Dante Augusto Blauth

Localização do locutor em ambiente de videoconferência utilizando sinal de áudio e vídeo

São Leopoldo

Fevereiro de 2010

Dante Augusto Blauth

Localização do locutor em ambiente de videoconferência utilizando sinal de áudio e vídeo

Dissertação submetida a avaliação como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada.

Orientador:
Marta Becker Villamil

UNIVERSIDADE DO VALE DO RIO DOS SINOS

São Leopoldo

Fevereiro de 2010

Agradecimentos

Esse é o único capítulo da dissertação no qual o autor pode expressar algum sentimento, portanto farei uso dessa oportunidade. A comunidade científica se organizou de forma que novas pesquisas são baseadas em conhecimento comprovado por pesquisas anteriores. Na minha dissertação utilizo técnicas propostas por diversos pesquisadores, sem as quais eu não chegaria muito longe, mas esse reconhecimento e agradecimento está oficializado pelas referências bibliográficas e citações ao decorrer do texto de todos autores cujos trabalhos me beneficiei.

Portanto, esse capítulo é necessário para deixar registrado as pessoas que foram importantes para o desenvolvimento do trabalho mas que acabaram não aparecendo nos demais capítulos. Um agradecimento especial ao professor Cláudio Rosito Jung, por ter me selecionado como bolsista, por seu incrível conhecimento científico, por sua habilidade em gerenciar projetos de pesquisa e por ter sido meu orientador por 19 meses. Agradeço a todos os professores do Programa de Pós-Graduação em Computação Aplicada da Unisinos, mas em especial aos professores Luiz Paulo Luna de Oliveira e Adelmo Luis Cechin, pela fantástica didática empregada nas aulas decorrente do sólido conhecimento que possuem em suas áreas. Agradeço ao Vicente Minotto, bolsista da graduação que trabalhou comigo no projeto; além de um ótimo programador é também um grande amigo, o que propiciou ao projeto bons momentos de diversão. Agradeço a Hewlett Packard, pelo seu programa de pesquisa, que financiou meu curso e ainda me fez experimentar situações que antes via distantes de mim, como videoconferências com pesquisadores de várias partes do mundo e reuniões de projetos de pesquisas para desenvolvimento de novos produtos.

Resumo

A localização do locutor ativo em ambientes de videoconferência traz benefícios importantes, como transmitir apenas o sinal relativo a este locutor, reduzindo a quantidade de informações trafegadas na rede, eliminar ruídos indesejados e ainda possibilitar focar a câmera no locutor. Embora essa tarefa não seja inovadora, ainda está sendo aperfeiçoada, visto que ambientes reais de videoconferência possuem ruído, problemas de iluminação, etc., o que torna o objetivo mais difícil de ser alcançado de maneira satisfatória. O presente trabalho apresenta uma metodologia para localização do locutor ativo em videoconferência, utilizando o sinal de áudio capturado por um arranjo de microfones e o sinal de vídeo capturado por uma web-câmera. No tratamento do sinal de vídeo, utiliza-se um algoritmo para detecção de faces, que é modificado para para diminuir seu custo computacional. O sinal de áudio é processado no domínio de frequência, utilizando-se o algoritmo SRP-PHAT para localizar a fonte sonora. A fusão dos dois sinais é feita através de uma função ponderada e o resultado consitui os observáveis de um HMM (Modelo Escondido de Markov), desenhado para melhorar a coerência temporal da localização. O resultado dos experimentos mostra que o método diminuiu em torno de 90% os erros de localização em comparação à localização que utiliza apenas sinal de áudio.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

DFT: Transfomada Discreta de Fourier (*Discrete Fourier Transform*).

FDP: Função Densidade de Probabilidade.

FFT: Transformada rápida de Fourier (*Fast Fourier Transform*).

LSF: Linhas Espectrais de Frequências.

TDE: Tempo de atraso estimado (*Time Delay Estimation*).

TDOA: Tempo de atraso da chegada (*Time Delay of Arrival*).

HMM: Modelos Escondidos de Markov (*Hidden Markov Models*).

RGB: É a abreviatura do sistema de cores aditivas formado por Vermelho (*Red*), Verde (*Green*) e Azul (*Blue*).

SRP: Potência da resposta direcionada (*Steered Response Power*)

PHAT: Transformada de Fase (*Phase Transform*)

VAD: Detecção da Atividade de Voz (*Voice Activity Detection*).

Sumário

1 Introdução

As técnicas que objetivam a comunicação entre pessoas dos mais diversos lugares do mundo sempre foram o foco de muitas pesquisas para inovações e aperfeiçoamentos. Atualmente, os ambientes de comunicação à distância envolvendo duas ou mais pessoas, com transmissão de sinais de áudio e vídeo, conhecidos como videoconferência, vêm recebendo bastante atenção dos pesquisadores.

Um dos problemas em aplicações de videoconferência é a presença de ruído de fundo, que pode degradar consideravelmente a qualidade do sinal de áudio transmitido. O ruído não apenas prejudica a clareza da comunicação, mas deixa a transmissão menos eficiente, pois dificulta a compactação dos dados. Ambientes de videoconferências mais sofisticados possuem um arranjo de microfones, que pode ser utilizado para amplificar o sinal sonoro proveniente de uma determinada localização espacial em detrimento do ruído de fundo (VEEN; BUCKLEY, 1988). Outro objetivo possível a sistemas de videoconferência é focar a câmera em direção do locutor ativo automaticamente, para melhorar o contato visual e tornar a comunicação mais eficiente.

Para ambos objetivos é desejado localizar e isolar a pessoa que está falando a cada instante de tempo (WANG; BRANDSTEIN, 1999; ZOTKIN et al., 2000). Em sistemas de videoconferência, informações de áudio e vídeo estão disponíveis para o desenvolvimento de técnicas de localização do locutor, e podem ser utilizadas de forma individual ou conjunta. Esse trabalho se propõe a elaborar um método robusto de localização do locutor ativo em uma videoconferência, utilizando para isso técnicas de Processamento de Sinais, Aprendizado de Máquinas e Visão Computacional.

1.1 O Problema

Localizar e isolar o locutor ativo em uma videoconferência não é um assunto inovador, mas em um ambiente real de videoconferência alguns modelos podem falhar em seu objetivo. Nos ambientes reais normalmente há outras fontes sonoras, como ar-condicionados, ventiladores,

ruído de computadores e talvez outras pessoas falando ao fundo. Todas essas fontes sonoras fazem parte do som capturado pelos microfones e constituem o elemento chamado ruído. O próprio sistema dos microfones também contribui em parte pelo ruído, que se soma ao sinal que realmente interessa, o do locutor ativo.

Ao sair da fonte sonora até chegar ao microfone, o som percorre vários caminhos. Normalmente há um caminho direto, sem barreiras entre o locutor e o microfone, mas também há inúmeros outros caminhos onde a onda sonora encontra obstáculos e é refletida produzindo ecos denominados reverberação, e após ser refletida algumas vezes a onda sonora finalmente encontra o microfone e se soma ao sinal capturado. Dessa forma, a estimativa da fonte sonora utilizando apenas a informação de áudio é prejudicada, e a informação visual pode ser utilizada para melhorar a estimativa da fonte sonora. De fato, a fonte sonora de interesse em aplicações de videoconferência são os participantes, e técnicas de detecção de pessoas (ou faces) a partir de sequências de vídeo podem ser utilizadas para auxiliar na localização da fonte sonora.

1.2 Objetivo

O objetivo desse trabalho foi desenvolver um algoritmo para conseguir localizar o locutor ativo em um ambiente com as condições reais de uma videoconferência. Foi utilizado para essa tarefa o sinal de áudio vindo de um arranjo de microfones e o sinal de vídeo proveniente de uma câmera.

Para que o objetivo geral fosse atingido, alguns objetivos específicos foram perseguidos neste trabalho:

- Estudar e avaliar algoritmos já existentes de localização por áudio.
- Estudar e implementar algoritmos de detecção de faces em vídeo, introduzindo melhorias para reduzir o custo computacional.
- Estudar e implementar técnicas de análise multimodal para integrar as informações de áudio e vídeo.
- Avaliar o custo computacional das técnicas implementadas/desenvolvidas.

Deve-se salientar que o problema de análise multimodal envolve diversos problemas complexos (análise por áudio, análise por vídeo e a combinação dos dois).

1.3 Contribuição

A interação homem-máquina tem se aperfeiçoado a cada ano que passa. Com o constante aumento da capacidade de processamento dos computadores, mais informações podem ser processadas para uma tomada de decisão. Essa evolução permite que outras fontes de informação possam ser utilizadas para aumentar a certeza no reconhecimento de determinado padrão, princípio básico da Visão Computacional. A contribuição desse trabalho é o desenvolvimento de um método de localização do locutor utilizando, para aprimorar a eficiência do resultado, a informação do vídeo. Fazendo analogia ao comportamento do ser humano, que utiliza informação de todos os sentidos para interagir com o mundo a sua volta (MARAGOS; POTAMIANOS; GROS, 2008), com a informação de áudio e vídeo o método se torna mais robusto do que utilizando apenas uma dessas fontes de informação, quando a fusão utilizada propicia esse resultado. O 'como' utilizar essas duas fontes de informação se torna fundamental para o resultado da localização. O método precisou ter um custo computacional viável para sua empregabilidade.

1.4 Estrutura do texto

O trabalho está estruturado da seguinte forma: o capítulo 2 apresenta conceitos que são utilizados no decorrer do trabalho e explica, de forma sucinta, as técnicas empregadas. No capítulo 3 é feita a revisão da bibliografia existente para os temas abordados no trabalho: detecção de pele e face usando vídeo, detecção de atividade de voz através de áudio, localização da fonte sonora e análise multimodal utilizada na fusão dos sinais de vídeo e áudio. No capítulo 4 é apresentado o modelo proposto, desenvolvido com base na revisão bibliográfica. O capítulo 5 apresenta os resultados experimentais e o capítulo 6 apresenta as conclusões e possíveis direções para trabalhos futuros.

2 *Conceitos Básicos*

Esse capítulo apresenta alguns conceitos básicos nas áreas de processamento de sinais/imagens, visão computacional e reconhecimento de padrões necessários para o melhor entendimento do algoritmo proposto neste trabalho. Técnicas mais específicas ao objetivo do projeto e que ainda estão sendo estudadas e aperfeiçoadas são vistas no capítulo de Revisão Bibliográfica.

2.1 Visão Computacional

O início do trabalho se faz com técnicas de processamento de sinais, os quais vêm de fontes de vídeo e áudio. Os sinais de vídeo são fornecidos por uma câmera, que produz uma sequência de imagens (quadros) ao longo do tempo. Uma imagem digital é formada por um conjunto de pixels, que são os menores elementos presentes em uma imagem (GONZALEZ; WOODS, 1992). Em uma imagem colorida, cada pixel é representado por um vetor de três elementos, que carrega a informação de cor do pixel, sempre associado um espaço de cor específico, e é baseado nesse vetor que se faz o reconhecimento de padrões, merecendo assim uma atenção especial.

Existem diferentes espaços de cores (formas diferentes de codificar a informação de cor de um pixel), entre eles os mais comuns são o RGB e o YCbCr (GONZALEZ; WOODS, 1992). O sistema RGB (Figura 2.1) é composto por cores aditivas, cujas primitivas são o Vermelho (*Red*), Verde (*Green*) e Azul (*Blue*). O YCbCr (Figura 2.2) é um espaço de cor obtido por uma tranformação linear do RGB para isolar a componente de luminosidade, comumente utilizado em sistemas de vídeo. A componente Y corresponde à luminosidade, sendo essencialmente uma cópia a preto e branco da imagem. As componente Cb e Cr contêm informação sobre as componentes azul e vermelha respectivamente. A componente da cor verde é omitida pois pode ser calculada a partir da Cb e Cr. A separação da componente de luminosidade das componentes de cor é particularmente importante para reconhecimento de padrões baseados na cor, por exemplo a detecção de pele, pois busca eliminar as diferenças causadas pelas condições de

luminosidades entre diversos ambientes (SCHMUGGE et al., 2007). Mais detalhes sobre o problema de detecção de cor de pele serão vistos no capítulo seguinte.

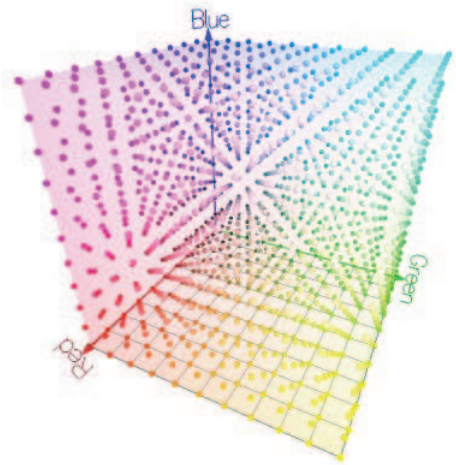

Figura 2.1: Espaço de cor RGB, onde os eixos representam as cores vermelho (R), verde (G) e azul (B) respectivamente. Fonte: http://www.couleur.org

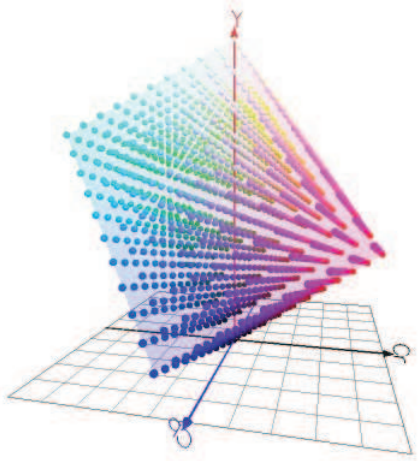

Figura 2.2: Espaço de cor YCbCr, obtido por uma tranformação linear do RGB, onde o Y é o componente de luminosidade. Fonte: http://www.couleur.org

Dentro da área de visão computacional, técnicas de reconhecimento de padrões são comumente empregadas para a análise de mais alto nível da cena (como a identificação de estruturas em uma imagem ou vídeo). Em particular, a técnica de aprendizado de máquinas chamada Adaboost (FREUND; SCHAPIRE, 1997), cujo nome vem da junção entre as palavras em inglês *ADAptive* e *BOOSTing*, é bastante utilizada em problemas de detecção de objetos específicos. O algoritmo Adaboost está relacionado ao problema da construções de classificadores mais robustos combinando regras simples. Os classificadores são construídos um após o outro, e seus objetos de treinamento são amostrados de forma aleatória, inicialmente com distribuição uniforme e posteriormente com distribuição proporcional a sua dificuldade de classificação. No final do processamento, o resultado é um classificador forte (*strong classifier*) composto por classificadores fracos (*weak classifiers*) ponderados. Um exemplo de aplicação do Adaboost relevante para este trabalho é o a algoritmo para detecção de faces proposto por Viola e Jones (2001), que será visto em mais detalhes no capítulo seguinte.

2.2 Processamento de Áudio

O sinal de áudio é normalmente obtido através de um ou mais microfones, e é representado digitalmente por um vetor numérico contendo uma amostragem do sinal contínuo amostrado em espaços de tempo constantes (OPPENHEIM; SCHAFER, 1989). A literatura sobre proces-

samento de áudio é vasta, e uma grande quantidade de algoritmos se baseia na manipulação no domínio de frequência.

A representação de sinais do domínio de frequência (também conhecido como domínio de Fourier) é feita através da Transformada Discreta de Fourier (ou DFT, de *Discrete Fourier Transform*), que mapeia um vetor n-dimensional no domínio tempo em outro vetor n-dimensional no domínio da frequência. Intuitivamente, cada elemento de um sinal transformado pela DFT representa o peso de uma onda senoidal, cuja frequência é dada pelo índice do elemento. Do ponto de vista computacional, há implementações bastante eficientes da DFT, chamadas de FFTs (ou *Fast Fourier Transforms*), que permitem o processamento de sinais em tempo-real.

Para entender o funcionamento da FFT, antes é preciso revisar o que é a DFT. A DFT é definido por

$$A_r = \sum_{k=0}^{N-1} X_k e^{-2\pi irk/N} \quad r = 0, ..., N-1 \tag{2.1}$$

onde A_r é o *r-ésimo* coeficiente da DFT e X_k a *k-ésima* amostra do sinal no domínio de tempo, que consiste de N amostras, e $i = \sqrt{-1}$. Por conveniência, (2.1) pode ser escrita como

$$A_r = \sum_{k=0}^{N-1} (X_k) W^{rk} \quad r = 0, ..., N-1 \tag{2.2}$$

onde

$$W = e^{-2\pi i/N} \tag{2.3}$$

Os X_ks são valores de uma função em pontos discretos no tempo; o indexador r é as vezes chamado de 'frequência' da DFT. A DFT tem também sido chamado de *discrete Fourier transform* ou *discrete time, finite range Fourier transform*.

A FFT é um algoritmo que torna possível o cálculo da DFT mais rapidamente que os demais algoritmos existentes (COCHRAN et al., 1967). O custo computacional de um algoritmo tradicional para cálculo da DFT é de N^2 operações aritméticas, sendo N a quantidade de amostras a serem utilizadas na transformada. Utilizando a FFT, o custo computacional cai para $2N \log_2 N$ operações aritméticas. Ou seja, para um conjunto de 1024 amostras, o esforço computacional da FFT será 50 vezes menor do que um algoritmo tradicional de DFT. Esse baixo custo computacional permitiu o uso prático da Transformada de Fourier em diversas tecnologias, com seu

cálculo em tempo real, principalmente no campo de processamento de sinais.

Suponha um conjunto de N amostras de um sinal (tal como X_k mostrado na Figura 2.3(a)). Dividindo-se em duas funções Y_k e Z_k, cada qual contendo metade dos pontos ($N/2$). A função Y_k é composta pelos pontos das posições pares $(X_0, X_2, X_4, ...)$ e Z_k é composta pelos pontos ímpares $(X_1, X_3, X_5, ...)$. Essas funções são mostradas na Figura 2.3(b) e (c), e podem ser escritas formalmente como

$$Y_k = X_{2k},\ Z_k = X_{2k+1}, \quad k = 0, 1, 2, ..., \frac{N}{2} - 1. \tag{2.4}$$

Sendo Y_k e Z_k sequências de $N/2$ pontos cada, elas tem sua transformada discreta de Fourier definida por

$$B_r = \sum_{k=0}^{\frac{N}{2}-1} Y_k e^{-4\pi irk/N},\ C_r = \sum_{k=0}^{\frac{N}{2}-1} Z_k e^{-4\pi irk/N}, \quad r = 0, 1, 2, ..., \frac{N}{2} - 1. \tag{2.5}$$

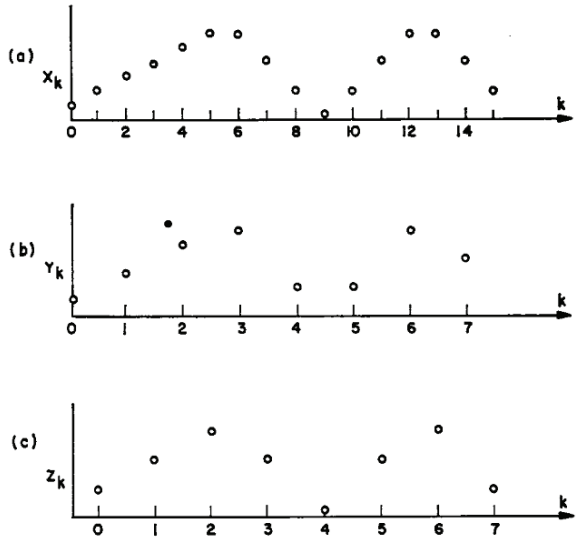

Figura 2.3: Decomposição da série de amostras no domínio de tempo em duas séries distintas, cada qual possuindo metade das amostras. Uma das séries com as amostras de posições pares e outra com as de posições ímpares. Fonte: (COCHRAN et al., 1967).

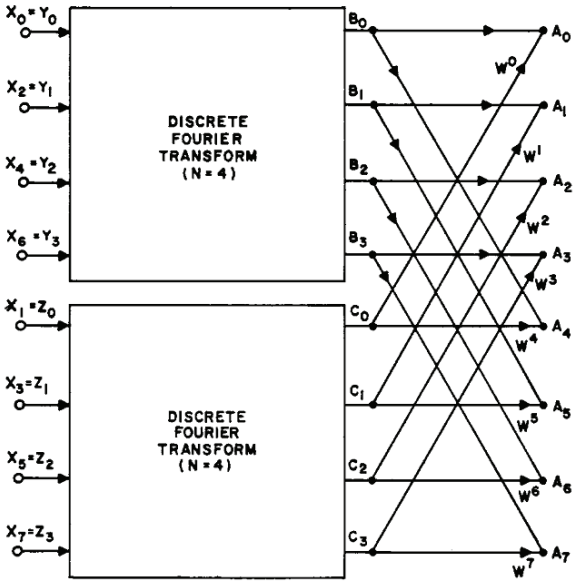

Figura 2.4: Fluxograma ilustrando a redução de uma DFT de N amostras em duas DFTs de $N/2$ amostras cada. Cada ponto, ou nodo, representa uma variável, e as setas que terminam nestes pontos contribuem para seu valor. As contribuições são aditivas, e o peso de cada contribuição, se outra além da unidade, é indicado pela constante escrita perto da cabeça da seta de transmissão. Então, neste exemplo, o valor A_7 é igual a $B_3 + W_7 \times C_3$. Fonte: (COCHRAN et al., 1967).

A transformada discreta de Fourier a ser encontrada é A_r, a qual pode ser escrita em termos de pontos de posições pares e ímpares

$$A_r = \sum_{k=0}^{\frac{N}{2}-1} \left\{ Y_k e^{-4\pi irk/N} + Z_k e^{\frac{-2\pi ir}{N}[2k+1]} \right\}, \quad r = 0,1,2,...,N-1 \tag{2.6}$$

ou

$$A_r = \sum_{k=0}^{\frac{N}{2}-1} Y_k e^{-4\pi irk/N} + e^{-2\pi ir/N} \sum_{k=0}^{\frac{N}{2}-1} Z_k e^{-4\pi irk/N} \tag{2.7}$$

a qual, utilizando (2.5), pode ser escrita como

$$A_r = B_r + e^{-2\pi ir/N} C_r, \quad 0 \leq r < \frac{N}{2} \tag{2.8}$$

Para valores de r maiores que $N/2$, B_r e C_r repetem periodicamente os valores obtidos quando $r < N/2$. Portanto, substituindo-se r por $r + N/2$ em 2.8 obtem-se

$$A_{r+N/2} = B_r + e^{-2\pi i[r+\frac{N}{2}]/N} C_r, \quad 0 \leq r < \frac{N}{2} \tag{2.9}$$

que simplificando fica

$$A_{r+N/2} = B_r - e^{-2\pi ir/N} C_r \quad 0 \leq r < \frac{N}{2} \tag{2.10}$$

Utilizando a Equação (2.3), (2.8) e (2.10) podem ser escritas como

$$A_r = B_r + W^r C_r, \quad 0 \leq r < \frac{N}{2} \tag{2.11}$$

e

$$A_{r+N/2} = B_r - W^r C_r, \quad 0 \leq r < \frac{N}{2} \tag{2.12}$$

Assumindo que existe um método que calcula a transformada discreta de Fourier em um tempo proporcional ao quadrado do número de amostras (N^2), agora pode-se utilizar o algoritmo visto acima para calcular as transformadas de Y_k e Z_k, consumindo um tempo proporcional a $2(N/2)^2$, e utilizando (2.11) e (2.12) para encontrar A_r com mais N operações. Isso está

ilustrado no fluxograma da Figura 2.5. Os pontos da esquerda são os valores de X_k, ou seja, Y_k e Z_k, e os pontos na direita são os pontos da transformada discreta de Fourier A_r.

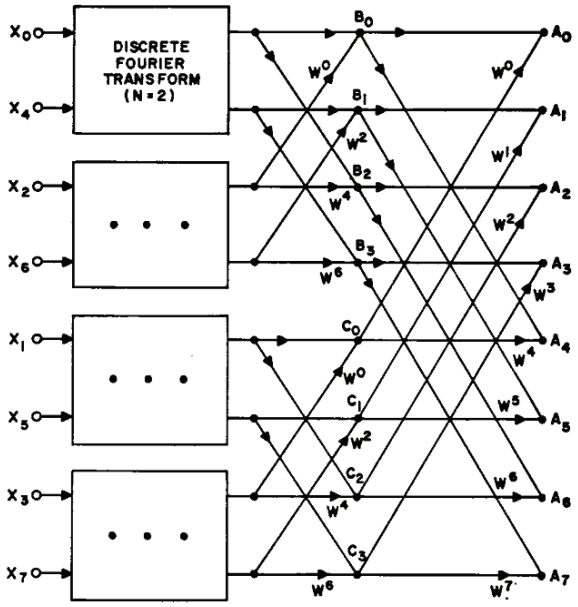

Figura 2.5: Fluxograma ilustrando a redução do cálculo da DFT iniciado na Figura 2.4. Fonte: (COCHRAN et al., 1967).

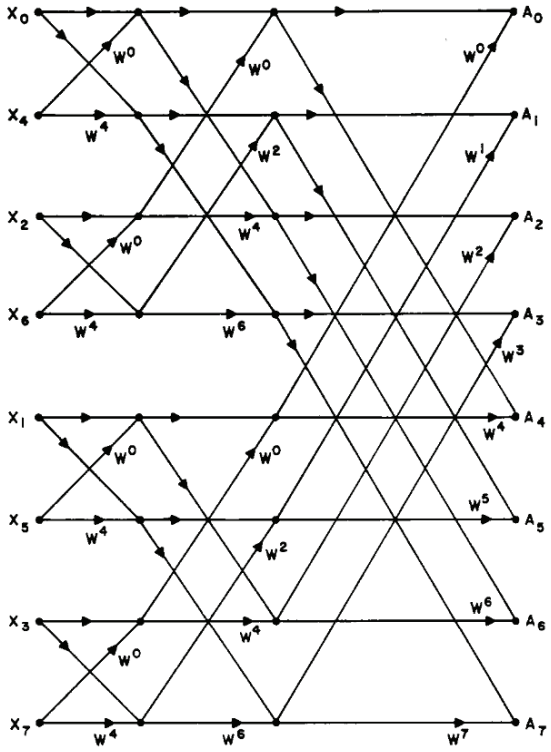

Figura 2.6: Fluxograma ilustrando o cálculo da DFT quando as operações envolvidas são completamente reduzidas em multiplicações e adições. Fonte: (COCHRAN et al., 1967).

Cooley e Tukey (1965) propuseram que, desde que Y_k e Z_k são transformados e que o cálculo da DFT de N amostras pode ser reduzido através do cálculo das DFTs de duas sequências de $N/2$ amostras cada, o cálculo de B_k (ou C_k) pode ser reduzido para o cálculo de sequências de $N/4$ amostras. Essas reduções pode ser feitas tantas vezes quanto o número de amostras for divisível por 2. Portanto, se $N = 2^n$ então podem ser feitas n reduções, aplicando (2.4), (2.11) e (2.12) primeiro para N, depois para $N/2$,..., e finalmente para uma função de dois pontos. A transformada discreta de Fourier de uma função de um ponto é, naturalmente, a própria amostra. As reduções sucessivas de uma DFT de oito pontos é iniciada na Figura 2.4, depois continuada nas Figuras 2.5 e 2.6. Na Figura 2.6 as operações têm sido completamente reduzidas à multiplicações e adições complexas. No fluxograma existem 8×3 pontos e $2 \times 8 \times 3$ setas, correspondendo respectivamente a 24 adições e 48 multiplicações. Metade das multiplicações podem ser omitidas desde que a transmissão indicada pela seta é unitária.

Mais informação sobre a FFT pode ser extraída da Figura 2.6. Por exemplo, se a sequência de amostras de entrada X_k é armazenada na ordem $X_0, X_4, X_2, X_6, X_1, X_5, X_3, X_7$, como mostrado

na Figura 2.6, o cálculo da DFT pode ser feito sem a necessidade de consumir mais memória do computador, apenas utilizando a memória alocada para as amostras de entrada X_k (*in place*). Os resultados intermediários são armazenados nas mesmas posições dos dados originais de entrada. Para essa versão do algoritmo que está sendo apresentada, a ordenação inicial da sequência das amostras, X_k, foi necessária para o cálculo *in place*. Essa ordenação é devido aos movimentos repetidos das variáveis das posições ímpares de uma sequência para o final da sequência durante cada estágio da redução, como mostrado nas Figuras 2.4, 2.5 e 2.6. Essa ordenação foi chamada de 'bit invertido' devido às amostras serem posicionadas em ordem relativa a posição com os bits invertidos, como pode ser visto na Figura 2.7.

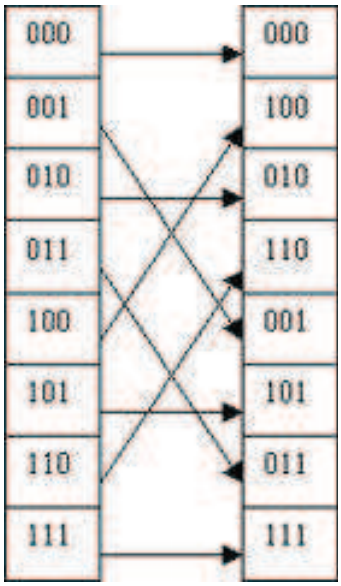

Figura 2.7: Ordenação por bit invertido em uma sequência de oito amostras.

Neste trabalho, algoritmos de localização por áudio foram pesquisados e muitos deles desenvolvidos no domínio de Fourier. Um exemplo é o algoritmo SRP-PHAT, proposto por DiBiase (2000), que será visto com maiores detalhes no capítulo seguinte.

2.3 Análise Multimodal

Os humanos percebem o mundo natural de forma multimodal, através de seus sensores para visão, audição e tato. Eles entendem o mundo multimodal aparentemente sem esforço, embora exista uma enorme quantidade de informações sendo processadas pelo cérebro para essa tarefa. Técnicas computacionais, apesar dos avanços recentes, continuam significativamente inferiores à compreensão multimodal dos seres humanos (MARAGOS; POTAMIANOS; GROS, 2008). A Análise Multimodal é o conjunto de técnicas voltadas para integrar e interagir com as diversas

fontes de informação de naturezas distintas, como no caso desse trabalho, informações de áudio e vídeo.

Uma abordagem bastante utilizada para realizar Análise Multimodal, particularmente em aplicações que envolvem o parâmetro tempo, é baseada em Modelos Escondidos de Markov (*Hidden Markov Models*, ou HMMs) (RABINER, 1989). Um HMM pode ser utilizado para descrever um sistema dinâmico, que no decorrer do tempo pode sofrer mudanças em seu estado (ou possivelmente continuar no mesmo estado) de acordo com um conjunto de probabilidades associadas a cada par de estados - a matriz de probabilidade de transição de estado. No HMM, cada estado é "escondido" pois não é observado explicitamente, mas pode ser descrito por um outro conjunto de processos estocásticos que produzem uma sequência de observações (RABINER, 1989). Os elementos desse segundo conjunto são os símbolos observáveis, a partir dos quais se pode inferir o estado do sistema a cada instante de tempo.

Os processos escondidos consistem de um conjunto de estados conectados por transições com probabilidades (autômato finito), enquanto que os processos observáveis (não escondidos) consistem de um conjunto de saídas ou observações, cada qual pode ser emitido por cada estado de acordo com alguma função de densidade de probabilidade. Devido sua capacidade de modelar o sinal de áudio com bons resultados na prática, o HMM tem se tornado a abordagem dominante para o processamento de áudio e reconhecimento da fala.

O conjunto dos símbolos observáveis pode ser contínuo ou discreto. Em particular, um HMM para as observações de símbolos discretos é caracterizado por:

- N, o número de estados do modelo. Embora os estados sejam escondidos, em muitas aplicações práticas existe algum significado físico inerente ao conjunto de estados do modelo. Sejam S_1, S_2,...,S_N os N estados do modelo.

- M, o número de símbolos distintos observáveis por estado. Os símbolos observáveis correspondem à saída física do sistema que está sendo modelado, e são denotados como $V = \{v_1, v_1, ..., v_M\}$.

- A probabilidade de transição entre os estados $A = \{a_{ij}\}$, onde

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i],\ 1 \leq i, j \leq N, \tag{2.13}$$

  ou seja, a_{ij} representa a probabilidade de o sistema estar no estado S_j no tempo $t+1$ dado que o sistema se encontra no estado S_i no tempo t. Para o caso especial onde qualquer estado pode alcançar qualquer outro estado em uma simples etapa, tem-se $a_{ij} > 0$ para todo i, j.

- A distribuição de probabilidade dos símbolos observáveis no estado j, $B = \{b_j(k)\}$, onde

$$b_j(k) = P[O_t = v_k | q_t = j],\ 1 \leq j \leq N,\ 1 \leq k \leq M. \tag{2.14}$$

- A distribuição de probabilidade do estado inicial $\pi = \{\pi_i\}$, onde:

$$\pi_i = P[q_1 = i],\ 1 \leq i \leq N \tag{2.15}$$

Pode-se observar que uma completa especificação de um HMM requer especificação de dois parâmetros do modelo, N e M, e a especificação dos três conjuntos de medidas de probabilidade A, B e π. Por conveniência é utilizada a notação compacta $\lambda = (A, B, \pi)$ para indicar o conjunto completo de parâmetros do modelo.

Para fixar os conceitos expostos acima sobre HMM, o exemplo "urna e bola" descrito em Rabiner (1989) é brevemente exposto. Tem-se três urnas, cada qual contém várias bolas coloridas, em um universo de M cores distintas de bolas. Nesse exemplo, uma pessoa escolhe uma urna inicial S_1 de acordo com algum processo randômico. Dentro dessa urna S_1, uma bola é escolhida aleatoriamente, a cor da bola escolhida é registrada na observação e a bola colocada novamente na urna. Uma nova urna é então selecionada de acordo com um processo aleatório associado à urna corrente, e o processo de seleção da bola é então repetido.

No exemplo "urna e bola", cada urna é um estado escondido ($N = 3$), visto que não se sabe de qual urna as bolas foram retiradas. Os símbolos observáveis são as bolas coloridas (M), e a distribuição das bolas coloridas em cada urna corresponde às distribuições $b_j(k)$. O processo de mudança de urna a cada iteração corresponde às probabilidades de transição a_{ij}, e o processo de escolha da primeira urna está relacionado com as probabilidade do estado inicial π. A Figura 2.8 apresenta uma representação gráfica desse HMM através de um grafo, onde cada vértice representa um estado, e as arestas (ponderadas) as probabilidades de transição.

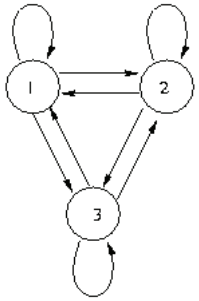

Figura 2.8: HMM de três estados representando as três urnas.

A transição de uma urna para outra (evento não observável ou escondido) depende de uma

distribuição de probabilidade associada a urna corrente. As possíveis transições estão representadas pelas setas e suas probabilidades são comumente representadas em forma de matriz. Cada estado (urna) possui também sua distribuição de probabilidade dos eventos observáveis, aqui representado pela escolhas das bolas coloridas.

Apesar de toda teoria exposta aqui sobre os Modelos Escondidos de Markov, a mesma não teria aplicação prática se os três problemas básicos não tivessem solução (RABINER, 1989). Os problemas são os seguintes:

- *Problema 1*: Dada a sequência de observações $O = (O_1, O_2, ..., O_T)$ e o modelo $\lambda = (A, B, \pi)$, como pode ser eficientemente calculado $P(O|\lambda)$, a probabilidade da sequência de observações dado o modelo? É conhecido como problema da avaliação e é útil para testar a probabilidade da sequência observada ter sido produzida por determinado HMM, possibilitando selecionar o modelo mais coerente com a observação. A solução mais popular para esse cálculo é o procedimento *forward-backward*, que pode ser encontrado em Baum e Eagon (1967) e Baum e Sell (1968).

- *Problema 2*: Dada a sequência de observações $O = (O_1, O_2, ..., O_T)$ e o modelo λ, como escolher a sequência de estados correspondente $Q = (q_1, q_2, ..., q_T)$ que melhor explica as observações? Este é problema da busca da melhor sequência de estados, procurando descobrir a parte escondida do modelo. O problema é resolvido geralmente utilizando um procedimento próximo ao ótimo, o algoritmo de Viterbi, que pode ser visto em detalhes em Viterbi (1967), Forney G.D. (1973) e sua implementação em Lou (1995).

- *Problema 3*: Como ajustar os parâmetros do modelo $\lambda = (A, B, \pi)$ para maximizar $P(O|\lambda)$? É o problema do treinamento e é de longe o mais difícil de ser calculado. É necessária uma sequência de observações para ajustar os parâmetros do modelo chamada de sequência de treinamento. Não existe maneira conhecida de resolver analiticamente o conjunto de parâmetros do modelo que maximiza a probabilidade da sequência de observações de uma maneira fechada. Entretanto pode-se escolher $\lambda = (A, B, \pi)$ tal que sua probabilidade $P(O|\lambda)$ é localmente maximizada, utilizando um procedimento iterativo como o método de Baum-Welch, que pode ser encontrado em Baum et al. (1970).

A discussão sobre HMM, focada para o problema da análise multimodal utilizando os sinais de áudio e vídeo, é retomada no capítulo seguinte.

3 *Revisão Bibliográfica*

Como já foi dito anteriormente, a localização de um locutor não é uma pesquisa inovadora, mas ainda é uma área que precisa ser aperfeiçoada para seus resultados serem mais robustos. Alguns trabalhos existentes se assemelham com o objetivo deste, como o de Wang e Brandstein (1997) e o de Lo et al. (2004). Entretanto, a diferença está nas diferentes técnicas empregadas para resolver as principais tarefas do trabalho, que podem ser discriminadas em localização e rastreamento de pessoas, localização por áudio e fusão dessas duas fontes de informação. Portanto, o texto que se segue manterá essa organização com a finalidade de focar melhor cada objetivo.

Esse capítulo apresenta uma breve revisão bibliográfica sobre técnicas existentes na literatura que abordam os problema da localização da fonte sonora usando informação de áudio, vídeo e análise multimodal. Na parte da análise por vídeo, serão revisados basicamente algoritmos de detecção de faces em imagens, que correspondem à fonte sonora de interesse em aplicações de videoconferência. Na parte de análise de áudio, serão abordadas técnicas que realizam a detecção de atividade de voz, e técnicas de localização com base em arranjos de microfones. Finalmente, são avaliadas algumas abordagens para combinação de dados de áudio e vídeo através de análise multimodal.

3.1 Detecção de faces em vídeo

Um tema muito pesquisado atualmente na área de Visão Computacional é a detecção e reconhecimento de faces em imagens e vídeos. A detecção de face é uma pré-condição para o algoritmo de reconhecimento da mesma.

Dentro das pesquisas de detecção de face, uma abordagem bastante utilizada é de reconhecimento de determinados padrões geométricos que determinam uma face, como o trabalho de Viola e Jones (2001). Esse trabalho sugere um classificador em cascata, onde cada nível da cascata é formado por padrões geométricos selecionados por um treinamento feito pelo algoritmo

de aprendizado AdaBoost. Exemplos de formas geométricas que, individualmente, representam os classificadores fracos podem ser vistos na Figura 3.1. A Figura 3.2 mostra as formas geométricas pertencentes ao classificador forte do primeiro nível da cascata, encontrado nos experimentos feitos por Viola e Jones (2001), que podem caracterizar uma face.

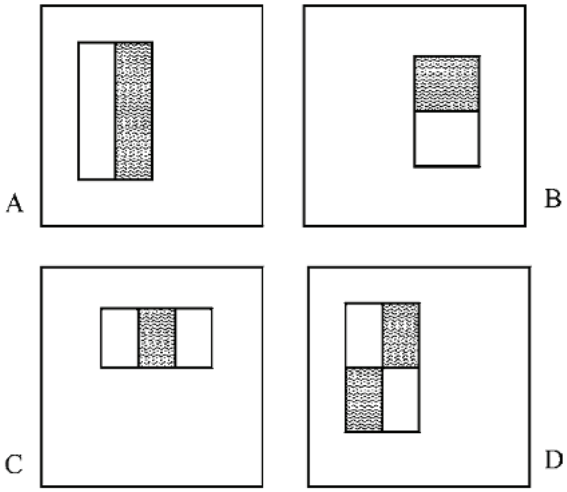

Figura 3.1: Exemplo de formas geométricas que, individualmente, representam os classificadores fracos. Fonte Viola e Jones (2001).

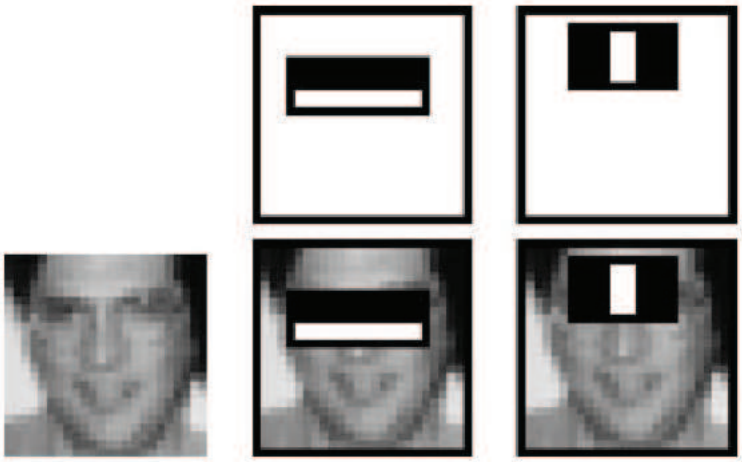

Figura 3.2: Primeira e segunda forma geométrica selecionadas para detectar face pelo algoritmo proposto por Viola e Jones (2001).

O algoritmo varre com uma janela deslizante toda a áera da imagem a procura regiões candidatas a conter uma face. Em cada posição da janela deslizante, as formas geométricas selecionadas pelo AdaBoost são utilizadas para determinar a existência ou não de face. O cálculo das formas geométricas baseia-se em uma diferença dos valores médios dos pixels pertencentes a parte clara pelos pertencentes a parte escura. Se a diferença for maior que a determinada então a forma geométrica foi 'encontrada'.

Em uma sequência de vídeo, o processamento para calcular todas as formas geométricas necessárias para determinar uma face, em todas as posições possíveis da imagem consome muito tempo, inviabilizando utilização do algoritmo. Para contornar esse problema, Viola e Jones (2001) desenvolveram o conceito da imagem integral, que é uma representação da imagem original onde cada pixel contém o valor que é a soma do valor de todos os pixels acima e a esquerda, como mostrado na Figura 3.3. Com base na imagem integral, a janela deslizante pode calcular o valor das formas geométricas do classificador sem precisar somar todos os pixel contidos em determinado retângulo da forma geométrica, mas sim utilizando apenas quatro pontos de referência, como mostra a Figura 3.4. O conceito de imagem integral também é utilizado nesse trabalho para o cálculo das áreas da imagem cobertas com pixels com cor de pele, apresentado em detalhes no capítulo de Metodologia.

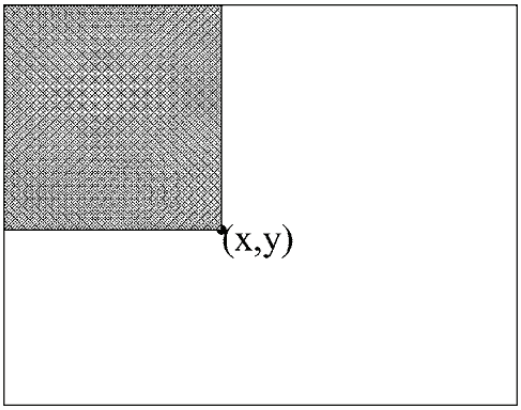

Figura 3.3: O valor da imagem integral no ponto (x, y) é a soma do valor de todos os pixels acima e a esquerda. Fonte Viola e Jones (2001).

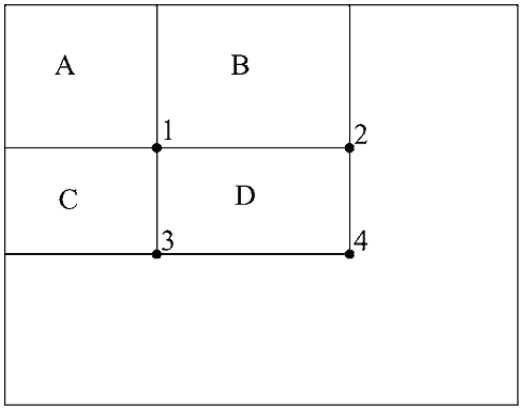

Figura 3.4: A soma dos pixels dentro do retângulo *D* pode ser calculada com quatro pontos de referência: 4+1-(2+3). Fonte Viola e Jones (2001).

Outra abordagem é a utilização da informação radiométrica para detectar faces. Essa técnica parte do pré-suposto que uma face é constituída de pele humana, cuja a resposta espectral (canais R, G e B) é conhecida e possível de ser classificada em uma imagem digital.

Como trabalho utilizando essa abordagem pode ser citada a pesquisa desenvolvida por Hsu, Mottaleb e Jain (2002), onde trabalharam com pixels de *pele* e *não-pele* no espaço de cor YCbCr, para encontrar a região do espaço que caracteriza a cor de pele independente da intensidade da iluminação (componente Y). A Figura 3.5 mostra a localização dos pixel *pele* no espaço de cor YCbCr.

Hsu, Mottaleb e Jain (2002) concluíram que, na prática, a cor de pele é não-linearmente dependente da luminância e não independente como outros estudos mencionaram (Saber e Tekalp (1998) e Sobottka e Pitas (1998)). A solução dele foi ajustar os limites da região onde os pixels

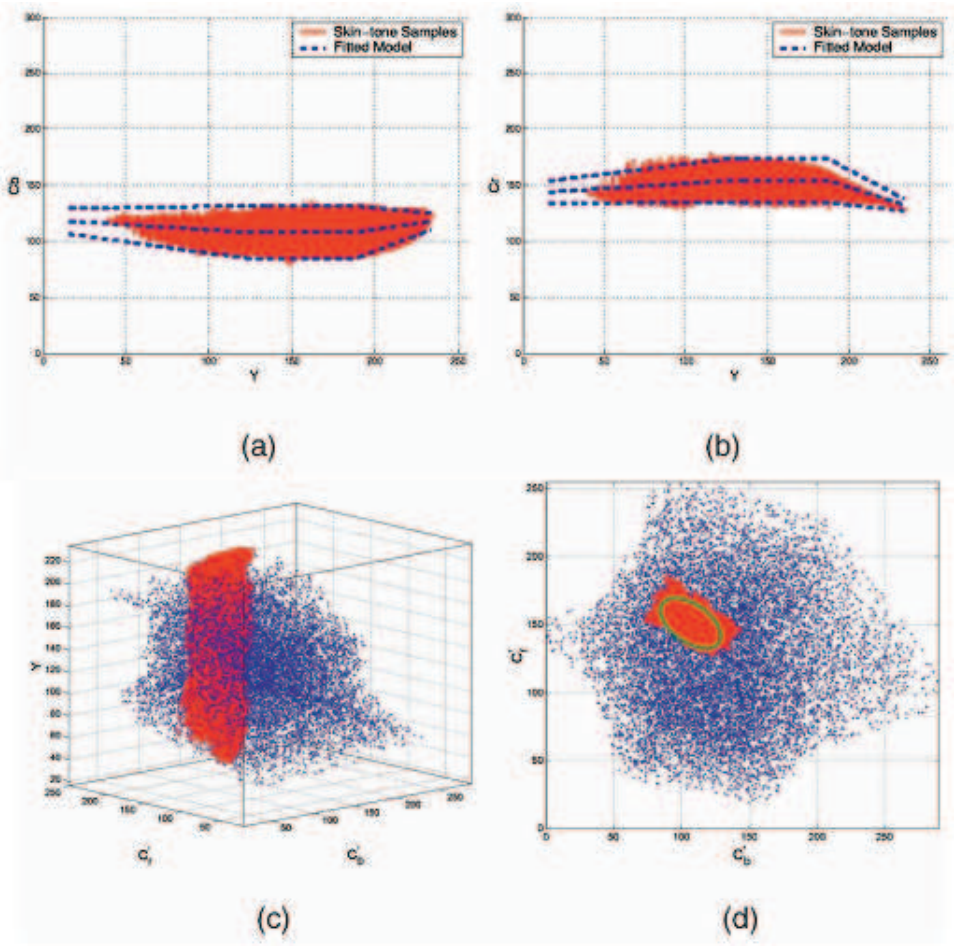

Figura 3.5: Região de cor de pele (em vermelho) no espaço de cor YCbCr. (a) No sub-espaço YCb. (b) No sub-espaço YCr. (c) Espaço YCbCr. (d) No sub-espaço CbCr (HSU; MOTTALEB; JAIN, 2002).

de *pele* aparecem no espaço YCbCr, formando um sólido, como pode ser visto nas Figuras 3.5(a) e 3.5(b).

O objetivo principal dessa abordagem é delimitar, em algum espaço de cor, a região de valores possíveis para classificar um pixel como *pele*, para ser utilizado na detecção de pessoas em imagens. Além de muitos objetos, fora o ser humano, poderem possuir a mesma resposta espectral e ser confundido com pele, a diferença de luminosidade complica, e muito, a delimitação em um espaço de cor da classe *pele*. Dependendo da intensidade da fonte luminosa, a imagem pode adquirir tonalidades diferentes, como pode ser visto na Figura 3.6.

Devido a esse problema de luminosidade, a detecção de pele em uma imagem (ou vídeo) é frequentemente realizada em três passos:

1. Transformar a cor do pixel de RGB para outro espaço de cor que contenha a componente de luminosidade;

Figura 3.6: Diferença na imagem devido a variações na intensidade da luz. Fonte Martinez e Benavente (1998).

2. Retirar a componente de luminosidade do espaço de cor e usar apenas os dois componentes de cor restantes no processo de classificação;

3. Classificar de acordo com algum modelo de distribuição da cor de pele.

A mesma pessoa pode ser fotografada em ambientes diferentes, onde fontes luminosas de naturezas distintas estão presentes. O resultado é que a resposta espectral dessa mesma pessoa acaba variando dependendo da origem da luz. A Figura 3.7 mostra um exemplo de imagens obtidas com diferentes fontes de luz.

Figura 3.7: Diferença na imagem devido a variações do tipo de luz. Da esquerda para direita, a fonte luminosa foi incandescente, luz do dia, por do sol e fluorescente, respectivamente. Fonte Marszalec et al. (2000).

Com a popularização da técnica de retirar a componente de luminosidade do modelo, Shin, Chang e Tsap (2002) fizeram uma pesquisa para comparar a classificação da pele no RGB com oito transformações populares do espaço de cor. A pesquisa procurou respostas para as seguintes perguntas:

1. A transformação de RGB para outro espaço de cor ajuda na classificação?

2. Retirar do modelo a compoente de luminosidade ajuda na classificação?

3. Qual o melhor espaço de cor para classificação?

Os espaços de cor contemplados na pesquisa foram, além do RGB: NRGB (RGB Normalizado), CIEXYZ, CIELAB, HSI, SCT, YCbCr, YIQ e YUV. O RGB serviu de base para a transformação e como parâmetro de comparação dos resultados. A metodologia empregada foi calcular a separabilidade das classes *pele* e *não-pele* em imagens nos nove espaços de cor. Para cada espaço de cor foi feita sua versão integral com a componente de luminosidade (3D) e sem a componente de luminosidade (2D), para verificar se há vantagem em retirar essa componente do modelo. Para calcular a separabilidade das classes foram criados dois indicadores baseados na análise do histograma e dois baseados na análise da matriz de dispersão das classes *pele* e *não-pele*. Os valores dos pixels com cor de pele utilizados nas análises foram retirados do banco de imagens AR (MARTINEZ; BENAVENTE, 1998) e do banco UOPB (MARSZALEC et al., 2000); os pixels de *não-pele* para a análise foram retirados do banco de imagens da Universidade de Washington (BERMAN; SHAPIRO, 1999).

Os oito espaços de cor (fora o RGB) foram analisados em suas versões 2D e 3D pelos quatro indicadores, ou seja, cada espaço de cor carregou consigo oito observações de sua perfomance, metade referente a análise 2D e a outra metade na 3D. Como resposta a questão um, notaram que em quatro dessas oito observações, nenhum espaço de cor se mostrou superior ao RGB, e os que foram superior, a diferença foi mínima. Na questão dois os resultados mostraram que em três dos quatro indicadores, a retirada da componentes de luminosidade diminui significativamente (com 95% de confiança) a separabilidade entre as classes, concluindo-se que é mais interessante manter essa componente.

Cada espaço de cor foi analisado sua versão 3D e 2D, totalizando portanto 18 espaços de cor. O resultado da análise mostrou que o espaço de cor RGB 3D foi o melhor de todos em três dos quatro indicadores calculados, respondendo, assim, a pergunta três. Como final da pesquisa concluíram que (1) a transformação do espaço de cor e (2) a retirada da componente de luminosidade do modelo não incrementa a performance da tarefa de detecção de pele.

Dois anos após a pesquisa de Shin, Chang e Tsap (2002), Jayaram et al. (2004) refizeram o mesmo estudo utilizando desta vez, um indicador baseado em uma distribuição normal e um na análise do histograma. Os bancos de imagens utilizados para os testes foram os mesmos da pesquisa anterior. As perguntas respondidas nessa pesquisa foram:

1. A transformação de RGB para outro espaço de cor ajuda na classificação?
2. Retirar do modelo a compoente de luminosidade ajuda na classificação?
3. O modelo escolhido para a cor de pele faz diferença?
4. Qual o melhor espaço de cor para classificação?

Para comparar a performance dos indicadores nos espaços de cor, foi utilizada a área abaixo da curva ROC (*Receiver Operating Characteristic*). Como o resultado de sistemas de classificação em classes geralmente são contínuos, ou seja, produzem um valor situado dentro de um determinado intervalo contínuo, como [0;1], é necessário definir um ponto de corte, ou um limiar de decisão, para se classificar e contabilizar o número de predições positivas e negativas (nesse caso, decidir entre *pele* ou *não-pele*). Como este limiar pode ser selecionado arbitrariamente, a melhor prática para se comparar o desempenho de diversos sistemas é estudar o efeito de seleção de diversos limiares sobre a saída dos dados. Para cada ponto de corte são calculados valores de sensibilidade (o quanto o classificador acerta para *pele*) e especificidade (o quanto o classificador acerta para *não-pele*), que podem então serem dispostos em um gráfico denominado curva ROC, que apresenta no eixo das ordenadas os valores de sensibilidade e nas abscissas o complemento da especificidade, ou seja, o valor (1-especificidade).

Tabela 3.1: A performance dos nove espaços de cor calculada através da área abaixo da curva ROC, em suas versões 3D e 2D, com critérios de separabilidade das classes baseados em uma distribuição normal e na análise do histograma. Fonte Jayaram et al. (2004).

	Normal		Histograma	
Espaço de cor	3D	2D	3D	2D
CIELAB	0,907	0,908	0,977	0,960
CIEXYZ	0,908	0,914	0,959	0,935
HSI	0,929	0,934	0,980	0,973
NRGB	0,915	0,920	0,955	0,962
RGB	0,908	0,921	0,960	0,949
SCT	0,944	0,943	0,982	0,968
YCbCr	0,908	0,861	0,964	0,886
YIQ	0,908	0,861	0,966	0,887
YUV	0,908	0,861	0,964	0,886

A Tabela 3.1 mostra a performance dos nove espaços de cor calculada através da área abaixo da curva ROC, em suas versões 3D e 2D, com critérios de separabilidade das classes baseados em uma distribuição normal e na análise do histograma. Com base na Tabela 3.1, em relação a pergunta um, notaram que a transformação do espaço de cor ajudou em duas das oito vezes quando utilizando 2D e modelando com a distribuição normal para classificação, seis das oito vezes quando utilizando 3D e modelando com o histograma para classificação, três das oito vezes quando utilizando 3D e modelando com a distribuição normal, quatro das oito vezes quando utilizando 2D e modelando com o histrograma. Concluíram que a performance devido à transformação do espaço de cor estava presente mas não era consistente. Para resposta da pergunta dois, notaram que para ambos modelos (histograma e distribuição normal), a performance com a componente de cor (3D) foi melhor do que sem (2D). Na análise pelo histograma,

a performance utilizando a componente de cor foi significativamente melhor que no 2D com intervalo de confiança de 95%. A pergunta três se refere a modelagem da cor de pele e compararam o desempenho o modelo do histograma com o da distribuição normal. Verificaram que o modelo baseado na análise do histograma foi melhor em um intervalo de confiança de 95%, concluindo que a escolha do modelo faz diferença no resultado da classificação. Quanto ao melhor espaço de cor para ser utilizado na tarefa de classificação, pergunta quatro, observaram uma melhor performance usando SCT, HSI ou CIELAB, em suas versões 3D.

A última versão dessa pesquisa de comparação de espaços de cor foi feita por Schmugge et al. (2007). Mantiveram as mesmas questões de 2004 e também a comparação através da curva ROC das mesmas oito transformações do espaço de cor em suas versões 3D e 2D. A métrica continuou com o mesmo indicador baseado na distribuição normal e com o mesmo da análise do histograma. O aspecto mais importante nessa pesquisa de 2007 é o amadurecimento do aspecto estatístico para explicação dos resultados, o que a torna uma fonte mais consistente para futuros projetos que a utilizem como base. Os bancos de imagens utilizados para testes foram os mesmos, apenas com o acréscimo do dbSkin[1].

Os resultados encontrados podem ser vistos na Tabela 3.1. Para resposta da pergunta um, comparando com o desempenho do RGB verificaram que quando modelado com a distribuição normal utilizando 3D melhorou o desempenho em três das oito vezes e piorou em uma das oito vezes; quando modelado com o histograma utilizando 3D melhorou em seis das oito vezes e piorou em duas das oito vezes; quando modelado com a distribuição normal utilizando 2D melhorou em três das oito vezes e piorou em cinco das oito vezes; quando modelado com o histograma utilizando 2D melhorou em duas das oito vezes e piorou em 6 das oito vezes. Quanto a resposta da pergunta dois, que trata de retirar a componente de luminosidade do modelo, notaram que a performance da separabilidade das classes em 3D é maior que utilizando 2D, mas também notaram que a diferença da performance de 3D para 2D foi muito menor com os espaços de cor HSI e SCT do que com o YCbCr, YiQ e YUV, suspeitando que a componente de luminosidade acrescenta muito mais informação para a separabilidade nos espaços de cor YCbCr, YIQ e YUV. Na questão três verificaram que o modelo baseado no histograma foi significativamente melhor que o baseado na distribuição normal, com um intervalo de confiança de 95%. Quanto ao melhor espaço de cor para ser utilizado na tarefa de classificação, pergunta quatro, observaram que a melhor performance foi obtida usando SCT ou HSI, em suas versões 3D.

Existem também abordagens na detecção de faces que utilizam tanto a informação ge-

[1]Banco de imagens desenvolvido pela Universidad de Chile para treinamento de identificação de pele, disponível em http://agami.die.uchile.cl/skindiff/

Tabela 3.2: A performance dos nove espaços de cor, em suas versões 3D e 2D, com critérios de separabilidade das classes baseados em uma distribuição normal e na análise do histograma, foi calculada através da área abaixo da curva ROC. Fonte Schmugge et al. (2007).

	Normal		Histograma	
Espaço de cor	3D	2D	3D	2D
CIELAB	0,889	0,899	0,908	0,894
CIEXYZ	0,862	0,848	0,895	0,876
HSI	0,844	0,854	0,947	0,939
NRGB	0,875	0,878	0,894	0,897
RGB	0,862	0,876	0,898	0,905
SCT	0,913	0,906	0,932	0,912
YCbCr	0,862	0,851	0,907	0,856
YIQ	0,862	0,851	0,902	0,855
YUV	0,862	0,851	0,907	0,856

ométrica da imagem quanto a radiométrica. Wu e Ai (2008) também trabalharam com a imagem no espaço de cor YCbCr, encontrando a região no sub-espaço CbCr (componentes de cor) que caracteriza a cor da pele (Figura 3.8).

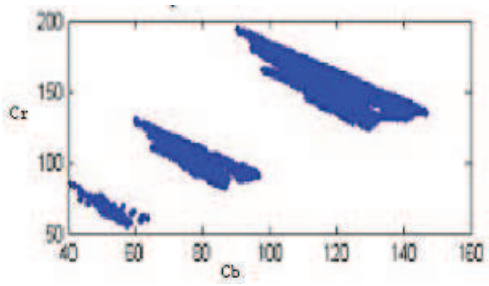

Figura 3.8: Regiões de cor de pele no sub-espaço de cor CbCr.

Com a imagem no espaço de cor YCbCr, Wu e Ai (2008) criaram uma imagem de probabilidade de um pixel pertencer à classe *pele* (Figura 3.9(c)). Depois, aplicando um limiar para decisão, fabricaram uma imagem binária determinando a região constituída por pele (Figura 3.9(d)). Em seguida, com algumas operações morfológicas obtiveram a região candidata a conter uma face. Para finalizar, utilizaram o algoritmo de classificação AdaBoost para o reconhecimento dos padrões geométricos que caracterizam uma face, criado por Viola e Jones (2001), e o aplicaram na área candidata encontrada (Figura 3.10).

Wimmer e Radig (2006) propuseram trabalhar com classificadores de cor de pele dinâmicos. Eles viram que nos trabalhos anteriores sempre existia uma região fixa do espaço de cor, RGB ou não, que representava os pixels de cor de pele. A idéia deles foi transformar essa região de forma que melhorasse o desempenho do classificador. Assim criaram a região cuboidal de cor de pele, onde os limites do espaço são definidos levando em consideração a média e a var-

Figura 3.9: Segmentação da imagem em cor de pele. (a) Imagem original. (b) Imagem no espaço de cor YCbCr. (c) Imagem de similaridade. (d) Imagem binária.

Figura 3.10: Imagem binária após operações morfológicas de fechamento.

iância da imagem; criaram a região elipsoidal, que é definida pela distância de Mahalanobis entre a média e a cor a ser classificada; criaram também a região por regras, que é definida por algoritmos de aprendizado tipo ID3, C4.5 ou J4.8. A justificativa deles é que dependendo das condições específicas das imagens, determinado modelo de região de cor de pele seria eficiente.

3.2 Detecção de atividade de voz através de áudio

Muitas aplicações que trabalham com transmissão de sinal de áudio necessitam otimizar o fluxo de informações. Uma técnica bastante utilizada é transmitir o som da voz apenas quando o locutor está falando. Essa técnica é chamada de *Voice Activity Detection* (Detecção da Atividade de Voz), ou simplemsnte VAD.

A maioria dos algoritmos convencionais de VAD assumem que as estatísticas do ruído de fundo são estacionárias em um longo período de tempo, normalmente empregando o uso de heurísticas para determinar o que é voz e o que é ruído. Essas técnicas foram e são bastante utilizadas para a telefonia celular, onde o consumo da banda de transmissão do sinal é um ponto crítico para o crescimento do serviço. Devido também a essa grande demanda, muitas técnicas foram desenvolvidas e aperfeiçoadas.

O serviço telefonia celular Pan-Européia foi introduzida em 1991, mas os algoritmos de codificação da transmissão da voz do locutor já tinham sido selecionado em 1989 pela CEPT Groupe Spécial Móbile (GSM), o qual especificou a implementação da Transmissão Descontínua (DTX). Quando o DTX está presente, a transmissão da voz pelo telefone celular é interrompida quando não há sinal de voz ativa (FREEMAN et al., 1989). O componente mais crítico

do DTX é, portanto, o Detector de Atividade de Voz. De acordo com Freeman et al. (1989) a distinção entre fala e ruído, em situações onde a razão entre sinal e ruído (SNR) é baixa, só podia ser feita levando em consideração as características espectrais do sinal de entrada. Partia-se da suposição que o ruído de fundo é estacionário por um período relativamente longo, onde as características espectrais do ruído são similares ao longo do tempo. Um sinal de voz seria então um desvio nesse comportamento espectral similar e contínuo.

Na revisão da literatura existente nesse campo da ciência, um algoritmo bastante citado é o G.729 (SALAMI et al., 1997), devido a sua atual importância no mercado e é frequentemente utilizado na comparação de desempenho de novos algoritmos VAD. O G.729 na verdade é um padrão para compressão de dados de áudio (voz). É utilizado para transmissão da voz por telefones celulares e Voice over IP (VoIP). Foi desenhado para transmitir dados a 8Kb/s, codificando o sinal a cada 10ms, o que corresponde a 80 amostras, ou seja, 8000 amostras por segundo (SALAMI et al., 1997). A versão G.729b desse algoritmo utiliza um algoritmo VAD, pois constataram que parte considerável de uma conversa normal é constituída por silêncio, com uma média de 60% do tempo em uma conversação entre pessoas. Durante esse silêncio, o microfone fica capturando ruído do ambiente. A maioria das fontes de ruído carregam menos informação que a fala, podendo assim ter uma maior taxa de compressão (BENYASSINE et al., 1997). O algoritmo VAD do G.729b leva em consideração os parâmetros abaixo:

- Distorção Espectral ΔS: soma dos quadrados da diferença entre o LSF (Linhas Espectrais de Frequências) do conjunto de amostras corrente com o LSF médio do rúido de fundo.
- Diferença de todas bandas de energia: calculado pela diferença entre energia do frame corrente e a média da energia do ruído de fundo.
- Diferença das bandas de baixa energia: calculado pela diferença entre as bandas de baixa energia (0-1kHz) do frame corrente com a média das bandas de baixa energia do ruído de fundo.

Esses parâmetros de diferenças se encontram em um espaço Euclidiano multi-dimensional, onde são delimitadas regiões as quais são classificadas sons de voz e não voz. Após cada iteração do algoritmo de VAD, as estatísticas dos parâmetros do ruído de fundo são atualizados.

Sohn e Sung (1998) aplicaram uma regra de decisão estimando parâmetros desconhecidos utilizando o critério da máxima verossimilhança. Eles empregam um modelo estatístico no qual a fala e o ruído são variáveis randômicas independentes com distribuição Gaussiana, então os coeficientes da Transformada Discreta de Fourier (DFT) de cada processo (conjunto de amostras

processado) são variáveis randômicas assintoticamente independentes com distribuição Gaussiana. O vetor de coeficientes de fala, ruído e fala com ruído são representados como S, N e X, respectivamente.

Os parâmetros estatísticos do ruído são assumidos serem conhecidos *a priori*. As duas hipóteses de detecção de atividade de voz são:

$$\begin{cases} H_0 : fala - ausente : X = N \\ H_1 : fala - presente : X = N + S \end{cases} \tag{3.1}$$

As funções conjuntas de probabilidades condicionadas em H_0, em H_1 e θ são dadas por:

$$\begin{cases} p(X|H_0) = \prod_{k=0}^{L-1} \frac{1}{\pi\lambda_N(k)} exp\{-\frac{|X_k|^2}{\lambda_N(k)}\} \\ p(X|\theta, H_1) = \prod_{k=0}^{L-1} \frac{1}{\pi[\lambda_N(k)+\lambda_S(k)]} exp\{-\frac{|X_k|^2}{\lambda_N(k)+\lambda_S(k)}\} \end{cases} \tag{3.2}$$

onde λ é a variância do sinal e L a quantidade de amostras utilizadas no cálculo do DFT.

Mukherjee e Gwee (2007) propuseram um modelo que separa o sinal de áudio (*frame*) em três bandas de frequência e a energia de cada banda $E_i(k)$ é comparada com o limiar da energia do ruído para detectar um pico. O limiar da energia do ruído é da forma $\bar{E}_i + k \times \sigma_i$, onde $\bar{E}_i$ e σ_i são a energia média e o desvio padrão das primeiras 100 amostras na i-ésima banda. Os valores de K e as condições para a existência de um pico $P_i(k)$ na i-ésima banda estão listados na Tabela 3.3.

	Frequência (Hz)	Critério para existência de pico
Banda 1	0 - 750	$E_1(k) \geq \bar{E}_1 + 2.0 \times \sigma_1$
Banda 2	750 - 1875	$E_2(k) \geq \bar{E}_2 + 1.5 \times \sigma_2$
Banda 3	1875 - 4000	$E_3(k) \geq \bar{E}_3 + 1.2 \times \sigma_3$

Tabela 3.3: Limiares da energia do ruído por bandas de frequência

Um pico na banda das frequências mais baixas $P_1(k)$ é importante pois a maior parte da informação da fala está concentrada nessa banda. Os outros dois picos $P_2(k)$ e $P_3(k)$ são úteis para detecção da fala em região de frequências maiores mas com baixa energia a qual teria sido normalmente ofuscada pelo sinal de baixa frequência.

3.3 Arranjos de microfones e localização da fonte sonora

Um arranjo de microfones (Figura 3.11) baseia-se na disposição organizada e conhecida um conjunto de microfones para que a captura do som de todos eles auxilie a calcular a localização ou criar um filtro espacial na fonte sonora de interesse. Nessa seção são discriminadas as principais categorias de técnicas para localização da fonte sonora com utilização de um arranjo de microfones. Essas técnicas estão fundamentadas em princípios já utilizados por radares e sonares para localização de objetos e obstáculos. Para a maioria delas é necessário assumir dois pressupostos (BRANDSTEIN; WARD, 2001):

- As bandas de comprimento de onda incidentes no arranjo de microfones devem ser estreitas.
- A fonte sonora deve estar localizada longe o suficiente do arranjo de microfones para serem modeladas como sinais paralelos para cada microfone (Figura 3.11).

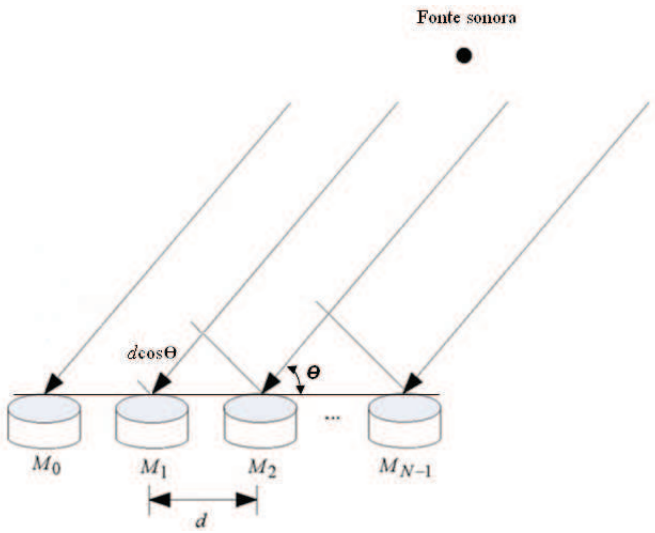

Figura 3.11: Arranjo de microfones. Adaptado de Jung et al. (2007).

No caso da localização de um locutor em um ambiente de videoconferência a situação torna-se complicada, dado que no interior de uma sala existe sempre um ruído de fundo. O sinal da fala é considerado de banda larga, e não estreita como o primeiro pressuposto acima. Existe também o problema de reverberações excessivas dentro da sala, o que torna o problema sempre mais complexo do que numa situação de espaço livre. O segundo pressuposto assume que o sinal vem sob a forma de onda plana, e muitas vezes no processamento com o arranjo de microfones isso não acontece, podendo a onda ser esférica devido a pequena distância entre o emissor e o receptor.

Quando a distância da fonte sonora até o arranjo de microfones é muito maior que a largura do próprio arranjo, as ondas sonoras são consideradas planas ao encontrar com o arranjo, sendo sua curvatura ignorada pelo modelo, como é visto na Figura 3.11 (BRANDSTEIN; WARD, 2001). A essa situação se dá o nome de Campo Distante (*Far Field*) e muitos algoritmos de localização são baseados com esse pressuposto. Em contrapartida existe a situação onde a fonte sonora está próxima do arranjo de microfones e a onda sonora não pode mais modelada como plana, sendo essa situação denominada como Campo Próximo (*Near Field*) e está ilustrada na Figura 3.12.

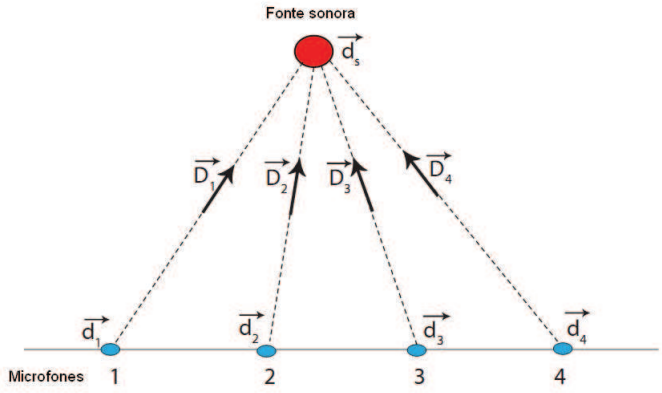

Figura 3.12: Direção de chegada (DOA) em um arranjo de quatro microfones. Fonte: Do (2009).

3.3.1 Estratégias de localização da fonte sonora

De acordo com Brandstein e Ward (2001), podemos agrupar as técnicas de localização da fonte sonora utilizando arranjo de microfones em:

Localização baseada em *Beamforming*

Segundo Veen e Buckley (1988) o termo *beamforming* deriva do fato dos filtros espaciais antecessores foram desenhados para formar feixes semelhantes a lápis (Figura 3.13) com o objetivo de receber um sinal (ondas) de uma localização específica e atenuar sinais de outras localizações. O *beamforming* é aplicável tanto para radiação de energia quanto para recepção.

Esse conjunto de técnicas baseia-se em encontrar a direção do sinal que possua maior energia (SRP - *steered response power*) obtida pela formação de um *beamforming*. A idéia dessas técnicas é utilizar um conjunto de sensores alinhados cuja posição é conhecida e analisar o sinal recebido por eles, como demonstrado pela Figura 3.11. Dependendo da inclinação da fonte

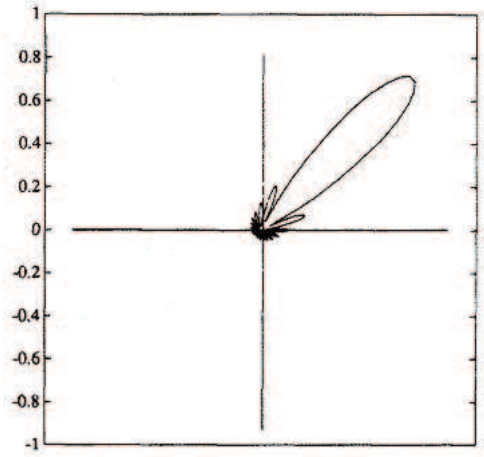

Figura 3.13: Formato de um *Beamformer*. Fonte: Veen e Buckley (1988).

sonora em relação a posição dos sensores haverá atraso no recebimento do sinal em alguns sensores, devido ao som percorrer uma distância maior. Precisa-se varrer todas as direções com o *beamforming* e a direção na qual o sinal for maximizado será a da fonte sonora. Na prática, isso se traduz em encontrar o atraso no recebimento do som por um sensor em relação a outro para corrigir os sinais recebido de forma que quando somados amplifique o sinal da fonte sonora de interesse devido a interferência construtiva da onda (Figura 3.14). Um efeito contrário se dará às fontes ruidosas que se localizam em direções diferentes à fonte de interesse. O resultado dessa operação é criar um campo onde o sinal seja amplificado e ignorar sons vindos de outras localidades, explicando assim o formato da Figura 3.13.

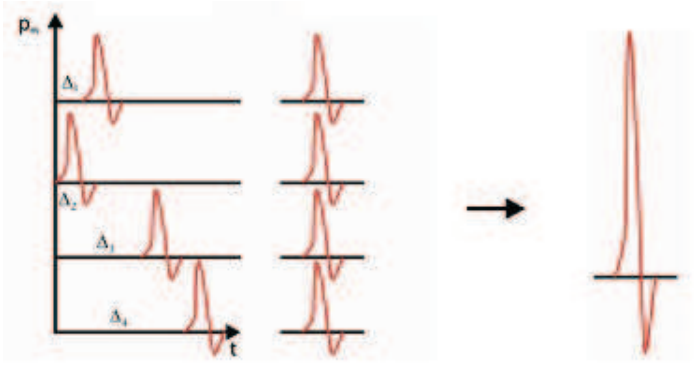

Figura 3.14: Sinais recebidos nos microfones. Retirando o atraso da chegada do sinal é possível amplificar o sinal da fonte sonora desejada, originando o *Beamforming*.

O método mais intuitivo e conhecido desse grupo de técnicas é o Atraso e Soma (*Delay and Sum*), que consiste em incluir atrasos no sinal recebido em um dos microfones com o objetivo que a soma das ondas dos microfones possua um efeito construtivo, como mostra a Figura 3.14. O atraso exato é definido quando o maior efeito construtivo da onda é encontrado. É definido como:

$$y(t, q_s) = \sum_{n=1}^{N} x_n(t + \Delta_n) \tag{3.3}$$

onde Δ_n é o atraso quando o foco do *beamforming* está para determinada direção, e q_s a localização espacial da fonte.

Localização baseada em Estimadores espectrais de alta resolução

As técnicas agrupadas nesse item incluem os métodos modernos de *beamforming* adaptados da análise de alta resolução espectral. Cada uma dessas técnicas é baseada em uma matriz de correlação espaço-espectral derivada dos sinais recebidos dos sensores. Esses algoritmos tendem ser significativamente menos robustos que os métodos convencionais de *beamforming* (BRANDSTEIN; WARD, 2001).

Localização baseada em TDOA (Tempo de atraso da chegada)

TDOA vem do nome *Time Delay Of Arrival* que significa Tempo de Atraso da Chegada. A localização baseada em TDOA utiliza o atraso da chegada do som nos diferentes microfones do arranjo e a informação da posição do sensor possibilita calcular uma curva hiperbólica onde a fonte sonora pode ser encontrada. A intersecção das curvas geradas pelos demais pares de sensores definem o lugar exato da fonte (BRANDSTEIN; WARD, 2001).

Trabalhando com um par de microfones é possível encontrar a Direção de Chegada (*Direction Of Arrival* ou DOA) da fonte sonora. Analisando a Figura 3.11 conclui-se que sabendo a distância d entre o par de microfones M_1 e M_2, a frequência de amostragem Fs de captura do sinal, a velocidade de propagação das ondas acústicas c, e calculando o atraso τ (em amostras) com que o sinal chega a cada microfone, poder-se-á estimar a direção da fonte sonora através da expressão:

$$\theta = acos\left(\frac{c \times \tau}{d \times Fs}\right) \tag{3.4}$$

No caso do Campo Distante, todos microfones do arranjo possuem a mesma DOA. No caso do Campo Próximo, para os M elementos do arranjo existem M DOAs (DO, 2009).

O método mais comum baseado em TDOA é o de Correlação Cruzada (CC), que está definido na equação abaixo para sinais de dois microfones:

$$c_{12}(\tau) = \int_{-\infty}^{\infty} x_1(t)x_2(t+\tau)dt \tag{3.5}$$

onde $x_1(t)$ e $x_2(t)$ são os sinais do par de microfones e τ o atraso no sinal de chegada entre os microfones. O valor de τ irá variar para que maximize $c_{12}(\tau)$, encontrando assim o TDOA.

Para reduzir o tempo de computação necessário a correlação é implementada no domínio de frequência para aproveitar os algoritmos rápidos da FFT (Fast Fourier Transform). Depois de realizada a correlação é calculada a transformada inversa de Fourier para voltar ao domínio do tempo e obter-se o TDOA.

$$R_{12}(\tau) = \int_{-\infty}^{\infty} X_1(\omega)X_2^*(\omega)e^{j\omega\tau}d\omega \tag{3.6}$$

Na expressão anterior, o valor de τ que maximiza $R_{12}(\tau)$ corresponde ao TDOA e está no domínio de tempo contínuo. No entanto, neste tipo de aplicação, o tempo é discreto dado que o sistema é digital. Assim, fazer os cálculos no domínio de frequência é computacionalmente vantajoso, dado que a transformada discreta de Fourier pode ser calculada com recurso ao rápido algoritmo FFT.

Em tempo discreto, a expressão equivalente à (3.6) é:

$$R(n) = \sum_{k=1}^{K} x_1(k)x_2(k+n) \quad n = \{-N,...,N\} \tag{3.7}$$

Significando que a correlação é realizada numa janela de K amostras. O valor N depende da frequência Fs com que os sinais dos microfones são amostrados e representa o número de amostras que são necessárias analisar para procurar o máximo da correlação.

$$N = \frac{Fs \times d}{c} \tag{3.8}$$

Na equação 3.8 o valor d é a distância entre os dois microfones e c é a velocidade do som. Assim, como se pode observar, quanto maior for a frequência de amostragem ou a distância entre microfones, maior será a resolução com que se calcula a direção da fonte sonora, uma vez que o número de amostras que representam o TDOA será maior. No entanto, quanto maior for a distância entre microfones, maior terá que ser a distância entre estes e a fonte sonora, para que a situação seja de onda plana.

3.3.2 Algoritmos robustos para localização da fonte sonora

Considerando $x_n(t)$ o sinal proveniente de uma fonte sonora $s(t)$ recebido no microfone n, o modelo do sinal recebido pode ser expresso como (BRANDSTEIN; WARD, 2001):

$$x_n(t) = s(t) + v_n(t) \tag{3.9}$$

onde $s(t)$ é o sinal recebido no microfone n e v_n é o ruído do ambiente produzido por outras fontes sonoras, como ventiladores ou outros equipamentos, somado com o ruído produzido pelo próprio sistema do microfone. Na presença de superfícies que possam refletir o som, as ondas sonoras produzidas por uma única fonte se propagam em diversos caminhos. Isso gera o efeito de reverberação, onde o som reflete em objetos produzindo ecos, modificando a onda sonora original.

O sinal recebido pelo *n-ésimo* microfone pode ser expresso como:

$$x_n(t) = s(t) \star h_n(q_s, t) + v_n(t) \tag{3.10}$$

onde $h_n(q_s, t)$ é a resposta ao impulso total sendo resultado de dois filtros: a resposta do impulso na sala e a resposta do canal do microfone. O símbolo $\star$ define uma operação de convolução. Caracteriza-se por todos os caminhos acústicos entre a origem e a localização do microfone, incluindo o caminho direto entre eles.

A Figura 3.15 ilustra, em uma janela de 10ms, a resposta que foi medida em uma típica sala de videoconferência. O componente do caminho direto e alguns componentes refletidos fortes foram identificados.

A proposta do Tempo de Atraso Estimado (TDE) é avaliar a disparidade temporal entre o caminho direto no sinal recebido em dois microfones. Para essa finalidade é preciso reescrever a equação da resposta do impulso em termos do componente do caminho direto. A Equação (3.10) é modificada para:

$$x_n(t) = \frac{1}{r_n} s(t - \tau_n) \star g_n(q_s, t) + v_n(t), \tag{3.11}$$

onde r_n é a distância entre a origem e o microfone, τ_n é o tempo de atraso do caminho direto e $g_n(q_s, t)$ é a resposta do impulso modificada que engloba a resposta original menos o componente do caminho direto. O modelo do sinal do microfone é agora expresso explicitamente em

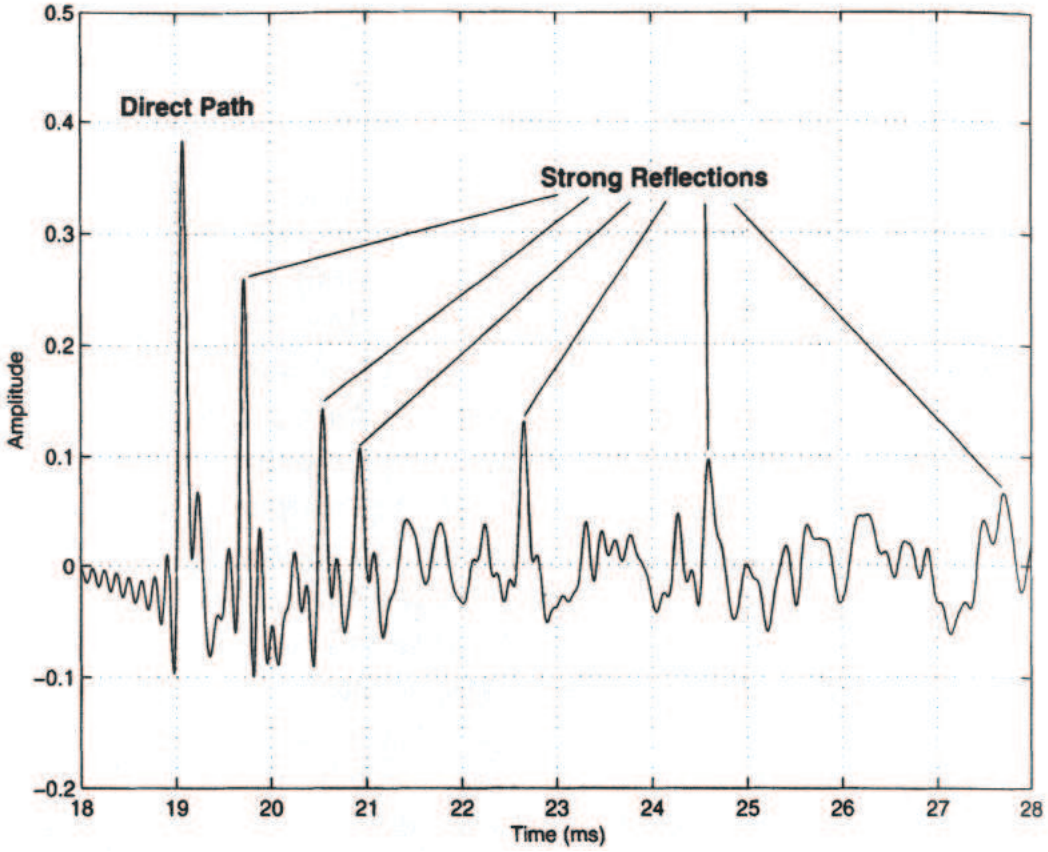

Figura 3.15: Segmento de 10ms da resposta de um impulso medido em uma típica sala de videoconferência. O caminho direto da fonte até o microfone e alguns caminhos refletidos estão destacados. Fonte: Brandstein e Ward (2001).

termos dos parâmetros de interesse.

Correlação Cruzada Generalizada (GCC)

A função de Correlação Cruzada Generalizada (GCC) é definida como a correlação cruzada de duas versões filtradas de $x_1(t)$ e $x_2(t)$ da Equação (3.11). Com a Transformada de Fourier destes filtros representadas por $G_1(\omega)$ e $G_2(\omega)$, respectivamente, a função GCC pode ser expressa no domínio de Fourier como:

$$R_{12}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty}(G_1(\omega)X_1(\omega))(G_2(\omega)X_2(\omega))^* e^{j\omega\tau}d\omega \tag{3.12}$$

Organizando a ordem dos filtros e definindo a função de peso dependente da frequência, $\Psi_{12} \equiv G_1(\omega)G_2(\omega)^*$, a função GCC pode ser expressa como:

$$R_{12}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\Psi_{12}(\omega)X_1(\omega)X_2(\omega)^* e^{j\omega\tau}d\omega \tag{3.13}$$

Idealmente, $R_{12}(\tau)$ será máximo na posição de atraso desejado. O TDOA estimado é calculado como:

$$\widehat{\tau}_{12} = argmax R_{12}(\tau) \tag{3.14}$$

Os possíveis valores do TDOA estão restritos a um intervalo finito, o qual é determinado pela separação física entre os microfones.

Correlação Cruzada Generalizada com PHAT (GCC-PHAT)

O objetivo da função de peso Ψ_{12} é enfatizar o valor do GCC para o valor correto do TDOA em ambientes com reverberação. Várias funções têm sido investigadas, e a função de peso PHAT (*Phase Transform*) proposta em Knapp e Carter (1976) e definida por

$$\Psi_{12}(\omega) = \frac{1}{|X_1(\omega)X_2^*(\omega)|} \tag{3.15}$$

funciona bem em condições reais. A função de peso PHAT enfatiza igualmente todas as frequências.

Potência da resposta direcionada com PHAT (SRP-PHAT)

Utilizando a técnica do SRP como fundamento, a Equação (3.3) pode ser reescrita no domínio de Fourier aplicando-se o filtro $G(\omega)$ como:

$$Y(\omega, q) = \sum_{n=1}^{N} G_n(\omega)X_n(\omega)e^{j\omega\Delta_n} \tag{3.16}$$

onde $X_n(\omega)$ e $G_n(\omega)$ são a Transformada de Fourier do *n-ésimo* sinal de microfone e seu filtro associado, respectivamente. O termo Δ_n é responsável por fase-alinhar, visando direcionar com o atraso apropriado para localização da fonte q.

O objetivo do algoritmo SRP-PHAT, proposto por DiBiase (2000), é combinar as vantagens do SRP com a robusteza oferecida pela função de peso PHAT. A função SRP-PHAT pode ser escrita como:

$$P(q) = \sum_{l=1}^{N}\sum_{k=1}^{N}\int_{-\infty}^{\infty} \Psi_{lk}(\omega)X_l(\omega)X_k^*(\omega)e^{j\omega(\Delta_k-\Delta_l)}d\omega \tag{3.17}$$

onde $\Psi_{lk}(\omega) = G_l(\omega)G_k(\omega)^*$ é análogo ao GCC de dois canais com função de peso definido pela Equação 3.13. Representa o somatório do resultado de todas as combinações possíveis entre dois microfones do array. A versão correspondente da função de peso PHAT, definida pela Equação (3.15), para vários microfones é:

$$\Psi_{lk}(\omega) = \frac{1}{|X_l(\omega)X_k^*(\omega)|} \tag{3.18}$$

3.4 Análise multimodal

A tecnologia digital nos provê de informação mutimídia cujo volume e complexidade tem se expandido rapidamente. A rápida explosão da informação multimídia aumenta a dificuldade de encontrar informação relevante, o que tem demandado um grande esforço para desenvolver ferramentas para detecção automática, reconhecimento e análise semântica do conteúdo mult-mídia. O presente trabalho trata apenas de dados audiovisuais, portanto a análise multimodal nesse contexto tem foco de relacionar a informação vinda de imagens, no caso vídeo, com informação vinda do áudio.

Dadas diversas fontes de dados de naturezas distintas, a fusão dos sensores, em particular áudio e vídeo, é necessária para reduzir a dependência de um sensor com um parâmetro *apriori* inválido, reduzir a incerteza ao estimar parâmetros devido a erros na modelagem dos sensores no mapeamento mundo-para-sinal e reduzir incertezas devido a ruídos no sinal (MARAGOS; POTAMIANOS; GROS, 2008). Sensores de naturezas diferentes podem ser complementares para o reconhecimento de determinados eventos, o que torna sua fusão necessária.

Segundo Maragos, Potamianos e Gros (2008), os maiores desafios que se encontram ao relacionar os dados de áudio e vídeo são:

- O volume de dados é muito grande.
- Diferentes taxas temporais: enquanto em um vídeo é comum se trabalhar com frequência de 25-30 frames por segundo, com áudio se trabalha normalmente com 44.100 amostras por segundo.
- Informações assíncronas: por exemplo, em um jogo de futebol a cena gol aparece antes do grito de gol do narrador.

Segundo Hennecke, Stork e Prasad (1996) e Pan et al. (1998), a fusão dos descritores pode ser feita em três níveis distintos:

- Ao nível de sinal: normalmente utilizada para reduzir incertezas na medição do sinal de um sensor. Busca retirar ruídos indesejáveis para o processamento e limita-se a trabalhar com sensores que possuem mesmo formato de sinal, portanto não é muito útil para aplicações multimídias.

- Ao nível de feições: a fusão de informações multimídia geralmente enfatiza como complementar eficientemente informação de diferentes sensores, por exemplo, de áudio e vídeo. É relativamente interessante trabalhar nesse nível pois feições de diferentes meios podem ser complementares para a tomada de decisão.

- Ao nível de decisão: nesse nível a fusão não leva em conta as dependências entre as feições vindas de diferentes sensores, portanto torna-se menos interessante.

Hennecke, Stork e Prasad (1996) denomina os três níveis de Fusão Antecipada (*Early Fusion*), Fusão Intermediária (*Intermediate Fusion*) e Fusão Tardia (*Late Fusion*) respectivamente.

A integração de feições extraídas de diversas fontes não é uma tarefa trivial. Os dois maiores problemas encontrados nesse processo são:

- A decisão: qual a decisão final se várias fontes (vídeo e som) provêm informação contraditória?

- A sincronização: em uma integração multimodal é preciso sincronizar os dados para análise.

A análise das informações audiovisuais é normalmente alcançada utilizando abordagens determinísticas ou probabilísticas. Enquanto os métodos determinísticos agrupam cenas consecutivas utilizando medidas apropriadas ((LIENHART; PFEIFFER; EFFELSBERG, 1997), (SARACENO; LEONARDI, 1998)), os métodos probabilísticos utilizam os Modelos Escondidos de Markov (HMM) para representarem seus estados contidos (neste caso *fala* e *não-fala*) ((FERMAN; TEKALP, 1999), (WOLF, 1997)). Embora essas duas abordagens ainda estão sendo aperfeiçoadas, a escolha por HMM se faz mais adequada devido ao fato do comportamento randômico de qualquer linguagem natural, a qual não permite colocar regras determinísticas para modelar este comportamento, como foi descrito em Alatan, Akansu e Wolf (2001).

Na classificação de dados multimídia, muitas técnicas vêm do reconhecimento de padrões clássicos. Os classificadores Bayesianos são bastante utilizados nessa tarefa, por três razões básicas:

- São bastante eficientes em lidar com dados de muitas dimensões.
- Podem gerenciar limites não lineares.
- São simples de utilizar.

Finalmente, redes Bayesianas são ferramentas bastante flexíveis. Essas redes permitem modelar qualquer grafo de dependência entre variáveis randômicas.

Quando muitas fontes de dados de observações são levadas em conta, os HMMs podem ser adaptados. Se os fluxos de dados são sincronizados e compartilham a mesma frequência, uma primeira solução é fundir os descritores a cada instante com o objetivo de criar descritores multimodal maiores, mas essa técnica é restrita para a fusão de descritores de mesma natureza. Quando os descritores são de naturezas diferentes, como o caso do áudio e vídeo, os dois fluxos de dados podem ser assumidos independentes e modelados pela suas próprias HMMs, mas depois são combinados nos pontos de sincronização (BOURLARD; DUPONT, 1996) (DUPONT; LUETTIN, 2000). A Figura 3.16 ilustra a fusão de dois descritores de mesma natureza em uma HMM e a Figura 3.17 ilustra dois descritores de naturezas diferentes em suas próprias HMMs, onde depois seus resultados são combinados nos pontos de sincronização.

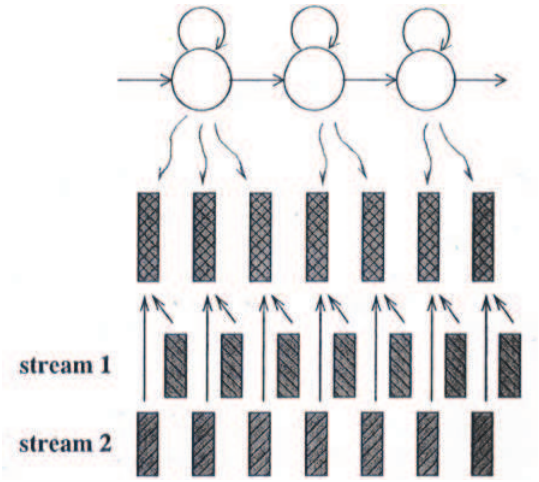

Figura 3.16: Fusão de descritores síncronos com HMM. Fonte: Maragos, Potamianos e Gros (2008).

Os dois casos extremos de HMM de vários descritores são o síncrono e o assíncrono. No síncrono, os dois modelos de Markov possuem um compartilhamento na sequência de estados e podem ser considerados como sincronizados a cada instante. No assíncrono não existe sincronização (com exceção no início e no fim do processo) e o modelo é equivalente ao modelo síncrono no produto do espaço de estado, e normalmente chamado de modelos do produto.

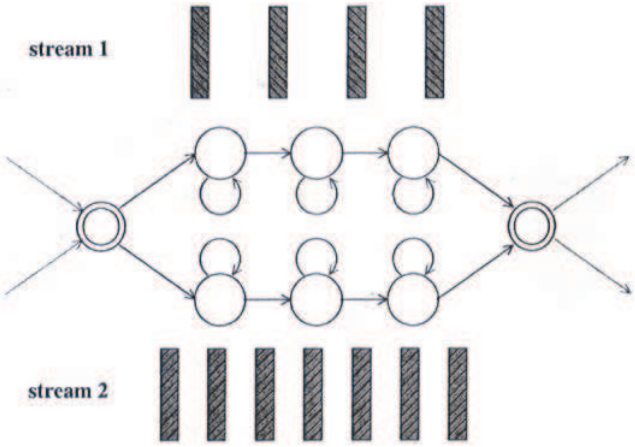

Figura 3.17: Fusão de descritores assíncronos com HMM. Fonte: Maragos, Potamianos e Gros (2008).

Considera-se um par de sequências $y^{(1)}$ e $y^{(2)}$ onde cada uma possui T amostras $y^{(i)} = (y_1^{(i)}, y_2^{(i)}, ..., y_T^{(i)})$. Então o modelo HMM síncrono para vários descritores podem ser representados por uma sequência de estados escondidos comum $x = (x_1, x_2, ..., x_T)$, com x_t utilizando valores de um único conjunto de estados (Equação 3.19).

$$p(y^{(1)}, y^{(2)}|x) = p(x_0) \prod_{t=1}^{T} p(x_t|x_{t-1}) p(y_t^{(1)}|x_t) p(y_t^{(2)}|x_t) \tag{3.19}$$

No caso do modelo HMM com descritores assíncronos, cada modalidade tem seu próprio conjunto de estados escondidos $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_T^{(i)})$, com $x_t^{(i)}$ utilizando valores de diferentes conjuntos de estados (Equação 3.20).

$$p(y^{(1)}, y^{(2)}|x^{(1)}, x^{(2)}) = p(x_0^{(1)}, x_0^{(2)}) \prod_{t=1}^{T} p(x_t^{(1)}, x_t^{(2)}|x_{t-1}^{(1)}, x_{t-1}^{(2)}) p(y_t^{(1)}, y_t^{(2)}|x_t^{(1)}, x_t^{(2)}) \tag{3.20}$$

O produto HMM resultante permite o modelo assíncrono, desde que a cada momento seja possível obter qualquer combinação dos estados unimodais.

3.5 Considerações sobre as técnicas avaliadas

Com base na revisão bibliográfica realizada, pode-se concluir que o algoritmo de detecção de faces proposto em Viola e Jones (2001) apresenta uma boa taxa de detecção para faces frontais, parecendo uma boa escolha para a localização de pessoas em aplicações de videoconferência. Embora essa técnica seja relativamente rápida comparada com outros detectores de

face, a inclusão de um algoritmo de pré-processamento baseado em informações de cor de pele é importante para reduzir o espaço de busca, diminuindo o tempo de execução da detecção de faces.

Com relação à localização de áudio usando arranjos de microfones, a técnica SRP-PHAT parece ser adequada a ambientes reais, pois é menos sensível a problemas de reverberação (BRANDSTEIN; WARD, 2001). Entretanto, como todas as possíveis combinações entre dois microfones são usadas, o tempo de execução pode crescer rapidamente à medida que o número de microfones é aumentado. Por outro lado, o uso de mais microfones também insere redundância na localização, e os resultados tendem a ser melhores.

Finalmente, a fusão de dados de áudio e vídeo usando HMMs se mostrou adequada em trabalhos relacionados. Devido a sua capacidade de modelar o sinal de áudio com bons resultados na prática, a idéia é explorar HMMs neste trabalho para realizar a combinação dos resultados dos processamentos de áudio e vídeo.

4 *Modelo Proposto*

Para o objetivo de identificar o locutor em uma vídeoconferência é necessário reunir uma série de técnicas específicas que contribuem para esse objetivo. A seguir, as técnicas utilizadas são descritas em maiores detalhes.

4.1 Detecção dos membros da vídeoconferência usando informação de vídeo

A informação buscada no vídeo é a presença dos membros da videoconferência. Isso é feito encontrando as respectivas faces no ambiente, onde cada face é uma potencial fonte sonora a ser combinada com a informação de áudio. A detecção de faces em sequência de vídeo é feita com a utilização do algoritmo proposto em Viola e Jones (2001) combinado com classificação de cor de pele utilizando imagens integrais. Essa abordagem já foi empregada por Schramm (2009), que utilizou o algoritmo de Viola e Jones (2001) treinado para encontrar faces, adaptando a janela deslizante que busca pelos padrões geométricos (Figura 3.1) para deslizar apenas nas áreas classificadas pela presença de cor de pele, otimizando e muito o custo computacional do algoritmo. Um dos motivos da rapidez do algoritmo foi a utilização da imagem integral para cálculo das áreas classificadas como *pele*. A imagem integral é uma técnica também elaborada em Viola e Jones (2001) para calcular as formas geométricas dentro da janela deslizante a um custo computacional baixo.

Para aprimorar o algoritmo de detecção de faces no vídeo proposto em Viola e Jones (2001), foi feita uma pesquisa sobre detecção de pele em imagens digitais com o objetivo de minimizar a área a ser pesquisada pelo algoritmo em busca das faces. Existem várias pesquisas para o modelo de distribuição dos pixels relativos a pele nos diversos espaços de cor (KAKUMANU; MAKROGIANNIS; BOURBAKIS, 2007). Schmugge et al. (2007) descreve que a forma mais utilizada para elaborar o modelo de cor de pele é converter o espaço de cor RGB para um espaço de cor que contenha um componente de luminosidade e retirá-lo do modelo, assim eliminando as diferenças de luminosidade das imagens, mantendo apenas os componentes de cor,

mas conclui que a falta do componente de luminosidade no modelo diminui a performance do classificador.

A idéia principal do modelo de cor de pele proposto é utilizar um conjunto de classificadores simples e rápidos, os quais combinados podem produzir bons resultados na classificação. De fato, esse é o objetivo do algoritmo AdaBoost, que combina de forma ponderada classificadores fracos construindo classificadores fortes, sendo esses mais complexos. O primeiro passo para construção de um classificador forte baseado no AdaBoost é a definição dos classificadores fracos, os quais devem ser simples e rápidos de processar. No modelo, um conjunto de planos no espaço de cor RGB foram escolhidos, formando os limites de decisão entre as classes *pele* e *não-pele*, compondo o conjunto de classificadores fracos. Para formar os planos, foram gerados vetores unitários de acordo com a equação

$$\boldsymbol{u}(\alpha,\beta) = (\sin\alpha, \cos\alpha\cos\beta, \cos\alpha\sin\beta)^T, \tag{4.1}$$

onde α e β são os ângulos de $\boldsymbol{u}$ em cordenadas esféricas, como mostrada na Figura 4.1.

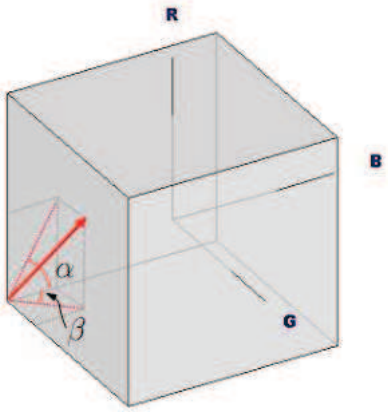

Figura 4.1: Vetores unitários utilizados para gerar os planos no espaço de cor RGB.

Para $0 \leq \alpha < \pi$, $0 \leq \beta < \pi$, os vetores $\boldsymbol{u}(\alpha,\beta)$ possuem suas coordenadas α e β de acordo com

$$\alpha_i = \frac{i\pi}{N_a}, \quad \beta_j = \frac{j\pi}{N_a}, \quad i = 0,..,N_a, \quad j = 0,...,N_a, \tag{4.2}$$

formando vetores uniformemente deslocados, gerando N_a^2 vetores, onde N_a é o número de amostras (deslocamentos) em cada eixo. Estes vetores unitários resultantes são utilizados para construir um conjunto de N_a^2 classificadores fracos $h_{ij}(\boldsymbol{x})$, dados por

$$h_{ij}(\boldsymbol{x}) = \begin{cases} 1, & \text{se } p_{ij}\boldsymbol{u}(\alpha_i,\beta_j)^T\boldsymbol{x} \geq p_{ij}T_{ij} \\ -1, & \text{senão} \end{cases}, \tag{4.3}$$

onde $\boldsymbol{x} = (R,G,B)^T$ é o vetor de cor no espaço de cor RGB, T_{ij} é o limiar e $p_{ij} \in \{-1,1\}$ é um valor de paridade. É importante salientar que ambos T_{ij} e p_{ij} são obtidos quando o AdaBoost é aplicado. Neste trabalho foi utilizado $N_a = 18$, portanto $361 = 19^2$ classificadores fracos foram gerados.

Depois de aplicar o AdaBoost, um classificador forte $H(\boldsymbol{x})$ é obtido através de

$$H(\boldsymbol{x}) = \begin{cases} 1, & \text{se } \sum_{k=1}^{N_w} w_k h_{i_k j_k}(\boldsymbol{x}) \geq 0 \\ 0, & \text{senão} \end{cases}, \tag{4.4}$$

onde $h_{i_k j_k}$ são os classificadores fracos selecionados com correspondentes pesos w_k, e N_w é o número de classificadores fracos utilizados para construir o classificador forte.

Para treinar o classificador, um conjunto de amostras contendo exemplos positivos (pixels de *pele*) e exemplos negativos (pixels de *não-pele*) foi necessário. Foi utilizado o banco de dados dbSkin, da Universidad de Chile, que contém 103 imagens com respectivo conjunto verdade[1]. Um exemplo é mostrado na Figura 4.2 e a disposição dos pixels de *pele* no espaço de cor RGB das imagens do banco de dados é mostrada na Figura 4.3.

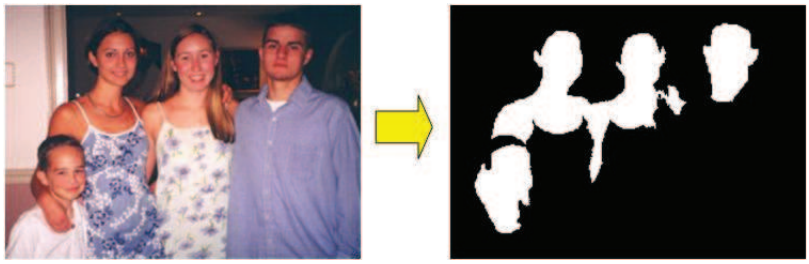

Figura 4.2: Exemplo de imagem do banco de dados dbSkin, onde a segunda imagem representa em branco o conjunto verdade das regiões de pele da primeira imagem.

Utilizar todos os pixels de todas as imagens poderia tornar tendencioso o classificador para a cor que aparece muitas vezes em um fundo homogênio, principalmente se for em uma imagem de grande dimensão. Para evitar tal tendência, foi coletado um número fixo de amostras (900, definido empiricamente) de cada imagem do banco de dados, utilizando uma grade sobreposta na imagem. Esta amostragem uniforme minimizou o problema da cor dominante, mas produz cerca de 15% de seus pixels pertencendo a classe *pele*, pois há muito mais pixels de fundo do

[1] Imagem binária, marcando a posição dos pixels de *pele*.

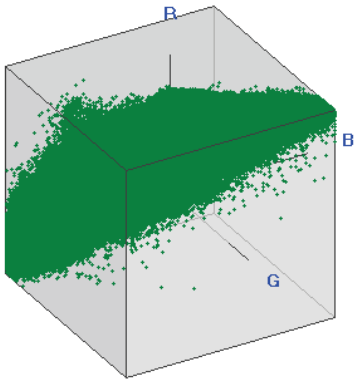

Figura 4.3: Disposição dos pixels de *pele* no espaço de cor RGB das imagens do banco de dados dbSkin.

que pele em uma imagem deste banco de dados, o que deixa o classificador tendencioso para a classe *não-pele*. Baseado no treinamento realizado em Viola e Jones (2001), foi utilizado no treinamento um conjunto com a proporção 2:1 entre *não-pele* e *pele* (em Viola e Jones (2001) essa proporção foi entre as classes *não-face* e *face*), selecionando pixels de *não-pele* de forma randômica em cada imagem. No total, cerca de 41.000 amostras foram utilizadas para o treinamento, e outras 9.000 amostras para o teste de performance dos classificadores fortes encontrados.

Os classificadores fortes resultantes da aplicação do AdaBoost são utilizados para restringir as áreas onde o algoritmo de Viola e Jones (2001) deve procurar por faces, evitando, assim, processamento desnecessário. Em cada imagem I, o teste dos classificadores fortes é aplicado para cada pixel da imagem, gerando uma imagem binária S, com valor 1 nos pixes considerados *pele*, e 0 nos demais. Dada uma sub-região retangular W, representando as fronteiras de uma candidata a face, é esperado que exista um percentual significante de pixels no interior do retângulo classificados como *pele*. Por outro lado, olhos, cabelos e o cenário de fundo não são detectados como *pele*, portanto esse percentual não pode ser também muito alto. Dada a janela $W(u,v)$ retangular $m \times n$, centralizada no pixel (u,v), é considerada candidata a face se

$$T_1 \leq R(u,v) \leq T_2 \tag{4.5}$$

onde

$$R(u,v) = \frac{1}{nm} \sum_{(x,y) \in W(u,v)} S(x,y) \tag{4.6}$$

é a razão de pixels detectados como *pele* e $0 \leq T_1 < T_2 \leq 1$ são os limiares que representam os

percentuais mínimo e máximo de pixels com cor de pele para considerar uma região candidata a face.

Como normalmente não há informação *a priori* de onde as faces podem estar, a janela W é posicionada em todas as possíveis sub-regiões da imagem. Entretanto, para selecionar faces de diferentes escalas, janelas com tamanhos variáveis foram empregadas. A implementação do teste descrito em (4.5) teria um tempo de processamento muito alto (nm somatórios para cada posição e escala da janela). Portanto, tais somatórios foram calculados utilizando o conceito de imagens integrais (VIOLA; JONES, 2001), e o somatório substituído por quatro somas simples, resolvendo o problema do custo computacional deste cálculo.

4.2 Localização do locutor através da direção da fonte sonora

Para que seja possível a localização da fonte sonora em um ambiente de vídeoconferência é necessária a utilização de um arranjo de microfones. É necessário mais de um microfone para que seja feita uma triangulação entre os microfones e a fonte sonora, assim calculando-se a direção da mesma. O primeiro passo dessa tarefa é a captura do sinal de um arranjo de microfones, portanto foi necessária a aquisição de um equipamento apropriado, no caso, composto por uma placa PCI HDSP 9632 do fabricante Hammerfall DSP System. A placa possui a capacidade de trabalhar com até 16 microfones e suporta uma frequência de amostragem de 192kHz por microfone (Figura 4.4).

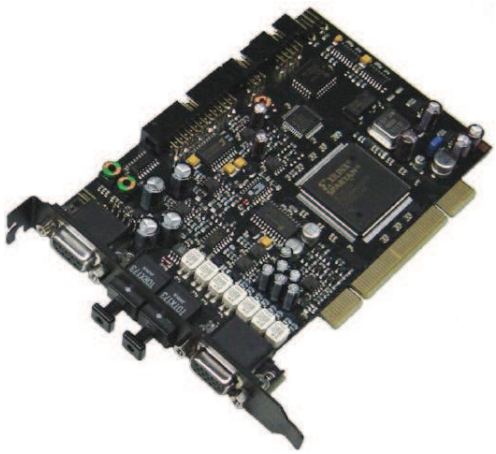

Figura 4.4: Placa PCI HDSP 9632 do fabricante Hammerfall DSP System.

Também foi necessário adquirir o arranjo de microfones, que no caso contou com seis unidades. O módulo adquirido amplifica e digitaliza o sinal dos microfones individualmente e os envia para a placa PCI HDSP 9632. O equipamento adquirido foi o OctaMic II da RME Intelligent Audio Solutions (Figura 4.5). Ele possui a capacidade para oito entradas analógicas

(microfones) operando cada uma com resolução de até 24bits e frequência de amostragem de até 192kHz.

Figura 4.5: OctaMic II da RME Intelligent Audio Solutions.

O algoritmo utilizado para a localização da fonte sonora é o SRP-PHAT, devido sua maior robustez em ambientes com presença de ruído e reverberação. O sinal de cada microfone é capturado em *buffers*[2], cujo tamanho é calculado de acordo com a precisão e frequência que se deseja. Devido ao cálculo do SRP-PHAT ser desenvolvido no domínio de frequência, a primeira operação realizada com o sinal é o FFT. O algoritmo do FFT utiliza uma quantidade de amostras tal que $B = 2^n$, onde B é a quantidade de amostras a ser utilizada na transformada e n um número inteiro positivo, portanto o *buffer* utilizado na captura do som precisa ser uma potência de 2.

O SRP-PHAT do experimento utiliza o conceito de Campo Próximo (*Near Field*) para o cálculo da localização, visto que a largura do arranjo e a distância do mesmo até o locutor são semelhantes. Os locutores foram posicionados em linha a uma distância fixa de um metro do arranjo. O arranjo possui os microfones dispostos também em linha, igualmente espaçados entre si em 15 centímetros. No centro do arranjo foi posicionada uma câmera típica de conferência na Internet. A Figura 4.6 mostra uma foto do ambiente onde foram realizados os testes.

4.3 Localização do locutor utilizando áudio e vídeo

O modelo proposto para integração do sinal de áudio e vídeo neste trabalho se enquadra na Fusão Intermediária, definida em Hennecke, Stork e Prasad (1996), onde feições reconhecidas em sinais de naturezas diferentes contribuem para a tomada de decisão final, que é a localização do locutor. A escolha pela fusão nesse nível se deve ao fato das feições encontradas no sinal de áudio e vídeo serem complementares, já que a voz do locutor deverá sair da mesma região espacial onde sua face é detectada.

Os sinais são primeiramente sincronizados a cada coleta de amostras de áudio para o cálculo do SRP-PHAT. Esse tempo pode variar de acordo com a precisão necessária e com a frequência

[2]Memória reservada para armazenar dados durante algum processamento.

Figura 4.6: Ambiente onde foram realizados os testes.

de amostragem.

O HMM foi modelado de tal forma que cada estado escondido representasse um setor no espaço, no qual o locutor poderia ser encontrado. Seus símbolos observáveis são as possíveis respostas do SRP-PHAT, ponderadas pelas regiões onde foram encontradas faces pela câmera. A fusão de sinais caracteriza-se nesse trabalho por uma função de peso, atribuídas pelo sinal de vídeo. Para que os resultados sejam coerentes, foi feita uma calibração espacial do arranjo de microfone com a detecção de faces da câmera.

Para o registro das fontes de dados de áudio e vídeo foi feito um mapeamento espacial das áreas de cobertura da câmera e do microfone, subdividindo o espaço em $S_1, S_2, ..., S_N$, identificados no HMM como possíveis estados do modelo. Após o mapeamento vem a calibração, assim o setor S_i do áudio e do vídeo representam a mesma porção do espaço. As Figuras 4.7, 4.8 e 4.9 representam esse mapeamento para o sinal de áudio, vídeo e ambos respectivamente.

Com o processamento do sinal de áudio para calcular o SRP-PHAT, tem-se como resultado um vetor representando em cada elemento um setor do espaço varrido pelo algoritmo. O valor máximo deste vetor determina o setor onde encontra-se a fonte sonora. Caso a quantidade de setores pesquisados pelo SRP-PHAT seja superior a quantidade de estados da HMM, então é preciso agrupar estes setores em bins. A Figura 4.7 mostra um exemplo de curva resultante do SRP-PHAT associado ao setor do espaço no qual foi realizado o cálculo.

Devido ao ruído do ambiente principalmente causado por reverberações, um relacionamento com outras fontes de informação se faz necessário, neste caso é com o sinal de vídeo. Com o processamento do sinal de vídeo pelo algoritmo de Viola e Jones tem-se a posição das faces encontradas (Figura 4.8). Após a posição ser alocada em algum setor mapeado servirá de

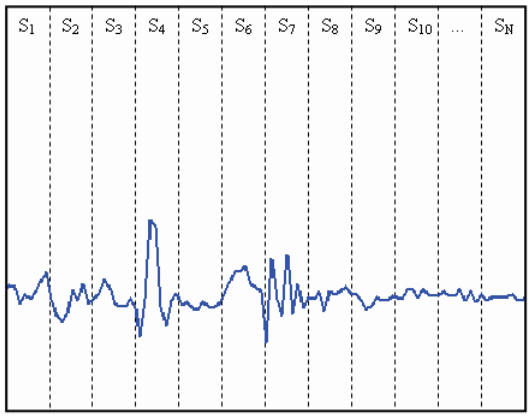

Figura 4.7: Exemplo de curva resultante do SRP-PHAT relacionada com os setores mapeados.

peso para os valores retornados pelo SRP-PHAT, aumentando assim as chances de encontrar a fonte sonora apenas onde foram encontradas faces.

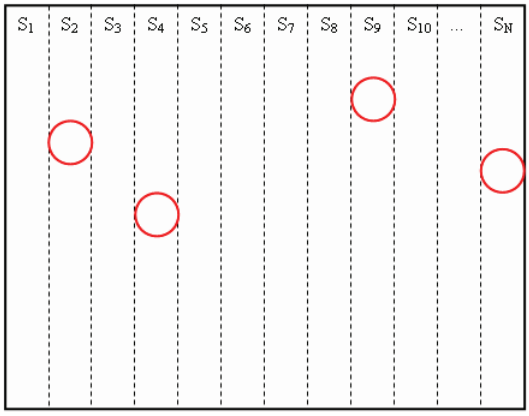

Figura 4.8: Exemplo de faces detectadas relacionadas com os setores mapeados.

A Figura 4.9 mostra uma possível situação esperada, onde o valor mais alto do SRP-PHAT coincide com uma posição onde também foi encontrada uma face. Com essa fusão dos sinais espera-se um aumento da robustez do algoritmo, permitindo que mesmo em momentos de pausa na fala, ruído ou oclusão da face, a localização do locutar não seja afetada.

Com o vetor $\boldsymbol{S} = (S(1), S(2), ..., S(N))$ de dimensão N que contém os valores obtidos pelo SRP-PHAT nas N posições, o observável do HMM proposto é um vetor $\boldsymbol{O} = (O_1, O_2)$, onde

$$O_1 = argmax(\boldsymbol{S}), \quad O_2 = \frac{S(O_1)}{\mu(\boldsymbol{S})} \tag{4.7}$$

e $\mu(\cdot)$ denota a média do vetor. É importante observar que O_1 indica a posição do maior pico da SRP-PHAT, o que é normalmente utlizado para localização baseada em arranjos de microfones. O segundo observável, O_2, é a razão entre a altura máxima e a altura média, e pode

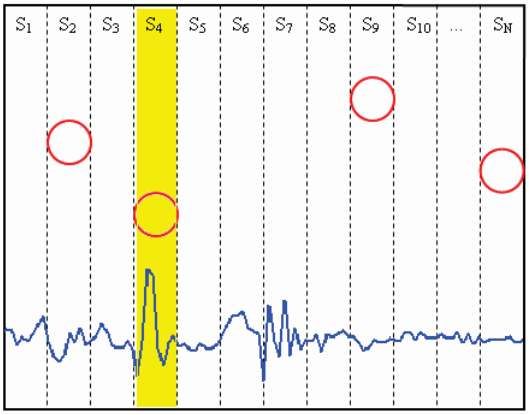

Figura 4.9: Possível situação esperada, onde o valor mais alto do SRP-PHAT coincide com uma posição onde também foi encontrada uma face.

ser interpretado como uma medida de confiança na extração do pico: em situações de fala, o maior pico tende a ser bem maior que todos outros valores, gerando um valor alto para O_2; por outro lado, em situações de silêncio, todos valores de $\boldsymbol{S}$ tendem a ser semelhantes, gerando um valor de O_2 baixo (próximo de 1).

Com as informações acima o modelo agora pode ser exposto.

X: Variável observável. Vetor $O = (argmax(\boldsymbol{S}), \ \frac{S(O_1)}{\mu(\boldsymbol{S})})$ resultante do SRP-PHAT.
N: Quantidade de setores no qual o espaço foi dividido.
M: Possíveis valores de X.

Ainda é preciso definir as probabilidades de transição entre os N estados e as probabilidades de acontecer os observáveis em cada estado. Nesse trabalho a utilização do algoritmo Baum-Welch não foi possível, devido a duas razões:

- Um treinamento implica em utilizar dados conhecidos com todos os comportamentos possíveis, pois como são vários estados, devido aos vários segmentos da área de cobertura, seriam muitas possibilidades, dificultando e muito a utilização de algoritmos para esse treinamento.
- O número de estados é variável de acordo com a precisão que se espera, então para cada nova resolução espacial precisaria um novo treinamento.

Como definir então as matrizes A e B de probabilidades? A solução encontrada foi modelar com o comportamento esperado em um ambiente real de videoconferência. As premissas adotadas foram:

- Probabilidade de transição entre os estados

  - É esperado que o locutor faça pequenos movimentos podendo ser localizado em setores vizinhos em um curto espaço de tempo.
  - Não é esperado que o locutor fique alternando para setores distantes.

- Probabilidade de ocorrer os observáveis em determinado estado
  - Para cada estado é esperado que o observável tenha uma maior probabilidade de ocorrer com um valor alto na confiança.

No HMM proposto, o observável é o vetor $O = (O_1, O_2)$, detalhado na Equação 4.7. Para o estado i, a distribuição de probabilidades dos observáveis $p_i(O_1, O_2)$ (matriz B do modelo HMM) é dada por:

$$p_i(O_1, O_2) = K_B \exp\left\{ -\frac{|O_1 - i|}{f(O_2)} + \frac{O_2 - O_2^{\max}}{\alpha} \right\}, \tag{4.8}$$

onde $f(O_2)$ é uma função que controla o espalhamento da exponencial para diferentes valores de O_2 e K_B o coeficiente de normalização. f deve ser monotonicamente descrescente no intervalo $[1, O_2^{\max}]$, de modo que a exponencial produza um pico saliente quando O_2 é alto (ou seja, quando há uma maior confiança no pico extraído pelo SRP-PHAT), e um decaimento mais suave quando O_2 é menor (ou seja, a probabilidade de ocorrência do pico quando a confiança é menor é mais distribuída entre as posições). Experimentalmente, foi definida uma função quadrática para f, dada por

$$f(O_2) = 0.1 + (2N - 0.1)\frac{(O_2 - O_2^{\max})^2}{(O_2^{\max} - 1)^2} \tag{4.9}$$

Ainda na Equação (4.8), α é um parâmetro que controla o decaimento de $p_i(O_1, O_2)$ à medida que O_2 diminui, assumindo que em situações de fala espera-se que confianças mais altas sejam produzidas no SRP-PHAT. Experimentalmente, definiu-se $\alpha = (o_2^{\max})/2$. A Figura 4.10 mostra a FDP resultante da matriz de ocorrência dos observáveis em um determinado estado.

A matriz de transição $\boldsymbol{A} = [a_{ij}]$ contém as probabilidades de transição entre quaisquer dois estados. Em situações de fala, assume-se que o locutor não se desloca muito rapidamente, de modo que a probabilidade maior é que ele fique em seu estado anterior (ou se mova para algum estado vizinho). No modelo proposto, essa hipótese foi modelada matematicamente através de outra função exponencial:

$$a_{ij} = K_A \exp\left\{\frac{|i-j|}{\beta}\right\}, \tag{4.10}$$

onde β controla o decaimento da exponencial e K_A é o coeficiente de normalização para o estado i. Experimentalmente, definiu-se $\beta = N/5$. A Figura 4.11 mostra a FDP resultante.

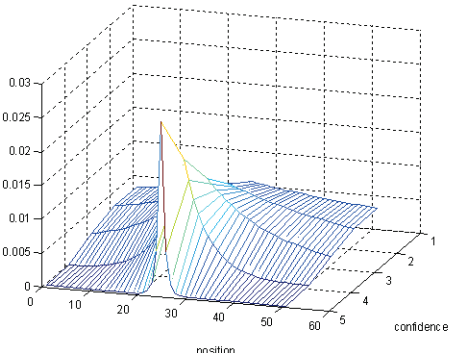

Figura 4.10: FDP da matriz de ocorrência dos observáveis em um determinado estado, sendo neste caso o estado 25. A abertura da exponencial decresce à medida que a confiança cresce.

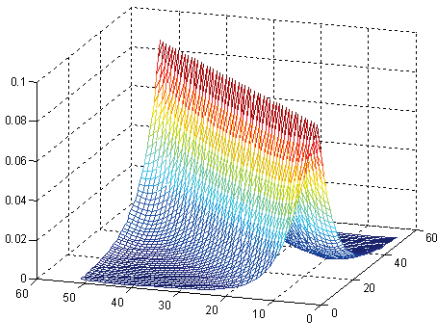

Figura 4.11: FDP da matriz de transição dos estados (A). Função exponencial para fortalecer a permanência no mesmo estado enquanto deixa menos provável a transição para estados mais distantes.

5 *Resultados*

5.1 Detecção dos membros da vídeoconferência usando informação de vídeo

A primeira parte do projeto é buscar a informação de todas as pessoas presentes em um ambiente de videoconferência, através da detecção das respectivas faces, utilizando apenas o vídeo como fonte de dados. Para essa tarefa foi utilizada a biblioteca OpenCV[1] que já possui implementado o algoritmo proposto por Viola e Jones (2001).

Com esse processamento tem-se detectado no vídeo as possíveis fontes sonoras de interesse. Ainda é preciso processar o sinal de áudio e fazer a análise multimodal desses sinais para concluir a localização da fonte sonora. Portanto o processamento do vídeo não pode ser muito custoso. A alternativa adotada nesse trabalho foi a redução do espaço de busca do algoritmo de detecção das faces nas regiões da imagem que possuam uma quantidade mínima de pixels classificados como *pele*.

Os planos do espaço de cor RGB que melhor classificam um pixel como *pele* encontrados pelo Adaboost podem ser vistos na Tabela 5.1, juntamente com seus pesos. Os dois planos compõem o primeiro nível da cascata de classificadores encontrados pelo AdaBoost. A separabilidade das classes *pele* e *não-pele* que eles representam pode ser visto na Figura 5.1, onde os gráficos mostram os histograma das classes *pele* (em azul) e *não-pele* (em vermelho) do primeiro e segundo plano respectivamente.

O desempenho dos classificadores dos níveis um e dois encontrados pode ser visto na Matriz de Confusão da Tabela 5.2. A Figura 5.2 mostra exemplo de imagem do banco de dados dbSkin,

[1] Biblioteca de código aberto com funções sobre Visão Computacional desenvolvida pela Intel.

Tabela 5.1: Planos e respectivos pesos selecionados pelo Adaboost.

	Plano RGB	Peso
Classificador fraco 1	$0.77R - 0.63G - 0.11B > 12.5$	1.12
Classificador fraco 2	$0.64R - 0.75G + 0.13B > 19.5$	1.05

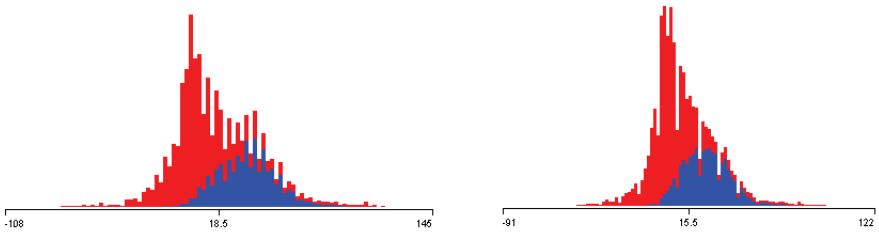

Figura 5.1: Distribuição das classes *pele* (azul) e *não-pele* (vermelho) no primeiro e segundo melhores planos encontrados pelo Adaboost, respectivamente.

Tabela 5.2: Desempenho dos classificadores (strong classifiers) encontrados

Primeiro nível				
	Classificado como			
	Pele	Não-pele	Total	Acerto
Pele	1021	47	1068	95,60%
Não-pele	2800	5132	7932	64,70%
Segundo nível				
	Classificado como			
	Pele	Não-pele	Total	Acerto
Pele	998	70	1068	93,44%
Não-pele	2638	5294	7932	66,74%

onde a segunda imagem representa em branco o conjunto verdade e a terceira imagem é a classificação da primeira imagem com os classificadores encontrados na pesquisa. A Figura 5.3 mostra alguns exemplos de imagens classificadas nos dois níveis do classificador encontrado.

Figura 5.2: Exemplo de imagem do banco de dados dbSkin, onde a segunda imagem representa em branco o conjunto verdade e a terceira imagem é a classificação da primeira imagem com os classificadores encontrados na pesquisa.

Em termos de tempo de processamento, o pré-processamento para identificação de regiões com cor de pele foi de aproximadamente 18 ms para imagens de tamanho 250×250. As imagens utilizadas são do banco de dados LFW (LEARNED-MILLER, 2007), destinado à detecção de faces em imagens digitais. O método proposto foi implementado em C++, e todos os experimentos foram feitos em um computador com processador Intel Core 2 Duo de 2.4 GHz, com 2 GB de memória RAM.

Figura 5.3: No topo a imagem original do banco dbSkin. No meio a imagem classificada usando o primeiro nível do classificador. Em baixo a imagem classificada usando os dois níveis do classificador.

Tabela 5.3: Comparação entre o método proposto e o algoritmo de Viola e Jones (2001).

Algoritmo	Falso pos.	Falso neg.	Tempo médio (ms)
Viola & Jones	35	20	63.07
Método proposto	22	22	47.16

A Tabela 5.3 resume a eficiência do algoritmo porposto utilizando 500 imagens do banco de dados LFW. Como esperado o número de falso positivos diminui quando a região de procura é reduzida, mas existiu um pequeno acréscimo no número de falso negativos, que ocorrem quando o teste de cor de pele falha e a face válida não é incluída na validação posterior utilizando as geometrias.

O ganho de performance no pré-processamento tende a aumentar quando imagens maiores são utilizadas e a face corresponde a uma porção menor da imagem. No banco de dados LFW, nas imagens de 250×250 as faces ocupam uma grande porção. Para ilustrar esse efeito foi avaliado o tempo de execução em um vídeo com e sem o pré-processamento da cor de pele com resolução de 640×480. Os tempos de processamento de cada frame do vídeo são mostrados no gráfico na Figura 5.4, onde pode ser visto que o algoritmo proposto leva 30% do tempo requerido pela aplicação direta do algoritmo do Viola e Jones. Também mostra o montante de tempo gasto pelo pré-processamento (em média 34 ms) e o tempo gasto no total do processamento (em média de 84 ms). A média de tempo requerido para o processamento do algoritmo de Viola e Jones (2001) é 301 ms.

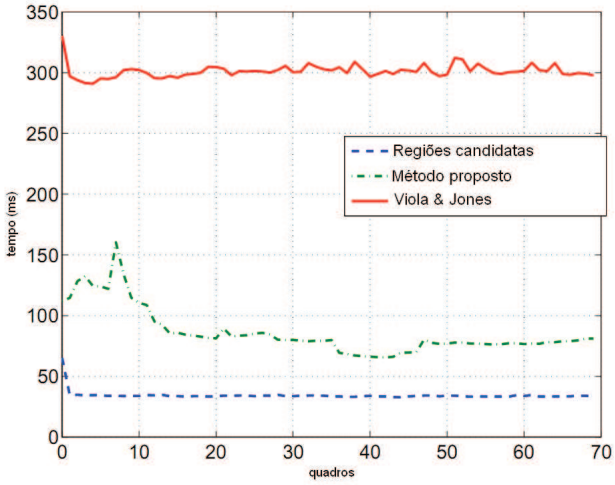

Figura 5.4: Tempos de execução para uma sequência de vídeo de tamanho 640×480.

5.2 Localização do locutor através da direção da fonte sonora

A localização do locutor atrevés do áudio foi feita utilizando-se um arranjo de seis microfones fixados na parede, organizados em uma linha reta, com espaço de 15 cm entre um microfone e o seguinte. Não houve preocupação em posicioná-los de tal forma que minimizasse efeitos de reverberação nem foram retirados obstáculos que reflitam a onda sonora, para que o resultado fosse o mais semelhante possível a um ambiente real de videoconferência. Também não foram retiradas nenhuma outra fonte de ruído do ambiente, tais como ar-condicionados, pessoas circulando e outros computadores ligados.

No cálculo do SRP-PHAT foi considerado a velocidade de propagação do som no ar de 343,4 m/s, que é a velocidade medida quando a temperatura do ar está em 20^oC. Nos experimentos foram utilizadas 4096 amostras de som para cada microfone para o cálculo do SRP-PHAT. A uma frequência de 44.100 Hertz isso significa um cálculo da posição do locutor a cada 0,0929 segundo.

O que observou-se foi um resultado satisfatório mas não ideal. O algoritmo SRP-PHAT localiza corretamente o locutor quando não existe obstáculo que provoque uma forte reverberação como uma parede, nem pode existir uma fonte de ruído muito alto competindo com a voz do locutor. Esses dois fatores são decisivamente perigosos para o resultado do SRP-PHAT quando há pausas na fala do locutor entre uma palavra e outra, originando hiatos na emissão de som. Momentos de silêncio durante uma conversação são normais, mas a localização do locutor precisa continuar consistente. A Figura 5.5 mostra três situações identificadas nos testes de performance do algoritmo, onde uma bolinha preta é desenhada no setor onde a fonte sonora foi detectada. As bolinhas em tonalidades de cinza mais fraco são as posições anteriores de-

tectadas pelo algoritmo. Na primeira imagem o locutor da esquerda é corretamente localizado pelo algoritmo, na segunda imagem o locutor da direita é localizado. Na terceira imagem, devido a reverberação do ambiente, a localização ficou incorreta e é possível de perceber que o SRP-PHAT localizou uma fonte sonora um pouco mais a esquerda. Depois descobriu-se que era o efeito da parede lateral da sala, que refletia a onda sonora.

Figura 5.5: Sequência de vídeo para análise da localização pelo áudio utilizando o SRP-PHAT. Na primeira imagem o locutor da esquerda é corretamente localizado pelo algoritmo, na segunda imagem o locutor da direita é localizado. Na terceira imagem, devido a reverberação do ambiente, a localização ficou incorreta.

O custo computacional do processamento do áudio também precisou ser analisado para garantir a viabilidade do projeto, ainda mais utilizando um arranjo de seis microfones, portanto seis fontes de áudio. O SRP-PHAT realiza operações relativamente simples dentro do domínio de frequências, então a preocupação foi analisar o custo computacional inerente do FFT e iFFT. Sendo n a quantidade de amostras envolvidas no cálculo do FFT, de acordo com a notação assintótica, o custo computacional do FFT pode ser definido como $O(n \times \log(n))$, ou seja, o tempo de execução é logarítmico e cresce ligeiramente a medida que n cresce (COCHRAN et al., 1967). Quando n duplica $\log(n)$ aumenta mas muito pouco; apenas duplica quando n aumenta para n^2. O que observou-se na prática foi o processamento total do áudio não ocupando mais de 25% do tempo de processamento de todo o algoritmo do projeto, portanto deixando viável essa solução de localização pelo áudio.

5.3 Localização do locutor utilizando áudio e vídeo

Seguindo o discutido na seção 4.3, foi preciso modelar as FDP das matrizes de transição entre estados e de ocorrência dos observáveis do HMM, de acordo com premissas que explicam o comportamento esperado do locutor, em um ambiente de videconferência.

Para o cálculo de qual estado escondido, ou setor do espaço, se encontra o locutor foi uti-

lizado o algoritmo de Viterbi (VITERBI, 1967), parametrizando-o para levar em consideração os dez últimos observáveis obtidos. Portanto o algoritmo precisa de $10 \times 0,0929$ segundo para conseguir calcular a primeira posição. Para o teste foram utilizados 51 estados, ou seja, o espaço de procura foi dividido em 51 setores.

O vetor S que contém os valores obtidos pelo SRP-PHAT é ponderado por um coeficiente w nos valores encontrados em regiões em que foram detectadas faces, forçando o algoritmo a selecionar uma dessas regiões para a posição do locutor ativo. Foi preciso tomar o cuidado para que mesmo se algum membro da videoconferência não estar com sua face detectada no momento em que estiver falando, devido a uma oclusão parcial ou total da face, mesmo assim ainda seja possível classificá-lo como locutor ativo utilizando apenas informação do sinal de áudio. O valor de w foi encontrado experimentalmente, onde $w = 0,25$ foi o peso que obteve os melhores resultados nos testes.

Para o teste de performance do modelo foram gerados cinco vídeos de um minuto cada. Não houve preocupação em modificar as características acústicas do ambiente, nem evitar o trânsito das demais pessoas, visando obter as condições mais próximas possível das reais de ruído e reverberação em uma videoconferência. Todos vídeos armazenam informações sobre o resultado do SRP-PHAT, resultado da detecção de faces e o resultado final do algoritmo, ou seja, a localização do locutor ativo. No teste foram utilizadas duas pessoas, as quais ficavam dialogando durante os 60 segundos de cada vídeo, a uma distância de um metro do arranjo de microfones. Foram utilizados seis microfones dispostos linearmente, espaçados entre si em 15 centímetros, com a câmera situada no centro do arranjo. Devido a proximidade dos locutores com o arranjo de microfones foi utilizado no algoritmo o conceito de Campo Próximo para a varredura do SRP-PHAT. Durante a gravação houve a preocupação de realizar atitudes comuns em videoconferência que podem fazer o algoritmo falhar, testando assim sua robusteza, por exemplo: ler um documento que oculte a face, virar para os lados, coçar o nariz, beber água, falar em tons variados e falar pausadamente.

Propositadamente, em três dos cinco vídeos gravados o ar-condicionado da sala permaneceu ligado, sendo a maior fonte de ruído presente no ambiente.

Após a gravação dos cinco vídeos, eles foram analisados para verificar os momentos de falha do algoritmo, comparando o resultado com outros algoritmos. Os métodos de localização do locutor analisados foram: SRP-PHAT, SRP-PHAT ponderado pela detecção de faces e o SRP-PHAT ponderado utilizando HMM. A Tabela 5.4 mostra o resultado da avaliação.

Notou-se que a localização baseada apenas no SRP-PHAT oscilava muito, causado principalmente por reverberações e pausas na fala. O problema se agravava com a presença do

ar-condicionado, quando praticamente dobrava o tempo no qual a localização do locutor se apresentou incorreta.

No método do SRP-PHAT ponderado pela detecção de faces, o problema de falhas causadas por reverberações no ambiente, em relação ao método anterior, praticamente desapareceu. Percebeu-se que o algoritmo oscilava em momentos onde nenhum dos locutores estava falando. Esse método também foi bastante sensível ao ruído do ar-condicionado, dobrando o tempo da localização incorreta. Em relação ao método que utilizou apenas o SRP-PHAT para a localização, ponderar pelas faces detectadas mostrou diminuir o erro de localização pela metade. Esse desempenho está resumido na Tabela 5.4.

O algoritmo proposto utiliza o SRP-PHAT ponderado pela detecção de faces em um modelo HMM e foi o terceiro algoritmo avaliado. Ele conseguiu resolver a fragilidade do algoritmo anterior de oscilar entre os locutores nos momentos de pausas do diálogo, pois o HMM evitou a transição abrupta entre os participantes quando o resultado do SRP-PHAT (O_2) não fossse alto o suficiente. Mesmo com a transição entre os locutores oscilando menos devido ao HMM, situações onde ocorreram transição repentina do locutor e por um breve instante foram corretamente reconhecidas pelo algoritmo, como o caso analisado de um participante responder apenas "sim"a uma pergunta do outro e depois continuar em silêncio ouvindo. Abaixo alguns fatos percebidos no teste.

- Em situações onde a face do locutor não era detectada mas apenas a voz, o algoritmo se mostrou robusto.
- A mudança de um locutor para outro foi ágil o suficiente para conseguir localizar uma rápida resposta "sim"dada pelo membro da videoconferência que estava apenas escutando até o momento.
- A transição de um locutor para outro demorou cerca de 0,1 segundo, que é o tempo para coleta de amostras do sinal dos microfones para um novo cálculo do SRP-PHAT.

A Figura 5.6 apresenta algumas importantes situações que ocorreram no teste nas quais o algoritmo proposto precisou resolver. Em cada imagem retirada dos vídeos realizados aparece no canto superior esquerdo a cena capturada pela câmera. No canto inferior esquerdo, dentro de um retângulo é desenhado um ponto que representa o resultado do SRP-PHAT, SRP-PHAT ponderado pela detecção de faces, detecção de faces e o SRP-PHAT ponderado com HMM respectivamente. Ao lado direito, o gráfico do SRP-PHAT e abaixo o SRP-PHAT ponderado pela detecção de faces. A Figura 5.6(a) foi retirada de um vídeo onde não havia a presença

Tabela 5.4: Desempenho dos algoritmos de localização do locutor. (*)Para apuração do percentual de localizações incorretas geradas pelos algoritmos foi considerada uma tolerância de três estados para a direita ou para a esquerda do estado do locutor ativo, assumindo-os também como corretos.

Vídeo 1: 60 segundos com ruído do ar-condicionado			
Métrica	SRP-PHAT	SRP-PHAT ponderado	SRP-PHAT pond. com HMM
Média do erro (μ)	3,71	3,69	1,56
Desvio padrão do erro (σ)	7,62	7,78	4,17
Localização incorreta(*)	14,29%	11,36%	2,24%
Vídeo 2: 60 segundos com ruído do ar-condicionado			
Métrica	SRP-PHAT	SRP-PHAT ponderado	SRP-PHAT pond. com HMM
Média do erro (μ)	3,82	3,85	1,45
Desvio padrão do erro (σ)	7,36	8,27	3,58
Localização incorreta(*)	17,09%	11,34%	1,86%
Vídeo 3: 60 segundos sem ruído do ar-condicionado			
Métrica	SRP-PHAT	SRP-PHAT ponderado	SRP-PHAT pond. com HMM
Média do erro (μ)	2,25	2,30	1,14
Desvio padrão do erro (σ)	5,10	5,99	2,52
Localização incorreta(*)	8,83%	5,41%	0,90%
Vídeo 4: 60 segundos sem ruído do ar-condicionado			
Métrica	SRP-PHAT	SRP-PHAT ponderado	SRP-PHAT pond. com HMM
Média do erro (μ)	2,67	1,77	1,41
Desvio padrão do erro (σ)	6,73	4,72	3,77
Localização incorreta(*)	7,43%	3,45%	1,90%
Vídeo 5: 60 segundos com ruído do ar-condicionado			
Métrica	SRP-PHAT	SRP-PHAT ponderado	SRP-PHAT pond. com HMM
Média do erro (μ)	4,24	2,91	1,84
Desvio padrão do erro (σ)	8,04	6,49	4,48
Localização incorreta(*)	18,94%	8,67%	3,72%

do ruído do ar-condicionado. A ausência do ruído é bastante evidente quando se compara com a Figura 5.6(b), que foi retirada de um vídeo onde o ar-condicionado estava ligado. Comparando os gráficos do SRP-PHAT entre as Figuras 5.6(a) e 5.6(b) percebe-se que na 5.6(a) o gráfico encontra-se muito menos perturbado e seu ponto máximo bem definido. Já na Figura 5.6(b) o gráfico está consideravelmente mais perturbado e seu ponto máximo já não tão bem definido. Na Figura 5.6, apenas as cenas (a) e (d) foram retiradas de vídeos sem a presença do ar-condicionado enquanto as demais, com a presença.

A Figura 5.6(c) retrata o momento no qual o locutor dois está coçando o nariz e sua face

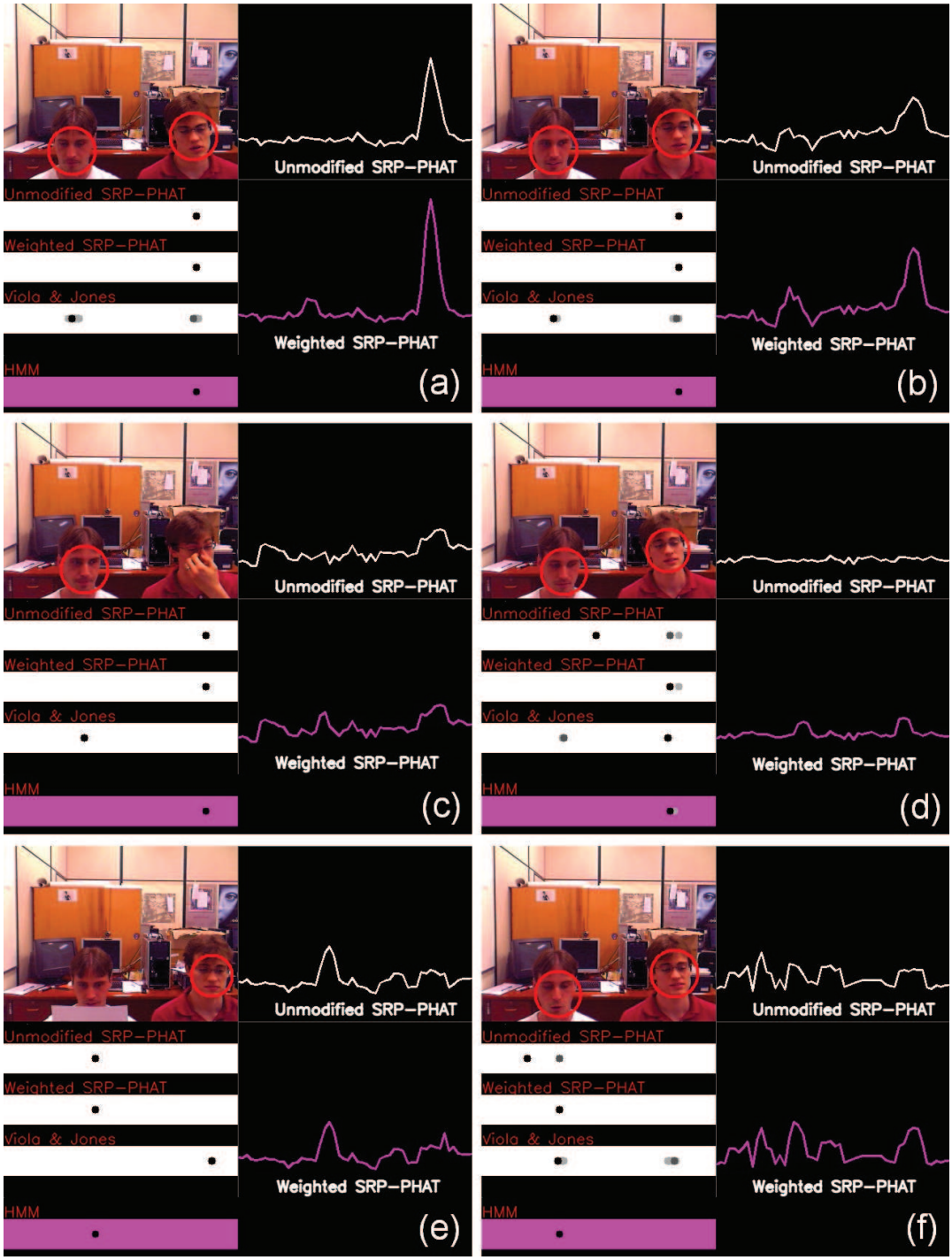

Figura 5.6: Exemplos de situações ocorridas nos vídeos gravados para o teste de performance do algoritmo. Em cada figura aparece no canto superior esquerdo a imagem capturada pela câmera. No canto inferior esquerdo o resultado do SRP-PHAT, SRP-PHAT ponderado pela detecção de faces, detecção de faces e o SRP-PHAT ponderado com HMM respectivamente. Ao lado direito, o gráfico do SRP-PHAT e abaixo o SRP-PHAT ponderado pela detecção de faces. As figuras representam: (a) diálogo normal sem presença do ar-condicionado; (b) diálogo normal com a presença do ar-condicionado; (c) face não detectada por estar coçando o nariz; (d) pausa no diálogo; (e) leitura durante a videoconferência; (f) erro do SRP-PHAT devido a reverberação.

deixa de ser detectada pelo algoritmo, entretanto o valor do SRP-PHAT é suficiente para manter a localização correta. Na Figura 5.6(d) houve uma pausa no diálogo e o algoritmo SRP-PHAT evidentemente não consegue localizar o locutor; devido a um O_2 baixo, o HMM mantém consistente a localização não oscilando para outros estados. Na Figura 5.6(e) o locutor dois está lendo um documento e sua face não é detectada pelo algoritmo, mas mesmo assim a localização final continua correta. A Figura 5.6(f) apresenta o momento mais crítico da vídeoconferência, com a presença de uma forte reverberação provocada por um obstáculo próximo; no gráfico do SRP-PHAT é possível notar um pico mais a esquerda de onde está o locutor ativo (locutor dois), fornecendo uma localização incorreta; devido a ponderação pelas faces detectadas, o resultado do SRP-PHAT da Figura 5.6(f) é reforçado nessas regiões e a localização se torna correta.

6 *Discussão e Trabalhos Futuros*

A solução encontrada para a localização do locutor em uma videconferência conseguiu alcançar satisfatoriamente os objetivos propostos. O nível de ruído e reverberação do ambiente é determinante para o sucesso ou fracasso de técnicas que utilizem apenas o som como fonte de informação. Esses algoritmos se comportam bem em ambientes onde não há outras fontes sonoras competindo com o locutor, nem obstáculos que reflitam a onda sonora causando reverberação. Ambientes assim, chamados de ideais, são pouco comuns de serem encontrados na prática.

A utilização de uma outra fonte de informação para a localização do locutor, além do som, se faz necessária, quando é preciso uma maior robustez no resultado. No algoritmo proposto foi utilizada a informação de vídeo, com o objetivo de fornecer mais subsídios para o algoritmo aumentar a precisão da localização. Do vídeo foi extraído a localização das faces presentes no ambiente, o que contribuiu para corrigir algumas deficiências na localização por áudio, observadas na presença de ruído e reverberação excessivas.

O algoritmo de Viola e Jones (2001), utilizado para a detecção de faces em uma sequência de vídeo, consumia muito tempo de processamento, o que impossibilitava sua utilização em conjunto com o processamento de áudio por computador convencional de uso doméstico. O algoritmo percorria toda a imagem com uma janela deslizante em busca de feições que determinam uma face e, dependendo da resolução do vídeo, consumia todo o tempo de processamento do computador. A solução encontrada para conseguir integrar o processamento de áudio com o vídeo em tempo real foi limitar o espaço de busca das faces apenas nas regiões do vídeo constituídas por determinada quantidade de pixels com cor de pele. Essa solução fez o algorimto de detecção de faces funcionar consumindo um terço do tempo do processamento original. Mesmo utilizando seis microfones, o tempo de processamento para o sinal de áudio foi menos expressivo que o de vídeo e não causou risco para realização do projeto. O processamento dos HMMs utilizados foi o que menos consumiu tempo de processamento, não causando impacto significativo no tempo total de uso do processador.

Alguns testes preliminares no início do trabalho mostraram que o SRP-PHAT é menos eficiente para localizar mais de um locutor ativo simultaneamente. Essa deficiência fez restringir o escopo desse primeiro trabalho apenas para a localização de um locutor ativo, ficando como proposta para uma futura pesquisa encontrar uma solução para a localização espacial de mais de um locutor falando simultaneamente. Outra limitação do método proposto é a detecção de pessoas no ambiente através das faces, obrigando-as a estarem falando de frente para a câmera. O método seria mais robusto se essa detecção não dependesse exclusivamente da face, ficando também como proposta para um próximo trabalho.

Referências Bibliográficas

ALATAN, A. A.; AKANSU, A. N.; WOLF, W. Multi-modal dialog scene detection using hidden markov models for content-based multimedia indexing. *Multimedia Tools Appl.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 14, n. 2, p. 137–151, 2001. ISSN 1380-7501.

BAUM, L. E.; EAGON, J. A. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, v. 73, p. 360–363, 1967. ISSN 0002-9904.

BAUM, L. E. et al. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 41, n. 1, p. 164–171, 1970. ISSN 00034851. Disponível em: <http://www.jstor.org/stable/2239727>.

BAUM, L. E.; SELL, G. R. Growth transformations for functions on manifolds. *Pacific J. Math.*, v. 27, p. 211–227, 1968. ISSN 0030-8730.

BENYASSINE, A. et al. Itu-t recommendation g.729 annex b: a silence compression scheme for use with g.729 optimized for v.70 digital simultaneous voice and data applications. *Communications Magazine, IEEE*, v. 35, n. 9, p. 64–73, Sep 1997. ISSN 0163-6804.

BERMAN, A. P.; SHAPIRO, L. G. A flexible image database system for content-based retrieval. *Comput. Vis. Image Underst.*, Elsevier Science Inc., New York, NY, USA, v. 75, n. 1-2, p. 175–195, 1999. ISSN 1077-3142.

BOURLARD, H.; DUPONT, S. A new ASR approach based on independent processing and recombination of partial frequency bands. In: *Proc. ICSLP '96*. Philadelphia, PA: [s.n.], 1996. v. 1, p. 426–429. Disponível em: <citeseer.nj.nec.com/bourlard96new.html>.

BRANDSTEIN, M.; WARD, D. *Microphone Arrays: Signal Processing Techniques and Applications*. 1. ed. [S.l.]: Springer, 2001. Hardcover. ISBN 3540419535.

COCHRAN, W. et al. What is the fast fourier transform? *Proceedings of the IEEE*, v. 55, n. 10, p. 1664–1674, Oct. 1967. ISSN 0018-9219.

COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, v. 19, n. 90, p. 297–301, 1965. Disponível em: <http://dx.doi.org/10.2307/2003354>.

DIBIASE, J. *A High Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments*. Tese (Doutorado) — Brown University, 2000.

DO, H. T. H. *Real-time SRP-PHAT source location implementations on a large-aperture microphone array*. Dissertação (Mestrado) — Brown University, 2009.

DUPONT, S.; LUETTIN, J. Audio-visual speech modeling for continuous speech recognition. *Multimedia, IEEE Transactions on*, v. 2, n. 3, p. 141–151, Sep 2000. ISSN 1520-9210.

FERMAN, A. M.; TEKALP, A. M. Probabilistic analysis and extraction of video content. In: *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*. [S.l.: s.n.], 1999. v. 2, p. 91–95 vol.2.

FORNEY G.D., J. The viterbi algorithm. *Proceedings of the IEEE*, v. 61, n. 3, p. 268–278, March 1973. ISSN 0018-9219.

FREEMAN, D. et al. The voice activity detector for the pan-european digital cellular mobile telephone service. In: . [S.l.: s.n.], 1989. p. 369–372 vol.1. ISSN 1520-6149.

FREUND, Y.; SCHAPIRE, R. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, p. 119–139(21), 1997.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.]: Addison-Wesley Publishing Company, 1992.

HENNECKE, M.; STORK, D.; PRASAD, K. V. Visionary speech: looking ahead to practical speech reading systems. In: *Speechreading by Humans and Machines: Models, Systems, and Applications*. New York: NATO/Springer-Verlag: Springer, 1996. p. 331–350.

HSU, R.; MOTTALEB, M. A.; JAIN, A. Face detection in color images. v. 24, n. 5, p. 696–706, May 2002.

JAYARAM, S. et al. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. [S.l.: s.n.], 2004. v. 2, p. II–813–II–818 Vol.2. ISSN 1063-6919.

JUNG, K. K. et al. Object tracking for security monitoring system using microphone array. In: . [S.l.: s.n.], 2007. p. 2351–2354.

KAKUMANU, P.; MAKROGIANNIS, S.; BOURBAKIS, N. A survey of skin-color modeling and detection methods. *Pattern Recognition*, v. 40, n. 3, p. 1106 – 1122, 2007. ISSN 0031-3203.

KNAPP, C.; CARTER, G. The generalized correlation method for estimation of time delay. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 24, n. 4, p. 320–327, Aug 1976. ISSN 0096-3518.

LEARNED-MILLER, G. B. H. M. R. B. E. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.* Amherst, October 2007.

LIENHART, R.; PFEIFFER, S.; EFFELSBERG, W. Video abstracting. *Communications of the ACM*, v. 40, p. 55–62, 1997.

LO, D. et al. Robust joint audio-video localization in video conferencing using reliability information. In: *IEEE Trans. on Instrumentation and Measurement*. [S.l.: s.n.], 2004. p. 1132–1139.

LOU, H.-L. Implementing the viterbi algorithm. *Signal Processing Magazine, IEEE*, v. 12, n. 5, p. 42–52, Sep 1995. ISSN 1053-5888.

MARAGOS, P.; POTAMIANOS, A.; GROS, P. *Multimodal Processing and Interaction: Audio, Video, Text*. [S.l.]: Springer Publishing Company, Incorporated, 2008. ISBN 0387763155, 9780387763156.

MARSZALEC, E. et al. Physics-based face database for color research. *Journal of Electronic Imaging*, SPIE, v. 9, n. 1, p. 32–38, 2000. Disponível em: <http://link.aip.org/link/?JEI/9/32/1>.

MARTINEZ, A.; BENAVENTE, R. The ar face database. *CVC Technical Report #24*, June 1998.

MUKHERJEE, K.; GWEE, B.-H. A 32-point fft based noise reduction algorithm for single channel speech signals. In: . [S.l.: s.n.], 2007. p. 3928–3931.

OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

PAN, H. et al. A hybrid nn-bayesian architecture for information fusion. In: *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*. [S.l.: s.n.], 1998. v. 1, p. 368–371 vol.1.

RABINER, L. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, v. 77, n. 2, p. 257–286, Feb 1989. ISSN 0018-9219.

SABER, E.; TEKALP, A. M. Frontal-view face detection and facial feature extraction using color, shape, and symmetry based cost functions. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 19, n. 8, p. 669–680, 1998. ISSN 0167-8655.

SALAMI, R. et al. Description of itu-t recommendation g.729 annex a: reduced complexity 8 kbit/s cs-acelp codec. In: . [S.l.: s.n.], 1997. v. 2, p. 775–778 vol.2.

SARACENO, C.; LEONARDI, R. Identification of story units in audio-visual sequences by joint audio and video processing. In: *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*. [S.l.: s.n.], 1998. v. 1, p. 363–367 vol.1.

SCHMUGGE, S. J. et al. Objective evaluation of approaches of skin detection using roc analysis. *Comput. Vis. Image Underst.*, Elsevier Science Inc., New York, NY, USA, v. 108, n. 1-2, p. 41–51, 2007. ISSN 1077-3142.

SCHRAMM, R. *Detecção de Faces e Rastreamento da Pose da Cabeça*. Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos, 2009.

SHIN, M.; CHANG, K.; TSAP, L. Does colorspace transformation make any difference on skin detection? In: *Applications of Computer Vision, 2002. (WACV 2002). Proceedings. Sixth IEEE Workshop on*. [S.l.: s.n.], 2002. p. 275–279.

SOBOTTKA, K.; PITAS, I. A novel method for automatic face segmentation, facial feature extraction and tracking. v. 12, n. 3, p. 263–281, June 1998.

SOHN, J.; SUNG, W. A voice activity detector employing soft decision based noise spectrum adaptation. In: . [S.l.: s.n.], 1998. v. 1, p. 365–368 vol.1. ISSN 1520-6149.

VEEN, B. V.; BUCKLEY, K. Beamforming: a versatile approach to spatial filtering. *ASSP Magazine, IEEE*, v. 5, n. 2, p. 4–24, April 1988. ISSN 0740-7467.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. *cvpr*, IEEE Computer Society, Los Alamitos, CA, USA, v. 01, p. 511, 2001. ISSN 1063-6919.

VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, v. 13, n. 2, p. 260–269, Apr 1967. ISSN 0018-9448.

WANG, C.; BRANDSTEIN, M. Multi-source face tracking with audio and visual data. In: . [S.l.: s.n.], 1999. p. 169–174.

WANG, C.; BRANDSTEIN, M. S. A hybrid real-time face tracking system. In: *Proc. of ICASSP98*. [S.l.: s.n.], 1997. p. 3737–3740.

WIMMER, M.; RADIG, B. Adaptive skin color classificator. *ICGST International Journal on Graphics, Vision and Image Processing*, Special Issue on Biometrics, 2006.

WOLF, W. Hidden markov model parsing of video programs. In: *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*. [S.l.: s.n.], 1997. v. 4, p. 2609–2611 vol.4.

WU, Y.; AI, X. Face detection in color images using adaboost algorithm based on skin color information. *International Workshop on Knowledge Discovery and Data Mining*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 339–342, 2008.

ZOTKIN, D. et al. An audio-video front-end for multimedia applications. In: . [S.l.: s.n.], 2000. v. 2, p. 786–791 vol.2. ISSN 1062-922X.