o)$ onde X_q são as variáveis que desejamos estimar e X_o são as variáveis observadas.

Algoritmos de inferência para redes bayesianas seguem a linha de algoritmos de programação dinâmica. Alguns algoritmos de inferência em redes bayesianas são parecidos com o algoritmo forward para HMM[5]. A idéia geral desses algoritmos é usar a propriedade de Markov local, que diz que um nó é condicionalmente independente de seus não descendentes conhecido os seus pais. Intuitivamente, estimar um nó é fácil se conhecemos os seus pais, mas se os pais são desconhecidos, podemos estimá-los facilmente se conhecemos os avós. Esse argumento recursivo nos leva ao nó raiz, que é conhecido pois a distribuição do nó raiz é um parâmetro da rede bayesiana.

Um algoritmo simples e comum de inferência em redes bayesianas é o de eliminação de variáveis, onde eliminamos as variáveis indesejadas distribuindo as somas dentro dos produtos. Um algoritmo mais eficiente é o de passagem de mensagem.

[5]Um HMM é uma rede bayesiana, como veremos na seção de redes bayesianas dinâmicas. Uma rede bayesiana também pode ser transformada em um HMM, bastando para isso criar mega-estados, o que na prática é ineficiente.

Todos eles são baseados em operadores forward e backward abstratos [Murphy 02], no espírito dos operadores forward e backward dos algoritmos de HMM.

Algoritmos para redes bayesianas são particularmente eficientes para polytrees (só existe um caminho entre dois nós), onde podemos fazer a inferência com custo linear no número de nós. Quando a rede bayesiana não é uma polytree, existe mais de um caminho entre dois nós, como na figura 2.7, e intuitivamente a influência entre os nós é exercida simultaneamente entre caminhos diferentes, o que dificulta a inferência. Pode ser provado [Russell 02] que a inferência em redes bayesianas é um problema *NP-difícil*[6].

No caso de redes bayesianas que não são polytrees, uma técnica comum para a inferência é o agrupamento (clustering), onde os nós são combinados para que a DAG resultante seja uma polytree [Charniak 91]. Um problema prático dessa técnica é que o número de variáveis cresce exponencialmente com o agrupamento.

Muitas vezes é necessário aplicar métodos de inferência aproximados. Métodos comuns são os algoritmos de Monte Carlo ou de filtragem de partículas, onde aproximamos distribuições contínuas a partir de um conjunto finito de partículas.

Treinamento

No treinamento tentamos obter qual a rede bayesiana que mais provavelmente gerou os dados observados. Há duas formas de treinamento em redes bayesianas: O treinamento de parâmetros (onde já conhecemos ou arbitramos o DAG e queremos estimar somente as distribuições de probabilidades condicionais) ou o treinamento estrutural (onde só conhecemos as variáveis aleatórias, os nós, e queremos estimar quais as arestas do DAG e suas distribuições condicionais).

O treinamento estrutural é mais delicado pois o número de DAG's é super-exponencial no número de nós. Um método comum é executar uma busca local usando o BIC (Bayesian Information Criterium) como função de custo [Murphy 02].

No treinamento de parâmetros já temos o DAG definido, bastando estimar quais as distribuições de probabilidades condicionais. O treinamento pode ser feito por máxima verossimilhança ou por métodos bayesianos. O treinamento é um problema de otimização não linear e pode ser resolvido por métodos gerais de otimização, como por exemplo, o método do subida pelo gradiente. Como em HMM, o método EM é muito utilizado. O algoritmo EM é um algoritmo iterativo em que assumimos um modelo inicial θ e um dado oculto não observado. No passo **E** calculamos a esperança das variáveis ocultas dado o modelo θ. No passo **M** assumimos que o valor das

[6] Pode ser provado que a inferência em redes bayesianas é *#P-difícil*, que é mais difícil que problemas *NP-completos*[Russell 02].

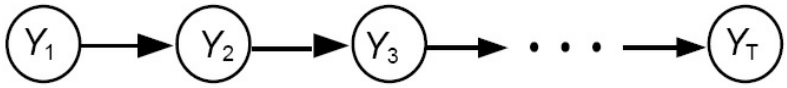

Figura 2.9: Cadeia de Markov expressa como uma rede bayesiana

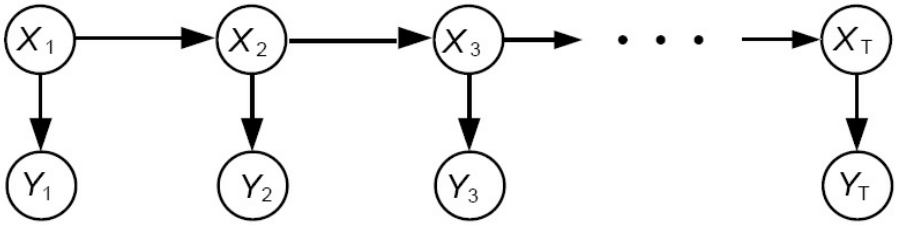

Figura 2.10: HMM expressa como uma rede bayesiana

variáveis ocultas é a esperança calculada no passo **E** e a partir daí calculamos um novo modelo θ_n. O algoritmo EM é útil quando é fácil calcular os passos **E** e **M**, o que é o caso quando as distribuições condicionais de uma rede bayesiana são lineares gaussianas. Pode ser provado que o EM converge linearmente para um máximo local. Maiores detalhes sobre o algoritmo EM pode ser encontrado na seção de HMM ou em [Murphy 02], [Nechyba 00], [Ghahramani 98].

2.9 Redes Bayesianas Dinâmicas (DBN)

Processos estocásticos, na forma de sequência de dados ou séries temporais, são muito importantes e com grandes aplicações práticas. Uma rede bayesiana que modela um processo estocástico é chamada de rede bayesiana dinâmica, que é uma rede bayesiana em que uma célula básica (rede bayesiana simples) se repete ao longo do tempo. O termo dinâmico se refere a um processo gerado por um sistema dinâmico, e não que a rede sofra alguma alteração ao longo do tempo.

Nas figuras 2.9 e 2.10 temos exemplos de redes bayesianas dinâmicas. A replicação de uma célula básica (Y ou $X \rightarrow Y$) ao longo do tempo forma a rede bayesiana dinâmica.

Cadeias de Markov, HMM e Sistemas Dinâmicos Lineares são casos particulares importantes de uma rede bayesiana dinâmica. A figura 2.9 mostra uma cadeia de Markov representada como uma rede bayesiana dinâmica. A estrutura $Y_i \rightarrow Y_{i+1}$

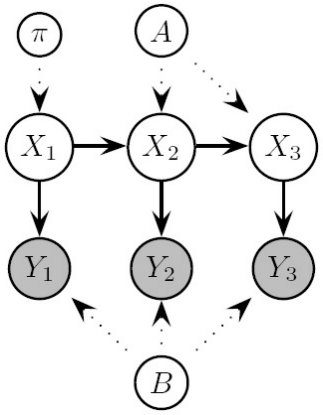

Figura 2.11: HMM e seus parâmetros, obtida em [Murphy 02]

indica que Y_i é pai de Y_{i+1}, isto é, $P(Y_{i+1}|Y_i) = P(Y_{i+1}|Y_1, Y_2, \ldots, Y_i)$, que é a propriedade básica de uma cadeia de Markov. Os parâmetros da rede bayesiana dinâmica são a distribuição condicional $P(Y_{i+1}|Y_i)$ que é a matriz de transição de cadeia de Markov e a distribuição a priori $P(Y_1)$.

A figura 2.10 mostra um HMM representado como uma rede bayesiana dinâmica. A estrutura $X_1 \rightarrow X_2 \rightarrow \cdots \rightarrow X_T$ é uma cadeia de Markov com a matriz de transição A e distribuição a priori $P(X_1) = \pi$. Já a estrutura $X_i \rightarrow Y_i$ indica que X_i é pai de Y_i com distribuição condicional $P(Y_i|X_i)$ na forma de uma matriz B ou uma gaussiana. A figura 2.11 mostra os parâmetros dessa rede bayesiana dinâmica.

Se as variáveis aleatórias X e Y são contínuas, a figura 2.10 também representa um sistema dinâmico linear (ou modelo de filtro de Kalman), cujas distribuições condicionais costumam ser da forma $P(X_{i+1}|X_i) = \mathcal{N}(A \cdot X_i + \mu_X, \sigma_X)$ e $P(Y_i|X_i) = \mathcal{N}(C \cdot X_i + \mu_Y, \sigma_Y)$ e distribuição a priori $P(X_1) = \mathcal{N}(\mu_1, \sigma_1)$.

Cálculos comuns em redes bayesianas como o cálculo da distribuição conjunta, a inferência e o treinamento também são os problemas básicos numa rede bayesiana dinâmica. Todos esses problemas em redes bayesianas dinâmicas podem ser resolvidos usando algoritmos para redes bayesianas comuns, levando em conta que a estrutura repetida de uma rede bayesiana dinâmica pode simplificar a notação e estrutura dos algoritmos.

2.10 Redes Bayesianas Causais

Algo muito interessante sobre redes bayesianas é que elas são um bom modelo para tratar de causalidade em termos matemáticos rigorosos. Já foi dito que causalidade é uma intuição útil para o significado das arestas do grafo de uma rede bayesiana, mas isso não é necessariamente verdadeiro, porém na grande maioria das aplicações a rede bayesiana é construída de forma que as arestas têm o significado de causalidade. Quando uma rede bayesiana é construída sobre relações causais ela é chamada de rede bayesiana causal ou grafo causal.

Uma rede bayesiana não é necessariamente causal. Por exemplo, seja uma rede bayesiana causal:

$$\textit{Chuva}\ (Ch) \longrightarrow \textit{Grama Molhada}\ (GM)$$

onde são conhecidos os parâmentros $P(Ch)$ e $P(Ch|GM)$. Podemos criar a rede bayesiana "invertida" não causal $GM \rightarrow Ch$, bastando para isso calcular $P(Ch|GM)$ com o teorema de Bayes e calcular a distribuição marginal $P(GM)$.

Uma rede bayesiana causal têm vantagens em relação à uma rede bayesiana não causal. Na rede bayesiana causal o modelo tem apelo e significado intuitivo e é mais estável, no sentido de que mudanças externas levam a pequenas mudanças na estrutura da rede bayesiana causal. Relações causais são mais estáveis que relações probabilísticas pois relações causais descrevem restrições físicas do mundo (ontologia) e relações probabilísticas descrevem nossas crenças sobre o mundo (epistemologia) [Pearl 00].

Pearl [Pearl 00] sugere que a rede bayesiana é um bom modelo para exprimir causalidade em termos matemáticos e cria uma definição de rede bayesiana causal, que seria uma rede bayesiana capaz de exprimir relações causais. Essa definição usa o conceito de ação (DO) em uma rede bayesiana. A ação de definir externamente o estado específico de um nó (variável aleatória) elimina a influência dos pais desse nó, o que justifica o corte das arestas que ligam os pais do nó a ele próprio, gerando um grafo "mutilado". Segundo Judea Pearl uma rede bayesiana é causal se para todo conjunto de nós do grafo em que fizermos a ação (DO), o grafo mutilado é uma rede bayesiana com probabilidades consistentes e onde se verifica as relações de independência condicional de uma rede bayesiana. Para mais detalhes, consultar [Pearl 00].

Como um exemplo da generalidade, utilidade e apelo intuitivo de modelos gráficos, a figura 2.12, tomada de um artigo de psicologia, é um modelo gráfico misto onde as variáveis aleatórias são representadas por retângulos. Se criarmos um mega-estado que é a união das variáveis *Uncouscious Cause of Thought* e *Uncouscious Cause of Action*, a aresta indireta *Unconscious Path* desaparece e o modelo gráfico misto se

torna uma Rede Bayesiana Causal.

2.10.1 Causalidade

Dos dicionários Aurélio Eletrônico e Random House temos as seguintes definições de Causalidade: (1) Qualidade da relação de causa e efeito ou (2) denota a relação lógica entre um evento (causa) e outro evento (efeito) que é uma consequência direta do primeiro (causa).

Esse entendimento informal é suficiente para o dia a dia, porém a análise filosófica da causalidade tem se provado extremamente difícil. Segundo David Hume (1711-76) todo o conhecimento vêm da experiência, que é codificada mentalmente como correlação, o que não implica em causalidade. Decidir se uma correlação é ou não é uma relação causal pode ser uma tarefa difícil. Não é claro nem mesmo como os seres humanos adquirem conhecimento sobre causalidade.

Causalidade não pode ser definida somente com a análise de dados, são necessários argumentos lógicos que definem o que é ou não causal. *Causalidade é um hábito aprendido pela mente, quase tão ficcional como ilusões óticas ou transitórias como o condicionamento de Pavlov* [Pearl 96].

Apesar de todos os problemas filosóficos e físicos (com exceção da terceira lei da termodinâmica, que diz que a entropia sempre aumenta, todas as leis físicas são simétricas, não levando em conta causas e efeitos), o conceito de causalidade é essencial para o ser humano.

2.10.2 Livre Arbítrio

Na seção anterior vimos que podemos errar em definir uma relação como causal. Um exemplo interessante é a questão do livre arbítrio. Segundo [Wegner 03] e [Wegner 99], o livre arbítrio é a percepção errada de que os pensamentos causam as ações humanas. Não é o eu consciente ou o pensamento, mas sim processos cerebrais inconscientes que causam a ação, conforme a figura 2.12.

Entre os vários argumentos e experimentos para Wegner justificar sua tese, um dos mais interessantes são os experimentos de Libet [Libet 85], [Wegner 99]. Nesses experimentos, resumidamente, pessoas são instruídas a apertar, quando tiverem vontade, um entre dois botões à sua escolha. É medido o tempo quando a pessoa toma a decisão, quando ela aperta o botão e também sua atividade cerebral. As medidas indicam que a pessoa aperta o botão por volta de 100 a 150 mili-segundos depois da tomada de decisão, mas surpreendentemente, a decisão consciente de apertar o botão foi precedida por 300 a 350 ms pelo "potencial de prontidão" (readiness poten-

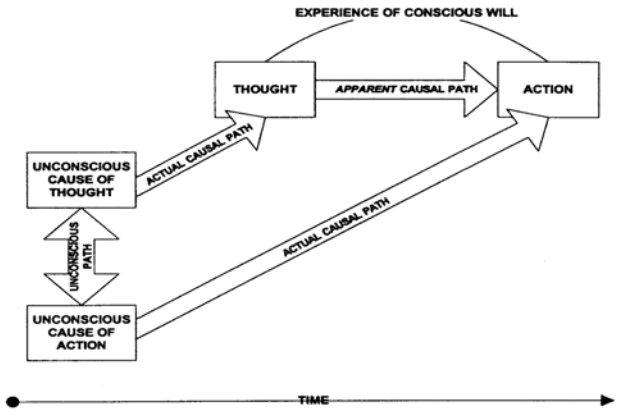

Figura 2.12: Experiência de Livre Arbítrio, de [Wegner 99]

tial) que se origina na área motora suplementar (SMA), que é uma região cerebral envolvida na preparação de movimentos. Esse experimento indica que processos cerebrais inconscientes iniciaram a decisão que seria tomada conscientemente 350ms depois. Esse experimento foi inspiração para a figura 2.12 obtida de [Wegner 99].

Recentemente, em experimento semelhante [Soon 08] usando ressonância magnética funcional (fMRI), foi medida a atividade cerebral relativa à tomada de decisão no córtex pré-frontal e parietal até dez segundos antes da tomada de decisão consciente em si.

Indo ainda mais longe, Wegner ([Wegner 05], [Wegner 07], [Wegner 08]) sugere que o ser humano é um robô que imagina que é um ser com livre arbítrio, com uma "alma". Apesar desse tipo de discussão fugir ao escopo e objetivo dessa tese, ela serve como argumento filosófico de que a modelagem matemática do movimento e comportamento humanos é possível e não está irremediavelmente destinada ao fracasso.

Capítulo 3

Trabalhos Relacionados

Vamos descrever alguns artigos que foram inspirações para esse trabalho. Existem relativamente poucos trabalhos sobre síntese de movimentos de cabeça a partir da voz que utilizam métodos estatísticos. Alguns destes artigos se baseiam na idéia de colagem de pedaços reais de movimentos, como o Mood Swings. Outros trabalhos, como o Rigid Head Motion Animation, Prosody-Driven Head-Gesture e Trajectory Model, usam um modelo probabilístico para treinar o modelo a partir de um banco de dados real de movimento e de voz associada para então sintetizar o movimento de cabeça somente a partir da voz, encontrando qual o movimento de cabeça mais provável dado a voz, segundo o modelo treinado.

Alguns trabalhos, como Motion Texture e Voice Puppetry não lidam exatamente com o nosso problema, mas desenvolveram técnicas úteis para a síntese de movimentos de cabeça a partir da voz ou ainda se mostram úteis no entendimento do problema.

3.1 Voice Puppetry

Matthew Brand [Brand 99] desenvolveu um método para predizer um sinal dado outro sinal relacionado e aplicou esse método para gerar uma animação facial dado uma trilha de áudio. O sistema foca em movimentos faciais e não da cabeça, com os participantes, durante o treinamento, sendo instruídos a movimentarem a cabeça o mínimo durante a fala.

Os descritores usados são o conjunto de pontos da face e suas velocidades para o movimento e uma mistura de descritores LPC e RASTA_PLP para o som.

O modelo utilizado é o HMM entrópico, que leva a modelos com estrutura mais esparsa. O método de Brand é constituido de quatro passos, como mostra a figura

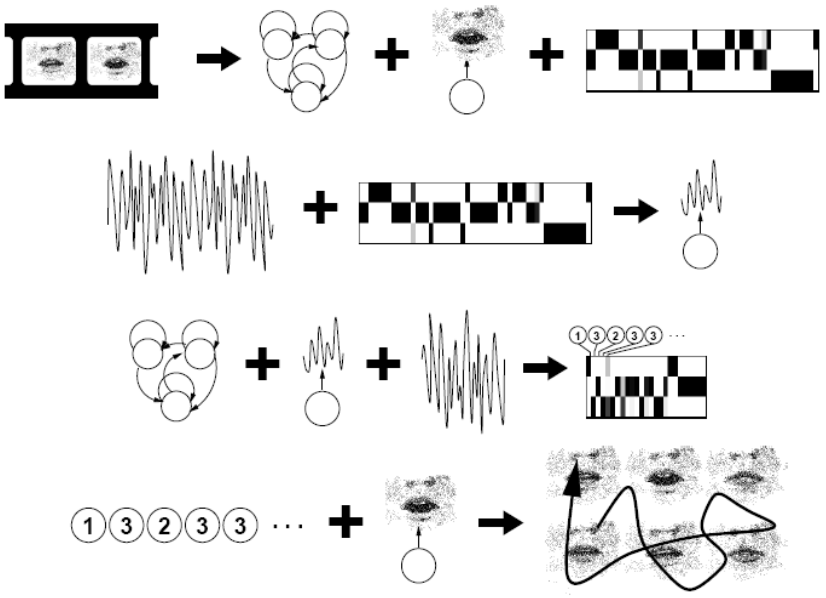

Figura 3.1: Voice Puppetry, retirada de [Brand 99]

3.1, retirada de [Brand 99]. No treinamento (primeira linha), o HMM é treinado com os dados de movimento do vídeo gerando uma matriz de transição, a densidade de observação e o diagrama de ocupação de estados do HMM no dado de treinamento. No remapeamento (segunda linha), a matriz de ocupação obtida no treinamento é combinada com o sinal de áudio para obtermos a densidade de observação do áudio. Na análise (terceira linha), dado um novo sinal de áudio e o HMM formado pela matriz de transição obtida no treinamento e a densidade de observação obtida no remapeamento, calculamos a sequência mais provável de estados com o algoritmo de Viterbi. Na síntese (quarta linha) é calculada a melhor trajetória de posições faciais de acordo com a sequência de estados mais provável obtida na análise e da densidade de observação obtida no treinamento. A síntese não é calculada simplesmente amostrando o HMM, o que levaria a movimentos bruscos, mas sim resolvendo um sistema linear levando em conta a média e variância de posição e velocidade de cada estado do HMM, gerando uma trajetória suave.

3.2 Motion Texture

Li, Wang e Shum [Li 02] propuseram um modelo em dois níveis para síntese de movimento de dança de um boneco articulado. Num primeiro nível o movimento (texton) é representado por um sistema dinâmico linear (LDS) da forma:

$$\begin{cases} X_{t+1} = A_t.X_t + V_t \\ Y_t = C_t.X_t + W_t \end{cases}$$

onde X_t é a variável oculta, representada com um vetor de dimensão de 12 a 15. V_t e W_t são ruídos Gaussianos. Y_t é a observação das rotações em 19 juntas corporais e a posição global, totalizando um vetor de dimensão 60, pois cada rotação é representada por um mapa exponencial (de dimensão 3).

No segundo nível de modelo é assumido que uma cadeia de Markov governa a probabilidade de transição entre LDS's.

O treinamento do modelo é feito com algoritmo EM a partir de mais de vinte minutos de movimento capturado, a uma taxa de $60Hz$, de um dançarino profissional.

No treinamento feito pelos autores foram obtidos 246 textons (LDS) diferentes. A síntese de movimento pode ser feita simplesmente amostrando o modelo em dois níveis ou pode ser feito ainda com síntese restrita obrigando que o ponto final de um texton seja o ponto inicial do texton seguinte. A síntese restrita é feita resolvendo um sistema linear, gerando um movimento suave e sem descontinuidades entre um texton e o seguinte.

3.3 Mood Swings

Nesse trabalho, Erika Chuang [Chuang 04], [Chuang 05], sintetiza a expressão facial e o movimento de cabeça a partir de um sinal de voz. Também é selecionado qual o tipo da expressão da voz, entre os três modos possíveis: neutro, raiva e alegria. Para a expressão facial foi usado um modelo bilinear. A síntese do movimento de cabeça, que é o nosso caso de interesse, é feito "colando" pedaços de movimentos de cabeça.

Inicialmente o som de teste é segmentado a partir da frequência fundamental (também chamada de $F0$, pitch ou tom) do sinal de voz. Como a frequência fundamental $F0$ é zero nas regiões onde o som é não vocalizado[1] e valor diferente de zero nas regiões vocalizadas, o som é segmentado em cada início (onset) de $F0$, onde o som não vocalizado passa a ser vocalizado. Segundo a autora, essa segmentação do

[1] som articulado sem vibração das cordas vocais, do inglês unvoiced ou voiceless

sinal de voz faz sentido prático pois foi observado empiricamente que movimentos significativos da cabeça acontecem logo após o início de $F0$.

Depois é feito o casamento de $F0$ em dois estágios. No primeiro estágio são comparados sentenças inteiras pois, segundo a autora, o conteúdo emocional e pessoal é expresso através dessas sentenças inteiras. Os descritores (features) da voz usados no primeiro estágio são: ritmo de fala, média de comprimento entre regiões com e sem voz e $F0$ (máximo, mínimo, média e desvio padrão). É comparado a distância euclidiana entre esses descritores no som de teste e em todas as sequências inteiras do banco de dados. Daí achamos as M sequências mais próximas, que são consideradas as sequências do banco de dados que têm o estilo expressivo desejado. No segundo estágio é calculada uma função distância entre cada segmento do som de teste (segmentado a partir de $F0$) e os segmentos (também obtidos com $F0$) das M sequências obtidas no primeiro estágio. Essa função distância depende de vários parâmetros, como as características geométricas dos segmentos (valores máximo e mínimo em cada segmento, máximo e mínimo da derivada, curvatura, etc.), da diferença de energia entre os segmentos e ainda penaliza segmentos de comprimentos diferentes. Para cada segmento de teste são mantidas apenas as K melhores escolhas segundo a função distância.

Outra etapa é o cálculo do melhor caminho. Dos N segmentos de $F0$ do som de entrada e das K melhores escolhas para cada segmento de teste obtidas no casamento é formado uma treliça (trellis) e o melhor caminho é achado com programação dinâmica (algoritmo de Viterbi), onde o custo de transição de um segmento para o próximo é calculado de forma que se encoraje tanto uma transição suave entre segmentos vizinhos (posição, velocidade e aceleração próximas) como a escolha de segmentos consecutivos no banco de dados original e desencoraje a repetição do mesmo segmento.

Por fim é feito uma colagem dos segmentos selecionados onde os segmentos são reamostrados para o tamanho requerido e as posições iniciais e finais dos segmentos são conectadas e suavizadas (motion blending).

3.4 Visual Prosody

Em [Costa 01] foi proposto um método para animação facial dirigida por voz. A aquisição dos dados de movimento facial foi feita baseada no algoritmo KLT (Kanade, Lucas e Tomasi).

Os descritores de áudio utilizados são a frequência fundamental $F0$, a variância de $F0$ e a potência média nas faixas de frequência de $0 - 1.5KHz$ e de $1.5 - 4KHz$.

Os descritores de áudio e vídeo são unidos em um único vetor para o treinamento, que é feito a partir do modelo de misturas de gaussianas (GMM).

Na síntese de movimentos, dado um som de teste, obtemos com o GMM a estimativa X_v, que é o valor esperado dos descritores de vídeo dado os descritores do som de teste.

Da estimativa X_v são obtidos os movimentos pré-definidos mais prováveis (entre três movimentos possíveis), que então são concatenados com um Expression Blender, que dá coerência espacial e temporal para os movimentos. Os autores não entram em detalhes de como é feito a escolha do movimento pré-definido nem como esses movimentos são concatenados (expression blender).

3.5 Rigid Head Motion Animation

Nos artigos [Busso 05] e [Busso 07], Carlos Busso et alii desenvolveram um sistema para síntese de movimentos naturais de cabeça baseado em HMM. O banco de dados audiovisual utilizado foi feito com uma atriz com 102 marcos em sua face. A atriz expressa quatro emoções básicas: Neutra, tristeza, alegria e raiva. O sistema de captura VICON com três câmeras captura a posição $3D$ dos marcadores com uma taxa de amostragem de 120 frames/segundo e o som com $48KHz$. Das posições $3D$ dos marcadores são obtidos os ângulos de Euler da rotação da cabeça (três graus de liberdade).

Os descritores (features) utilizados para o sinal de voz são a frequência fundamental $F0$, a energia RMS e suas respectivas primeira e segunda derivadas, totalizando um descritor do som de dimensão 6. $F0$ é suavizado e interpolado para evitar zeros em regiões não vocalizadas.

Para a modelagem do movimento inicialmente é feito a quantização vetorial LBG [Linde 80] dos ângulos de Euler obtidos do banco de dados, obtendo-se K agrupamentos (clusters), com suas respectivas médias e variâncias.

Para cada cluster foi criado um HMM (com topologia left-right com dois ou três estados) e observação contínua modelada com uma mistura de duas gaussianas. O estado do HMM é o movimento $3D$ (rotação) e a observação é o som (dimensão 6). O sistema lida com os estados emocionais treinando HMM's diferentes para cada um desses estados. É utilizado um bigrama para modelar as transições entre os possíveis HMM's. Por fim é feito uma suavização com uma interpolação cúbica esférica com quatérnions e é aplicado ruído branco à sequência de movimento.

3.6 Trajectory Model

Gregor Hofer el alii desenvolveram um sistema de movimento de cabeça guiado pelo voz. De um total de 25 minutos de vídeo, os dados de movimento são obtidos com um sistema de captura de movimentos com uma taxa de amostragem de 500Hz. Os dados de voz são obtidos com um microfone a uma taxa de amostragem de 44kHz. As features de voz são a frequência fundamental ($F0$), os doze primeiros coeficientes de MFCC, a energia e suas respectivas primeira e segunda derivadas, totalizando um vetor de 42 dimensões. Os descritores para os movimentos da cabeça são os ângulos de Euler e suas primeira e segunda derivadas, totalizando um vetor de dimensão 9.

Os dados de movimento são manualmente classificados entre 4 alternativas possíveis de mudança corporal, movimentos laterais em volta de um eixo, pausa e movimento padrão (movimento lento ou indistinto).

O treinamento é feito com HMM [Hofer 07] ou com HMM de trajetórias [Hofer 07b]. Um HMM de trajetórias foi inicialmente proposto no contexto de síntese de voz com HMM's [Tokuda 00] e é um HMM que leva em conta a velocidade e aceleração do movimento. A síntese de um HMM cuja probabilidade de emissão é uma gaussiana é essencialmente irregular e descontínua pois a amostragem da gaussiana não leva em conta qual o último valor amostrado. Já a síntese de um HMM de trajetórias é suave pois leva em conta a velocidade e aceleração da trajetória do movimento.

O treinamento é feito com os descritores de voz e movimento. Já na síntese, do som de teste é calculado quais os tipos de movimentos mais prováveis segundo o modelo treinado.

3.7 Prosody-Driven Head-Gesture

Sargin et al. [Sargin 08] desenvolveram um método de análise conjunta de movimentos de cabeça e voz para a síntese de movimentos de cabeça a partir da voz. Os descritores usados para cada frame de som são a frequência fundamental ($F0$), sua derivada e a energia. Para o movimento, os descritores são os ângulos de Euler.

A análise é dividida em dois estágios. No primeiro estágio são obtidos os padrões básicos de som e movimento. No segundo estágio os padrões básicos obtidos no primeiro estágio são conjuntamente analisados para a detecção de padrões recorrentes.

Para obter os padrões elementares do primeiro estágio é usado um HMM com estrutura paralela left-right com M ramos paralelos e em cada ramo temos N estados, como indicado na figura 3.2, obtida de [Sargin 08]. A observação é modelada por uma Gaussiana com matriz de covariância diagonal. Os autores utilizam os valores $N_g = 10$ e $M_g = 5$ para o movimento de cabeça e $N_s = 5$ e $M_s = 5$ para o som.

Cada um dos M ramos do HMM é considerado um padrão básico entre os N padrões possíveis.

Inicialmente os HMM's de som e movimento são treinados separadamente com seus respectivos dados de treinamento para a obtenção dos seus parâmetros. Na posse dos HMM's já treinados e seus respectivos dados de treinamento, obtemos a sequência de estados mais provável com o algoritmo de Viterbi. Essa sequência de estados mais provável indica a sequência de ramos (ou padrões básicos) do dado de treinamento. Obtida a sequência de ramos tanto para o som como para o movimento do banco de dados, passamos à segunda estapa de treinamento.

Na segunda etapa treinamos conjuntamente os padrões discretos obtidos na primeira etapa com um HMM discreto paralelo multistream que tem a matriz de transição de estados também representada pela figura 3.2, obtida de [Sargin 08]. Um HMM multistream tem uma matriz de transição de estados única para as streams, mas com probabilidades de observação independentes para cada stream.

As matrizes de observação para o som e movimento, assim como a matriz de transição são determinadas no treinamento com os dados obtidos na primeira etapa.

Na síntese, dado um som de entrada, obtemos os seus padrões com o HMM já treinado na primeira etapa. Com esses padões de som e o HMM já treinado da segunda etapa, obtemos qual a observação de movimentos (padrões) discreta mais provável. Daí voltamos à primeira etapa para a síntese de movimentos com os HMM's correspondentes aos padrões de movimento obtidos na segunda etapa.

Por fim é necessário a suavização dos ângulos de Euler obtidos com a amostragem dos HMM's. Isso é feito com um filtro mediano sobre 11 amostras (frames) de ângulos de Euler seguido de uma suavização gaussiana sobre 15 frames.

3.8 Linking Head Motion and Speech Acoustics

Hani C. Yehia et al. [Yehia 02] pesquisaram a relação entre movimentos faciais e de cabeça com a voz, mostrando que a acústica pode ser estimada do movimento facial e vice-versa.

Para análise de movimentos faciais, as features acústicas usadas são os parâmetros LSP (line spectrum frequency pairs) pois são fortemente relacionados com as frequências ressonantes do trato vocal e, consequentemente, com a geometria do trato vocal [Yehia 98].

Os autores [Yehia 02] mostraram que existe uma forte correlação entre a frequência fundamental ($F0$) e o movimento de cabeça e mostraram ser possível estimar $F0$ do movimento de cabeça e vice-versa. Foram utilizados estimadores lineares para verificar o quanto da variância de $F0$ pode ser inferida do movimento de cabeça e

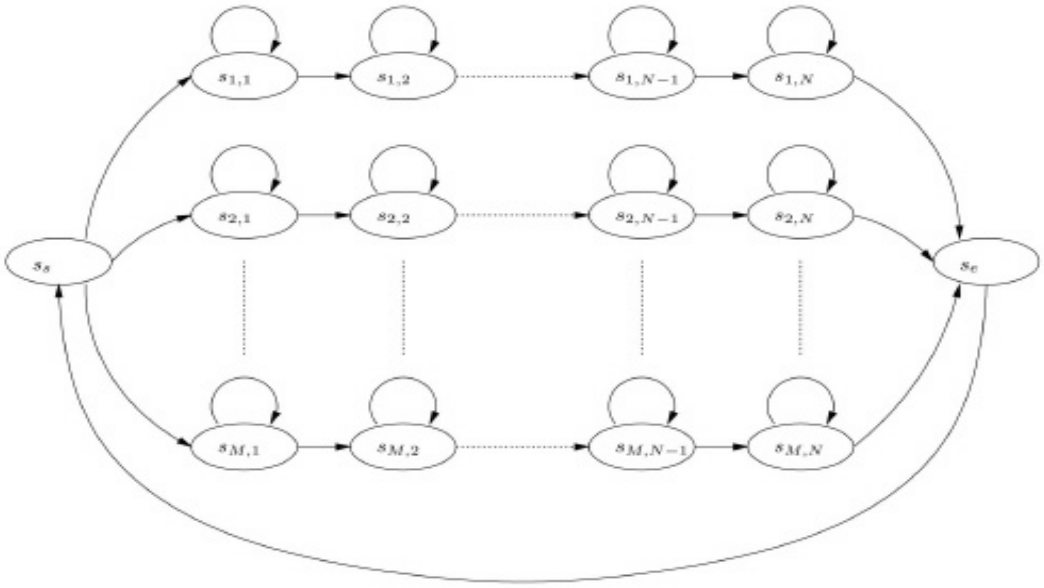

Figura 3.2: Estrutura de um HMM paralelo letf-right, obtida de [Sargin 08]

vice-versa. Do movimento de cabeça para estimar $F0$ foi verificado que 88% e 73% (nos bancos de dados EVB e TK [Yehia 98], respectivamente) da variância de $F0$ é observada no movimento de cabeça. De $F0$ para a estimação do movimento de cabeça, somente 50% e 25 % (nos bancos de dados EVB e TK [Yehia 98]) da variância do movimento de cabeça é observada em $F0$. Esses valores indicam que estimar o movimento de cabeça de $F0$ é um problema "um para muitos" e que novas restrições devem ser adicionadas para uma melhor estimação do movimento de cabeça.

Carlos Busso et al. [Busso 07] aplicaram CCA (Canonical Correlation Analysis) e acharam valores bastante altos ($r > 0.85$) de correlação entre o movimento de cabeça e o movimento sintetizado pelo seu método, indicando uma boa performance do sistema.

Gregor Hofer el al. [Hofer 07] também aplicou CCA entre descritores de movimento e som, mas obtiveram valores baixos ($r = 0.08$) de correlação, o que indica que propriedades temporais dos dois sinais têm de ser levadas em conta.

3.9 Talking Faces

Cosatto et al. [Cosatto 03] propuseram um método para síntese de "faces falantes" e aplicaram seu esquema para várias aplicações interativas, entre elas um leitor de notícias e um leitor de emails.

O método utilizado segue a linha do artigo Mood Swings, em que pedaços reais de movimentos são concatenados aplicando-se o algoritmo de Viterbi para cálculo de melhor caminho segundo uma função custo que leva em conta vários fatores desejáveis na concatenação dos segmentos.

Os autores também consideram conhecido o texto falado no sinal de áudio, que é utilizado para adicionar ao movimento a prosódia visual. Tanto em [Graf 02] como em [Cosatto 03], a partir de várias horas de gravação, foram analisados quantitativamente o movimento facial e de cabeça que acompanham o sinal de voz e investigado sua relação com a estrutura prosódica do texto. Essas estatísticas sobre prosódia visual são utilizadas na síntese para o cálculo da prosódia visual apropriada.

3.10 Técnicas Empregadas

Nos artigos anteriores vimos vários métodos para resolver o problema de síntese de movimento de cabeça. Nesta seção vamos relacionar quais as idéias dessas técnicas que foram utilizadas na nossa proposta.

Algumas técnicas são promissoras e provavelmente fornecem bons resultados, mas não foram implementadas. Dentre elas podemos citar o HMM entrópico [Brand 99], que pode ser um modelo eficiente e simples para a síntese do movimento de cabeça e também a técnica de síntese de HMM suavizada de Voice Puppetry [Brand 99] e de Trajectory Model [Hofer 07b], que podem gerar uma síntese com HMM sem uma suavização arbitrária e ainda podem ser úteis se desejamos obrigar que a síntese passe por certos pontos, como no caso de animação por key points e junção de segmentos criados por diferentes modelos.

Várias técnicas foram implementadas e testadas. O modelo em dois níveis baseado em LDS, assim como o método de síntese restrita de Motion Texture [Li 02] é utilizado como uma opção para modelagem e síntese no nosso modelo. O uso de mapa exponencial, como sugere [Li 02], também foi empregado. O uso de $F0$ como um dos descritores de som é sugerido em vários artigos e também foi aproveitado.

A idéia de uma síntese baseada em colagem [Chuang 05], [Cosatto 03] também foi utilizada como uma opção para a síntese ou treinamento no nosso modelo. Também foi aplicada a idéia de um modelo de dois níveis, como [Sargin 08], onde um primeiro nível descobre o padrão e o segundo nível de modelagem relaciona esses padrões.

Capítulo 4

Método Proposto

O nosso objetivo é relacionar som e movimento de cabeça para uma posterior síntese do movimento de cabeça a partir do som. Vimos no capítulo 3, de Trabalhos Relacionados, vários métodos para resolver esse problema e antecipamos certas idéias e técnicas que iremos usar.

Neste capítulo vamos discorrer sobre como os dados são obtidos e quais os descritores usados. Vamos mostrar como analisamos e segmentamos o som e movimento. Também vamos apresentar nosso modelo e descrevemos o seu treinamento e síntese. Por fim apresentamos alguns resultados e aplicações.

4.1 Introdução

O nosso método de síntese do movimento de cabeça a partir do som consiste em duas etapas básicas: Treinamento e Síntese.

No treinamento, o nosso dado de entrada é um conjunto C_{train} composto de T sequências de movimentos de cabeça e seu correspondente conjunto de sons S_{train}.

$$C_{train} = \{C_1, C_2, \ldots, C_T\}$$
$$S_{train} = \{S_1, S_2, \ldots, S_T\}$$

onde as sequências C_i podem ter comprimentos diferentes.

No treinamento obtemos os parâmetros "ótimos" θ^* do modelo (de estrutura conhecida) que melhor expliquem a relação entre movimento e voz, segundo o método de máxima verossimilhança:

$$\theta^* = \arg\max_{\theta} P(C_{train}, S_{train} \mid \theta)$$

Na síntese, dados um som S_{in} e os parâmetros θ do modelo, devemos obter o movimento C_{in} que melhor explica a entrada S_{in} segundo o modelo de parâmetros θ.

4.2 Dados

4.2.1 Banco de Dados

Foi implementado no MATLAB um sistema de modelagem e rastreamento de dados baseado no KLT [Cunha 04]. Nesse sistema o usuário marca, com o mouse, no primeiro frame do vídeo (imagem acima e à esquerda na figura 4.1) pontos de arestas (que tem valor alto de gradiente) da face e o sistema se encarrega de rastrear esses pontos ao longo do vídeo, a forma do objeto rígido (a face) e também determina a posição (a rotação e translação 2D) em todos os outros frames do vídeo, como mostra a figura 4.1. Um problema desse tipo de técnica é que muitas vezes são necessários ajustes nos parâmetros ou uma nova escolha de pontos iniciais para que o rastreamento seja feito corretamente, rotações altas não são capturadas apropriadamente e a precisão do valor da rotação e translação é influenciada pelos movimentos faciais. Maiores detalhes dessa técnica e sua implementação podem ser encontradas em [Cunha 04].

Outro banco de dados que também utilizamos é o obtido no site de Hani Yehia [Yehia 02b], que foi usado em seus artigos [Yehia 98] e [Yehia 02]. Uma característica desse banco de dados é sua precisão, pois foi obtido com o sistema de captura de movimentos com marcadores OPTOTRACK [Yehia 02] com taxa de amostragem de 60 frames por segundo.

4.2.2 Descritores dos Dados

Para o movimento de cabeça, os descritores comumente usados são os ângulos de Euler. No sistema implementado podemos utilizar ângulos de Euler, mas preferimos o uso de mapa exponencial [Grassia 98], pois existe a vantagem do mapa exponencial ser uma representação localmente linear [Li 02].

O dado de movimento é expresso pela rotação (mapa exponencial 3D) e a translação (3D ou 2D, no caso de dados obtidos com o programa de rastreamento baseado no KLT). Além da translação e rotação também podem ser usados conjuntamente os seus vetores velocidade e aceleração.

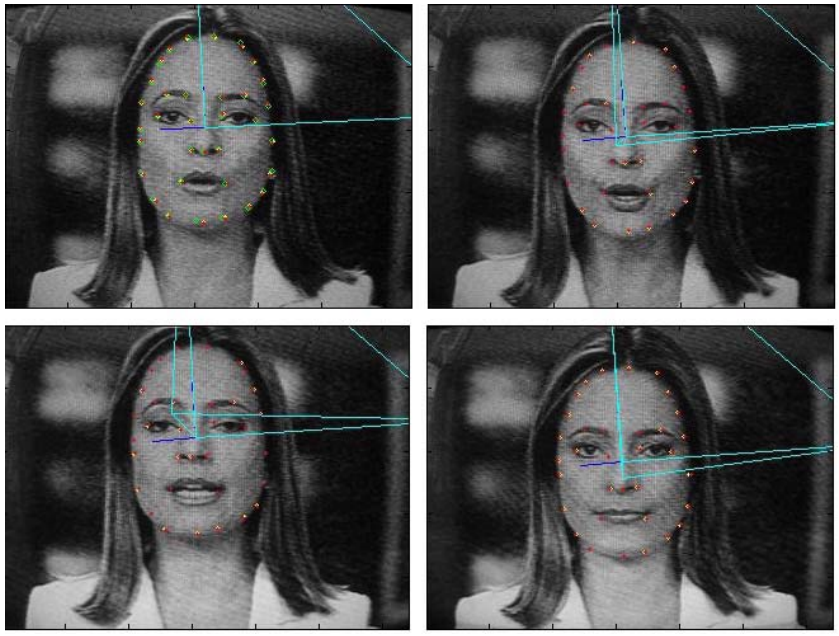

Figura 4.1: Exemplo de rastreamento e captura de movimento rígido da cabeça

O dado do movimento de cabeça C pode ser expresso da forma:

$$C = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N} \\ r_{3,1} & r_{3,2} & \cdots & r_{3,N} \\ t_{1,1} & t_{1,2} & \cdots & t_{1,N} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,N} \\ t_{3,1} & t_{3,2} & \cdots & t_{3,N} \end{bmatrix}$$

onde N é o número de frames de vídeo. $r_{i,j}$ é rotação de coordenada i no frame j e $t_{i,j}$ é a coordenada i da translação no frame j.

Se desejamos também usar a velocidade dC como descritor, concatenamos verticalmente C e dC, obtendo uma matriz de dimensão $12 \times N$. Na notação do MATLAB: $C = [C; dC]$.

Para os descritores de sinal de áudio existem várias opções e ainda não estão bem definidos na literatura quais os melhores descritores de áudio num dado problema [Brand 99]. Opções de descritores de áudio que podem ser utilizados na implementação são o MFCC (Mel-frequency cepstral coefficients), o espectro (obtido com a transformada de Fourier), a energia RMS, a frequência fundamental (F0) e LSP (Line

Spectrum Pairs). Além desses descritores também é comum o uso de suas primeira e segunda derivadas. Na implementação, todos esses descritores são calculados para o som de entrada e o usuário escolhe quais efetivamente vão ser usados. É claro que os descritores usados no treinamento devem ser os mesmos usados na síntese.

Por exemplo, se utilizamos como descritores de voz somente a frequência fundamental $F0$ e a energia E, a matriz S que representa os descritores de voz é da forma $S = [F0;\ E]$, isto é:

$$S = \begin{bmatrix} F0_1 & F0_2 & \cdots & F0_N \\ E_1 & E_2 & \cdots & E_N \end{bmatrix}$$

onde N é o tamanho do vetor de descritores de som.

Por fim, apesar de não ser absolutamente necessário, reamostramos a matriz de som S para que ela tenha o mesmo número de colunas que a matriz de movimento de cabeça C.

Existem vários pacotes para processamento de voz (e obtenção de descritores) para o MATLAB. Efetivamente usamos o toolbox de Malcolm Slaney [Slaney 98] e o de Dan Ellis [Ellis 06] para o cálculo de MFCC e espectro e rotinas de Hani Yehia [Yehia 02b] para o cálculo de F0 e LSP.

4.3 Segmentação e Classificação do Som

O objetivo final de nosso trabalho é a partir do som obter o movimento de cabeça. Uma primeira tarefa é segmentar e classificar o som para a posterior síntese. É possível fazer uma estimativa global do movimento a partir do som sem nenhuma segmentação explícita ou modelagem do som em unidades menores mais significativas, como em [Brand 99]. No entanto, correntemente se tenta segmentar o som em unidades mais simples para daí obter o movimento de cabeça.

Uma idéia é segmentar o som com o uso de seus descritores de forma que a segmentação do movimento de cabeça seja herdado da segmentação do som, como em [Chuang 05], que segmenta o som a partir das descontinuidades (onsets) de F0.

Uma outra possibilidade para a segmentação do som é com o uso da matriz de similaridade [Foote 01], [Cooper 03]. A similaridade entre dois vetores é o cosseno do ângulo entre eles. Nesse tipo de técnica se calcula a matriz de similaridade entre os vetores de descritores do som (suavizados) ao longo do tempo e daí segmentamos o som onde há máxima variação do descritor. A segmentação depende do descritor utilizado e da sua suavização. Podemos ainda segmentar com um algoritmo de menor caminho para DAG, onde o custo de um segmento é calculado com a matriz de similaridade e o custo de segmentação é um parâmetro fixo [Jensen 06]. Na

seção 4.7 apresentamos uma aplicação de segmentação de som usando matrizes de similaridade.

Uma característica interessante para a síntese de movimentos é se além de segmentarmos o som, também pudermos classificá-lo. Na segmentação descrita anteriormente com o método de similaridade, o som não é em geral classificado, mas é possível a sua classificação com o uso de uma medida de distância apropriada [Pampalk 04], [Pampalk 06].

Também é possível segmentar o som (e movimento) com o uso de texturas de movimento [Li 02]. Essa técnica foi criada originalmente para modelar movimentos, mas também pode ser usada para os descritores de som.

Uma outra alternativa para segmentação e classificação de sons é o uso de HMM, que tem seu uso respaldado pelo sucesso em reconhecimento de voz [Rabiner 93]. Em reconhecimento de voz a unidade básica usada são os fonemas, que são modelados por HMM's com estrutura left-right de 3 a 5 estados.

4.4 Modelo em Dois Níveis

Propomos, assim como Sargin [Sargin 08], que a segmentação de voz e movimento de cabeça sejam feitos independentemente. Isso é interessante pois em geral a tentativa de modelar som e movimento em unidades menores (ou padrões) leva a segmentações não coincidentes pois as características dos sinais de som e movimento são essencialmente diferentes.

Num primeiro nível de modelamento vamos descobrir, independentemente, quais os padrões de movimento e som do nosso banco de dados. Já no segundo nível do modelo vamos relacionar som e movimento. Vamos descrever a seguir esses dois níveis do modelo.

4.4.1 Nível 1: Padrões de Som e Movimento

A obtenção de padrões de som e movimento pode ser feita por vários métodos. Implementamos dois possíveis métodos: com o modelo do artigo Motion Texture [Li 02], baseado em LDS ou com HMM. Outras opções são possíveis, como por exemplo utilizando AR-HMM (HMM auto-regressivo), HMM de trajetória [Hofer 07b] ou técnicas baseadas em similaridade [Pampalk 06].

Vamos descrever em maiores detalhes o método implementado para obtenção de padrões de som e movimento baseado em HMM, que é semelhante ao método proposto por [Sargin 08]. Foi utilizado o HMM toolbox [Murphy 05] para os cálculos com HMM.

Nesse método supomos que um padrão pode ser modelado por um HMM com estrutura left-right, como em reconhecimento de voz [Rabiner 93]. A função de probabilidade de observação do HMM é contínua, definida pela distribuição normal com matriz de covariância diagonal, que tem a vantagem de ter menor número de parâmetros. Só é aconselhável o uso de uma matriz completa de covariância quando o banco de dados é suficientemente grande.

O número de estados N da estrutura left-right define o comprimento mínimo que cada padrão deve conter. Se desejamos capturar M possíveis padrões, devemos definir um HMM com uma estrutura de M HMM's letf-right paralelos, cada um com N estados. O exemplo a seguir esclarece a estrutura do HMM usada na implementação.

Exemplo 6 *Matrizes de um HMM letf-right*

Se o número de estados $N = 3$ e o número de padrões $M = 2$, temos que a matriz de transição de estados A e a matriz de probabilidade inicial B do HMM devem ser da forma:

$$A = \begin{bmatrix} a_1 & 1-a_1 & 0 & 0 & 0 & 0 \\ 0 & a_2 & 1-a_2 & 0 & 0 & 0 \\ \alpha_1 & 0 & a_3 & \alpha_2 & 0 & 0 \\ 0 & 0 & 0 & a_4 & 1-a_4 & 0 \\ 0 & 0 & 0 & 0 & a_5 & 1-a_5 \\ \beta_1 & 0 & 0 & \beta_2 & 0 & a_6 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} b_1 \\ 0 \\ 0 \\ 1-b_1 \\ 0 \\ 0 \end{bmatrix}$$

onde a_{ij} indica a probabilidade de transição do estado i para o estado j. b_i indica a probabilidade do estado inicial ser i. Temos também que $\alpha_1 + \alpha_2 + a_3 = 1$ e $\beta_1 + \beta_2 + a_6 = 1$.

A matriz de transição de estados A também pode ser representada graficamente com sua máquina de estados, conforme a figura

A matriz B indica que o estado inicial do HMM é o estado 1 (com probabilidade b_1) ou o estado 4 (com probabilidade $1 - b_1$). A matriz A indica que se o estado inicial for o 1, o próximo estado será o 2 (com probabilidade $1 - a_1$) ou continuará sendo o estado 1 (com probabilidade a_1). Ao chegar ao estado 3, ele se repetirá (com probabilidade a_3) ou irá para o estado 1 ou 4 (com respectivas probabilidades α_1 e α_2). O padrão **1** é composto pelos estados 1, 2 e 3 e o padrão **2** é composto pelos estados 4, 5 e 6.

Treinamento

O treinamento desse HMM é feito com o algoritmo de Baum-Welch, discutido em 2.5.1, que é o método tradicional de treinamento para HMM's.

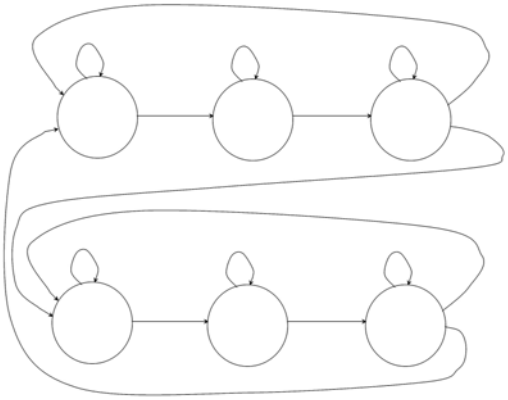

Figura 4.2: Máquina de estados da matriz A do Exemplo 6.

No treinamento obtemos os parâmetros do HMM: A matriz de transição de estados A, a matriz de probabilidade inicial B e também a média e a matriz diagonal de covariância da função de observação gaussiana.

Obtenção de Padrões

Obtidos os parâmetros do HMM no treinamento, queremos descobrir quais os padrões mais prováveis para o dado de treinamento. Estes padrões podem ser obtidos com o algoritmo de Viterbi, discutido em 2.5.1, que nos fornece a sequência de estados mais prováveis para cada uma das amostras do banco de dados de treinamento.

No exemplo a seguir mostramos em detalhes como achar esse padrões.

Exemplo 7 *Obtenção de padrões*

Vamos supor que o algoritmo de Viterbi no HMM do exemplo 6 nos fornece a sequência de estados

$$\prec 1, 1, 2, 2, 2, 3, 4, 5, 5, 6 \succ$$

Temos do HMM do exemplo 6 que o padrão discreto **1** é composto pela sequência de estados 1, 2, 3 e o padrão discreto **2** é composto pela sequência de estados 4, 5, 6. Logo temos que a sequência de estados inicial $\prec$1, 1, 2, 2, 2, 3$\succ$ indica o padrão **1** e a sequência final $\prec$4, 5, 5, 6$\succ$ indica o padrão **2**. Daí temos que a sequência de padrões relativa à essa sequência de estados é:

$$\prec 1, 1, 1, 1, 1, 1, 2, 2, 2, 2 \succ$$

Figura 4.3: Relação entre segmentos de som e movimento

isto é, os 6 primeiros frames são representados pelo padrão **1** e os 4 últimos frames são representados pelo padrão **2**.

4.4.2 Nível 2: Relação entre os Padrões de Som e Movimento

No nível 2 os dados de entrada são somente os padrões discretos de movimento e som obtidos no nível 1. A relação entre som e movimento vai ser modelada com um HMM onde a observação discreta é o padrão de som e o estado (que emite essa observação) é uma lista de padrões de movimento que ocorrem enquanto dura esse padrão de som. Por exemplo, se os padrões de som e movimento obtidos no nível 1 são os exibidos na figura 4.3, o HMM que relaciona som e movimento tem seus estados e observações expostos na figura 4.4. Para cada padrão de som não é relacionado somente um padrão de movimento, mas a lista de padrões de movimentos que ocorrem enquanto dura o padrão de som.

O número total de observações é o número de padrões possíveis de som, definido no primeiro nível. Já o número de estados do HMM é o número total de listas de padrões de movimento. No exemplo da figura 4.4 temos o total de 3 observações (3 padrões de som) e 5 estados (os estados são as listas: 1, 1-2, 2-3, 3-2-3 e 3).

Treinamento

O treinamento do HMM do segundo nível do modelo é bem simples, bastando aplicar o algoritmo de Baum-Welch (EM). Nesse caso, ao contrário do primeiro nível do modelo, a observação é discreta.

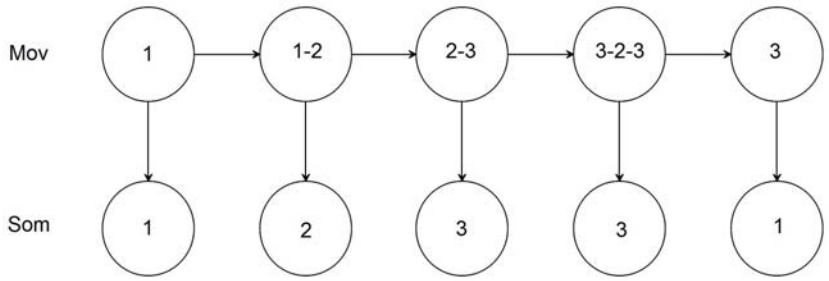

Figura 4.4: Exemplo de HMM relacionando som e movimento

4.5 Síntese

Na síntese, o sinal de entrada é somente o sinal sonoro e devemos obter o movimento de cabeça a partir desse sinal e do modelo previamente treinado. A síntese é composta dos sete passos descritos a seguir.

1. Do sinal sonoro de entrada devemos calcular os descritores do som, que devem ser os mesmos usados no treinamento.

2. Calcular os padrões de som que melhor explicam os descritores do sinal sonoro de entrada. Com isso obtemos uma trilha, como na segunda linha da figura 4.3, que indica para cada frame, qual o padrão de som mais provável. Esse cálculo depende de qual o tipo de modelo utilizado no primeiro nível do treinamento. Se usamos HMM, isso é feito com o algoritmo de Viterbi no HMM de som já treinado no primeiro nível de modelagem.

3. Com o HMM obtido no treinamento do segundo nível do modelo, obter a sequência de listas de padrões de movimento mais provável dado os padrões de som do passo 2. Isso é feito com o algoritmo de Viterbi, que fornece qual a sequência de estados (cada estado é uma lista de padrões de movimento que ocorrem enquanto dura um padrão de som) mais provável dado a sequência de observação (padrão de som) no HMM do segundo nível.

4. Obtenção dos Padrões de Movimento e seus respectivos comprimentos. No passo 3 foi obtido uma sequência de listas de padrões de movimento. Para obtermos a sequência de padrões devemos eliminar o primeiro elemento de cada lista, exceto na primeira lista de padrões de movimento. No exemplo 8 mostramos em detalhes como isso é feito. Precisamos ainda obter o comprimento de cada padrão de movimento. O nosso método não nos fornece

o comprimento exato de cada padrão de movimento, mas apenas estimativas baseadas nas durações dos padrões de som. O exemplo 8 mostra esses limites. Na seção 4.5.1 discorremos sobre as opções implementadas para a escolha desse comprimento.

5. Sintetizar o padrão de movimento no modelo de movimento do nível 1. Obtidos o padrão de movimento e seu comprimento devemos voltar ao nível 1 do modelo treinado para sintetizar o movimento. As possíveis opções para essa síntese são discutidas na seção 4.5.2.

6. Suavização da síntese obtida no passo 5. A suavização implementada consiste de dois filtros: inicialmente aplicamos o filtro mediano sobre 11 frames e em seguida o filtro gaussiano sobre 15 frames, como em [Sargin 08]. A suavização da síntese não é sempre necessária e depende do modelo utilizado. Por exemplo, se utilizamos a técnica de LDS (Motion Texture) ou HMM onde os descritores de movimento são somente as velocidades, não é necessário uma suavização. No entanto, se utilizamos HMM's e a posição da cabeça faz parte do descritor, a suavização é bastante necessária. Isso pode ser visto claramente nas figuras 4.5 e 4.6, obtidas respectivamente de [Busso 07] e [Sargin 08]. Pode ser notado que a síntese original da gaussiana é bastante "ruidosa" e que o comportamento local da síntese do movimento é basicamente determinado pela suavização, fazendo com que muitas vezes o movimento não seja realístico, como apontado em [Chuang 04].

7. União de extremos entre os segmentos sintetizados. O ponto inicial de cada segmento do movimento é definido pelo seu respectivo modelo (ou pelo ponto incial do segmento escolhido, no caso de colagem). Para que não haja descontinuidade entre as junções de segmentos sintetizados por diferentes padrões é necessário que o segmento posterior seja transladado para uma perfeita continuidade entre o ponto final do segmento anterior e o ponto inicial do segmento posterior. Se a síntese for pela técnica de LDS é possível obrigar que o ponto final do segmento anterior combine com o ponto inicial do segmento posterior a partir da resolução de um sistema linear, não sendo necessário uma translação.

Terminada a síntese do movimento, é interessante criar um vídeo com o movimento e o som de entrada. Para isso podemos usar um software como o Maya, que pode gerar um vídeo de um modelo de cabeça 3D se movendo de acordo com as rotações e translações já calculadas. A opção escolhida foi obter um modelo simples

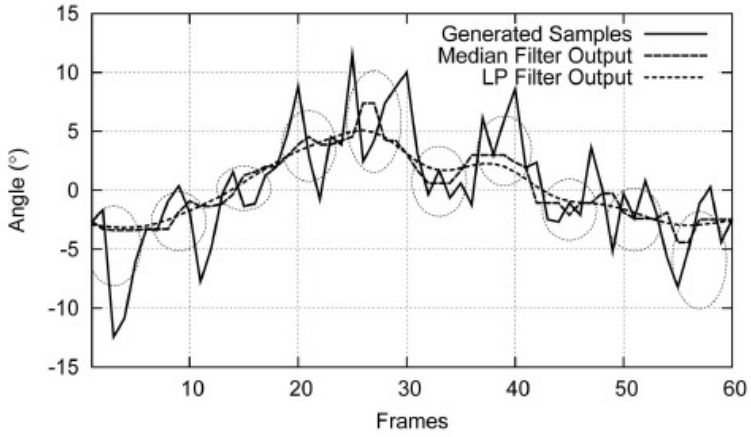

Figura 4.5: Suavização efetuada em [Sargin 08]

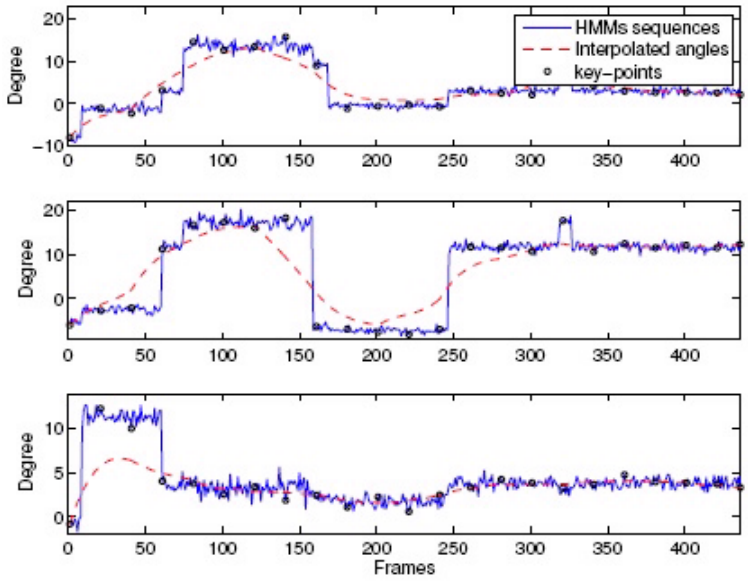

Figura 4.6: Suavização efetuada em [Busso 07]

de uma cabeça 3D e gerar o vídeo com o MATLAB. Obtivemos em [Nancy 97] o modelo de cabeça 3D "Nancy". Com isso podemos gerar o vídeo desse modelo se movendo de acordo com os dados obtidos na síntese e o som original. Nas figuras 4.10 e 4.11 temos amostras de vídeos gerados com a "Nancy".

Exemplo 8 *Obtenção dos Padrões de Movimento e seus comprimentos*

Se, como na figura 4.4, a sequência de observações de padrões de som obtida no passo 2 é $\prec 1, 2, 3, 3, 1 \succ$ e a sequência da lista padrões de movimento obtida no passo 3 é $\prec 1, 1-2, 2-3, 3-2-3, 3 \succ$, temos que a sequência padrões de movimento é $\prec 1, 2, 3, 2, 3 \succ$. Essa sequência de padrões é obtida eliminando o primeiro elemento de cada lista, exceto na primeira lista, isto é, eliminamos o 1 na segunda lista $(1-2)$, eliminamos o 2 na terceira lista $(2-3)$, eliminamos o 3 na quarta lista $(3-2-3)$ e é eliminado o 3 da quinta lista (3). Nota-se que o último padrão de uma lista é sempre o primeiro padrão da próxima, desde que a segmentação de som e movimento nunca coincida, o que é forçado na nossa implementação. Esse padrão que se repete indica qual o padrão de movimento nos pontos de segmentação do som.

Neste método não temos a duração exata de cada padrão de movimento, mas apenas limites provenientes da duração de cada padrão de som. Por exemplo, se na sequência de sons $\prec 1, 2, 3, 3, 1 \succ$ cada padrão dura 100 frames, temos que a sequência de padrões de movimentos $\prec 1, 2, 3, 2, 3 \succ$ tem os seguintes limites: O padrão **1** tem início no frame 1 e fim entre os frames 100 e 200, pois o padrão de movimento era **1** quando o som trocou do padrão **1** para o **2**. O padrão de movimento **2** tem início no último frame do padrão anterior +1 e fim entre os frames 200 e 300, e assim por diante.

4.5.1 Opções de escolha do comprimento dos padrões no passo 4 da síntese

Várias opções são possíveis para definir o tamanho de cada padrão de movimento. Foram implementadas algumas opções:

1. O tamanho de cada padrão será o tamanho médio desse mesmo padrão calculado no treinamento e multiplicado por um fator de ajuste de forma que a soma do tamanho de todos os padrões seja o tamanho total requerido na síntese, que é definido pelo tempo total do som de teste. Esta opção, ao contrário das outras, não necessariamente respeita os limites de duração individuais herdados dos padrões de som.

2. O tamanho de cada padrão é o escolhido na opção 1, se ele é factível (se respeita os limites de duração herdados dos padrões de som). Caso contrário, escolhemos o tamanho factível mais próximo ao tamanho dado na opção 1.

3. O tamanho escolhido é definido pelo tamanho do segmento do banco de dados desse padrão que é mais próximo ao calculado na opção 1. Se necessário reamostramos o segmento para que ele seja factível. Se a síntese for pelo método de colagem, esse será o segmento escolhido para colagem.

4. O tamanho escolhido é definido pelo segmento factível do banco de dados do padrão escolhido que tenha mínima distância entre seu ponto inicial e o ponto final do segmento sintetizado anteriormente.

5. O tamanho escolhido é definido pelo segmento do banco de dados desse padrão que tenha mínima distância entre seu ponto inicial e o ponto final do segmento sintetizado anteriormente. O segmento escolhido é reamostrado, se ele não for factível.

Tanto nessa opção como na anterior, a síntese é feita segmento a segmento. Nas três primeiras opções, é escolhido o tamanho (e segmento do banco de dados, no caso de colagem) de todos os padrões para só então ser feita a síntese do movimento.

4.5.2 Opções de modelos no passo 5 da síntese

Foram implementadas basicamente três técnicas para síntese de movimento no primeiro nível de modelamento:

- Motion Texture [Li 02], baseado na técnica de LDS.

- HMM, onde os descritores podem ser somente a posição ou somente a velocidade ou ainda uma mistura de posição, velocidade e aceleração. Para que o segmento a ser sintetizado tenha o tamanho requerido no passo 4 da síntese, repetimos a síntese do padrão desejado um número fixo de vezes para escolher aquele com comprimento mais próximo do desejado e, se necessário, reamostramos esse segmento para o tamanho requerido.

- Colagem, isto é, gerar movimentos de cabeça usando pedaços de movimentos reais, como em [Chuang 04] ou [Cosatto 03], e não a síntese de movimentos sintéticos gerados por um modelo, como nas técnicas anteriores. Nesse caso não sintetizamos o movimento no passo 5 da síntese, mas escolhemos no banco de dados (do padrão de movimento definido no passo 3 da síntese) aquele com o comprimento requerido no passo 4 da síntese.

4.6 Resultados

Antes de apresentar os resultados propriamente ditos, convém ressaltar algumas diferenças entre nosso método e alguns outros que serviram de inspiração.

A idéia de uma síntese baseada em colagem [Chuang 05], [Cosatto 03] tem o ganho de uma síntese realista, porém tem a desvantagem de ser uma técnica essencialmente "ad hoc". Usamos a idéia de síntese com colagem, mas guiada por um modelo. Notamos que problemas que acontecem no nosso método na junção de segmentos provavelmente são minimizados em [Chuang 05] e [Cosatto 03], mas com o custo da elaboração de funções de casamento bastante complexas.

Empregamos a idéia de um modelo de dois níveis de [Sargin 08], onde um primeiro nível descobre o padrão e o segundo nível relaciona esses padrões. No entanto, generalizamos o tipo de modelo utilizado no primeiro nível e no segundo nível usamos um modo diverso para relacionar tais padrões. Segundo nossos experimentos, a técnica de [Sargin 08] tende a gerar, na síntese, padrões de movimento com segmentações muito próximas às segmentações do som, o que não ocorre nos dados de treinamento nem no nosso método.

Apresentamos a seguir alguns resultados de experimentos com o método proposto.

Nas figuras 4.7, 4.8 e 4.9, a primeira linha mostra as 3 primeiras coordenadas (rotação) do banco de dados utilizado para treinamento, a penúltima linha mostra o sinal do som de entrada para a síntese e na última linha temos os descritores desse som. A segunda linha das figuras 4.7 e 4.8 indicam a primeira coordenada do movimento sintetizado e do movimento real (que não foi usado no treinamento) associado ao som de entrada. A segunda, terceira e quarta linhas da figura 4.9 mostram as três primeiras coordenadas (rotação) do movimento sintetizado e do movimento real (que não foi usado no treinamento) associado ao som de entrada. As marcas assinaladas nas sub-figuras correspondem aos pontos de segmentação.

Na figura 4.7 temos alguns resultados da síntese onde o banco de dados de treinamento consiste das amostras de número 9 a 55 de [Yehia 02b]. O sistema foi treinado com um HMM left-right de movimento com $M = 10$ e $N = 3$ (número de ramos e número de estados em cada ramo, respectivamente). Já o HMM de som tem uma estrutura com $M = 3$ e $N = 6$. No segundo nível de treinamento foram obtidos o total de 71 tipos de movimentos (lista de padrões). O cálculo do tamanho de cada padrão de movimento foi feito com a opção 3, conforme 4.5.1. A síntese do movimento (passo 5) foi feita pelo método de colagem, conforme 4.5.2. No vídeo correspondente da síntese foram observados movimentos realistas. Um erro observável pode ocorrer no final das sentenças, que algumas vezes tem movimentos intensos, como pode ser observado na segunda linha da figura 4.7. Esse movimento intenso não é comum no

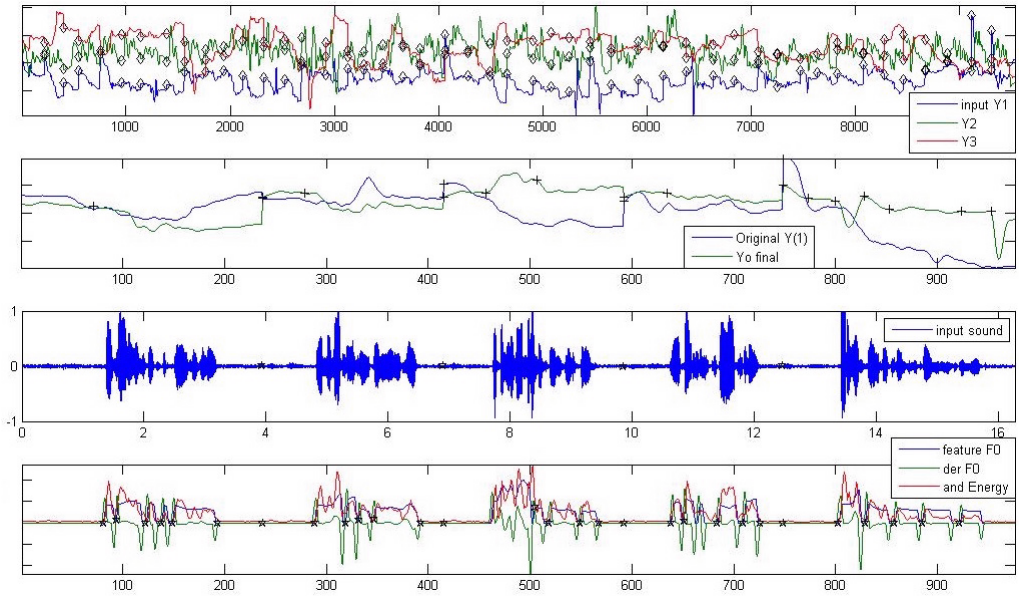

Figura 4.7: Resultado da Síntese com Colagem e Modelagem com HMM

final das sentenças do banco de dados.

Na figura 4.8 temos os resultados da síntese com o mesmo modelo treinado usado para a figura 4.7, só que o movimento foi obtido (passo 5) com a síntese no HMM, conforme 4.5.2. Note na segunda linha da figura 4.8 que a síntese com HMM é bastante ruidosa, como já foi observado nas figuras 4.5 e 4.6. A linha em vermelho indica a suavização dessa curva (passo 7 da síntese).

Na figura 4.9 temos resultados da síntese onde o banco de dados de treinamento consiste das amostras de número 9 a 15 de [Yehia 02b]. Utilizamos LDS como o modelo de movimento no nível 1 e a síntese é feita com a técnica de colagem (4.5.2). O movimento (no nível 1) é modelado com um total de 10 LDS e o som foi modelado utilizando um HMM com 3 padrões possíveis. No segundo nível de treinamento foram obtidos um total de 24 tipos de movimentos (lista de padrões). As linhas 2, 3 e 4 da figura 4.9 mostram o resultado da síntese. Notamos no vídeo dessa síntese que os resultados são convincentes.

Na tentativa de apresentar o vídeo final, nas figuras 4.10 e 4.11 temos vinte amostras do vídeo criado a partir do primeiro exemplo da figura 4.7. Na figura 4.10 temos os dados sintetizados e na figura 4.11 temos os dados originais de movimento.

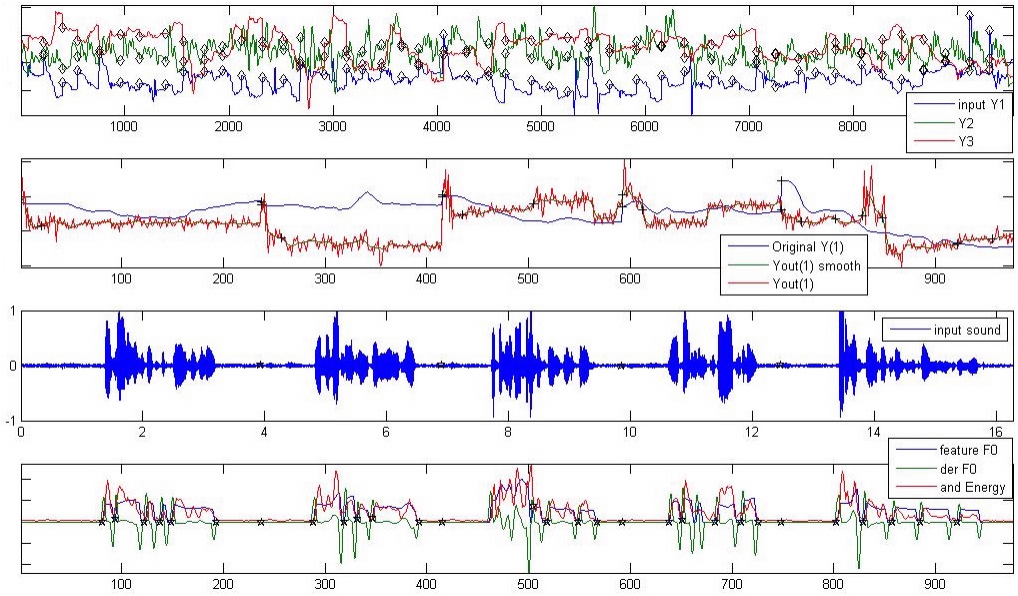

Figura 4.8: Resultado da Síntese e Modelagem com HMM

4.7 Aplicações

Apresentamos nesta seção uma aplicação de segmentação de som com o uso de matrizes de similaridade [Foote 01]. Foi implementado no MATLAB um sistema de exibição de fotos guiada pelo som onde a segmentação do som é calculada a partir da matriz de similaridade, mostrada na segunda linha da figura 4.12. A similaridade entre os pontos i e j do som é dada pelo cosseno do ângulo entre os descritores das posições i e j do som. As regiões brancas na figura 4.12 indicam alta similaridade e as regiões escuras indicam baixa similaridade. A dissimilaridade (novelty) num ponto i é calculada com a matriz de dissimilaridade da seguinte forma:

$$Novelty = Simil(Fut) + Simil(Pas) - 2Simil(Fut, Pas)$$

onde $Simil(Fut)$ é média (calculada com um filtro gaussiano) da similaridade numa janela posterior ao ponto i.

Os gráficos de dissimilaridade para diversos tamanhos de janela são apresentados nas últimas colunas da figura 4.12, onde quanto maior a janela, mais suavizado é o gráfico da dissimilaridade. A segmentação é feita nos máximos mais significativos do gráfico da dissimilaridade.

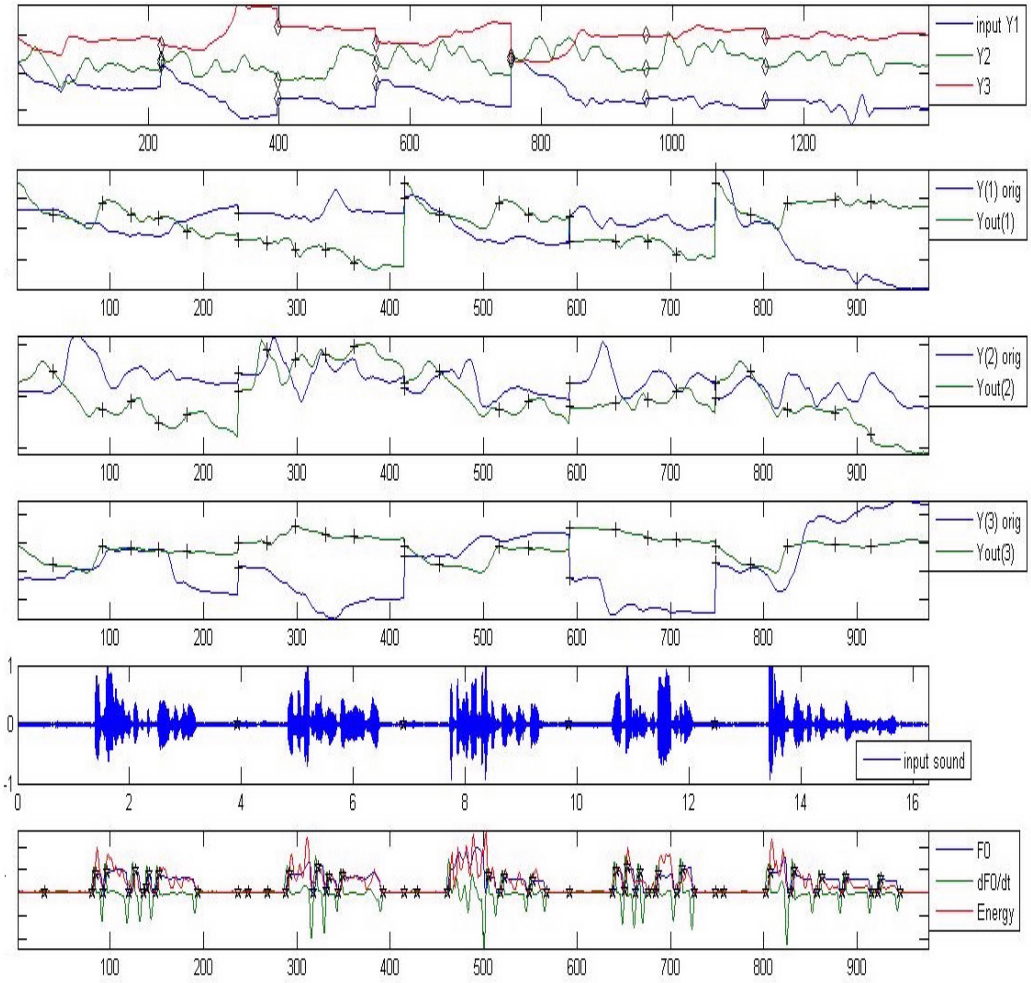

Figura 4.9: Resultado da Síntese com Colagem e Modelagem com LDS

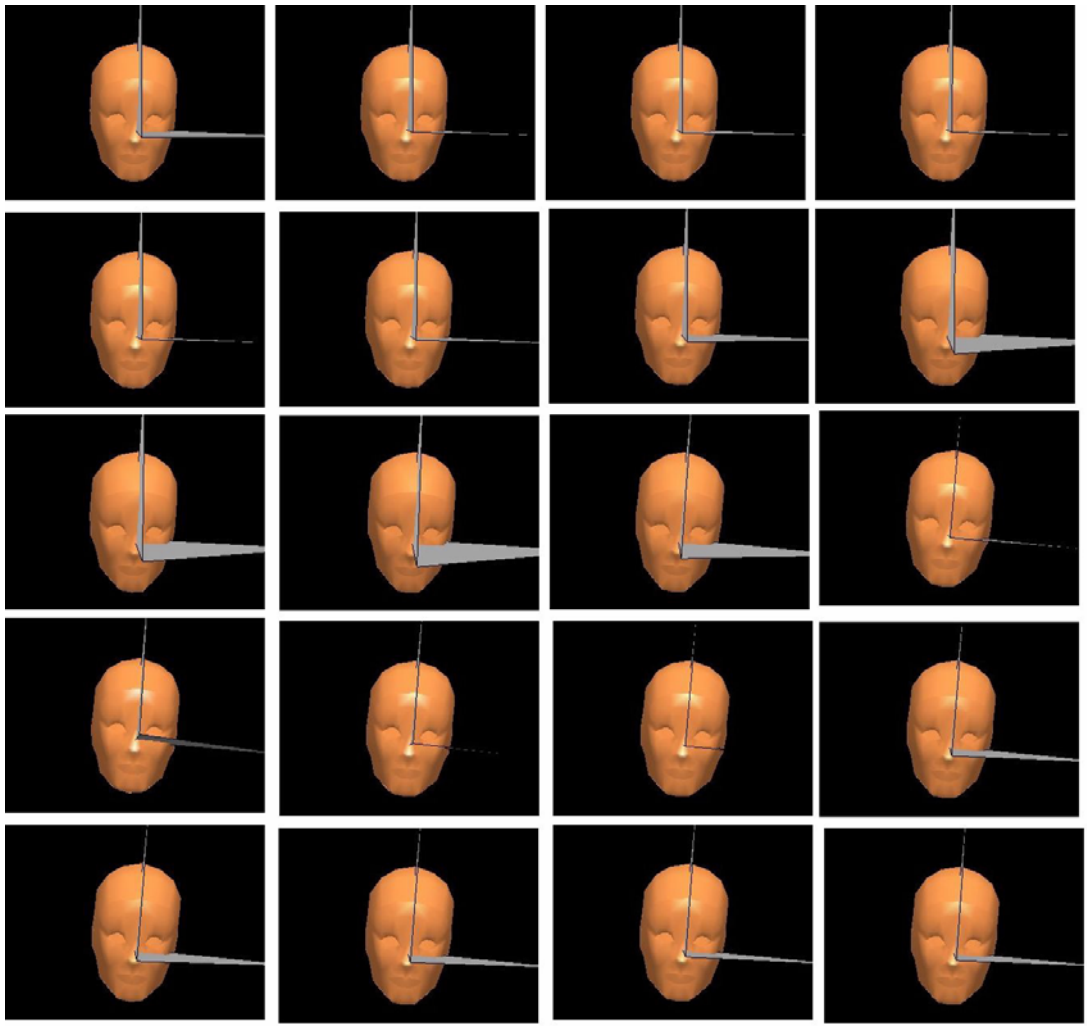

Figura 4.10: Amostras do vídeo do primeiro exemplo sintetizado na figura 4.7

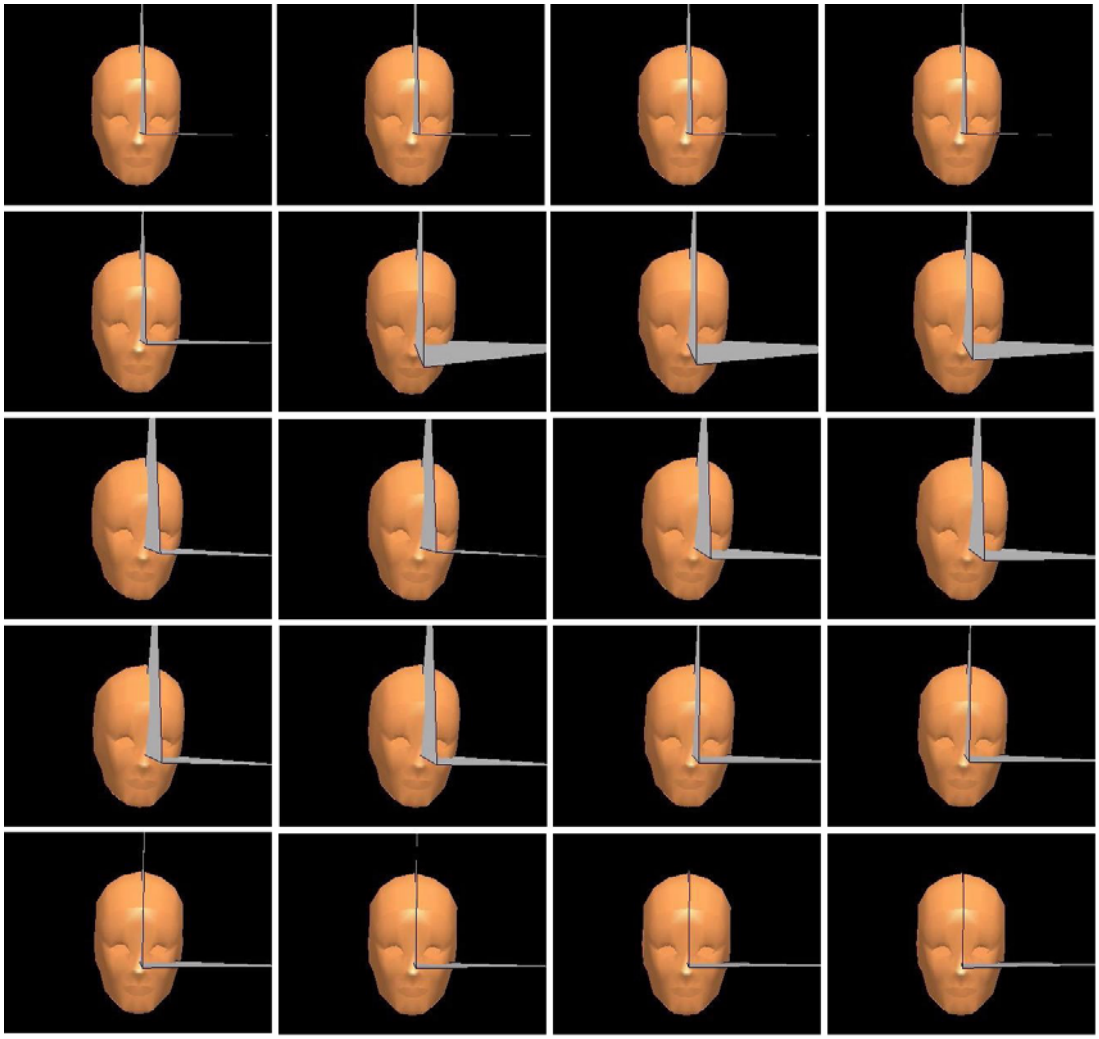

Figura 4.11: Amostras do vídeo com os dados originais do primeiro exemplo da figura 4.7

Também foi implementado a possibilidade do cálculo da segmentação com a utilização de um algoritmo de menor caminho para DAG [Jensen 06].

No sistema de exibição de fotos guiada pelo som é exibida uma nova foto nos pontos de segmentação do som. As fotos são amostradas de uma cadeia de Markov de fotos independente do som. Ainda há a possibilidade de obrigar a segmentação do som em certos pontos (no tempo) fixos definidos pelo usuário.

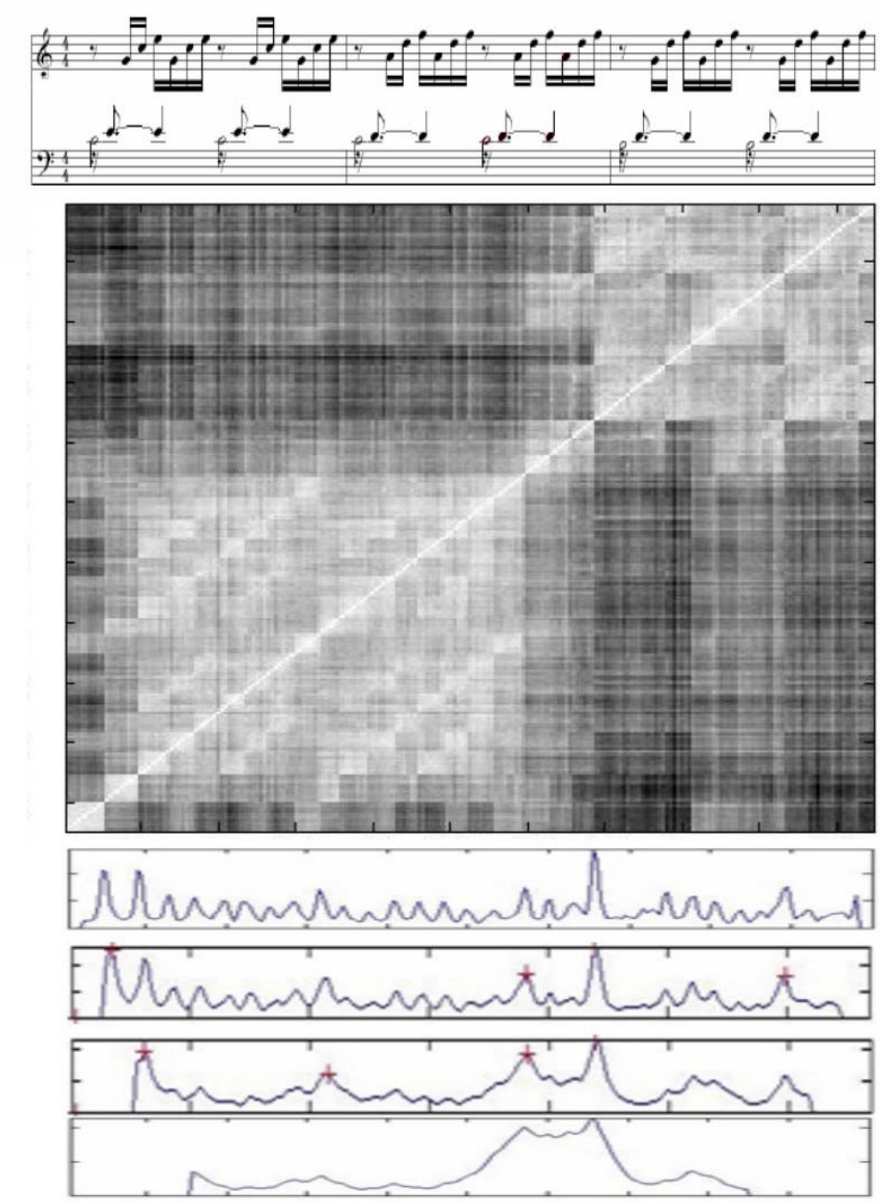

Figura 4.12: A matriz de similaridade e a dissimilaridade (novelty) das primeiras notas do prelúdio No 1 (BVW 846) de Bach

Conclusões

Esta tese apresenta um método para síntese de movimento de cabeça a partir de um sinal de voz como dado de entrada. Para isso usamos um modelo em dois níveis que relaciona voz e movimento a partir de dados reais capturados, como descrito no capítulo 4. Os modelos propostos fazem parte do arcabouço geral de redes bayesianas, examinadas no capítulo 2. Síntese de movimentos de cabeça ainda é um tópico relativamento novo, sem muitas referências bibliográficas. Artigos importantes dessa área e que foram inspirações para esse trabalho são discutidos no capítulo 3.

À primeira vista o problema proposto não é muito difícil, pois a dimensão do movimento de cabeça é baixa (6 dimensões). Movimentos faciais ou animação corporal, que aparentemente são fenômenos mais complexos e que envolvem dimensões bem maiores, têm a facilidade de terem uma correlação bem mais evidente com o som que os guia. No caso do movimento de cabeça essa correlação não é tão clara e ainda devemos levar em conta vários fatores como a semântica, a prosódia, estados emocionais e aspectos individuais do movimento.

O modelo proposto apresentou resultados razoavelmente realistas e exibe algumas propriedades interessantes como:

- Flexibilidade para incorporação de novos modelos de som e movimento sem alteração do arcabouço geral do modelo.
- Flexibilidade para incorporação de um terceiro nível no método para modelagem de conjunto de padrões de som e movimento.
- O uso de colagem de movimentos, como em [Chuang 04] e [Cosatto 03], nos permite a obtenção de movimentos realistas e sem sua dinâmica local regida por suavização, como em [Busso 07] e [Sargin 08]. No entanto, ao contrário de [Chuang 04] e [Cosatto 03], usamos modelos simples e sem técnicas "ad hoc".
- O método foi proposto e testado na relação de som e movimento de cabeça, no entanto ele tem potencial para relacionar vários outros tipos de sinais associados.

Algumas limitações do método ou da implementação também foram observadas:

- Muitas vezes o treinamento não nos fornece "bons" modelos, e em consequência os dados da síntese não são convincentes.
- Mesmo com um "bom" treinamento, movimentos "indevidos" ocorrem algumas vezes. Isso é particularmente notado no final das sentenças.
- A implementação não diferencia o silêncio final do inicial; porém é comum que antes do início do som haja um movimento de reposicionamento. Já no final da sentença é mais comum a cabeça permanecer relativamente imóvel.
- A translação usada na implementação (para não permitir descontinuidades na junção de segmentos) pode levar a posições finais não muito convincentes.

Trabalhos Futuros

A síntese realista de movimentos humanos, particularmente da cabeça, é bastante complexa. Muitos fatores devem ser levados em conta se desejamos uma animação de qualidade e consistência com a voz. A semântica do texto dito é essencial para uma síntese realmente apropriada. No contexto limitado deste trabalho vamos listar algumas idéias que podem melhorar a qualidade do nosso método.

- A animação tradicional é feita com key points, onde o animador determina a posição da cabeça em certos instantes chaves de tempo e depois completa a animação "interpolando" as posições definidas pelos key points. É importante para animadores que nosso método faça algo parecido, forçando certas posições específicas em tempos pré-definidos pelo usuário.

- Uma análise da qualidade da síntese, tanto teórica quanto prática (a partir de pesquisas onde observadores humanos classificam o realismo do movimento sintetizado comparado a movimentos reais). Essa análise é importante para determinar a qualidade do modelo e quais parâmetros e técnicas apresentam melhores resultados, assim como pode indicar outros tipos de técnicas que podem tornar o sistema mais eficiente.

- A suavização do movimento é necessária quando ele é sintetizado a partir de um HMM. Podemos evitar a suavização com o HMM de trajetórias [Hofer 07b] ou pelo método de síntese de HMM exposto em Voice Puppetry [Brand 99].

- O modelo em dois níveis proposto pode ser incrementado com o uso de HMM entrópico, como descrito em [Brand 99], que pode resultar em modelos mais simples (com menos parâmetros) que expliquem bem os dados. Podemos ainda usar redes bayesianas mais sofisticadas, como o HMM auto regressivo (ARHMM) ou o HMM de entrada e saída (IOHMM). Uma outra possibilidade para o aperfeiçoamento do método é o uso de um terceiro nível para modelagem de grupos de padrões de som ou movimento.

- A única opção implementada para evitar descontinuidades entre as junções de segmentos sintetizados por diferentes padrões é transladar o segmento posterior para a perfeita continuidade entre o ponto final do segmento anterior e o ponto inicial do segmento posterior. Essa translação nem sempre é desejável pois o ponto inicial faz parte do modelo. A soma de translações pode levar a movimentos não realísticos. Uma opção para evitar a translação é fazer uma suavização entre o fim de um segmento e o começo do seguinte.

- A síntese atualmente é feita a partir de um modelo ou por colagem. Uma opção interessante é misturar as duas técnicas numa mesma síntese. Um modo de se fazer isso é obtendo uma medida que nos informe se a síntese por colagem é boa. Se essa medida indica que um determinado segmento não é bem sintetizado por colagem, podemos sintetizar esse segmento a partir de um modelo.

- A escolha de segmentos para colagem ou a escolha do tamanho de cada segmento é feito de forma determinística. Isso eventualmente pode gerar repetições de movimentos, o que soa antinatural. Se essa escolha for feita de modo probabilístico, essas repetições ficarão mais raras. Um modo simples de se fazer isso é ao invés de escolher um segmento com mínima distância (ou tamanho), associar a esses segmentos uma medida de probabilidade inversamente proporcional a essa distância. Uma amostragem dessa distribuição nos dá uma maneira probabilística de escolha de segmentos (ou tamanhos) de um modelo para a síntese.

- A criação de banco de dados específicos para vários indivíduos pode levar a uma síntese bastante realista do estilo pessoal e além disso esses dados podem compor também um banco de dados geral que pode auxiliar na diversificação de movimentos.

- A criação de banco de dados relatando estados emocionais. Busso [Busso 07] e Erika [Chuang 05] fizeram pesquisas nesse campo e criaram ou modelos de estruturas semelhantes mas com diferentes parâmetros ou dividiram o banco de dados de acordo com os estados emocionais para a síntese de cada estado emocional específico. Uma opção interessante para a obtenção desse banco de dados é utilizar um equipamento como o Emotiv [Emotiv 09], que mede sinais neuronais a partir de EEG (Eletroencefalograma) e com o auxílio do software do sistema detecta estados emocionais em tempo real e ainda possui um giroscópio que nos fornece o movimento da cabeça. Infelizmente esse equipamento ainda não está sendo comercializado.

Referências Bibliográficas

[Applebaum 96] David Applebaum, *Probability and Information*, Cambridge University Press, 1996.

[Bechetti 99] Claudio Becchetti and Lucio Prima Ricotti, *Speech Recognition,* John Wiley&Sons, 1999.

[Bilmes 96] Jeff Bilmes, "What HMMs Can Do", IEICE Transactions in Information and Systems Vol.E89-D, No 3, March 2006, pp 869–891.

[Bilmes 97] Jeff Bilmes, "A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", Technical Report, University of Berkeley, ICSI-TR-97-021, 1997.

[Bishop 07] Christopher M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics), Springer, 1 edition, 2007.

[Brand 99] Brand, M.E., "Voice Puppetry", ACM SIGGRAPH, ISBN: 0-201-48560-5, pps 21-28, August 1999.

[Busso 05] C. Busso, Z. Deng, U. Neumann, and S.S. Narayanan, "Natural head motion synthesis driven by acoustic prosodic features", Computer Animation and Virtual Worlds, vol. 16, no. 3-4, pp. 283–290, July 2005.

[Busso 07] Carlos Busso et al., "Rigid Head Motion in Expressive Speech Animation: Analysis and Synthesis", IEEE Transactions on Audio, Speech and Language Preocessing, vol. 15, no. 3, pp. 1075–1086, March 2007.

[Cassell 94] J. Cassell et al., "Animated conversation: Rule-based generation of facial epression, gesture and spoken intonation, for multiple conversation agents", In Proceedings of ACM SIGGRAPH, pages 413–420. ACM Press, 1994.

[Charniak 91] Eugene Charniak, "Bayesian Networks without Tears", AI magazine, Volume 12. Issue 4. pp. 50-63, 1991.

[Chuang 04] E. Chuang, "Analysis, Synthesis, and Retargeting of Facial Expressions", PhD dissertation, Stanford University, 2004.

[Chuang 05] Erika Chuang and Cristoph Bregler, "Mood Swings: Expressive Speech Animation", ACM Transactions on Graphics, Vol. 24, No 2, April 2005, pp 331-347.

[Cooper 03] M. Cooper and J. Foote, "Summarizing Popular Music Via Structural Similarity Analysis", in Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2003).

[Cosatto 03] Cosatto, E.; Ostermann, J.; Graf, H.P.; Schroeter, J., "Lifelike talking faces for interactive services", Proceedings of the IEEE, Volume 91, Issue 9, Page(s): 1406-1429, Sept. 2003.

[Costa 01] Maurizio Costa et al., "Visual Prosody Analysis for Realistic Motion Synthesis of 3D Head Models", Proc. of ICAV3D'01 - International Conference on Augmented, Virtual Environments and 3D Imaging, pp. 343-346, Mykonos, Greece, May 30-June 01, 2001.

[Cunha 02a] Anderson M. Cunha and Luiz Velho, "Hidden Markov Models", Technical Report TR-2002-02 - VISGRAF Laboratory, IMPA, January 2002.

[Cunha 02b] Anderson M. Cunha, Ralph Teixeira, and Luiz Velho, "Discrete Scale Spaces", Proceedings of the International Symposium on Mathematical Morphology VI. CSIRO Mathematical and Information Sciences, pp. 241-251, April 2002.

[Cunha 03a] Anderson M. Cunha and Luiz Velho, "Metodos Probabilísticos para Reconhecimento de Voz", Technical Report TR-2003-04 - VISGRAF Laboratory, IMPA, June 2003.

[Cunha 03b] Anderson M. Cunha and Luiz Velho, "Reconhecimento de Digitos com HMM", Technical Report TR-2003-04 - VISGRAF Laboratory, IMPA, August 2003.

[Cunha 04] Anderson M. Cunha and Luiz Velho, "Rastreamento e Modelagem de um Objeto Rigido num Video", Technical Report TR-2004-03, VISGRAF Laboratory, IMPA, 2004.

[Deng 04] Z. Deng, C. Busso, S. Narayanan, and U. Neumann, "Audio-based head motion synthesis for avatar-based telepresence systems", ACM SIGMM 2004 Workshop on Effective Telepresence (ETP 2004), New York, NY, pp. 24–30, ACM Press, 2004.

[Dugad 96] R. Dugad and U. B. Desai, "A Tutorial on Hidden Markov Models". Tech. Report: SPANN-96.1, 1996.

[Ellis 06] Dan Ellis website, http://www.ee.columbia.edu/~dpwe/resources /matlab/rastamat/

[Emotiv 09] Emotiv webpage, Neuro-signal acquisition and processing wireless neuroheadset, http://www.emotiv.com/developers.html

[Foote 01] J. Foote, and S. Uchihashi, "The Beat Spectrum: A New Approach to Rhythm Analysis", in Proc. International Conference on Multimedia and Expo (ICME), 2001.

[Ghahramani 98] Z. Ghahramani, "Learning Dynamic Bayesian Networks", In C.L. Giles and M. Gori (eds.), Adaptive Processing of Sequences and Data Structures . Lecture Notes in Artificial Intelligence, 168-197. Berlin: Springer-Verlag. 1998.

[Graf 02] Hans Peter Graf, Eric Cosatto, Volker Strom, Fu Jie Huang, "Visual Prosody: Facial Movements Accompanying Speech", pp.0396, Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FG'02), 2002.

[Grassia 98] Grassia, F. Sebastian. "Practical Parameterization of Rotations Using the Exponential Map". Journal of Graphics Tools, 3(3):29-48, 1998.

[Haykin 01] Simon Haykin, *Redes Neurais*, Bookman, Porto Alegre, 2001.

[Heckerman 96] D. Heckerman, "A tutorial on learning with Bayesian networks", Microsoft Research tech. report, MSR-TR-95-06, 1996.

[Hertzmann 03] A. Hertzmann, "Machine Learning for Computer Graphics: A Manifesto and Tutorial", Proc. Pacific Graphics 2003. pp. 22-36. Invited paper. Banff, Alberta. October 2003.

[Hertzmann 04] A. Hertzmann, "Introduction to Bayesian Learning", SIGGRAPH 2004 Course Notes.

[Hill 01] H. Hill and A. Johnston, "Categorizing sex and identity from the biological motion of faces", Current Biology, vol. 11, no. 11, pp. 880–885, June 2001.

[Hofer 07] Gregor Hofer and Hiroshi Shimodaira, "Automatic head motion prediction from speech data", In Proc. Interspeech 2007, Antwerp, Belgium, August 2007.

[Hofer 07b] Gregor Hofer, Hiroshi Shimodaira, and Junichi Yamagishi, "Speech-driven head motion synthesis based on a trajectoy model", Poster at SIGGRAPH 2007.

[Hu 96] Jianying Hu, Michael Brown and William Turin, "HMM Based On-Line Handwriting Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 18, n. 10, October, 1996.

[James 81] Barry R. James, *Probabilidade: um curso em nível intermediário.* Projeto Euclides, IMPA, 1981.

[Jelinek 97] Frederick Jelinek, Statistical Methods for Speech Recognition. MIT Press, 1997.

[Jensen 06] K. Jensen, "Multiple scale music segmentation using rhythm, timbre and harmony", EURASIP Journal on Applied Signal Processing Special issue on Music Information Retrieval Based on Signal Processing, 2006.

[Kleinrock 75] Leonard Kleinrock, *Queueing Systems*, John Wyley & Sons, 1975.

[Krause 99] Paul J. Krause, "Learning probabilistic networks, The Knowledge Engineering Review", Cambridge University Press, Volume 13, Issue 4, pp. 321-351, 1999.

[Li 02] Yan Li et al., "Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis", Proceedings of SIGGRAPH 2002, pp. 465–472.

[Linde 80] Y. Linde, A. Buzo, and R. Gray. "An algorithm for vector quantizer design", IEEE Transactions on Communications, 28(1):84– 95, Jan 1980.

[Libet 85] Libet, B. "Unconscious cerebral initiative and the role of conscious will in voluntary action". Behavioral and Brain Sciences, 8:529-566, 1985.

[Munhall 04] K. G. Munhall, J. A. Jones, D. E. Callan, T. Kuratate, and E. Bateson, "Visual prosody and speech intelligibility: Head movement improves auditory speech perception", Psychological Science, vol. 15, no. 2, pp. 133–137, February 2004.

[Mumford 02] David Mumford, "Pattern Theory: the Mathematics of Perception", Proceedings of the International Congress of Mathematicians, Beijing, 2002, vol. 1, Higher Educ. Press, Bejing, 2002. ICM, 2002.

[Murphy 02] Kevin Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning", PhD Thesis, UC Berkeley, Computer Science Division, July 2002.

[Murphy 05] HMM toolbox. http://people.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[Nancy 97] Cindy Ballreich, Nancy humanoid body under Matlab, http://www.robots.ox.ac.uk/~wmayol/3D/nancy_matlab.html

[Nechyba 00] M. C. Nechyba, "Spring 2000 Lecture Notes - EEL6935: Machine Learning in Robotics II", 2000.

[Pampalk 04] E. Pampalk, "A Matlab Toolbox to Compute Similarity from Audio", Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04), 2004.

[Pampalk 06] E. Pampalk, "Computational Models of Music Similarity and their Application to Music Information Retrieval". Doctoral Thesis, Vienna University of Technology, Austria, March 2006.

[Pearl 88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kaufmann, 1988.

[Pearl 96] Judea Pearl. Faculty Research. Online Presentation, Transcript on Slides of 1996.

[Pearl 00] Judea Pearl. *Causality: Models, Reasoning, and Inference*, Cambridge University Press, 2000.

[Rabiner 93] Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of Speech Recognition,* Prentice-Hall, 1993.

[Russell 02] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach,* Prentice Hall; 2 edition, December 2002.

[Rabiner 89] L. R. Rabiner, "A Tutorial in Hidden Markov Models and Selected Applications in Speech Recognition", Proc. of the IEEE, 77(2):257–286, 1989.

[Roweis 99] Sam Roweis & Zoubin Ghahramani, "A Unifying Review of Linear Gaussian Models", Neural Computation 11(2) (1999) pp.305-345, 1999.

[Sargin 08] M.E. Sargin, Y. Yemez, E. Erzin, and A.M. Tekalp, "Analysis of Head Gesture and Prosody Patterns for Prosody-Driven Head-Gesture Animation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, No. 8, pp. 1330-1345, August 2008

[Shannon 48] C. E. Shannon, *A Mathematical Theory of Communication*, Reprinted with corrections from The Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, July, October, 1948.

[Slaney 98] Malcolm Slaney, "Auditory Toolbox Version 2", Technical Report #1998-010, Interval Research Corporation, 1998.

[Smyth 98] P. Smyth, "Belief networks, hidden Markov models, and Markov random fields: a unifying view", Pattern Recognition Letters, 1998.

[Soon 08] Soon, Chun Siong et al., "Unconscious determinants of free decisions in the human brain", Nature Neuroscience Published online: 13 April 2008.

[Tokuda 00] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorigthms for HMM-based speech synthesis", in Proc. ICASSP 2000, pp.1315–1318, Jun. 2000.

[Wegner 99] Wegner, D. M., & Wheatley, T. P. "Apparent mental causation: Sources of the experience of will", American Psychologist, 54, 480-492, 1999.

[Wegner 03] Wegner, D. M. "The mind's best trick: How we experience conscious will", Trends in Cognitive Science, 7, 65-69, 2003.

[Wegner 05] Wegner, D. M. "Who is the controller of controlled processes?", R. Hassin, J. S. Uleman, & J.A. Bargh (Eds.), The new unconscious (pp. 19-36). New York: Oxford University Press, 2005.

[Wegner 07] Wegner, D. M. "Dangers of brain-o-vision", Science, 315-1078, 2007

[Wegner 08] Wegner, D. M. "Self is magic". In J. Baer, J. C. Kaufman, & R. F. Baumeister (Eds.), Are we free? Psychology and free will (pp. 226-247). New York: Oxford University Press, 2008.

[Yehia 98] Yehia, Hani C., Rubin P., Vatikiotis-Bateson, E., "Quantitative Association of Vocal-Tract and Facial Behavior", Speech Communication, vol.26(1-2), pp.23-43, 1998.

[Yehia 02] Yehia, Hani C., Kuratate, T., Vatikiotis-Bateson, E., "Linking Facial Animation, Head Motion and Speech Acoustics", Journal of Phonetics. 30, 555-568, 2002.

[Yehia 02b] Hani C. Yehia website, http://www.cefala.org/~hani/