Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Identificação e Localização de pessoas em Smartspaces

Danilo Ávila Monte Christo Ferreira
Tales Mundim Andrade Porto

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Carla Denise Castanho

Brasília
2011

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Lamar

Banca examinadora composta por:

Prof. Dr. Carla Denise Castanho (Orientador) — CIC/UnB
Prof. Dr. Professor I — CIC/UnB
Prof. Dr. Professor II — CIC/UnB

CIP — Catalogação Internacional na Publicação

Ferreira, Danilo Ávila Monte Christo.

Identificação e Localização de pessoas em Smartspaces / Danilo Ávila Monte Christo Ferreira, Tales Mundim Andrade Porto. Brasília : UnB, 2011.
121 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2011.

1. palvrachave1, 2. palvrachave2, 3. palvrachave3

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Identificação e Localização de pessoas em Smartspaces

Danilo Ávila Monte Christo Ferreira
Tales Mundim Andrade Porto

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Carla Denise Castanho (Orientador)
CIC/UnB

Prof. Dr. Professor I
CIC/UnB

Prof. Dr. Professor II
CIC/UnB

Prof. Lamar
Coordenador do Bacharelado em Ciência da Computação

Brasília, 2 de maio de 2011

Dedicatória

Dedico a....

Agradecimentos

Agradeço a....

Resumo

A ciência...

Palavras-chave: palvrachave1, palvrachave2, palvrachave3

Abstract

The science...

Keywords: keyword1, keyword2, keyword3

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A computação ubíqua a tempos vem sendo tema de diversas pesquisas ao redor do mundo. Mark Weiser diz que o computador do futuro deve ser algo invisível (34, 35) proporcionando ao usuário um melhor foco na tarefa e não na ferramenta. A computação ubíqua tenta atribuir essa invisibilidade aos computadores buscando cada vez mais a diminuição do tamanho, a especificidade da tarefa e se acoplando aos objetos do dia-a-dia.

Um ambiente onde a computação ubíqua acontece em sua totalidade é chamado de *SmartSpace* (2). Esse ambiente provê ao usuário uma melhor forma de interagir com os computadores usando diversas tecnologias que estimulam a interatividade natural. Tais tecnologias são capazes de fornecer inteligência ao *SmartSpace*, necessária para concretizar a visão da ubicomp (4).

Para conseguir uma boa interação entre as diversas peças que compõem o *SmartSpace* é necessário que se tenha a disposição informações de contexto, como quem está no ambiente, onde está, o que está fazendo e outras que ajudam o sistema a definir o melhor ajuste dos equipamentos. Com uma base rica de informações de contexto, contendo os perfis dos usuários, garantimos uma maior acurácia na tomada de decisões. Informações de contexto como essas são complicadas de se obter devido a alta dinamicidade do ambiente, no qual usuários entram e saem a todo momento e interagem com diversos equipamentos.

A identificação de um usuário em um *SmartSpace* é feita por meio de sistema de reconhecimento automático. Há alguns anos, um grande número de pesquisas vem sendo desenvolvidas para criação deste tipo de sistema (3). Um dos motivos clássicos é que os métodos baseados em cartões de identificação e senhas não são altamente confiáveis. Estes podem ser perdidos, extraviados e até fraudados (28).

Um ambiente ubíquo capaz de reconhecer seus usuários, pode prover uma personalização automática do ambiente de acordo com as preferências do usuário e até mesmo prover um ambiente mais seguro com controle de acesso físico e prevenção de fraudes (3). Atualmente, os métodos de reconhecimento mais utilizados são baseados no uso de cartões magnéticos e senhas, que requerem sua utilização durante uma transação, mas que não verificam sua idoneidade (7).

Hoje em dia, várias técnicas de reconhecimento por meio de faces, íris, voz, entre outras, vêm sendo estudadas e utilizadas em sistemas de reconhecimento automático (28). O reconhecimento facial pode ser considerada como uma das principais funções do ser humano pois permite identificar uma grande quantidade de faces e aspectos psicológicos

demonstrados pela fisionomia. Pode ser considerada, também, como um problema clássico da computação visual pela complexidade existente na detecção e reconhecimento de características e padrões (3).

O reconhecimento facial vem se desenvolvendo junto a "quarta geração" de computadores através de sua aplicação na nova geração de interfaces que consiste na detecção e reconhecimento de pessoas (3).

É proposta então uma solução para o problema de localização e identificação de perfis de usuários em um *SmartSpace* utilizando como base o *middleware UbiquitOS* (10) integrado com o *Kinect*.

1.1 Organização do trabalho

Explicar a estrutura da monografia.

Capítulo 2

Rastreamento e Localização

Neste capítulo será mostrada uma abordagem conceitual sobre localização e rastreamento de objetos em um ambiente.

Sobre localização de objetos será mostrada uma visão geral de como obter a posicão do objeto relativa a uma camêra utilizando imagens de pronfundidade.

Sobre rastreamento de objetos será mostrada uma visão geral do processo e quais as dificuldades mais comuns encontradas. Mostraremos as técnicas mais comuns de detecção e rastreamento de objetos, as maneiras de representar os objetos rastreados e algumas características do objeto utilizadas.

2.1 Rastreamento

Rastreamento de objetos é uma importante tarefa do campo da computação visual. A proliferação de computadores com um alto poder computacional, a disponibilidade de câmeras de alta qualidade e preço acessível e a crescente necessidade de sistemas automáticos de análise de vídeos têm gerado um grande interesse em algoritmos de rastreamento de objetos (36).

Basicamente, rastreamento pode ser definido como o problema de estimar a trajetória de um objeto em um plano de imagem a medida em que se move na cena. Em outras palavras, um rastreador atribui *labels* para os objetos monitorados em diferentes quadros de um vídeo (36).

A detecção e o rastreamento de pessoas tem um grande potencial em aplicações em domínios tão diversos como animação, interação humano-computador, vigilância automatizada (monitorar uma cena para detectar atividades suspeitas), entre outros. Por esta razão, tem havido um crescimento notável na investigação deste problema.

O rastreamento de pessoas em um ambiente é considerada como uma tarefa complexa devido a:

1. complexidade do corpo humano;
2. alta dinamicidade do ambiente;
3. ruído nas imagens (36);
4. complexidade do movimento das pessoas;

5. oclusões parciais ou totais de pessoas;
6. variação na iluminação do ambiente (36);
7. processamento em tempo-real (36);

Algumas dessas dificuldades podem ser vencidas com a utilização de imagens de profundidade ao invés de imagens de cor ou intensidade. As imagens de profundidade, além de serem muito pouco sensíveis as variações de iluminação, provê um fácil entendimento da estrutura da cena, que pode ser utilizada para simplificar as tarefas de rastreamento. Além disso, as câmeras que provêm imagens de profundidade estão comercialmente disponíveis a um preço acessível (11).

Várias abordagens para rastreamento de objetos já foram propostas. Basicamente, elas se diferem na forma que tratam as seguintes perguntas (36):

- Qual representação do objeto é adequada para o rastreamento?
- Quais características na imagem devem ser utilizadas?
- Como o movimento, aparência e a forma do objeto deve ser modelada?

As respostas para estas perguntas dependem do contexto/ambiente onde o rastreamento será utilizado e do uso final das informações de rastreamento (36).

O rastreamento de pessoas geralmente inicia com o processo de segmentação da imagem da pessoa do resto da imagem. Depois, essas imagens segmentadas são transformadas em outras representações para reduzir a quantidade de informação ou para atender a um determinado algoritmo. Com isso, deve-se definir como as pessoas vão ser rastreadas *frame* a *frame* (24).

Basicamente, o processo de rastreamento pode ser dividido em duas etapas:

1. Detecção do objeto;
2. Rastreamento do objeto detectado;

Antes de falarmos mais sobre cada uma dessas etapas e os métodos existentes para cada, vamos falar sobre as maneiras existentes de representar os objetos rastreados e sobre as características nas imagens que podem ser utilizadas.

2.1.1 Representação do Objeto

Nos sistemas de rastreamento, os objetos rastreados devem ser representados de alguma maneira. Geralmente, as representações são baseados em suas formas. Existe uma forte relação entre a representação do objeto e o algoritmo de rastreamento escolhido (36). A representação é escolhida baseada no domínio da aplicação e as mais utilizadas são:

- **Pontos:** o objeto é representado por um ponto, como por exemplo a centroide (5) da Figura 2.1(a), ou por vários pontos (30), como por exemplo na Figura 2.1(b). Essa representação é mais adequada para rastreamento de objetos que ocupam uma pequena região na imagem (36);

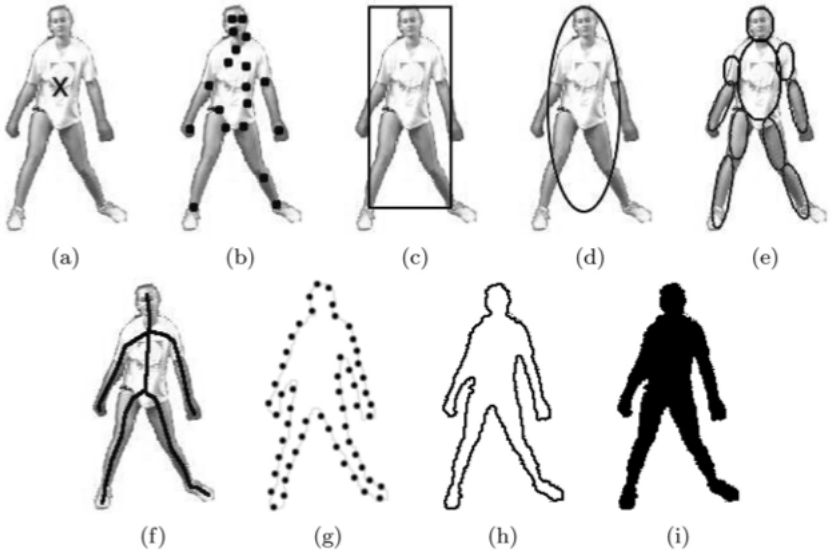


Figura 2.1: Representações de objetos rastreados. (a) Centroide, (b) múltiplos pontos, (c) representação retangular, (d) representação elíptica, (e) representação de múltiplas partes, (f) esqueleto do objeto, (g) contorno do objeto por pontos,(h) contorno completo do objeto, (i) silhueta do objeto (36)

- **Formas geométricas primitivas:** o objeto é representado por formas geométricas simples como um retângulo ou uma elipse, como mostrados nas Figuras 2.1(c) e (d) (6). Essa representação é mais adequada para simples objetos rígidos (36);
- **Silhueta e Contorno:** representação por contorno define os limites de um objeto, como mostrado nas Figuras 2.1(g) e (h). A região interna do contorno é chamada de Silhueta, como mostrado na Figura 2.1(i). Essa representação é mais adequada para rastrear objetos complexos de forma não rígida (36, 37). Ela é popular devido a sua simplicidade. A silhueta ou contorno de um objeto pode ser obtida definindo métodos de limiarização ou subtração, podendo ser utilizada tanto com imagens 2D quanto 3D. A representação 2D geralmente é mais simples (24);
- **Modelos de formas articuladas:** objetos articulados são compostos por partes do corpo que se ligam por meio de juntas. Para representar objetos articulados, utiliza-se figuras geométricas para cada parte do corpo, como mostrado na Figura 2.1(e) (36);
- **Modelos de esqueletos:** modelos de esqueletos são extraídos do objeto rastreado, como mostrado na Figura 2.1(f). Essa representação pode ser utilizada tanto para objetos articulados rígidos quanto não rígidos (36);

Para rastreamento de pessoas a representação por meio de contorno ou silhuetas são as mais adequadas (36).

O rastreamento de várias pessoas de maneira simultânea é considerada uma tarefa muito complexa. As representações das pessoas rastreadas podem se dividir ou fundir em

novas representações devido a possíveis oclusões ou ruídos na imagem, e a aparência do objeto pode variar devido a sombras e mudanças da iluminação (24).

2.1.2 Características para rastreamento

A seleção das características é uma tarefa crítica para o rastreamento e está fortemente relacionada com a representação do objeto. Em geral, a seleção procura as características mais singulares para que o objeto rastreado seja facilmente distinguido (36). As características mais comuns utilizadas atualmente são:

- **Cor:** a cor do objeto é influenciada principalmente por duas características: a distribuição da iluminação e a propriedade de reflectância do objeto. Geralmente, a representação *RGB* é utilizada para representar a cor (36);
- **Borda:** os limites de um objeto geram uma grande variação na intensidade na imagem e são menos sensíveis a variações na iluminação comparado com as cores. A detecção por meio das bordas é utilizada para identificar essas variações de intensidade na imagem. Os algoritmos que detectam as bordas do objeto geralmente as utilizam para representação dos mesmos (36);
- **Fluxo óptico:** é um campo denso de vetores de deslocamento que define a tradução de cada *pixel* em uma região. Ele é calculado a partir da restrição de luminosidade, que pressupõe a constância de brilho de *pixels* correspondentes nas *frames* consecutivas (14, 36);
- **Textura:** é a medida da intensidade da variação da superfície que quantifica propriedades como suavidade e regularidade. A Textura é menos sensível a variação da iluminação comparado com a cor (36);

De todas as características, a mais utilizada para rastreamento é a cor (36).

2.1.3 Detecção de objeto

Todo método de rastreamento requer um mecanismo de detecção de objetos que pode ser realizada a cada *frame* obtida ou na primeira vez que o objeto aparece no vídeo. Os método mais populares são:

- **Detector de pontos:** esses detectores são usados para encontrar pontos de interesses dentro da imagem que tem uma expressiva textura na sua respectiva localização. Pontos de interesse são amplamente usados no contexto do movimento e no rastreamento. A qualidade desejável para o ponto de interesse é que seja invariante diante das mudanças de iluminação e ângulo de visão da câmera (36).
- **Subtração de fundo:** é um método popular para segmentação de movimento, especialmente nas situações em que o plano de fundo é relativamente estático. Ele detecta as regiões de movimento na imagem obtendo a diferença *pixel* a *pixel* entre a *frame* corrente e a *frame* referente ao plano de fundo (15). Geralmente, um algoritmo de componentes conectadas é aplicado para obter regiões conectadas que correspondem a um objeto (36).

- **Segmentação:** o objetivo do algoritmo de segmentação é particionar a imagem em regiões com certo grau de similaridade. Todo algoritmo de segmentação tem dois problemas: o critério para definir uma boa partição e o método para arquivar particionamentos eficientes (31, 36).
- **Aprendizagem supervisionada:** a detecção de objetos pode ser feita pelo aprendizado automático de diferentes objetos de um conjunto de exemplos por meio de um mecanismo de aprendizado. Esse aprendizado requer o armazenamento de um um conjunto de *templates*. A partir desse conjunto de informações, o algoritmo gera uma função que mapeia as possíveis entradas para as saídas desejadas. Um problema padrão é a classificação onde a função gera um valor contínuo a partir de um determinado comportamento do objeto. No contexto da detecção de objetos as informações armazenadas são compostas por um par de características de objetos e uma classe associada onde ambos os valores são manualmente definidos. Seleção de características tem um papel importante no desempenho da classificação, portanto, é importante usar um conjunto de características que seja possível discriminar uma classe das outras (36).

2.1.4 Rastreamento de objeto

O objetivo do rastreamento de objetos é conhecer a trajetória do mesmo no tempo localizando sua posição em cada *frame*. O rastreamento de objetos também pode prover a região completa na imagem ocupada pelo objeto a cada instante (36).

As atividades de detecção de objetos e de estabelecimento de correspondências entre os objetos e as instâncias dos *frames* podem ser realizadas tanto separadamente como concomitantemente. No primeiro caso, as prováveis regiões de objetos são obtidas e o rastreador encontra correspondência entre os objetos e os *frames*. No último caso, as prováveis regiões e as correspondências são feitas juntas e estimadas pela atualização iterativa da localização do objeto e de regiões de informações obtidos dos *frames* anteriores (36).

Os métodos de rastreamento de objetos mais utilizados atualmente são:

- **Rastreamento de pontos:** os objetos detectados a cada frame são representados por pontos e a associação entre os pontos é feita com base no estados anterior do objeto que pode incluir a posição e o movimento. Essa abordagem requer um mecanismo externo para detectar o objeto a cada *frame* (36);
- **Rastreamento de *kernel***: os objetos são rastreados pelo cálculo do movimento do *kernel* em *frames* consecutivas. Esse movimento geralmente esta na forma de transformações paramétricas como translação e rotação. *Kernel* se refere ao formato ou aparência do objeto (36).
- **Rastreamento de silhuetas:** o rastreamento é feito estimando a região do objeto a cada *frame*. Esse método utiliza informação contida dentro da região do objeto. Esta informação pode ser na forma de densidade de aparência e modelos de forma que estão, geralmente, na forma de mapas de borda. Dado os modelos de objeto, silhuetas são rastreadas por qualquer forma de correspondência ou evolução de contorno. Ambos os métodos podem ser essencialmente considerados como segmentação de objetos aplicada no domínio temporal utilizando os priores gerados a partir dos *frames* anteriores (36).

Com os objetos rastreados, para obtermos a posição dos mesmos devemos obter informações de profunidade. Na próxima seção falaremos sobre como essas informações são obtidas utilizando imagens de profundidade.

2.2 Localização

Calcular a distância de vários pontos na cena relativa a posição da camêra é uma importante tarefa de sistemas de computação visual (16). Para isso, deve-se obter informações de profundidade dos objetos em interesse. Essas informações podem ser obitidas utilizando imagens de itensidade ou de profundidade.

Uma maneira comum de se obter informações de profundidade de imagens de itensidade é adquirir um par de imagens usando duas câmeras deslocadas entre si por uma distância conhecida. Como alternativa, duas ou mais imagens obtidas de uma camêra em movimento também pode ser utilizadas para calcular informações de profundidade (16).

Informações 3D também podem ser obitidas indiretamente através de imagens de intensidade utilizando sinais na imagem, como sombreamento e textura (16).

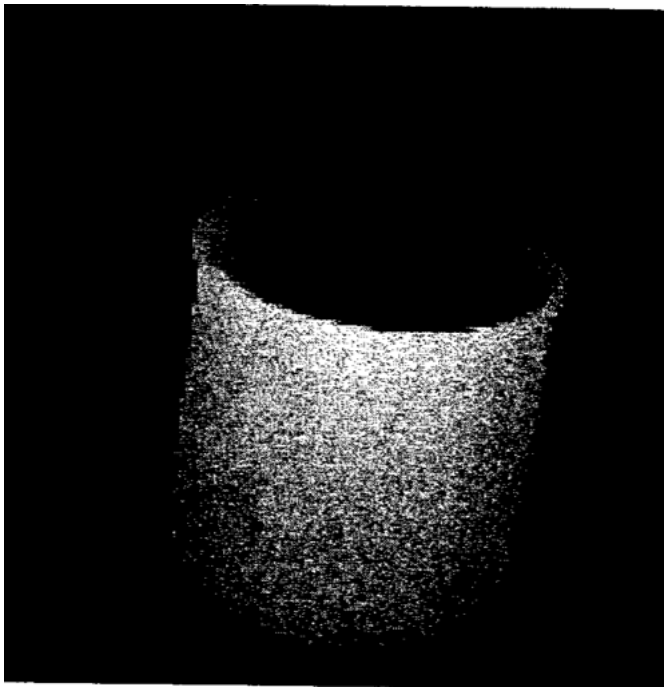


Figura 2.2: Imagem de profundidade de uma caneca de café (16).

Em contraste com imagens de itensidade, imagens cujo valor em cada pixel é uma função da distância do ponto correspondente na cena do sensor são chamadas de imagens de pronfunidade, exemplificada na Figura 2.2. Tais imagens pode ser adiquiras diretamente utilizando sensores específicos. Os dois métodos mais comuns para obter imagens de profundidade são (16):

1. **Radar:** a distância até o objeto é calculado observando a diferença de tempo entre o pulso eletromagnético transmitido e recebido. A informação de profundidade

também pode ser obtida através da dectecção da diferença de fase entre as ondas trasnsmitidas e as recebidas de um feixe de amplitude modulada (16);

2. **Triangulação:** utiliza as propriedades geométricas do triângulo para calcular a localização de objetos. Pode ser dividido em duas subcategorias: *lateration* e angulação. *Lateration* computa a posição de um objeto estimando sua distância de múltiplos ponto de referência. Calcular a posição de um objeto em duas dimensões requer estimativas de distância de três pontos não colineares como mostrado na Figura 2.3. Já em três dimensões são necessários quatro pontos não coplanares. Ângulação utiliza ângulos para determinar a distância do objeto. Em geral, ângulação em duas dimensões requer estimativas de dois ângulos e a estimativa da distância entre dois pontos de referência como mostrado na Figura 2.4 (13);

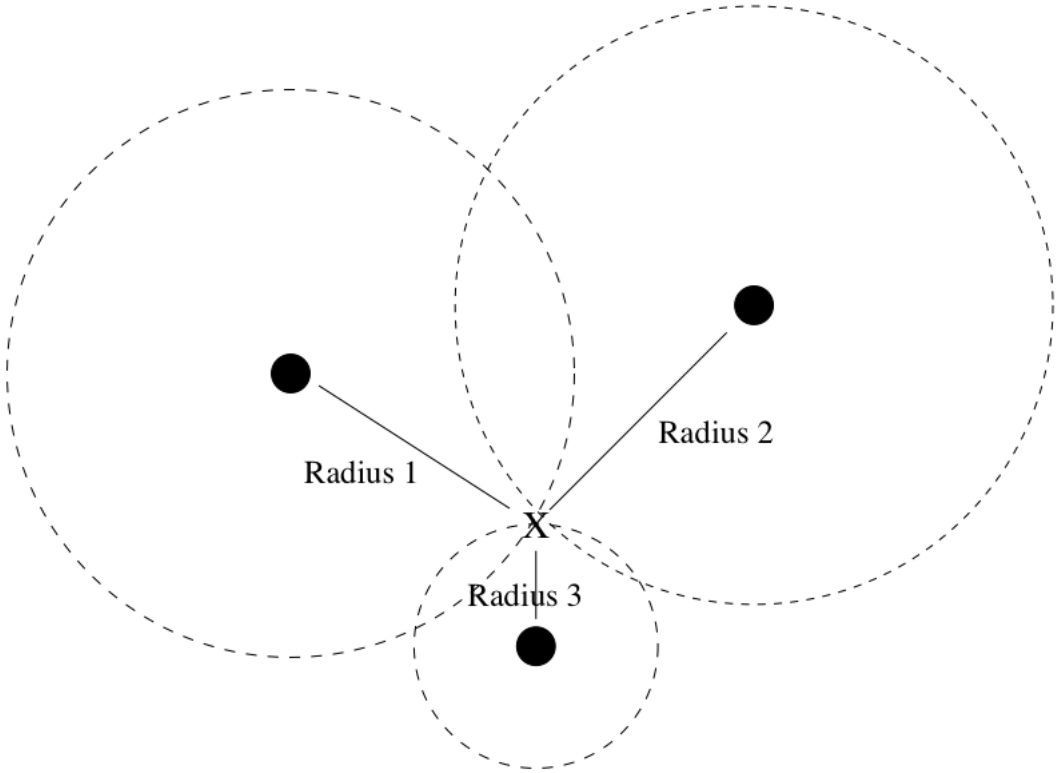


Figura 2.3: Determina a distância em duas dimensões utilizando *lateration*. Requer a distância entre o objeto X e três pontos de referência não colineares (13).

Imagens de profundidade são úteis devido a sua especificação explícita de valores de profundidade. Ao mesmo tempo acreditava-se que se a informação de profundidade fosse disponobilizada de maneira explícita, o processamento posterior da imagem seria facilitado. Tournou-se claro que a informação de profundiade ajuda, porém a tarefa básica de interpretação de imagens mantém todas as suas dificuldades (16).

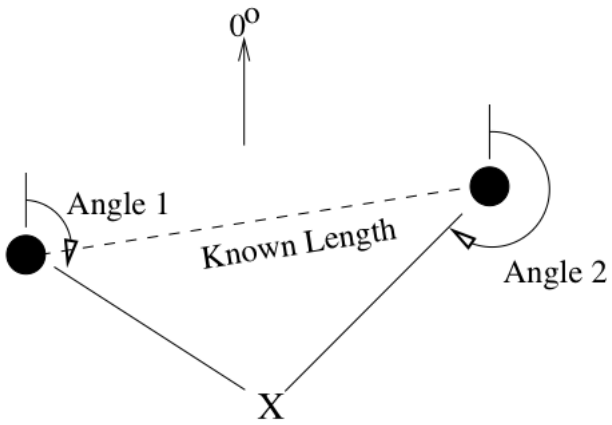


Figura 2.4: Exemplo de uma angulação em duas dimensões em que se localiza o objeto X utilizando ângulos relativos a um vetor de referência 0^{o} e a distância entre dois pontos de referência (13).

Capítulo 3

Reconhecimento Facial

Neste capítulo será apresentada uma abordagem conceitual sobre reconhecimento facial, uma vez que esse tópico consiste no alicerce de nosso trabalho.

Será apresentado também os conceitos gerais sobre biometria e sobre as características biométricas existentes.

Sobre reconhecimento facial, será apresentado os conceitos gerais e conceitos mais específicos das diferentes etapas do processo de reconhecimento: detecção de faces e reconhecimento das mesmas. Além desses conceitos, será apresentado alguns métodos utilizados atualmente em cada uma dessas etapas.

3.1 Biometria

As abordagens de identificação pessoal que utilizam "alguma coisa que você sabe", como Número de Identificação Pessoal (PIN - "Personal Identification Number"), ou "alguma coisa que você tenha", como um cartão de identificação, não são confiáveis o suficiente para satisfazer os requisitos de segurança de um sistema de transações eletrônicas porque não têm a capacidade de diferenciar um usuário legítimo de um impostor que adquiriu de forma ilegal o privilégio de acesso (18). Esta fragilidade pode ser evitada se utilizarmos o nosso corpo como chave do sistema. Alguns traços físicos ou comportamentais são muito mais complicados de serem forjados que uma cadeia de caracteres (17).

Biometria é uma tecnologia utilizada para identificação de um indivíduo baseado em suas características físicas ou comportamentais, baseia-se em "alguma coisa que você é ou faz" para realizar a identificação e, por isso, tem a capacidade de diferenciar entre um indivíduo legítimo de um impostor (18). As características físicas estão relacionadas a composição do corpo humano e seu formato e as comportamentais estão relacionadas a forma como o corpo humano faz algo (17). A figura 3.1 contém alguns exemplos desses dois tipos diferentes de características biométricas.

Teoricamente, qualquer característica física/comportamental pode ser utilizada para identificação caso siga alguns dos seguintes requisitos (21):

1. **universidade**: qualquer pessoa comum pode ser avaliada sobre essa característica;
2. **singularidade**: dada duas pessoas distintas, elas não podem ter a mesma característica dentro de uma proporção satisfatória;

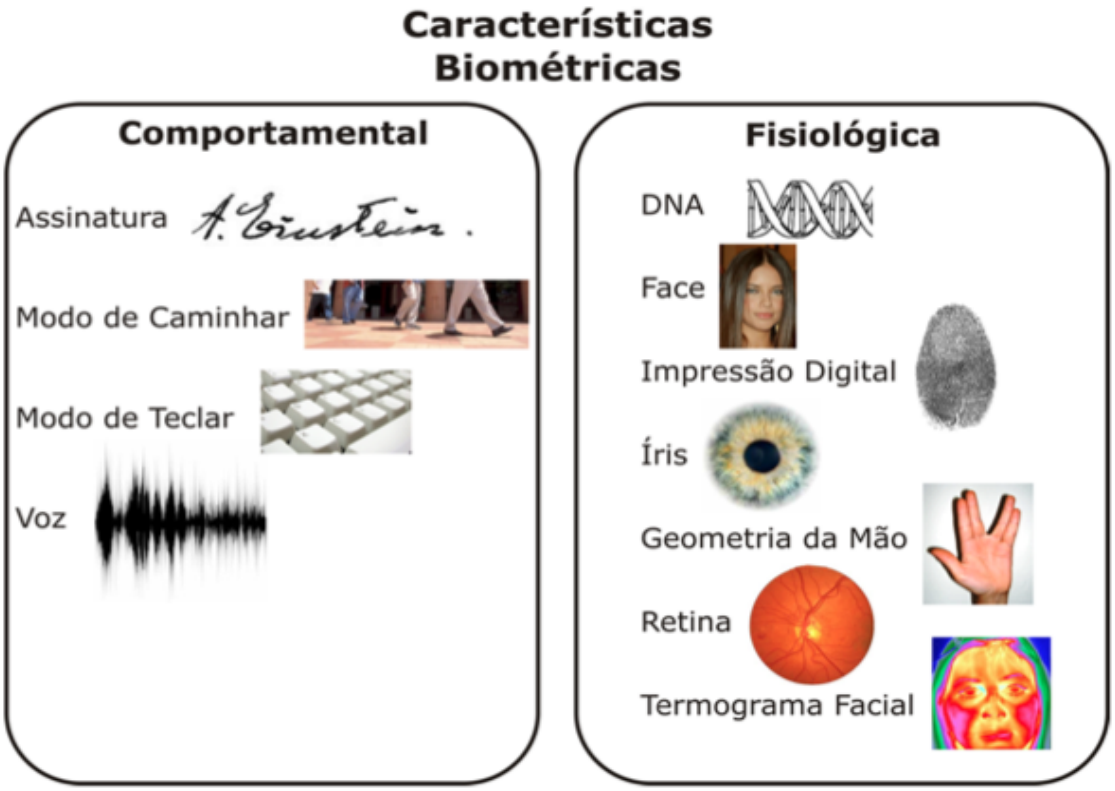


Figura 3.1: Exemplos de algumas características biométricas (17).

3. **permanência**: a característica não pode mudar significativamente de acordo com o tempo;
4. **exigibilidade**: pode ser mensurada quantitativamente;

Porém, na prática também são considerados outros requisitos (21):

1. **desempenho**: o processo de identificação deve apresentar um resultado aceitável;
2. **aceitação**: indica em que ponto as pessoas estão dispostas a aceitar o sistema biométrico;
3. **evasão**: refere-se a facilidade de ser adulterado;

São várias as vantagens que os sistemas biométricos têm em relação aos sistemas convencionais. Listamos as vantagens principais (17):

- características biométricas não podem ser perdidas ou esquecidas;
- características biométricas são difíceis de serem copiadas, compartilhadas e distribuídas;
- os sistemas biométricos necessitam que a pessoa esteja presente no local da autenticação;

Na prática um sistema biométrico deve ser capaz de (18):

1. atingir uma acurácia aceitável e com uma velocidade razoável;
2. não ser prejudicial aos indivíduos e ser aceito pela população alvo;
3. ser suficientemente robusto para métodos fraudulentos;

Novas técnicas de reconhecimento por meio de face, íris, retina e voz, entre outras, têm sido abordadas para aplicações em sistemas de reconhecimento automático (3, 28). Das nove características utilizadas atualmente a face é uma das mais populares (21). Nas tabelas 3.1 e 3.2 são mostradas as noves características e seus respectivos comportamentos baseados nos requisitos mencionados acima.

Tabela 3.1: Requisitos teóricos para algoritmos de reconhecimento facial (21).

Biometria	**Universidade**	**Singularidade**	**Permanência**	**Exigibilidade**
Face	Alta	Baixa	Média	Alta
Digital	Média	Alta	Alta	Média
Geometria da Mão	Média	Média	Média	Alta
"Hand Vein"	Média	Média	Média	Média
Iris	Alta	Alta	Alta	Média
"Retina Scan"	Alta	Alta	Média	Baixa
Assinatura	Baixa	Baixa	Baixa	Alta
Voz	Média	Baixa	Baixa	Média
Termograma	Alta	Alta	Baixa	Alta

Tabela 3.2: Requisitos práticos para algoritmos de reconhecimento facial (21).

Biometria	**Desempenho**	**Aceitação**	**Evasão**
Face	Baixa	Alta	Baixa
Digital	Alta	Média	Alta
Geometria da Mão	Média	Média	Média
"Hand Vein"	Média	Média	Alta
Iris	Média	Média	Alta
"Retina Scan"	Alta	Baixa	Alta
Assinatura	Baixa	Alta	Baixa
Voz	Baixa	Alta	Baixa
Termograma	Média	Alta	Alta

Os sistemas biométricos podem ser classificados em sistemas de verificação ou identificação. Sistemas de verificação são aqueles que autenticam a identidade dos usuários comparando-os com os próprios *templates*. Eles conduzem uma comparação "um para um" e determinam se o usuário é quem realmente diz ser. O maior desafio para esse tipo

de sistema é a acurácia. Geralmente, não é muito difícil satisfazer o requisito de tempo de resposta pois somente uma comparação "um para um" é feita (18).

Sistemas de identificação reconhecem um indivíduo pesquisando em todo o banco de dados procurando por uma correspondência. Eles conduzem uma comparação "um para muitos" para estabelecer a identidade do indivíduo. Ao contrário dos sistemas de verificação, nesse tipo de sistema tanto a acurácia quanto o tempo são os grandes desafios, por causa da necessidade de explorar todo o banco de dados. Geralmente, sistemas de identificação são mais complexos que sistemas de verificação (18).

Em um sistema biométrico exitem duas possíveis respostas, um usuário é ou não é quem afirma ser, sendo assim o sistema pode classificar o usuário como cliente ou impostor. Nessa tomada de decisão pode ocorrer dois tipos de erros: uma falsa aceitação, ao aceitar um impostor, (*False Acceptance* - FA) ou uma falsa rejeição (*False Rejection* - FR), ao rejeitar um cliente. Baseado nesses erros, duas taxas são utilizadas para avaliar sistemas biométricos: taxa de falsa aceitação (*False Acceptance Rate* - FAR) e taxa de falsa rejeição (*False Rejection Rate* - FRR) (17).

A FAR é a probabilidade de um sistema biométrico aceitar um impostor como cliente. Essa probabilidade é calculada pela equação 3.1 em que Nfa é o número de falsas aceitações e Ni é o número de impostores que tentaram acessar o sistema. A variação da taxa é representada pelo intervalo fechado $[0, 1]$, onde o valor 1 significa que todos os impostores foram falsamente aceitos e o valor 0 significa que todos impostores foram rejeitados. Logo quanto menor o FAR mais seguro o sistema é (17).

$$FAR = \frac{Nfa}{Ni}(17) \tag{3.1}$$

A FRR é a probabilidade de um sistema biométrico rejeitar um cliente e classifica-lo como impostor. Essa probabilidade é calculada pela equação 3.2 em que Nfr é o número de falsas rejeições e Nc é o número de clientes que tentaram acessar o sistema. A variação da taxa é representada pelo intervalo fechado $[0, 1]$, onde o valor 1 significa que todos os clientes foram falsamente rejeitados e o valor 0 significa que todos os clientes foram aceitos corretamente. Em sistemas cuja performance tem maior grau de prioridade que a segurança, deve-se reduzir a FRR para minimizar a ocorrência de falsas rejeições (17).

$$FRR = \frac{Nfr}{Nc}(17) \tag{3.2}$$

A partir dessas taxas de erro, pode-se obter outras medidas como a *Equal Error Rate* (ERR). Esta corresponde a taxa de erro na qual tanto a FAR quanto a FRR possuem o mesmo valor. Como muitos sistemas têm comportamentos diferentes, a ERR normalmente é utilizada para uma comparação mais rigorosa entre os sistemas. Quanto menor for a ERR, mais preciso é considerado o sistema (17).

3.2 Reconhecimento Facial

O reconhecimento facial é uma das atividades mais comuns realizadas diariamente por seres vivos dotados de certa inteligência. Essa simples atividade vem despertando o interesse de pesquisadores que trabalham com computação visual e inteligência artificial. O objetivo desses pesquisadores é de construir sistemas artificiais capazes de realizar o

reconhecimento de faces humanas e a partir desta capacidade, construir os mais diferentes tipos de aplicações: sistemas de vigilância, controles de acesso, definições automáticas de perfis, entre outros (25).

No anos 70, os estudos do reconhecimento facial eram baseados sobre atributos faciais mensuráveis como olhos, nariz, sobrancelhas, bocas, entre outros. Porém, os recursos computacionais eram escassos e os algoritmos de extração de características eram ineficientes. Nos anos 90, as pesquisas na área ressurgiram, inovando os métodos existentes (3, 18) e disseminando a técnica.

Um dos motivos que incentivou os diversos estudos sobre reconhecimento facial são as vantagens que o mesmo possui em relação a impressão digital e a íris. No reconhecimento por impressão digital a desvantagem consiste no fato que nem todas as pessoas possuem uma impressão digital com "qualidade" suficiente para ser reconhecida por um sistema. Já o reconhecimento por íris apresenta uma alta confiabilidade e larga variação, sendo estável pela vida toda. Porém, a desvantagem está relacionada ao modo de captura da íris que necessita de um alinhamento entre a câmera e os olhos da pessoa (3).

Basicamente existem duas particularidades que fazem da face uma característica biométrica bastante atrativa (17):

1. A aquisição da face é feita de forma fácil e não-intrusiva;
2. Possui uma baixa privacidade de informação: como a face é exposta constantemente, caso uma base de faces seja roubada, essas informações não representam algum risco e não possibilitam um uso impróprio;

Umas das maiores dificuldades dos sistemas de reconhecimento é tratar a complexidade dos padrões visuais. Mesmo sabendo que todas as faces possuem padrões reconhecidos, como boca, olhos e nariz, elas também possuem variações únicas que devem ser utilizadas para determinar as características relevantes. Outra dificuldade encontrada em relação a essas características é que elas possuem uma larga variação estatística para serem consideradas únicas para cada indivíduo. O ideal seria que a variância inter-classe seja grande e a intra-classe pequena, pois assim imagens de diferentes faces geram os códigos mais diferentes possíveis, enquanto imagens de uma mesma face geram os códigos mais similares possíveis. Portanto, estabelecer uma representação que capture as características ideias é um difícil problema (3).

Do ponto de vista geral, o reconhecimento facial continua sendo um problema aberto por causa de várias dificuldades que aumentam a variância intra-classe (18). Entre estas, destacamos as mais comuns (3):

- iluminação;
- ângulos e poses;
- expressões;
- cosméticos e acessórios;
- extração da face do contexto ou do fundo;

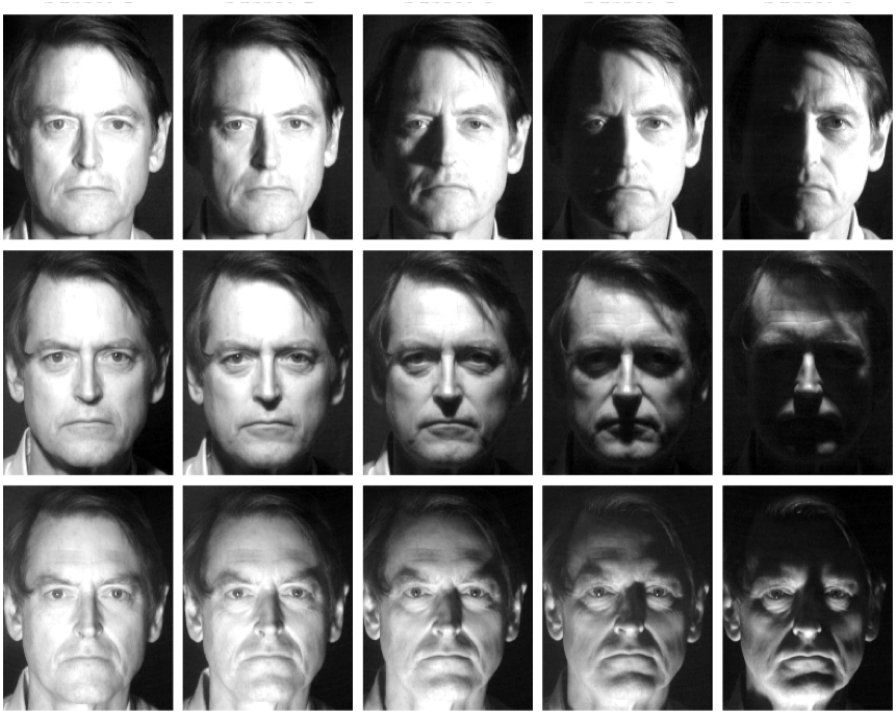


Figura 3.2: Exemplo de uma imagem de uma pessoa com a mesma expressão facial, vista do mesmo ponto de vista mas sobe diferentes condições de iluminação (26).

Na Figura 3.2, temos um exemplo de como uma mesma pessoa, com a mesma expressão facial, vista do mesmo ponto de vista, pode parecer drasticamente diferente quando as fontes de luz possuem diferentes direções (26).

No contexto de identificação, o reconhecimento facial se resume no reconhecimento de um "retrato" frontal, estático e controlado. Estático pois os "retratos" utilizados nada mais são que imagens, podendo ser tanto de intensidade quanto de profundidade e controlado pois a iluminação, o fundo, a resolução dos dispositivos de aquisição e a distância entre eles e as faces são essencialmente fixas durante o processo de aquisição da imagem (18).

Basicamente, o processo de reconhecimento facial pode ser divido em duas tarefas principais (18):

1. Detecção de faces em imagens;
2. Reconhecimento das faces encontradas;

Falaremos dessas duas tarefas separadamente nas próximas subseções.

3.2.1 Detecção de Faces em imagens

A primeira etapa para o reconhecimento de faces é a detecção de um rosto, e a partir daí a comparação do mesmo com modelos conhecidos pelo sistema (18, 25). Em um sistema de reconhecimento facial, tanto o tempo de resposta quanto a confiabilidade desta etapa influenciam diretamente no desempenho e o emprego deste sistema (25).

A detecção de faces é definida como o processo que determina a existência ou não de faces em uma imagem e uma vez encontrada alguma face, sua localização deve ser apontada através de um enquadramento ou através de suas coordenadas dentro da imagem (25). A Figura 3.3 representa um exemplo da detecção de uma face em uma imagem.

Figura 3.3: Exemplo de um processo de detecção de uma face em uma imagem.

O processo de detecção de faces geralmente é dificultado pelas seguintes razões mostradas a seguir:

1. **Pose**: as imagens de uma face podem variar de acordo com a posição relativa entre a câmera e a face (frontal, 45 graus, perfil, "de cabeça para baixo"), e com isso algumas características da face, como olhos e nariz, podem ficar parcialmente ou totalmente ocultadas (23).
2. **Presença de acessórios**: características faciais básicas importantes para o processo de detecção podem ficar ocultadas pela presença de acessórios, como óculos, bigode, barba, entre outros (23, 25).
3. **Expressões faciais**: embora a maioria das faces apresente estruturas semelhantes (olhos, bocas, nariz, entre outros) e são dispostas aproximadamente na mesma configuração de espaço, pode haver um grande número de componentes não rígidos e texturas diferentes entre as faces. Um exemplo são as flexibilizações causadas pelas expressões faciais (23, 25);
4. **Obstrução**: faces podem ser obstruídas por outros objetos. Em uma imagem com várias faces, uma face pode obstruir outra (23).
5. **Condições da imagem**: a não previsibilidade das condições da imagem em ambientes sem restrições de iluminação, cores e objetos de fundo (23, 25).

Atualmente, já existem diferentes métodos/técnicas de detecção de faces. Tais métodos podem ser baseados em imagens de intensidade e em imagens de cor ou também em imagens de três dimensões. Focaremos nos principais métodos de imagens de cor e de intensidade que serão utilizados neste trabalho. Estes podem ser divididos em 4 categorias:

1. **Métodos baseados em conhecimento:** métodos, desenvolvidos principalmente para localização facial, baseados em regras derivadas do conhecimento dos pesquisadores do que constitui uma típica face humana. Normalmente, captura as relações existentes entre as características faciais. É fácil encontrar regras que descrevem as características faciais. Por exemplo, uma face sempre é constituída por dois olhos simétricos, um nariz e uma boca. As relações entre essas características podem ser representadas pelas distâncias relativas e posições. Este método possui desvantagens em relação a construção do conjunto de regras. Se estas são muito gerais, corre-se o risco, de que o sistema que as utilizam, apresentar uma alta taxa de falsos positivos. Se são muito específicas, podem ser ineficazes ao tentar detectar faces por não satisfizerem todas as regras, diminuindo muito a precisão da detecção (19, 23);
2. **Métodos baseados em características invariantes:** esses algoritmos tem como objetivo principal encontrar as características estruturais que existem mesmo quando a pose, o ângulo e as condições de iluminação variam. E por meio dessas características localizar a face. São desenvolvidos principalmente para localização facial (23). A principal desvantagem desse método é que tais características invariantes podem ser corrompidas devido a algum tipo de ruído ou a fortes variações nas condições de iluminação, comprometendo a eficiência. A cor da pele e a textura da face são as principais características invariantes que podem ser utilizadas para separar a face de outros objetos (19);
3. **Métodos baseados em *templates*:** vários padrões comuns de um rosto são armazenados tanto para descrever o rosto como um todo quanto para descrever as características faciais separadamente. As correlações entre as imagens de entrada e os padrões armazenados são computados para detecção. Esses métodos são desenvolvidos para serem utilizados tanto para localização e como para detecção facial (23);
4. **Métodos baseados em aparência:** recebem este nome devido ao fato de não utilizarem nenhum conhecimento, a priori, sobre o objeto ou características a serem detectadas. Em contraste com os métodos baseado em *templates*, os modelos são retirados de um conjunto de imagens de treinamento que devem capturar a variabilidade da face. Esses modelos retirados são utilizados para detecção. Nesta classe de algoritmos surge os conceitos de aprendizado e treinamento, uma vez que as informações necessárias para realizar a tarefa de detecção são retiradas do próprio conjunto de imagens sem intervenção externa. São métodos desenvolvidos principalmente para detecção de faces (19, 23);

Um problema relacionado e muito importante é como avaliar a performance dos métodos de detecção de faces propostos. Para isso, muitas métricas foram adotadas como tempo de aprendizagem, número de amostras necessárias no treinamento e a proporção entre taxas de detecção e "falso alarme". Esta última é dificultada pelas diferentes definições para as taxas de detecção e falso alarme adotadas pelos pesquisadores (23).

Método Viola-Jones

O método que será utilizado nesse trabalho será o método *Viola-Jones*. Este é um método baseado em características para detecção de objetos que minimiza o tempo de computação e possui uma alta acurácia permitindo detecção de faces em tempo real. Esse método pode ser utilizado para construir uma abordagem de detecção facial rápida e eficaz (27). É o método utilizado na biblioteca *OpenCV*(*Open Source Computer Vision*) e muito utilizado atualmente.

O Método *Viola-Jones* para detecção facial utiliza quatro conceitos chaves (1, 27):

1. ***Haar features*:** simples características retangulares;
2. ***Integral image*:** uma nova representação da imagem que permite uma rápida avaliação de recursos e características;
3. **O método *AdaBoost***: um método de aprendizagem de máquina utilizado para construir um classificador, selecionando um pequeno número de características importantes usando *AdaBoost*;
4. **Classificador em cascata**: classificador que combina muitas características de maneira eficiente;

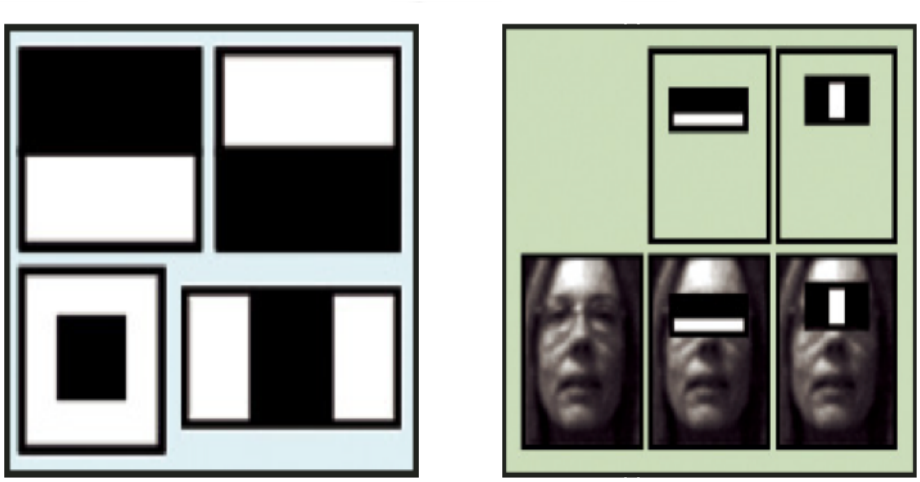


Figura 3.4: Exemplo de simples características retangulares (*Haar features*). Adaptada de (1).

A detecção facial em imagens é baseado em simples características retangulares (*Haar features*), exemplificada na Figura 3.4. Existem vários motivos para se usar essas características ao invés de usar diretamente os *pixels* da imagem. Uma das principais é que sistemas baseados em características são muito mais rápidos que sistemas baseados em *pixels* (27).

Tais simples características são remanescentes das funções de base *Haar*. O método utiliza três tipos de características, exemplificadas na Figura 3.5: características com dois, três ou quatro retângulos (27). A presença de uma característica em uma imagem é

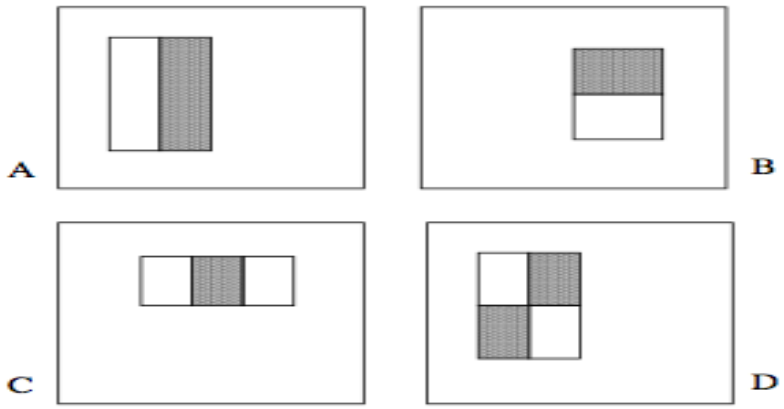


Figura 3.5: *Haar features* com dois, três e quatro retângulos (27).

determinada pela subtração da média dos valores dos *pixels* da região escura pela média dos valores dos *pixels* da região clara. Caso o valor seja maior que um limiar, então essa característica é tida como presente (1).

O Método *Viola-Jones* não trabalha diretamente com as intensidades da imagem. Para determinar a presença ou ausência de centenas de *Haar features* em cada posição de imagem e em várias escalas de forma eficiente, utiliza-se uma técnica chamada *Integral image*. Basicamente, o método consiste em acrescentar pequenas unidades juntas. Neste caso, pequenas unidades são valores de *pixels*. O valor "integral" para cada *pixel* é a soma de todos os *pixels* acima e a esquerda. Começando pelo canto superior esquerdo da imagem e atravessando para direita e para baixo, toda a imagem pode ser "integrada" com poucas operações por *pixels* (1, 27). Com a nova representação de imagem criada, qualquer *Haar feature* pode ser computada para qualquer escala e localização em um tempo constante (27).

Na Figura 3.6, após a "integração", o *pixel* (x, y) contém a soma de todos os valores de *pixels* dentro da região retangular sombreada no canto superior esquerdo de (x, y). Para calcular a média dos valores de *pixel* neste retângulo, basta dividir o valor em (x, y) pela área do retângulo (1).

A Figura 3.7 mostra como calcular os valores de um retângulo que não está localizado no canto superior esquerdo da imagem. Suponha que se deseja obter a soma dos valores do retângulo em D. De maneira intuitiva, pode-se realizar esse cálculo somando os valores dos *pixels* no retângulo formado por $A + B + C + D$, menos a soma nos retângulos $A + B$ e $A + C$, mais a soma dos valores de *pixel* em A. Em outras palavra, $D = A + B + C + D - (A + B) - (A + C) + A$ (1, 27).

Porém, usando *Integral image*, $A+B+C+D$ é o valor do ponto 4, $A+B$ é o valor do ponto 2, $A+C$ é o valor do ponto 3, e A é o valor do ponto 1. Então, com *Integral image*, você pode obter a soma dos valores de *pixel* de qualquer retângulo na imagem original usando somente três operações (1, 27):

$$(x_4, y_4) - (x_2, y_2) - (x_3, y_3) + (x_1, y_1) \tag{3.3}$$

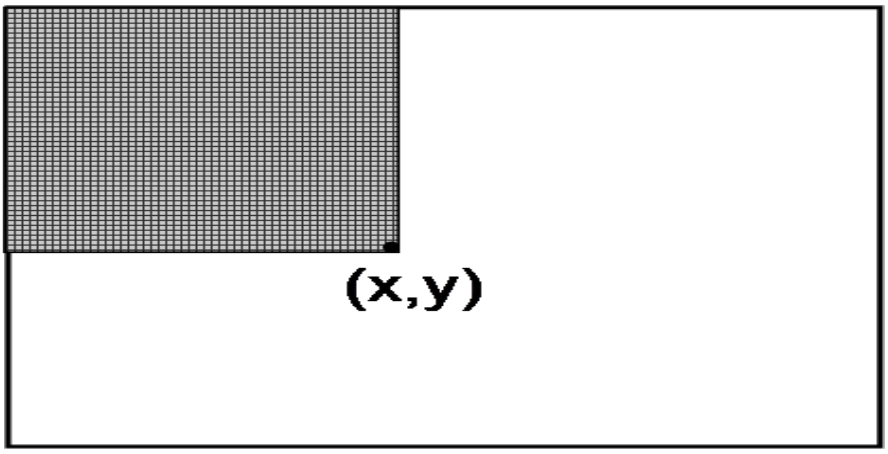


Figura 3.6: Exemplo da aplicação da técnica *Integral image*. Após a "integração" o *pixel* (x, y) contém a soma dos valores dos *pixels* do retângulo sombreado. Adaptada de (1).

Para selecionar os *Haar features* que serão utilizados e para definir os limiares, o método *Viola-Jones* utiliza um método de aprendizagem de máquina chamado *AdaBoost*. Este combina vários classificadores "fracos" para criar um classificador "forte".

Um classificador fraco é aquele que só obtém a resposta correta um pouco mais frequente que um "palpite aleatório". A combinação desses classificadores "fracos", onde cada um "empurra" a resposta final um pouco na direção certa, pode ser considerado como um classificador "forte". O método *AdaBoost* seleciona um conjunto de classificadores "fracos" para combinar e atribui pesos a cada um. Essa combinação ponderada resulta em um classificador "forte" (1).

Em qualquer região de uma imagem, o número total de *Haar features* é muito grande, muito maior que o número de *pixels*. Para assegurar uma classificação rápida, o processo de aprendizagem deve excluir o maior número de características disponíveis, e focar nas que são críticas. A seleção dessas características é alcançada através de uma simples modificação no método *AdaBoost*: o mecanismo de aprendizagem é construído de forma que cada classificador "fraco" retornado dependa de somente uma única característica. Como resultado, cada estágio do processo seleciona um novo classificador "fraco" o que pode ser visto como um processo de seleção de características. *AdaBoost* fornece um algoritmo de aprendizagem eficaz (27).

O método *Viola-Jones* combina uma série de classificadores *AdaBoost* na forma de uma cadeia de filtros, como na Figura 3.8, que recebe o nome de "Classificadores em Cascata". Cada filtro em si é um classificador *AdaBoost* com um número relativamente pequeno de classificadores "fracos" (1).

O limiar de aceitação, em cada nível, é definido "baixo" o suficiente para que passe por todos, ou quase todos, os exemplos de face do conjunto de treinamento (um grande banco de imagens contendo faces). Os filtros em cada nível são treinados para classificar imagens de treinamento que passaram por todas fases anteriores (1).

Durante a utilização, se alguma região de uma imagem falhar em passar em um desses filtros, esta é imediatamente classificada como "não face". Quando uma região de uma

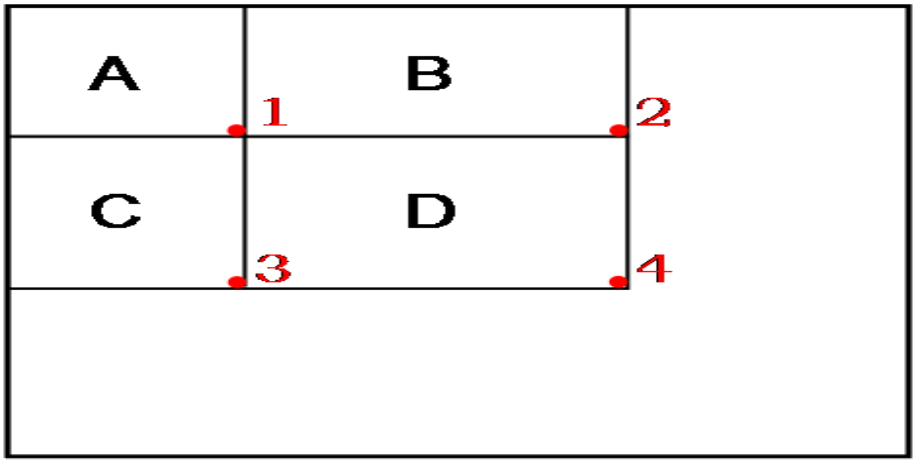


Figura 3.7: Exemplo utilizado para mostrar como calcular a soma dos valores de *pixel* de um retângulo que não está localizado no canto superior esquerdo da imagem utilizando *Integral image*. Adaptado de (1).

imagem passa por um filtro, ela vai para o próximo filtro na cadeia. As regiões das imagens que passarem por todos os filtros na cadeia são classificadas como "faces" (1).

O algoritmo utilizado para construção dos "Classificadores em Cascata" alcança um ótimo desempenho e, ao mesmo tempo, reduz drasticamente o tempo de computação. O aspecto chave é que os menores classificadores (filtros), e por isso mais eficientes, podem ser utilizados para rejeitar a maioria das regiões das imagens que não são faces antes que os classificadores mais complexos sejam utilizados (27). A ordem dos filtros no classificador é baseado nos pesos que o método *AdaBoost* define para cada filtro. Os filtros com maior peso são colocados no início, para eliminar as regiões das imagens que não são faces o mais rápido possível (1).

O método *Viola-Jones* é adequado para utilização em sistemas de detecção de faces em tempo real. Agora, o próximo passo para o processo de "Reconhecimento Facial" é comparar as faces encontradas com modelos conhecidos pelo sistema para realizar a identificação.

3.2.2 Reconhecimento das Faces encontradas

Na etapa de reconhecimento, as faces detectadas, serão comparadas com um banco de dados de faces conhecidas. Várias técnicas são usadas para acelerar essa comparação já que para identificar o usuário é necessário um grande número de comparações. Dentre as técnicas utilizadas existem duas variações principais, as que usam como entrada dados de imagens 2D (imagens de intensidade e imagens de cor) e as que usam como entradas dados de imagens 3D (imagens de profundidade).

Mais antiga e mais comum, as técnicas 2D são amplamente utilizadas e as principais são:

1. ***Eigenfaces*:** extrai as informações relevantes de uma face, codifica-a da maneira mais eficiente possível, e a compara com um banco de faces codificadas de maneira

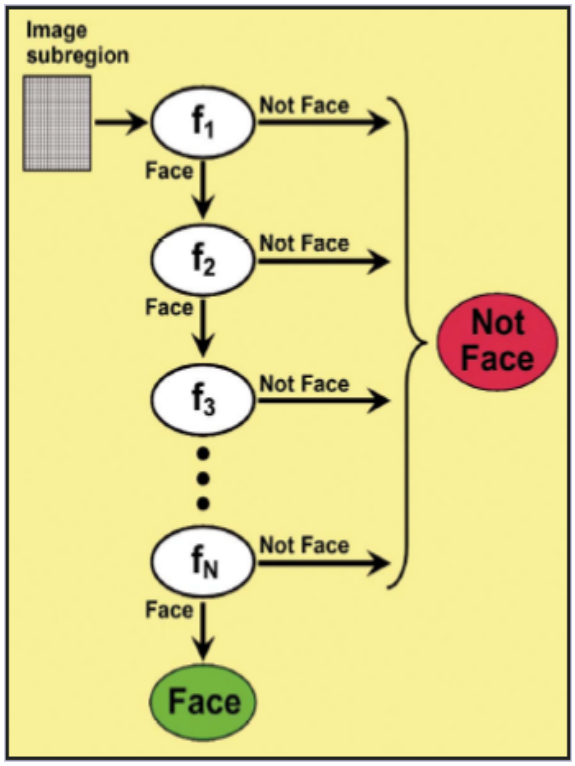


Figura 3.8: Ilustração de uma classificador em cascata composto com uma cadeia de filtros (1).

similar. Extrai as informações relevantes contidas em uma imagem de uma face de uma maneira simples captando a variação em uma coleção de imagens de face e usando essa informação para codificar e comparar imagens individuais de faces (22). É baseado em projeção linear das imagens em uma espaço de imagens de menor dimensão. Para redução dimensional utiliza PCA - "*Principal Component Analisys* (Análise de componente principal)" maximizando a dispersão total entre todas as imagens (26);

2. **Redes Neurais:** uma rede neural artificial é um modelo computacional capaz de, entre outras funções, armazenar, classificar padrões, realizar interpolação de funções não-lineares e apresentar soluções heurísticas para problemas de otimização. Isso é conseguido através de um processo denominado "aprendizado". O uso de redes neurais visa tornar o sistema de reconhecimento capaz de absorver pequenas variações ocorridas no momento da coleta de medidas faciais. Espera-se, portanto, mais robustez a falhas e que responda de forma mais confiável (25);

3. ***Fisher Faces*:** igual ao *Eigenfaces*, é um método que procura uma projeção linear das faces para um espaço dimensional menor onde os impactos causados pelas variações de luz e expressões faciais são minimizados. O método é derivado da discriminante linear de Fisher (FLD) (26).

O *Eigenfaces* mostrou um ótimo custo benefício, devido ao seu baixo custo computacional e a sua, relativamente alta, confiabilidade. Então, falaremos mais detalhadamente

como ele funciona.

Reconhecimento facial com *Eigenfaces*

Este é um algoritmo simples e fácil de implementar e os passos utilizados por ele também são utilizados em muitos métodos avançados. Os princípios básicos por trás dele, como PCA - "*Principal Component Analisys* (Análise de componente principal)" e *Distance-Based Matching* (Correspondência Baseada na Distância) aparecem cada vez mais na computação visual e em diversas aplicações de inteligência artificial (12).

Basicamente, as etapas do processo de reconhecimento são simples e bem definidas. Dada uma imagem de um rosto desconhecido e imagens do rosto das pessoas conhecidas executa as seguintes ações (12):

1. Computa a "distância" entre a nova imagem e cada uma das imagens já conhecidas.
2. Seleciona a imagem mais próxima do rosto em questão.
3. Se a "distância" da nova imagem para a imagem já catalogada for menor que o limite pré-definido, "reconhece" a imagem caso contrário classifica como "desconhecida".

Em *Eigenfaces*, a distância entre as imagens é medida ponto a ponto e é conhecida como "Distância Euclidiana". A distância euclidiana em duas dimensões entre os pontos P_1 e P_2 é dada pela fórmula $d_{12} = \sqrt{(d_x + d_y)}$, onde $d_x = x_2 - x_1$ e $d_y = y_2 - y_1$ e representada na Figura 3.9 (12).

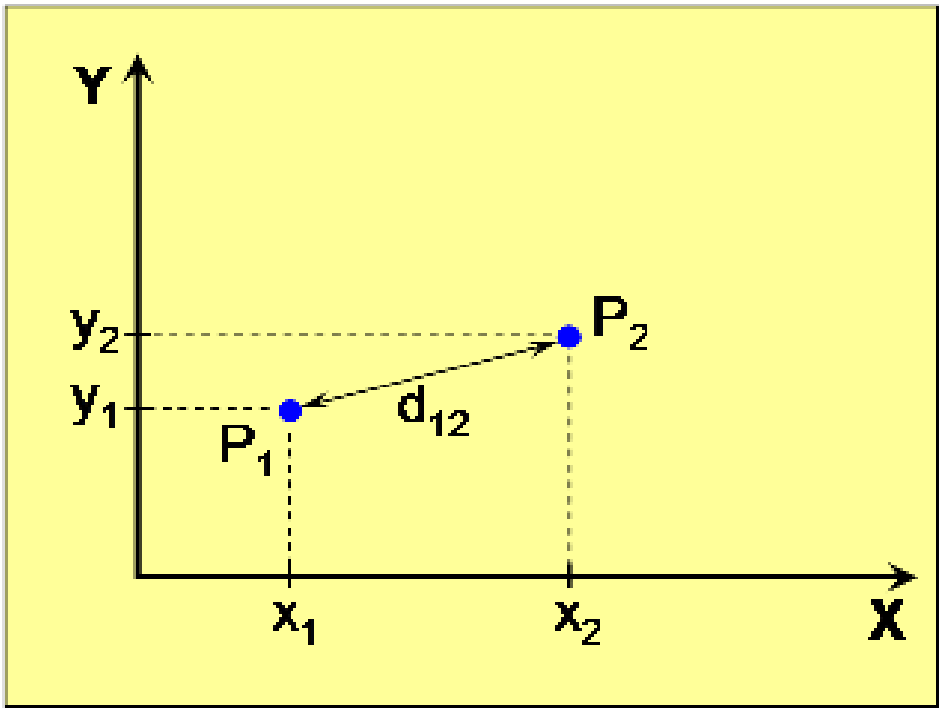


Figura 3.9: Distância euclidiana entre dois pontos em duas dimensões (12).

Imagens possuem "ruídos" e vamos definir ruído como qualquer coisa que atrapalhe na identificação, como por exemplo, as diferenças na luminosidade. Cada *pixel* possui uma intensidade de ruído diferente. Com cada *pixel* contribuindo para o ruído total, este se torna muito elevado comparado com a informação útil que se possa retirar da imagem, dificultando o processo de reconhecimento. Uma solução é diminuir a dimensionalidade da imagem, tornando assim o ruído menor e possibilitando extrair, da imagem, as informações importantes (12).

Redução do ruído utilizando PCA

O *Eigenfaces* utiliza o método PCA - "*Principal Component Analisys* (Analíse de componente principal)" para reduzir a dimensionalidade da imagem (12).

Para se ter uma ideia de como o PCA funciona, vejamos um caso especial chamado de *"least squares line fit"*. O lado esquerdo da Figura 3.10 mostra um exemplo de uma linha média entre três pontos, que são, no mapa em duas dimensões, Los Angeles, Chicago e Nova York. Estes três pontos do mapa são quase, mas não completamente, uma única linha. A linha tem apenas uma dimensão. Por isso, se conseguirmos substituir as localizações dos pontos em duas dimensões por localizações ao longo de uma única linha, vamos ter reduzido a dimensão do ponto em questão (12).

Como os pontos, nos quais se baseia o nosso exemplo, estão praticamente alinhados, uma linha pode ser traçada através deles com pouco erro. O erro no ajuste da linha é medido pela soma do quadrado da distância de cada ponto da linha. A linha de melhor ajuste é aquela que possui o menor erro (12).

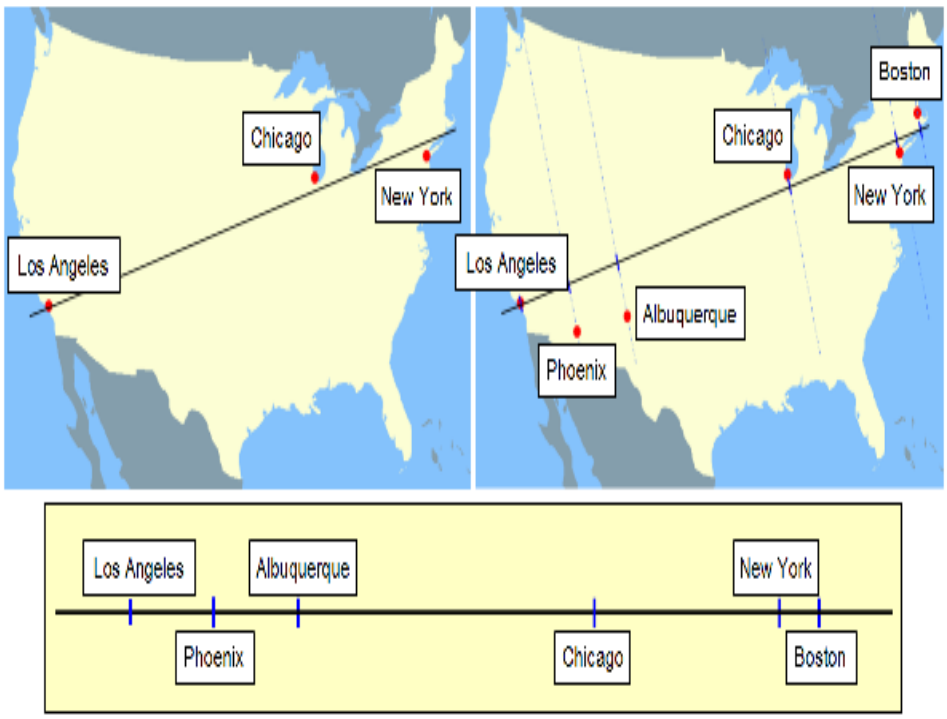


Figura 3.10: Exemplo para redução da dimensão do ponto (12).

Embora a linha encontrada acima seja um objeto de apenas uma dimensão, a mesma está localizada dentro de um espaço de duas dimensões e tem como orientação a sua inclinação. A inclinação da linha expressa algo importante sobre os três pontos, ela indica a direção em que os mesmos estão mais espalhados (12).

Se posicionarmos a origem do nosso plano cartesiano em algum lugar dessa linha, podemos escrever a equação da linha como uma simples $y = mx$, onde m é a inclinação da linha: dy/dx (12).

Quando ela é descrita desta maneira, a linha se torna um subespaço do espaço gerador definido pelo sistema de coordenadas (x, y). Esta descrição enfatiza o aspecto dos dados que estamos interessados, ou seja, a direção que mantém esses pontos mais separados um do outro (12).

Esta direção da separação máxima é chamada de "primeira componente principal" de um conjunto de dados. A próxima direção com máxima separação é a perpendicular a esta e é denominada "segunda componente principal". Em um conjunto de dados com apenas duas dimensões, podemos ter no máximo duas componentes principais (12).

Como em *Eigenfaces* cada imagem da face, de tamanho 50x50, é tratada como um ponto em espaço de 2500 dimensões, podemos ter muitas componentes principais (12).

No entanto, o número de componentes principais que podemos encontrar também é limitada pelo número de pontos. Para ver o porque disto, podemos pensar em um conjunto de dados que consiste de apenas um ponto. Não há sentido em ter uma linha na direção da separação máxima desse conjunto de dados, porque não há nada a separar. Agora, considere um conjunto de dados com apenas dois pontos. A linha que conecta esses dois pontos é a primeira componente principal. Mas não há uma segunda componente principal, porque não há nada mais a separar. Portanto, o número de componentes principais que podemos encontrar nunca será maior que o número de pontos menos um (12).

Voltando ao exemplo da Figura 3.10, agora que achamos um subespaço, precisamos de uma maneira de converter pontos do espaço gerador para pontos no subespaço. Esse processo recebe o nome de projeção. Quando você projeta um ponto em um subespaço, você define a ele a localização no subespaço que é a mais próxima da localização no espaço de dimensão maior. Por exemplo, para projetar um ponto de uma mapa de duas dimensões para uma linha você precisa achar o ponto na linha que é mais perto do ponto no mapa. (12).

As marcas azuis na Figura 3.10 mostram as localizações no subespaço das três cidades que definiram a linha. Outros pontos também podem ser projetados para esta linha. O lado direito da Figura 3.10 mostra a localização projetada para Phoenix, Albuquerque, Boston (12).

Em *Eigenfaces*, a diferença entre duas imagens é a distância euclidiana entre os pontos projetados em um determinado espaço. A projeção desses pontos em um subespaço de menor dimensão é a técnica utilizada para melhorar a relação sinal ruído (12).

Basicamente, a ideia explicada até agora consiste na transformação da imagem em um ponto no espaço, a redução da dimensão desse ponto, seguida do cálculo da distância entre os pontos.

As componentes principais - autovetor(*eingenvector*)

Em nossa definição de uma linha como um subespaço, usamos x e y como coordenadas para definir m, que é sua inclinação no espaço gerador. Quando m é um componente principal de um conjunto de pontos, ela é chamada de autovetor ou *eingenvector*, daí o nome *Eigenfaces* (12).

Para o reconhecimento facial, cada autovetor representa a inclinação de uma linha em um espaço de 2500 dimensões. Como no caso de duas dimensões, precisamos de todas as 2500 dimensões para definir a inclinação de cada linha. Embora seja impossível visualizar uma linha em muitas dimensões, podemos visualiza-las em uma simples imagem convertendo o ponto projetado para colocar o valor de cada *pixel* correspondente, ou seja, ao projetar o ponto em uma linha obtemos uma nova localização no espaço e ao converter essa nova localização de volta em imagem obtemos imagens *facelike* também chamadas de *eigenfaces* (12, 20).

O *Eigenfaces* é um método interessante para dar-nos alguma intuição sobre os componentes principais para o nosso conjunto de dados. O lado esquerdo da Figura 3.11 mostra as imagens das faces de dez pessoas. O lado direito mostra os seis primeiros componentes principais deste conjunto de dados, apresentados como *eigenfaces*. Uma *eigenface* da componente principal é uma imagem média de todas as *eigenfaces* que estão projetadas na mesma. Essas *eigenfaces* muitas vezes têm um olhar fantasmagórico, porque combinam elementos de várias faces. As regiões de *pixels* mais brilhantes e as regiões mais escuras em cada imagem são as que mais contribuem para as componentes principais (12).

As *eigenfaces* médias das componentes principais são usadas para que a partir delas possa se estimar a distância euclidiana entre a imagem que se deseja reconhecer e as imagens presentes no banco e a partir dessas distâncias dizer de qual delas a nova imagem mais se aproxima (20).

As limitações do *Eigenfaces*

As componentes principais encontradas pelo PCA apontam para a maior variação de dados. Uma das premissas do *Eigenfaces* é que a variabilidade das imagens subjacentes corresponde à diferença entre as faces. Esta suposição nem sempre é válida. A Figura 3.12 mostra as faces de dois indivíduos apresentadas em quatro diferentes condições de iluminação (12).

Na verdade, elas são imagens de faces de duas das dez pessoas mostradas na Figura 3.11. Quando a iluminação é muito variável esse algoritmo não é muito efetivo (12).

Outros fatores que podem aumentar a variabilidade da imagem em direções que tendem a diluir a identidade no espaço PCA incluem mudanças na expressão, ângulo da câmera e posição da cabeça (12).

A Figura 3.13 mostra como a distribuição de dados afeta o desempenho do *Eigenfaces*.

Quando os pontos referentes as imagens de cada indivíduo ficam aglutinadas e satisfatoriamente separadas das imagens do conjunto de imagens de outros indivíduos temos o melhor caso para o funcionamento do *Eigenfaces*.

Caso os pontos referentes as imagens dos indivíduos tenham uma variabilidade muito grande, a probabilidade de choque de imagens de dois indivíduos num mesmo ponto do subespaço PCA se torna muito grande tornando extremamente difícil separar os dois indivíduos (12).

Na prática, a projeção de determinadas imagens de uma pessoa no subespaço PCA provavelmente colidirá com projeções de imagens de outras pessoas. Como os autovetores (*eigenvector*) são determinados pela variabilidade dos dados, ficamos limitados a quão grande deve ser essa. Podemos tomar medidas para limitar, ou para gerir de outra forma, as condições ambientais que podem confundi-lo. Por exemplo, colocar a câmera na altura do rosto irá reduzir a variabilidade no ângulo da câmera (12).

As condições de iluminação, tais como iluminação lateral vinda de uma janela, são mais difíceis de controlar. Mas você pode considerar o acréscimo de inteligência em cima de reconhecimento facial para compensar isso. Por exemplo, se sabemos onde ele está localizado, e em que direção está olhando, ela pode comparar a imagem do rosto atual apenas com aquelas em situação semelhante (12).

Mesmo com sistemas altamente ajustados, sistemas de reconhecimento facial estão sujeitos a casos de identidade equivocada (12).

"Passo a passo" sobre *Eigenfaces*

A abordagem para o reconhecimento facial com *Eigenfaces* requer as seguinte operações de inicialização (22):

1. Adquirir um conjunto inicial de imagens para serem usadas como conjunto inicial de dados;
2. Treinar o algoritmo calculando a *eigenface* média;

Com o sistema inicializado, os seguintes passos devem ser seguidos para reconhecer novas imagens de faces (22):

1. Projetar a nova imagem na componente principal;
2. Calcular a distância euclidiana entre a *eigenface* média e a *eigenface* nova;
3. Comparar com as distâncias das outras imagens e dizer se é "conhecida" ou não de acordo com o limiar de proximidade.

Figura 3.11: Direita: imagens de rosto para dez pessoas. Esquerda: os seis primeiros componentes principais, visto como *Eigenfaces* (12).

Figura 3.12: Imagens das faces de dois indivíduos. A face de cada indivíduo é apresentada em quatro diferentes condições de iluminação. A variabilidade devido à iluminação aqui é maior do que a variabilidade entre os indivíduos. *Eigenfaces* tende a confundir as pessoas quando os efeitos de iluminação são muito fortes (12).

Figura 3.13: Como as distribuições de dados afetam o reconhecimento com *Eigenfaces*. Topo: O melhor cenário possível - pontos de dados para cada pessoa bem separados. Meio: O pior cenário - a variabilidade entre as imagens das faces de cada individuo é maior do que a variabilidade entre os indivíduos. Embaixo: um cenário realista - separação razoável, com alguma sobreposição (12).

Capítulo 4

Trabalhos Correlatos

Para prover serviços inteligentes para *SmartSpaces* é necessário adquirir informações de contexto, informações sobre a pessoas no ambiente e suas interações. Informações como número de pessoas, identidade, localização, postura, orientação da cabeça, entre outros.

Neste capítulo analisaremos alguns projetos que procuram obter informações de contexto como esta. De modo mais específico, informações sobre a identidade e localização das pessoas em um *SmartSpace*.

4.1 Projeto CHIL

O Projeto CHIL (*Computers in the Human Interaction Loop*) é composto por um time de quinze laboratórios internacionais de pesquisa acadêmica e industrial . Eles colaboram entre si no desenvolvimento de serviços que visam ajudar as pessoas de forma proativa durante suas atividades diárias e, em particular, durante sua interação com as outras pessoas.

Alguns dos protótipos que foram desenvolvidos no projeto incluem um *workspace* perceptivo e colaborativo, diversos serviços que facilitam a colaboração em reuniões e em salas de palestras, e um sistema perceptivo de assistência a um escritório virtual (32).

4.1.1 Rastreamento e Localização de Pessoas

A pesquisa sobre rastreamento de pessoas foi focada principalmente no rastreamento de pessoas dentro de *SmartSpaces*. O objetivo desse monitoramento foi determinar, para todos os pontos no tempo, as coordenadas dos ocupantes do *SmartSpaces* na cena em relação a uma *frame* de coordenadas. O que contradiz com a maioria das pesquisas de rastreamento visual, onde somente as coordenadas na imagem são estimadas (32).

Os sensores usados no *SmartSpace* incluem (32):

- um mínimo de quatro câmeras fixas instaladas nos cantos da sala, com campos de visão sobrepostos;
- uma câmera com grande ângulo de visão fixa com vista para o quarto inteiro;
- três arrays de microfones em forma de T de 4 canais;
- um microfone de Mark III de 64 canais;

Essa grande quantidade de sensores disponíveis pode ser vista como uma vantagem, pois podem oferecer uma grande redundância nas informações capturas que podem ser exploradas pelos algoritmos usados. Porém, isso pode também ser visto como um grande desafio, pois surgem problemas como sincronização dos dados, transferência de processamento distribuído, fusão de espaço-temporal, entre outros.

Do ponto de vista do áudio, é importante mencionar que o Projeto CHIL representa uma das primeiras tentativas de realizar e avaliar sistematicamente rastreamento acústico com uma rede distribuída de microfones (32).

O sistema de rastreamento e localização foi amplamente testado usando os dados dos seminários e reuniões CHIL (32).

Durante o projeto, muito progresso foi feito partindo de sistemas de única modalidade com a inicialização manual ou implícita, usando recursos simples, o que implica várias etapas de processamento manualmente concatenadas offline e acompanhamento de no máximo uma pessoa, para um sistema totalmente automático, com auto-inicialização, em tempo real, usando uma combinação de recursos, fusão dos *streams* de vários sensores de áudio e vídeo e capaz de rastrear alvos múltiplos (32).

Sobre os algoritmos de rastreamento visual, duas abordagens principais foram seguidas pelos vários sistemas de rastreamento desenvolvidos (32):

1. uma abordagem baseada em modelos, em que um modelo 3D do objeto rastreado é mantido.
2. uma abordagem orientada a dados, onde sistemas de rastreamento 2D operam de forma independente sobre os diferentes ângulos de visão das câmeras e os dados do rastreamento pertencentes a um mesmo alvo são coletados no formato de um rastreamento 3D.

Em termos de desempenho, a abordagem baseada em modelos geralmente prevê uma melhor acurácia, mas menos precisão do que a abordagem orientada a dados. O tratamento das oclusões e da associação dos dados dos sistemas de rastreamentos independentes são as desvantagens do modelo orientado a dados. Para diminuir o impacto desses problemas, começaram a detectar e rastrear rostos em vez de corpos inteiros (32).

Sobre os algoritmos de rastreamento por áudio, três abordagens principais foram seguidas. Entre elas, a que teve a melhor performance foi um sistema baseado em um *Joint Probabilistic Data Association Filter*, que mantém o controle de uma série de fontes de áudio, incluindo fontes de ruídos, resolvendo associações dos dados e atualizando o conjunto de posições (32).

As abordagens de rastreamento por meio de áudio e vídeo foram combinas em um rastreamento multimodal. Esse rastreamento multimodal é, notavelmente, baseado em filtro de partículas uma vez que permitem uma integração flexível de recursos através dos sensores (32).

No rastreamento multimodal, era esperado que a fusão dos diferentes tipos de dados proveria resultados mais precisos, eliminando, assim, os efeitos de decisões erradas tomadas por algum rastreamento monomodal. Porém, não aconteceu o que se esperava. O rastreamento multimodal é altamente dependente das tarefas e dados em mãos, e exige um cuidadoso equilíbrio na disponibilidade e qualidade dos dados (32).

4.1.2 Identificação das Pessoas

A fim de realizar identificação de forma natural e implícita, os sensores distribuídos no ambiente devem monitorar continuamente o espaço, e captura de dados áudio-visual das pessoas discretamente quando eles aparecem. Ou seja, o sistema de identificação de pessoas deve operar em segundo plano, sem necessitar de atenção e cooperação das pessoas. Dependendo da localização de uma pessoa e sua distância dos sensores, os dados recebidos podem variar (32).

Para reconhecimento facial, o sistema utiliza sequências de imagens fornecidas pelas várias câmeras no *SmartSpace*. A cada $200ms$ imagens das "caixas delimitadoras das faces" e posições dos centros dos olhos são fornecidas, como exemplificado na Figura 4.1. As faces, então, são alinhadas utilizando os centros dos olhos ou utilizando as "caixas delimitadoras das faces". Para obter robustez contra alguns erros, o sistema gera algumas imagens alinhadas adicionais modificando rótulos dos centros dos olhos ou os rótulos das "caixas delimitadoras das faces" (32).

Figura 4.1: Exemplo do sistema de identificação facial do Projeto CHIL. No canto inferior direito, pode-se observar a imagem da face extraída (32).

Uma das abordagens utiliza reconhecimento facial baseado na aparência utilizando transformada discreta de cosseno (DCT). Utiliza modelagem de variação intrapessoal de Gauss para avaliar a probabilidade de que a diferença de um rosto de uma galeria de imagens de faces é de fato intrapessoal. O sistema de reconhecimento utiliza um classificador do "vizinho" mais próximo (32).

Nesse projeto foi observado que selecionando somente as imagens frontais de faces, ao invés das amostras disponíveis, é prejudicial para a performance do sistema de reconhecimento (32).

As decisões obtidas do vários pontos de vista das várias câmeras são combinados por meio de uma regra de soma ponderada. Os pesos são determinado de acordo com a separação dos dois melhores resultados (32).

4.2 Identificação multimodal e localização de usuários em um ambiente inteligente

Esse trabalho propõe um sistema que realiza detecção de movimento, rastreamento de pessoas, reconhecimento facial, identificação baseado em características, localização baseada em áudio, módulos de identificação baseado em áudio, e fusão de todas essas informações utilizando filtro de partículas para obter um sistema robusto de identificação e localização. O sistema foi projetado para operar de uma maneira completamente automática, sem intervenção do usuário (29).

Os *streams* de dados são processados com ajuda de um *middleware* servidor-cliente genérico chamado *SmartFlow*, o que resulta em uma arquitetura portável para diferentes plataformas. Este *middleware* permite transporte de grande quantidade de dados de sensores para algoritmos de reconhecimento em nós distribuídos na rede (29).

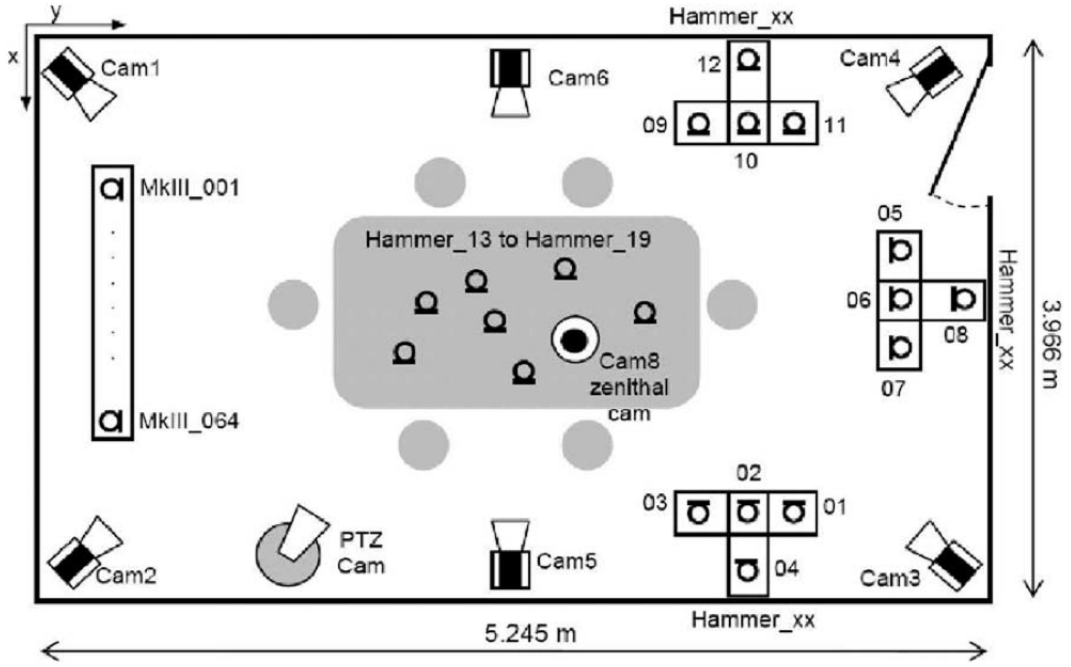


Figura 4.2: A configuração do *SmartSpace* (29).

Os sensores utilizados e as condições do *SmartSpace* deste projeto são descritos pela Figura 4.2. Os seguintes sensores foram utilizados (29):

- quatro câmeras nos cantos da sala (rotuladas como Cam1 a Cam4 na Figura 4.2);
- uma câmera *zenithal fish-eye* no telhado (rotulada como Cam8 na Figura 4.2);
- uma câmera ativa apontada e com zoom para a porta de entrada para capturar as faces das pessoas que entram na sala (rotulada como PTZ na Figura 4.2);
- um *array* de microfones NIST Mark III de 64 canais ;
- três *clusters* de microfone de 4 canais no formato de T;

- oito microfones no teto;

O projeto conecta diferentes métodos de rastreamento e reconhecimento para prover serviços de recohecimento e rastreamento multimodal no *SmartSpace*. A Figura 4.3 o fluxo de informação no sistema multimodal. O objetivo do sistema é identificar cada usuário ao entrar pela porta e rastreá-lo no ambiente. Para ter um sistema robusto que rastreia e identifica todas as pessoas ao mesmo tempo, utilizaram uma abordagem multimodal (29).

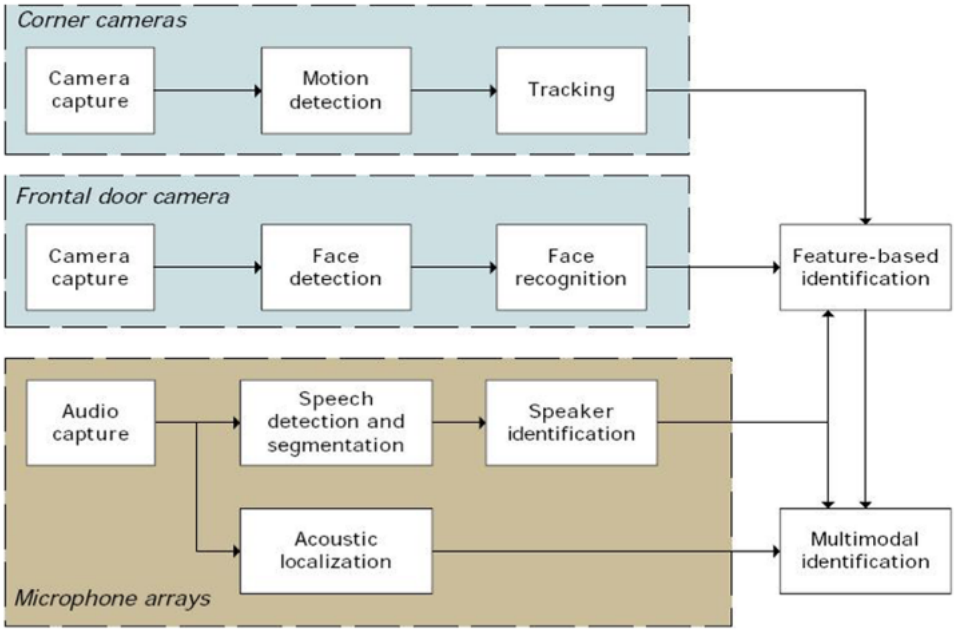


Figura 4.3: *Workflow* da informação na arquitetura do *SmartSpace* (29).

As câmeras no sistema são responsáveis pela detecção de movimento, rastreamento, detecção e reconhecimento facial que são as funcionalidades que nos interessam.

4.2.1 Detecção de movimento, Localização e Rastreamento

A detecção de movimento tenta separar o "primeiro plano" do "fundo"para o seu funcionamento. O método utilizado é baseado na detecção de objetos em movimento sob a suposição que imagens em uma cena sem objetos em movimento mostra regularidades, que pode ser modelada utilizando métodos estáticos. O conjunto de treinamento é construído por sequências pequenas de gravações *offline* feitas da sala vazia (29).

Para localização e rastreamento foi utilizado a abordagem de um mapa de ocupação probabilística (*probabilistic occupancy map* - POM) simplificado para ambientes internos, onde as trajetórias de movimentos são curtas e menos frequentes quando comparadas com trajetórias em ambientes externos (29).

4.2.2 Detecção e Reconhecimento Facial

A detecção de movimento é feita utilizando o método *Viola-Jones* já descrito anteriormente utilizando a biblioteca *OpenCV*.

Para realizar o reconhecimento facial utiliza uma técnica que aproveita a vantagem que o ambiente é constantemente monitorado e combina a informação de várias imagens para realizar o reconhecimento. As imagens das faces são fornecidas pelo módulo de detecção. Para cada sequência de imagens, as faces de um mesmo indivíduo são agrupadas. Então, para cada grupo, o sistema compara as imagens com a galeria de imagens (29).

Uma abordagem baseada em PCA (*Principal Component Analisys*) foi utilizado para comparação entre as imagens.

4.3 Captura de contexto dinâmico e *Arrays* de vídeos distribuídos para *SmartSpaces*

Esse trabalho apresenta detalhes de um projeto de pesquisa de longo prazo, onde *SmartSpaces* de uma gama de funcionalidades úteis são projetados, construídos e avaliados sistematicamente. Algumas das funcionalidades chave incluem: detecção de intrusos; rastreamento várias pessoas; pose do corpo e análise de postura; identificação de pessoas; modelagem do corpo humano; e análise do movimento (33).

O sistema proposto monitora o ambiente em baixa resolução de forma contínua, detectando somente a presença e localização das pessoas e suas dimensões. Formas de aquisição de imagens mais detalhadas são ativadas quando um evento ou atividade de potencial interesse é detectado. Esses eventos serão os focos de atenção do sistema (33).

O monitoramento de baixa resolução e de grande área do ambiente é alcançado graças a algumas câmeras de amplo ângulo de visão e estáticas. Com um pequeno número de câmeras PTZ (*pan/tilt/zoom*) ativas, múltiplos focos de atenção podem ser mantidos de forma simultânea (33).

Foi desenvolvido uma arquitetura de sistema para o *SmartSpace* chamada DIVA. Ela pode ser vista como uma rede inteligente e ativa de câmeras, onde várias câmeras ão controladas para prover uma ampla gama de funcionalidades (33).

O projeto inclui dois *SmartSpaces* separados, mas conectados. O primeiro é chamado de AVIARY foi projetado para ser uma pequena sala de conferências. O segundo chamado de MICASA foi projetado para ser uma pequena sala de aula. A Figura 4.4 mostra fotos dos dois *SmartSpaces* e como estão conectados, além da disposição dos sensores utilizados.

Os sensores na sala AVIARY incluem (33):

- uma rede de quatro câmeras omnidirecionais;
- quatro câmeras PTZ;
- quatro câmeras retilíneas estáticas;
- dois *arrays* de microfones com quatro microfones cada;

As quatro câmeras omnidirecionais (ODVSs) estão perto dos cantos de uma mesa de reunião, cobrindo o quarto inteiro de dentro para fora, como mostrado na Figura 4.4. As câmeras PTZ e retilíneas estão nos "vértices" da sala a $1.4m$ acima do chão. Os dois *arrays* de microfones foram instalados na parede e no teto (33).

Um computador é alocado para rastreamento, utilizando imagens das quatro câmeras omnidirecionais ou imagens das quatro câmeras retilíneas estáticas. Outro computador é

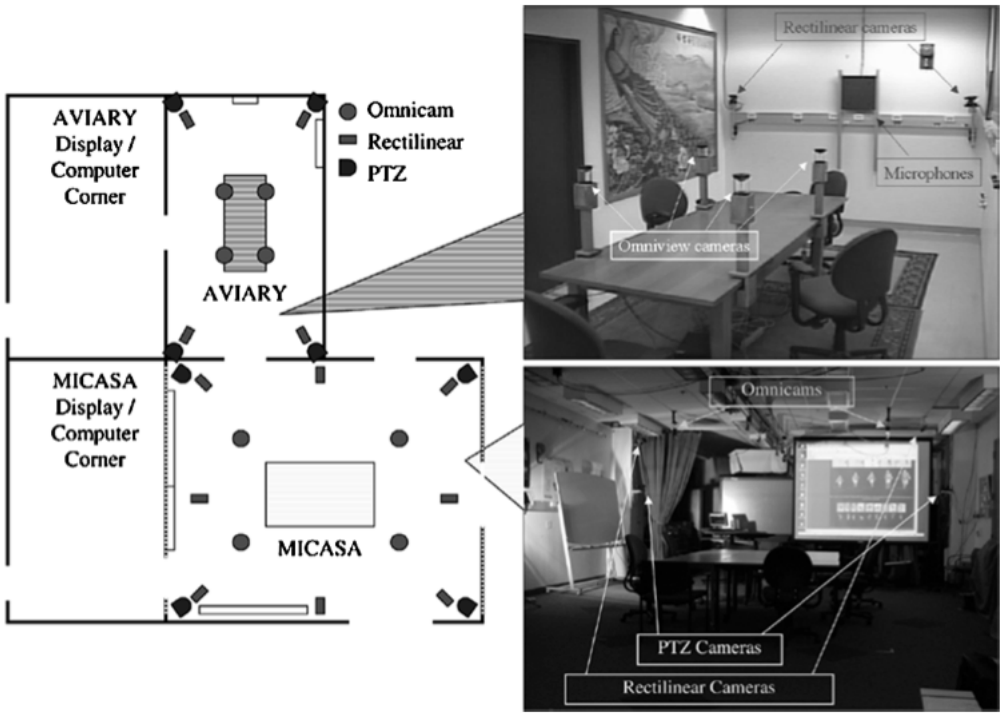


Figura 4.4: Representação dos *SmartSpaces* MICASA e AVIARY (33).

utilizado para analisar os eventos de áudio e de vídeo. Um terceiro computador é utilizado para arquivar *streams* de áudio e vídeo para posterior recuperação (33).

O *SmartSpace* MICASA é duas vezes maior que o AVIARY. Os sensores utilizados nessa sala são (33):

- uma rede de quatro câmeras omnidirecionais;
- quatro câmeras PTZ;
- oito câmeras retilíneas státicas;

As câmeras omnidirecionais são instaladas no teto, como mostrado na Figura 4.4. As câmeras PTZ e quatro câmeras retilíneas foram instaladas de maneira similar ao *SmartSpace* AVIARY. As quatro câmeras retilíneas restantes foram instaladas nas paredes como mostrado como mostrado na Figura 4.4. As câmeras nos vértices possuem maior campo de visão para cobrir toda a sala e fazem parte do array de câmeras para rastreamento (33).

4.3.1 Rastreamento, Detecção de faces e Reconhecimento facial

Foi desenvolvido um sistema em tempo-real que utiliza a rede de câmeras omnidirecionais que é responsável pelo rastreamento, detecção das faces e reconhecimento facial.

O rastreamento baseado na câmeras omnidirecionais é feito detectando a silhuetas das pessoas por meio da subtração do fundo com remoção de sombras.

O vídeo capturado é processado para detecção e reconhecimento de faces como mostrado na Figura 4.5. Para detecção de faces utiliza o método de segmentação de tom de pele. As imagens resultantes são classificadas para rejeitar as imagens sem faces. Então, o método *Eigenface* é utilizado tanto para classificação da face quanto para reconhecimento.

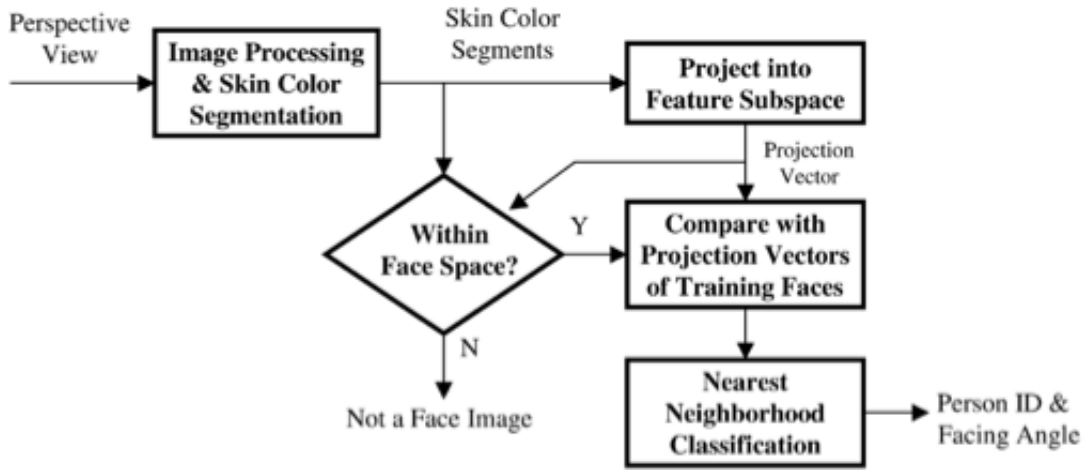


Figura 4.5: Método de detecção e reconhecimento facial (33).

Capítulo 5

Problema e Proposta

Observando a realidade de um *SmartSpace* fica claro que as informações como posição das pessoas e suas respectivas identidades são imprescindíveis para que decisões possam ser tomadas. Atualmente, a maioria das soluções encontradas para fornecer esse tipo de informação foram projetadas para funcionar em ambientes rigidamente definidos. Com isso, não seria adequado tentar incorporar soluções como estas em um ambiente com diferentes dimensões, condições de iluminação, posição dos móveis pois este se resultaria em um cenário diferente. Além disso, precisamos de uma solução que seja integrada com o middleware *UbiquitOS*, o qual gerência nosso *SmartSpace*.

Esse trabalho propõe, então, um sistema aberto de rastreamento, localização e identificação de pessoas no *SmartSpace* que forneça essas informações de contexto para o *UbiquitOS*. Tal sistema será chamado de TRUE, ***T****racker and* ***R****ecognizer* ***U****sers in* ***E****nvironment*.

O sistema *True* utilizará tanto imagens de cor quanto de profundidade. As imagens de profundidade serão utilizadas no rastreamento e localização dos usuários no ambiente, e as imagens de cor serão utilizadas no reconhecimento facial e no cadastro dos usuários. Então, os dispositivos presentes no ambiente deverão ser capazes de fornecer esses tipos de dados a um taxa e qualidade adequada.

Para obter esses dados do ambiente, será utilizado o sensor *Kinect* da Microsoft 5.6, um dispositivo bastante acessível e capaz de fornecer imagens de cor e de profundidade sincronizadas.

O sistema *True* será dividido em três módulos principais:

- **Módulo de Rastreamento**: parte do sistema responsável pelo rastreamento dos usuários no ambiente.
- **Módulo de Reconhecimento**: parte do sistema responsável por reconhecer os usuários rastreados.
- **Módulo de Registro**: parte do sistema responsável pelo cadastro de novos usuários e treino do algoritmo.

O Módulo de Registro será independente dos demais. Porém, os outros dois módulos deverão trocar informações entre si para centralizar todas as informações (localização e reconhecimento) de todos usuários rastreados no ambiente. A seguir, será explicado mais detalhadamente cada módulo e como deverá ser essa troca de informações.

5.1 Módulo de Reconhecimento

O Módulo de Reconhecimento, como se deduz do próprio nome, será responsável pelo reconhecimento facial dos usuários no ambiente. Para isso, será necessário detectar a face do usuário em questão em uma imagem e depois realizar o reconhecimento da mesma em tempo real.

Para realizar a detecção facial será utilizado o método *Viola-Jones* 3.2.1. Um método que pode ser utilizado para construir uma abordagem de detecção facial rápida e eficaz (27) em tempo real. Além disso, este método é implementado pela biblioteca *OpenCV* (*Open Source Computer Vision*) onde bons classificadores em cascata de *Haar features* são fornecidos, como por exemplo um classificador de faces frontais, utilizado nesse trabalho.

Para realizar o reconhecimento facial será utilizado *Eigenfaces* 3.2.2. Uma técnica bastante satisfatória quando utilizada sobre uma base de dados (faces) relativamente grande, permitindo ao sistema inferir, das imagens suas principais características e, partindo delas, realizar o reconhecimento das imagens utilizando um número bastante reduzido de cálculos (8), permitindo, assim, um reconhecimento em tempo real.

O reconhecimento será feito em imagens de usuários que serão passadas para Módulo de Rastreamento. Essa imagens serão compostas somente pela região da imagem em que o usuário se encontra, como mostrado na Figura (**colocar figura aqui.**). Basicamente, o processo de reconhecimento será realizado pelas seguintes etapas e ilustrado na Figura 5.1:

1. Obtém a imagem de entrada correspondente a imagem formada somente pelo usuário cujo reconhecimento foi requisitado.
2. Realiza detecção facial na imagem. Caso nenhuma face seja encontrada, retorna "vazio". Vale a pena ressaltar que no máximo uma face pode ser encontrada nesta imagem.
3. Pré-processamento da imagem: a imagem é convertida em escala de cinza, uma nova imagem é criada recortando a região da face encontrada, a imagem, então, é redimensionada e equalizada criando assim uma padrão de tamanho, brilho e contraste nas imagens aumentando a acurácia do reconhecimento.
4. Reconhecimento facial com *Eigenfaces* é realizado.
5. Retorna o nome da face "mais parecida" e a confiança do reconhecimento.

O Módulo de Reconhecimento será dependente do de Rastreamento. Ele ficará ocioso até que chegue uma requisição de reconhecimento de um determinado usuário. No caso, o autor da requisição será o Módulo de Rastreamento. A seção 5.3 explica mais detalhadamente a relação entre os dois módulos.

5.2 Módulo de Rastreamento

O Módulo de Rastreamento será responsável por rastrear os usuários no *SmartSpace*, determinar a localização física de cada um em relação ao *Kinect* e gerenciar suas identidades.

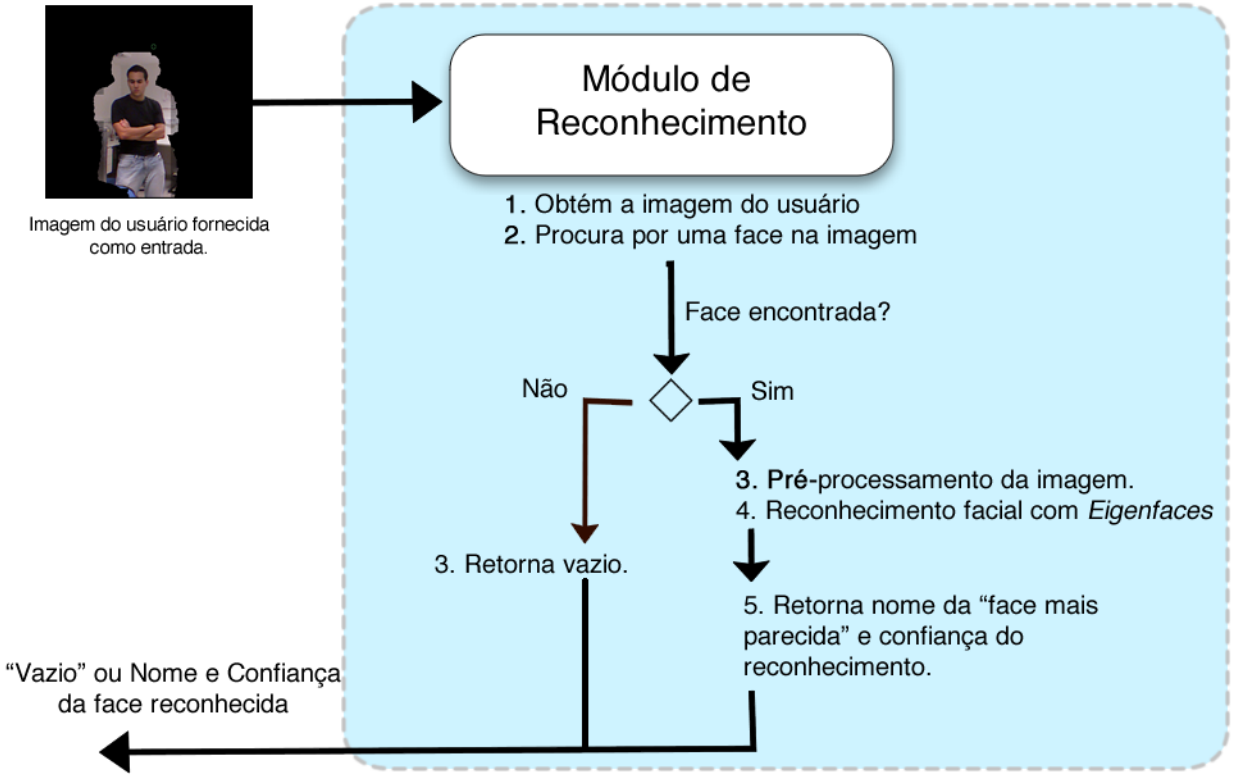


Figura 5.1: Representação das etapas propostas para o reconhecimento facial no Módulo de Reconhecimento.

Para realizar rastreamento e localização dos usuários será utilizado a implementação existente na biblioteca *OpenNI* (*Open Natural Interaction*). Trata-se de um *framework* que define *APIs* para o desenvolvimento de aplicações utilizando interação natural. Utilizando as imagens de profundidade, a detecção e o rastreamento dos usuários será feita utilizando subtração de fundo 2.1.3 e a representação dos usuários será feita por silhuetas 2.1.1.

O rastreamento e a localização serão feitas utilizando as imagens de profundidades providas pelo *Kinect*, tornando-os não susceptíveis as variações nas condições de iluminação. Essas imagens de profundidade nada mais são que *depth maps* (mapas de profundidade), em que cada pixel da imagem contém o valor estimado da distância em relação ao sensor. O *Kinect* fornece esses dados a uma taxa de $30fps$ (*frames* por segundo) com uma resolução $320px$ x $240px$.

Com esses mapas de profundidade, a biblioteca *OpenNI* consegue calcular as coodernadas (x, y, z) em relação ao *Kinect* de qualquer pixel na imagem. Ou seja, se tivermos a representação de um usuário rastreado na imagem, conseguiremos obter sua localização relativa ao *Kinect*. Então, fixando a posição do mesmo no ambiente, conseguiremos

estimar a localização de qualquer usuário rastreado em tempo real.

5.3 Relação Rastreamento e Reconhecimento

Até agora, foi mostrado como os Módulos de Rastreamento e de Reconhecimento funcionarão de maneira isolada, mas não como irão se relacionar. O Módulo de Rastreamento irá deter as informações sobre todos os usuários rastreados no ambiente e será responsável por requisitar reconhecimento ao Módulo de Reconhecimento, que deverá acontecer quando um novo usuário for detectado ou quando for necessário reconhecer um usuário já rastreado.

Basicamente, quando um novo usuário for detectado, a relação entre rastreamento e reconhecimento acontecerá de acordo com as seguintes etapas e representada na Figura 5.2:

1. O Módulo de Rastreamento detecta novo usuário, e obtém um número pré-definido de imagens sucessivas do novo usuário. Para cada imagem, ele cria uma nova imagem de cor contendo somente aquele usuário, como mostrado na Figura (**colocar a figura aqui**), e a envia para o Módulo de Reconhecimento.
2. Para cada imagem recebida, o Módulo de Reconhecimento tenta reconhecer o novo usuário e retorna "vazio" ou o nome e a confiança do reconhecimento.
3. O Módulo de Rastreamento verifica se a confiança é maior que um limiar pré-definido, se for ele incrementa o contador que armazena o número de vezes que o usuário foi reconhecido, armazena o nome obtido juntamente com a confiança e calcula qual nome será atribuído ao novo usuário.

Ao invés de tentar realizar o reconhecimento somente quando novos usuários são detectados, o sistema *True* continuará a tentar reconhecer os usuários já reconhecidos para melhorar a confiança no reconhecimento. Essas tentativas de reconhecer novamente os usuários ocorrerão em intervalos de tempo pré-definidos seguindo as mesmas etapas de quando um novo usuário for detectado. A única etapa que será diferente será a primeira: ao invés de obter várias imagens de um mesmo usuário, serão obtidas uma imagem de cada usuário rastreado e as mesmas serão enviadas ao Módulo de Reconhecimento.

Desta maneira, o Módulo de Rastreamento conseguirá reunir em um só lugar todas as informações sobre os usuários rastreados, como localização corrente, nome, confiança do reconhecimento, quantas vezes o usuário foi reconhecido e quais diferentes nomes foi atribuído ao mesmo.

5.4 Módulo de Registro

O Módulo de Registro será responsável por cadastrar novos usuários no sistema e treiná-lo para também reconhecer esse novo usuário. Basicamente, o processo de registro seguirá as seguintes etapas e ilustrada na Figura 5.3:

1. O novo usuário fica em uma posição fixa e frontal em relação ao *Kinect*.
2. O sistema obtém um número pré-definido de imagens frontais do usuário.

3. O usuário, então, deve rotacionar um pouco a face para a esquerda e o sistema obtém um número pré-definido de imagens do usuário. Depois, deve rotacionar um pouco para direita e o sistema obtém mais imagens do usuário.
4. As imagens obtidas são processadas: as imagens são convertidas em escala de cinza, novas imagens são criadas recortando a região da face encontrada, as imagens, então, são redimensionadas e equalizadas criando assim uma padrão de tamanho, brilho e contraste nas imagens.
5. Armazena-se as imagens.
6. O sistema é treinado para, também, reconhecer esse usuário.

Após o treinamento, o sistema *True* reiniciará para que o reconhecimento seja feito utilizando as novas informações obtidas com o treinamento.

5.5 *SmartSpace* Laico

O ambiente para o qual o sistema *True* será projetado, desenvolvido e testado chama-se LAICO (**LA**boratório de sistemas **I**ntegrados e **CO**ncorrente), um laboratório do Departamento de Ciência da Computação da Universidade de Brasília. O LAICO possui dimensões de, aproximadamente, $7,67m$ x $6,45m$ ilustrado pela Figura 5.4.

5.6 Kinect

O Kinect, mostrado na Figura 5.5, é o nome de um projeto da Microsoft para seu console de videogame Xbox 360, que tem ainda como colaboradora a empresa Prime Sense. O projeto visa criar uma nova tecnologia capaz de permitir aos jogadores interagir com os jogos eletrônicos sem a necessidade de ter em mãos um controle(*joystick*), inovando no campo da jogabilidade.

A Microsoft define o Kinect como "jogos sem necessidade de controle e experiência de entretenimento". Porém, vê-lo como um novo jeito de jogar é subestimar sua significância (9).

O Kinect possui as seguintes especificações técnicas:

- Sensor
  - Lentes com detecção de cores e profundidade
  - Microfone de voz
  - Motor de inclinação para ajuste do sensor
- Campo de visão
  - Campo de visão horizontal: 57 graus
  - Campo de visão vertical: 43 graus
  - Alcance físico da inclinação: (+/-) 27 graus

  - Um alcance máximo de aproximadamente 4.5 metros para câmera de profundidade.

- Fluxo de Dados
  - 320x240 16-bit depth a 30FPS
  - 640x480 32-bit color a 30FPS
  - 16-bit áudio a 16 kHz

Basicamente, é um hardware composto por câmeras que obtém imagens de cor, som e que utiliza iluminação infra-vermelha (IR) para obter imagens de profundidade (9). A Figura 5.6 mostra a organização interna do Kinect em alto nível.

Um chip personalizado processa os dados providos da câmera de profundidade que está correlacionado com as imagens de cor. Com isso, o software que utiliza o Kinect pode combinar cada pixel com sua profundidade. Os dados processados, são enviados para a máquina por meio de uma interface USB na forma de mapas de profundidades e imagens de cor (9).

Com os dados providos pelo sensor (mapas de profundidades e imagens de cor) e pelo preço acessível, ele se torna um dispositivo passível de ser usado em tarefas como rastreamento, localização e reconhecimento facial.

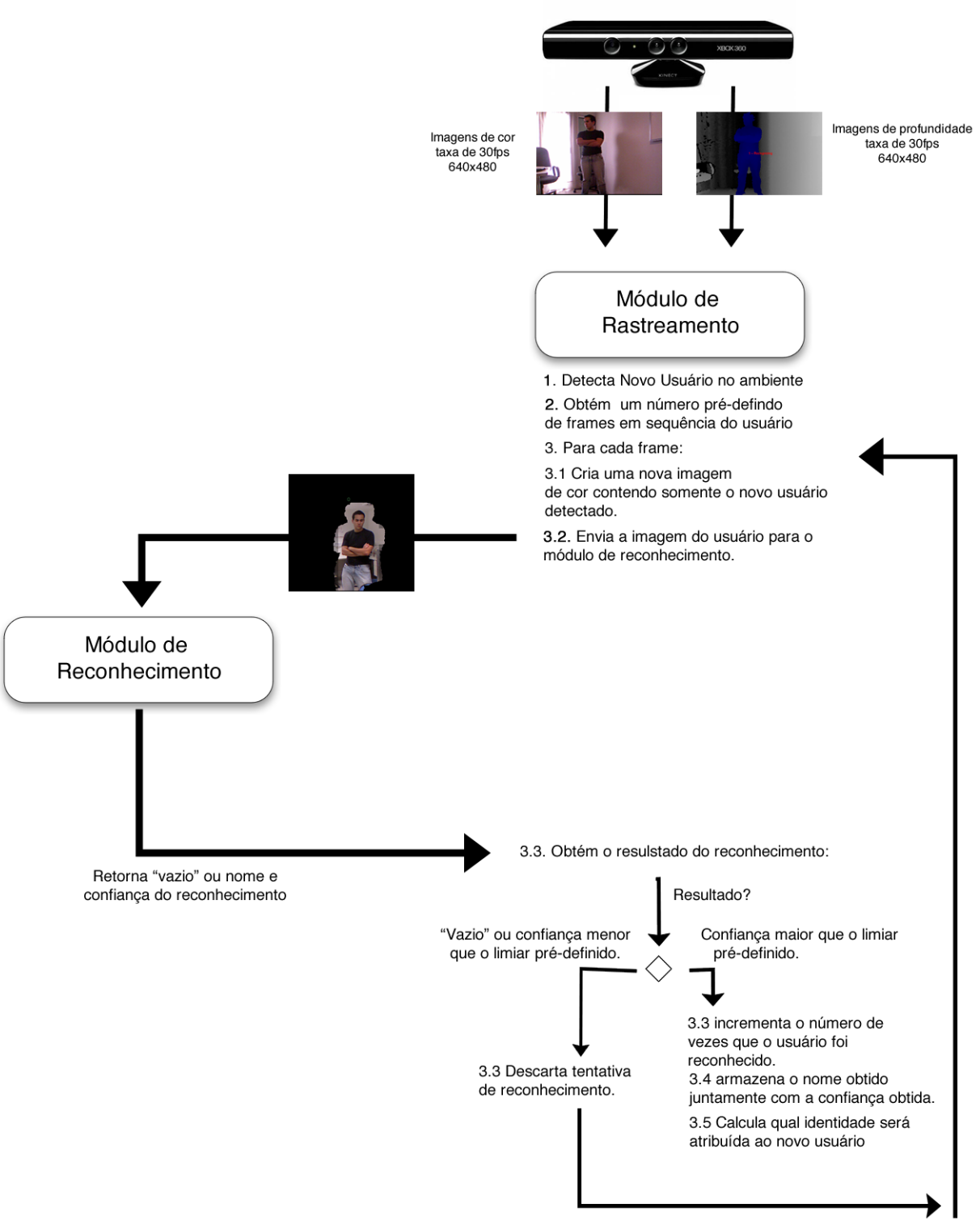
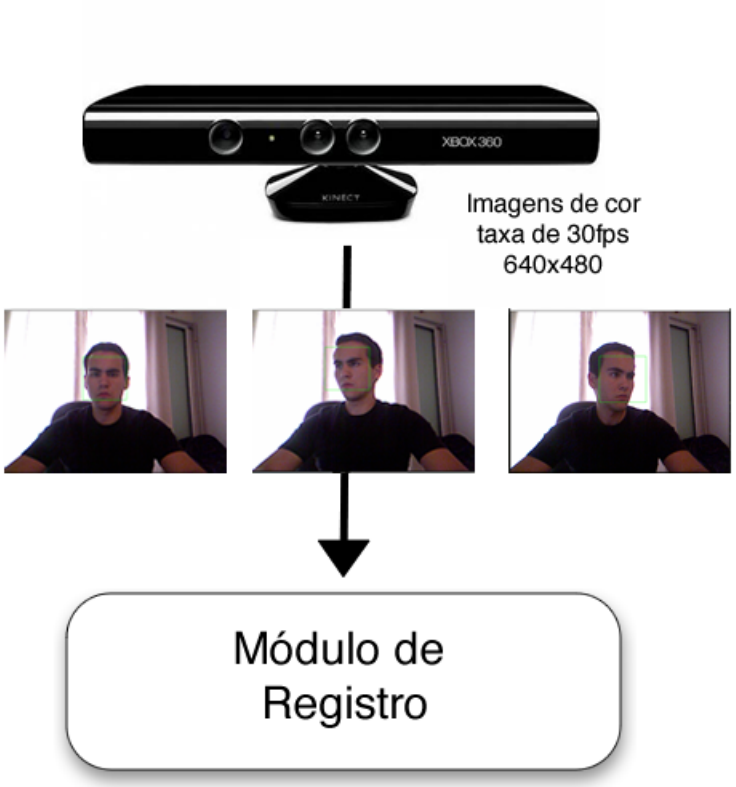


Figura 5.2: Representação da relação que o Módulo de Rastreamento terá com o Módulo de Reconhecimento quando um novo usuário for detectado.



1. Sistema obtém um número pré-definido de imagens frontais da face, imagens da face um pouco rotacionada para esquerda e direita.

2. Imagens das faces do novo usuário são pré-processadas.

3. Imagens são armazenadas.

4. O sistema é retreinado para também reconhecer o novo usuário.

Figura 5.3: Etapas de cadastro de um novo usuário no sistema.

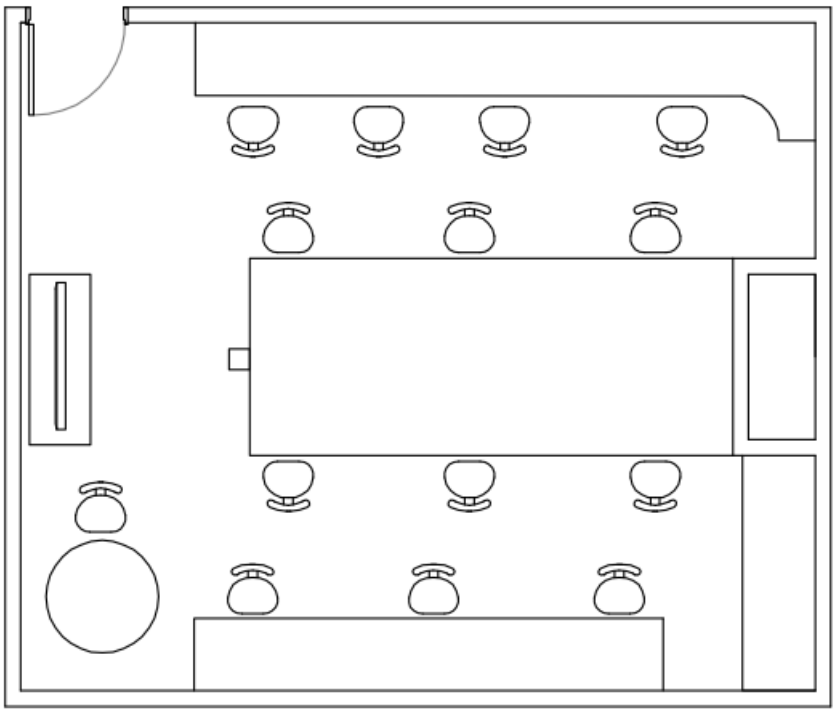


Figura 5.4: Planta do *SmartSpace* Laico.

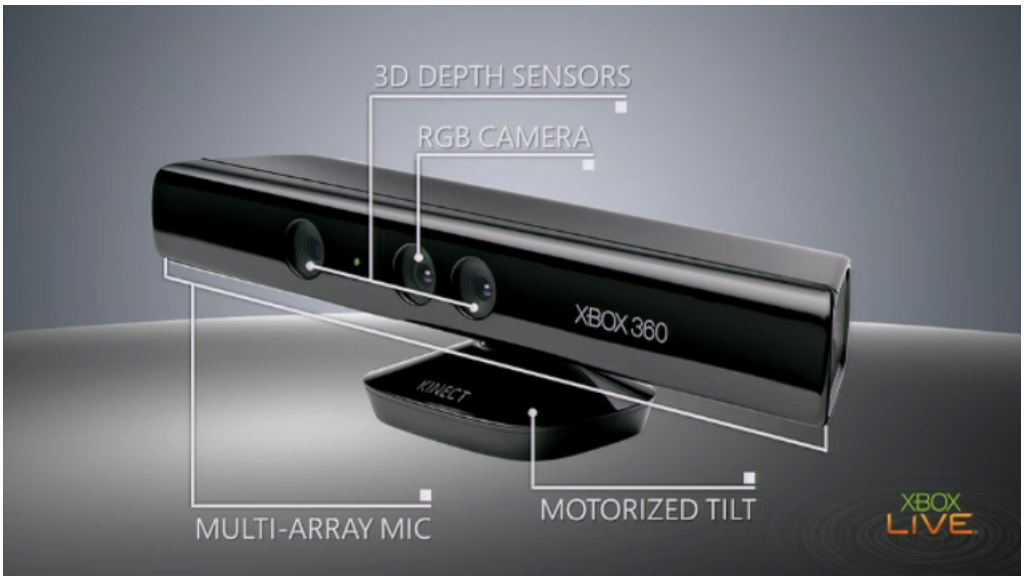


Figura 5.5: Sensor Kinect da Microsoft.

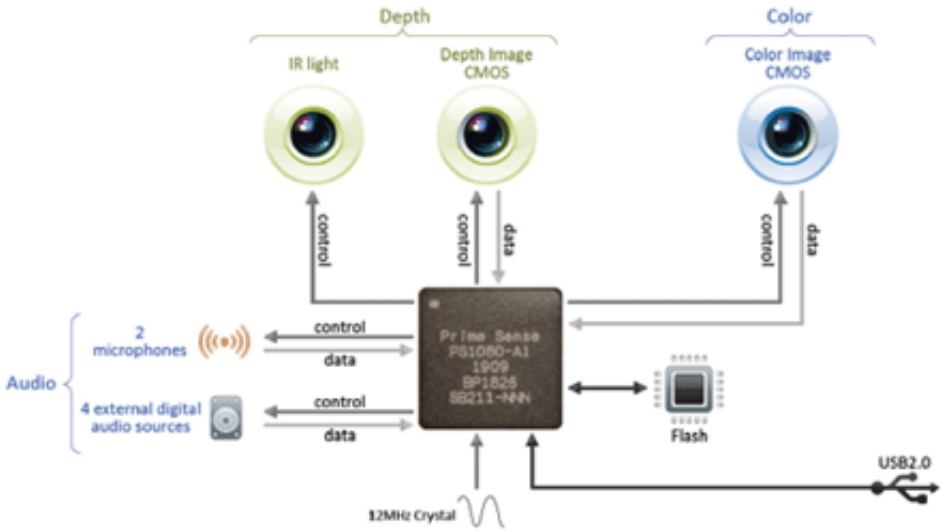


Figura 5.6: Organização interna do Kinect (9).

Referências

[1] How face detection works. *SERVO Magazine*, fevereiro 2007. vi, 19, 20, 21, 22, 23

[2] G. Abowd, C. Atkeson, and I. Essa. Ubiquitous smart spaces. *Georgia Institute of Technology, College of Computing*, 1998. 1

[3] Â. R. Bianchini. Arquitetura de redes neurais para o reconhecimento facial baseado no neocognitron. Master's thesis, Universidade Federal de São Carlos, 2001. 1, 2, 13, 15

[4] F. N. Buzeto. Um conjunto de soluções para a construção de aplicativos de computação ubíqua. Master's thesis, Departamento de Ciência da Computação, Universidade de Brasília, http://monografias.cic.unb.br/dspace/handle/123456789/257, 2010. 1

[5] C.J.Veenman, M.J.T. Reinders, and E. Backer. Resolving Motion Correspondence for Densely Moving Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, jan 2001. 4

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, may 2003. 5

[7] J. Daugman. Face and gesture recognition: Overview. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):675–676, 1997. 1

[8] T. S. Körting e N. L. D. Filho. Utilizando eigenfaces para reconhecimento de imagens. *Fundação Universidade Federal do Rio Grande*. 41

[9] H. Fairhead. All about kinect. vii, 44, 45, 49

[10] A. R. Gomes. Ubiquitos – uma proposta de arquitetura de middleware para a adaptabilidade de serviços em sistemas de computação ubíqua. Master's thesis, Departamento de Ciência da Computação, Universidade de Brasília, http://monografias.cic.unb.br/dspace/handle/123456789/110, 2007. 2

[11] N. Grammalidis, G. Goussis, G. Troufakos, and M. G. Strintzis. 3-d human body tracking from depth images using analysis by synthesis. *Department of Electrical and Computer Engineering University of Thessaloniki*, 2001. 4

[12] R. Hewitt. Face recognition with eigenface. *SERVO Magazine*, 2007. vi, vii, 24, 25, 26, 27, 28, 29, 30, 31

[13] J. Hightower and G. Borriello. Location sensing techniques. *University of Washington, Computer Science and Engineering*, agosto 2001. vi, 9, 10

[14] B. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence Laboratory*, 17:185–203, 1981. 6

[15] W. Hu, T. Tan, Fellow, IEEE, L. Wang, and S. Maybank. A survey on visual surveillance of a survey on visual surveillance of object motion and behaviors. *IEEE Transactions On Systems, Man, and Cybernetics—part C: Applications and Reviews*, 34(3):334–352, augusto 2004. 6

[16] R. Jain, R. Kasturi, and B. G. Schunck. *Machine vision*. McGraw-Hill, Inc., New York, NY, USA, 1995. vi, 8, 9

[17] S. A. D. Junior. Reconhecimento facial 3d utilizando o simulated annealing com as medidas surface interpenetration measure e m-estimator sample consensus. Master's thesis, Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná, 2007. vi, 11, 12, 14, 15

[18] Hong L. and Jain A. Integrating faces and fingerprints for personal identification. *IEEE Transactions on Pattern and Machine Intelligence*, 20(12):1295–1307, dezembro 1998. 11, 13, 14, 15, 16

[19] E . C. Lopes. Detecção de faces e características faciais. Technical report, Pontifícia Universidade Católica do Rio Grande do Sul. 18

[20] A. P. Pentland M. A. Turk. Face recognition using eigenfaces. *IEEE Computer Society Confer*, 1991. 27

[21] J. H. Saito M. Arantes, A. N. Ide. A system for fingerprint minutiae classification and recognition. In *Proceedings of the 9th International Conference on Neural Information Processing(ICONIP'O2)*, volume 5, pages 2474 – 2478. viii, 11, 12, 13

[22] A. Pentland M. Turk. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 23, 28

[23] N. Ahuja M. Yang, D. J. Kriegman. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, janeiro 2002. 17, 18

[24] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst.*, 81:231–268, March 2001. 4, 5, 6

[25] D. R. Oliveira. Reconhecimento de faces usando redes neurais e biometria. Master's thesis, São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), setembro 2003. 15, 16, 17, 23

[26] D. J. Kriegman P. N. Belhumeur, J. P. Hespanha. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *European Conference on Computer Vision*, 1996. vi, 16, 23

[27] M. Jones P. Viola. Robust real-time object detection. *Second International Workshop on Statistical and Computational Theories of Vision – Modeling, Learning, Computing, and Sampling*, julho 2001. vi, 19, 20, 21, 22, 41

[28] A. Jain S. Pankanti, R. M. Bolle. Guest editors' introduction: Biometrics-the future of identification. *Computer*, 33:46–49, 2000. 1, 13

[29] A. Ali Salah, R. Morros, J. Luque, C. Segura, J. Hernando, O. Ambekar, B. Schouten, and E. Pauwels. Multimodal identification and localization of users in a smart environment. *Journal on Multimodal User Interfaces*, 2(2):75–91, setembro 2008. vii, 35, 36, 37

[30] D. Serby, E. Koller-Meier, and L. Van Gool. Probabilistic Object Tracking Using Multiple Features. *17th International Conference on Pattern Recognition (ICPR)*, 2:184–187, 2004. 4

[31] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 7

[32] R. Stiefelhagen, K. Bernardin, H. Kemal Ekenel, and M. Voit. Tracking identities and attention in smart environments - contributions and progress in the chil project. In *FG*, pages 1–8, 2008. vii, 32, 33, 34

[33] M. M. Trivedi, K. S. Huang, and I. Mikic. Dynamic context capture and distributed video arrays for intelligent spaces. *Ieee Transactions On Systems, Man, and Cybernetics—part A: Systems and Humans*, 35(1):145 – 163, janeiro 2005. vii, 37, 38, 39

[34] M. Weiser. The world is not a desktop. *Interactions*, 1:7–8, Janeiro 1994. 1

[35] M. Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3:3–11, Julho 1999. 1

[36] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13+, December 2006. vi, 3, 4, 5, 6, 7

[37] A. Yilmaz, X. Li, and M. Shah. Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536, nov 2004. 5