

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia da Computação e Sistemas Digitais

FERNANDO FUMIHIKO SHIN IKE
THIAGO BIERRENBACH CARREIRO

Autenticação de Usuários por Biometria

São Paulo
2006

**FERNANDO FUMIHIKO SHIN IKE
THIAGO BIERRENBACH CARREIRO**

Autenticação de Usuários por Biometria

Projeto de Formatura apresentado à Escola Politécnica da Universidade de São
Paulo

Área de Concentração:
Engenharia de Computação e Sistemas Digitais

Orientador: Prof^a. Dr^a. Graça Bressan

São Paulo

2006

RESUMO

O estudo do reconhecimento facial tem recebido grande atenção por parte da comunidade de computação gráfica. As pessoas detectam e identificam rostos em uma cena com pequeno ou nenhum esforço. Entretanto, construir um sistema automatizado para realizar esta tarefa introduz certas dificuldades. A necessidade por um sistema de reconhecimento facial confiável tem aumentado bastante devido a uma demanda crescente por sistemas de monitoramento e identificação criminal. O principal desafio dos softwares de reconhecimento facial é identificar ou verificar uma ou mais pessoas em um cenário usando um banco de dados de imagens de rostos. O Sistema proposto faz um controle de acesso de pessoas dentro de um cenário simulando o prédio de Engenharia Elétrica da Universidade de São Paulo, levando em conta diferentes áreas de acesso, ou seja, dentro do ambiente físico (cenário) mencionado, contendo diferentes tipos de atores (usuários) e fazendo uso do reconhecimento facial para autenticação de pessoas de acordo com políticas de segurança pré-determinadas.

ABSTRACT

The study of the face recognition has received great attention from the computer vision community. The people detect and identify faces in a scene with little or no effort. However, to construct an automatized system to make this task introduces certain difficulties. The necessity for a reliable face recognition system has increased a lot due to an increasing demand for monitoring and criminal identification systems. The main challenge of face recognition software is to identify or to verify one or more people in a scene using a data base of images of faces. The considered System makes access control of people in a scene simulating the Electric Engineering building of São Paulo University, considering different access areas, that is, inside of the physical environment (scene) mentioned, with different types of actors (users) and making use of face recognition for authentication of people in accordance with pre-determined security politics.

Lista de Figuras

Figura 1: Diagrama de casos de uso.....	14
Figura 2: Diagrama de estados do acesso à sala de aula.....	15
Figura 3: Diagrama de estados do acesso ao laboratório.....	15
Figura 4: Diagrama de estados do acesso ao portão de entrada inicial.....	16
Figura 5: Diagrama de classes.....	16
Figura 6: Diagrama de seqüência de cadastro de usuário.....	17
Figura 7: Diagrama de seqüência do acesso ao portão de entrada.....	17
Figura 8: Diagrama de seqüência do acesso à sala de aula.....	18
Figura 9: Diagrama de seqüência do acesso ao laboratório.....	18
Figura 10: Arquitetura em camadas do software.....	19
Figura 11: Arquitetura do reconhecedor facial A4Vision.....	21
Figura 12: Arquitetura lógica do sistema.....	22
Figura 13: Visão geral do funcionamento do algoritmo de reconhecimento facial baseado em eigenface.....	39
Figura 14: Arquitetura do FaceVACS: criação do conjunto de características.....	44
Figura 15: Arquitetura do FaceVACS: registro e verificação.....	44
Figura 16: Exemplo de imagem.....	45
Figura 17: Posições dos olhos encontradas pelo algoritmo.....	46
Figura 18: Depois da normalização.....	47
Figura 19: Depois do pré-processamento.....	47
Figura 20: Extração de características locais.....	48
Figura 21: Transformação global, produzindo o conjunto de características da imagem do rosto.....	48
Figura 22: Combinando centros de grupos (em vermelho) para um conjunto de referência (pontos verdes são conjuntos de características criados das imagens).....	49
Figura 23: Dados de formato ruidoso vindos do sensor.....	51

Figura 24: Imagem do sensor, embora mais aplinada que a de cima, contém dados vazios (lacunas) e outliers.....	51
Figura 25: Vistas não-frontais resultam em obstruções e buracos no formato do rosto. Esta visão do formato do rosto esconde as lacunas.....	51
Figura 26: Rotacionando o formato, as porções faltantes do formato tornam-se evidentes.....	52
Figura 27: Remoção de outlier.....	53
Figura 28: Dados originais ruidosos.....	53
Figura 29: Resultado após combinar média e filtro médio (a estrutura em malha nessas imagens é um artefato de renderização).....	53
Figura 30: Exemplo de preenchimento de lacunas.....	54
Figura 31: Diferentes vistas de um rosto obtidas do sensor.....	54
Figura 32: Vista padrão depois da normalização.....	55
Figura 33: Qualidade baixa devido aos olhos estarem fechados.....	56
Figura 34: Qualidade baixa devido à posição não-frontal.....	56
Figura 35: Distribuição das notas para a qualidade em um conjunto de 1000 fotos típicas de passaporte. A taxa de rejeição é a porcentagem de imagens cuja nota para a qualidade está abaixo de um dado limite, de modo que eles seriam rejeitados devido à baixa qualidade da amostra.....	57
Figura 36: Curvas que mostram o efeito de rejeitar o registro de imagens com baixa qualidade. Os resultados foram obtidos com a máquina B2T6 em um conjunto de 2000 fotos típicas de passaporte (1000 imagens para registro e 1000 imagens para probe).....	58
Figura 37: Eigenface.....	60
Figura 38: Como as imagens são construídas em um vetor de rostos.....	62
Figura 39: Matriz de covariância.....	64
Figura 40: Exemplo de tela de captura e verificação.....	70

Sumário

1 – Introdução.....	8
1.1 – Objetivo do Projeto.....	8
1.2 – Motivação.....	8
1.3 – Organização.....	8
2 – Especificação dos Requisitos.....	10
2.1 – Cenário para Implementação do Sistema.....	10
2.2 – Atores do Sistema.....	10
2.3 – Requisitos Funcionais (funcionalidades) do Sistema.....	10
2.4 – Requisitos Não Funcionais do Sistema.....	11
2.5 – Especificação dos Casos de Uso.....	11
2.6 – Diagrama dos Casos de Uso.....	14
2.7 – Diagrama de Estados.....	15
2.8 – Diagrama de Classes.....	16
2.9 – Diagramas de Seqüência.....	17
3 – Arquitetura.....	19
3.1 – Arquitetura em camadas do software.....	19
3.2 – Descrição do conteúdo dos pacotes.....	20
3.3 – Arquitetura lógica do sistema.....	22
4 – Visão geral do reconhecimento facial.....	23
4.1 – Análise das alternativas.....	23
4.2 – Aspectos atuais do reconhecimento facial.....	25
4.3 – Algoritmos de reconhecimento facial.....	26
4.4 – Bases de dados para testes.....	29
5 – Reconhecimento.....	37
5.1 – Reconhecimento utilizando eigenfaces.....	37
5.2 – Reconhecimento no software FaceVACS.....	43
6 – Implementação e Testes.....	60
6.1 – Implementação.....	60
6.2 – Testes dos softwares de reconhecimento disponíveis.....	69

6.3 – Exemplo de tela de captura e verificação.....	70
7 – Considerações finais.....	71

Anexos:

Anexo I – Cronograma do Projeto.....	72
--------------------------------------	----

Lista de Referências.....	73
----------------------------------	----

1 – Introdução

1.1 – Objetivo do Projeto

O objetivo do projeto consiste em fazer um controle de acesso de pessoas dentro de um cenário simulando o prédio de Engenharia Elétrica da Universidade de São Paulo, levando em conta diferentes áreas de acesso, ou seja, dentro do ambiente físico (cenário) mencionado, contendo diferentes tipos de atores (usuários) e fazendo uso da biometria (reconhecimento facial) para autenticação de pessoas de acordo com políticas de segurança pré-determinadas pelos integrantes do grupo.

O tipo de autenticação biométrica usado será o reconhecimento facial que, posteriormente poderia ser combinado com outros tipos de biometrias (comportamental, por exemplo) para aumentar a confiabilidade do sistema.

1.2 – Motivação

A motivação deste projeto consiste nos benefícios que ele traria para um cenário como o prédio de Engenharia Elétrica da Universidade de São Paulo em níveis de segurança, já que este ambiente físico não possui nenhum controle de acesso (além da tradicional fechadura) às suas dependências, que incluem o portão de entrada inicial, diversas salas de aula, diversos laboratórios de pesquisa e desenvolvimento, e que, juntos possuem valores agregados financeiros e intelectuais elevados.

Além disso, os algoritmos de reconhecimento facial existentes atualmente ainda estão em fase de aprimoramento da eficiência e dos níveis de confiabilidade, ou seja, é uma área que ainda pode se desenvolver muito para fins de aplicações de segurança.

1.3 – Organização

No capítulo 2, este documento fala da especificação dos requisitos do sistema (projeto) e de seus usuários.

No capítulo 3, o documento descreve a arquitetura do sistema, através da arquitetura de software, arquitetura de hardware (física) e a alocação dos elementos de software na arquitetura de hardware. A arquitetura de software é representada através de pacotes e as dependências entre eles.

No capítulo 4, o documento analisa as alternativas de solução existentes e que poderiam ser utilizadas no projeto, enumerando os principais algoritmos usados pelos pesquisadores, analisa os requisitos do banco de dados a ser utilizado citando alguns exemplos de banco de dados de imagens faciais utilizados por pesquisadores, como o FERET, que utiliza faces pré-cadastradas para testes de desempenho do sistema.

No capítulo 5, o documento detalha dois algoritmos de reconhecimento facial (Algoritmo baseado em “eigenfaces” e outro algoritmo usado por um software comercial) e seus processos de captura.

No capítulo 6, o documento detalha as implementações e testes de algoritmos utilizados no reconhecimento facial do sistema, a sua integração com a camada de interface e também com o banco de dados de imagens faciais.

2 – Especificação dos Requisitos

2.1 – Cenário para Implementação do Sistema

O cenário para implementação do projeto consiste de uma parte específica do prédio de Engenharia Elétrica da Escola Politécnica da USP, incluindo laboratório e sala de aula ligados ao setor de computação.

A entrada nestes lugares será limitada por dois níveis (estágios) de acesso. O primeiro nível estará localizado no portão inicial de entrada do prédio e o segundo nível estará localizado nas entradas do laboratório e sala de aula do setor de computação.

2.2 – Atores do Sistema

Os atores do sistema são:

- Alunos de Engenharia da Computação da Escola Politécnica;
- Professores de Engenharia da Computação da Escola Politécnica;
- Funcionários em geral (incluindo pesquisadores e bolsistas) do setor de computação da Escola Politécnica;
- Visitantes;
- Pessoa não cadastrada.

2.3 – Requisitos Funcionais (funcionalidades) do Sistema

Os requisitos funcionais especificados pelo grupo são:

- 1) Cadastrar pessoas de acordo com um determinado tipo de ator
- 2) Permitir o acesso de alunos de Engenharia Elétrica da Escola Politécnica à sala de aula do prédio de Engenharia Elétrica.
- 3) Bloquear o acesso de alunos de Engenharia Elétrica da Escola Politécnica ao laboratório do prédio de Engenharia Elétrica.
- 4) Permitir o acesso dos professores de Engenharia Elétrica da Escola Politécnica à sala de aula do prédio de Engenharia Elétrica.
- 5) Permitir o acesso dos professores de Engenharia Elétrica da Escola Politécnica ao laboratório do prédio de Engenharia Elétrica.

- 6) Permitir o acesso de funcionários do prédio de Engenharia Elétrica ao laboratório do prédio de Engenharia Elétrica.
- 7) Bloquear o acesso de funcionários do prédio de Engenharia Elétrica à sala de aula do prédio de Engenharia Elétrica.
- 8) Bloquear o acesso de visitantes ao laboratório do prédio de Engenharia Elétrica.
- 9) Bloquear o acesso de visitantes à sala de aula do prédio de Engenharia Elétrica.
- 10) Permitir o acesso de pessoas cadastradas no portão inicial de entrada
- 11) Bloquear o acesso de pessoas não cadastradas no portão inicial de entrada

2.4 – Requisitos Não Funcionais do Sistema

Os requisitos não-funcionais especificados pelo grupo são:

1) Confiabilidade

Caso algum equipamento do sistema falhar, o sistema deve continuar funcionando normalmente.

2) Segurança

Caso uma parte do sistema falhar, o sistema deve ir para um estado considerado “seguro”.

3) Expansibilidade

O sistema deve ser flexível ou expansível a mais tipos de atores, áreas de acesso e políticas de segurança.

4) Disponibilidade

O sistema deve estar funcionando a maior parte do tempo possível para garantir o requisito de disponibilidade (aumentar a probabilidade de estar funcionando num determinado instante de tempo).

2.5 – Especificação dos Casos de Uso

Caso de uso 1: Cadastro de novo usuário no sistema

Descrição: Este caso de uso descreve o cadastro de um novo usuário no sistema.

Evento iniciador: Solicitação de cadastro de usuário.

Atores: Funcionário do portão de entrada inicial (porteiro).

Pré-condição: Sistema (terminal próximo ao portão de entrada inicial) aguardando entrada de dados de novo usuário.

Seqüência de eventos:

1. Funcionário solicita operação de cadastro de usuário.
2. Sistema solicita os dados relativos ao novo usuário.
3. Funcionário insere os dados (nome, tipo e imagem do rosto) do novo usuário.
4. Sistema insere o novo usuário e exibe o resultado.
5. Funcionário finaliza a operação.

Pós-condição: novo usuário (ator) cadastrado

Extensões:

1. Usuário a ser cadastrado já existe: sistema apresenta mensagem ao funcionário (passo 4).

Inclusão: busca usuário (passo 4)

Caso de uso 2: Acessar a Sala de Aula

Descrição: Este caso de uso descreve o controle de acesso de um ator qualquer cadastrado na entrada da sala de aula.

Evento iniciador: Entrada de dados no sistema por ator cadastrado.

Atores: Qualquer ator cadastrado no sistema

Pré-condição: Sistema aguardando entrada de dados de usuário.

Seqüência de eventos:

1. Ator cadastrado solicita acesso à sala de aula
2. Sistema adquire dados pessoais (biométricos) do ator e libera ou bloqueia o acesso à sala de aula de acordo com o tipo de ator.

Pós-condição: acesso à sala de aula liberado ou bloqueado ao ator cadastrado.

Extensões:

Inclusão:

1. busca dados pessoais (biométricos) na base de dados dos usuários com acesso à sala de aula. (passo 2)
-

Caso de uso 3: Acessar o Laboratório

Descrição: Este caso de uso descreve o controle de acesso de um ator cadastrado à entrada do laboratório.

Evento iniciador: Entrada de dados no sistema por um ator cadastrado.

Atores: Qualquer ator cadastrado no sistema

Pré-condição: Sistema aguardando entrada de dados de usuário.

Seqüência de eventos:

1. Ator cadastrado solicita acesso ao laboratório
2. Sistema adquire dados pessoais (biométricos) do ator e libera ou bloqueia o acesso ao laboratório de acordo com o tipo do ator.

Pós-condição: acesso ao laboratório liberado ou bloqueado ao ator cadastrado.

Extensões:**Inclusão:**

1. busca dados pessoais (biométricos) na base de dados dos usuários com acesso ao laboratório. (passo 2)
-

Caso de uso 4: Acessar o Portão de Entrada

Descrição: Este caso de uso descreve o controle de acesso de uma pessoa qualquer ao portão de entrada inicial.

Evento iniciador: Entrada de dados no sistema por uma pessoa qualquer.

Atores: Pessoa qualquer

Pré-condição: Sistema aguardando entrada de dados de usuário.

Seqüência de eventos:

1. Pessoa solicita acesso ao portão de entrada.

2. Sistema adquire dados pessoais (biométricos) da pessoa e libera ou bloqueia o acesso ao portão de entrada, caso ela estiver cadastrada ou não no sistema.

Pós-condição: acesso ao portão de entrada liberado ou bloqueado à pessoa.

Extensões:

Inclusão:

1. busca dados pessoais (biométricos) na base de dados dos usuários cadastrados.
(passo 2)
-

2.6 – Diagrama de Casos de Uso

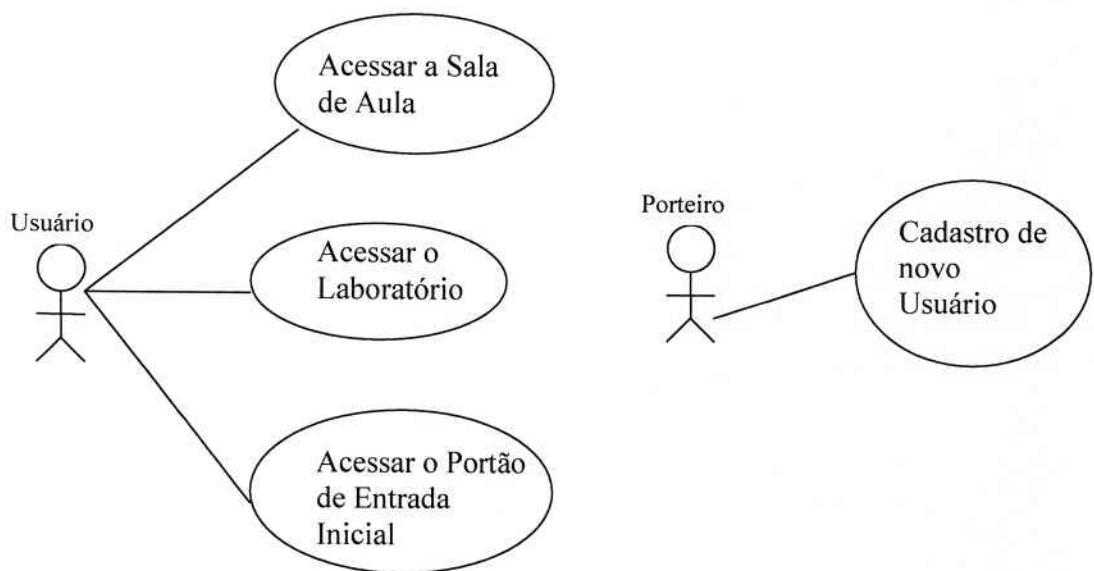


Figura 1: Diagrama de casos de uso

2.7 – Diagrama de Estados

Acesso à Sala de Aula

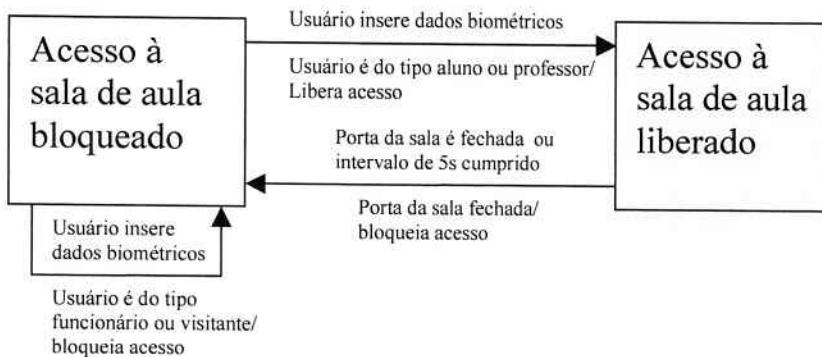


Figura 2: Diagrama de estados do acesso à sala de aula

Acesso ao Laboratório

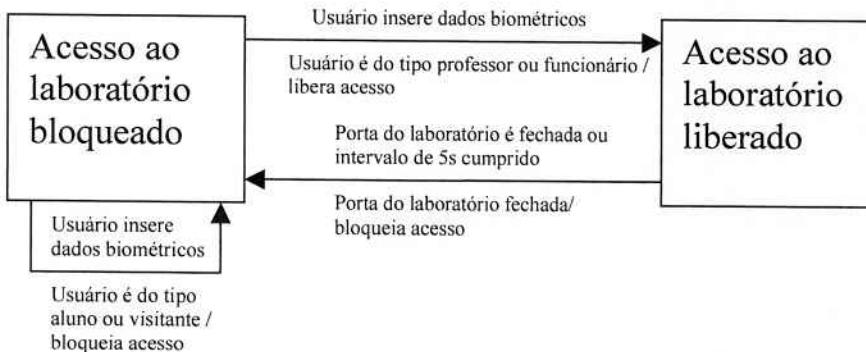


Figura 3: Diagrama de estados do acesso ao laboratório

Acesso ao Portão de Entrada Inicial

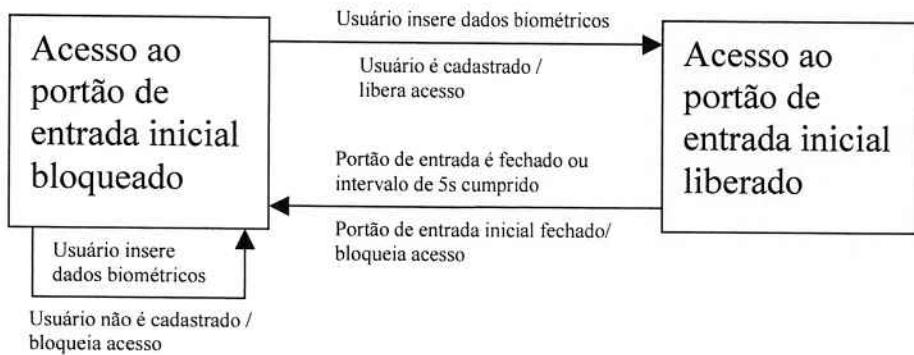


Figura 4: Diagrama de estados do acesso ao portão de entrada inicial

2.8 – Diagrama de Classes

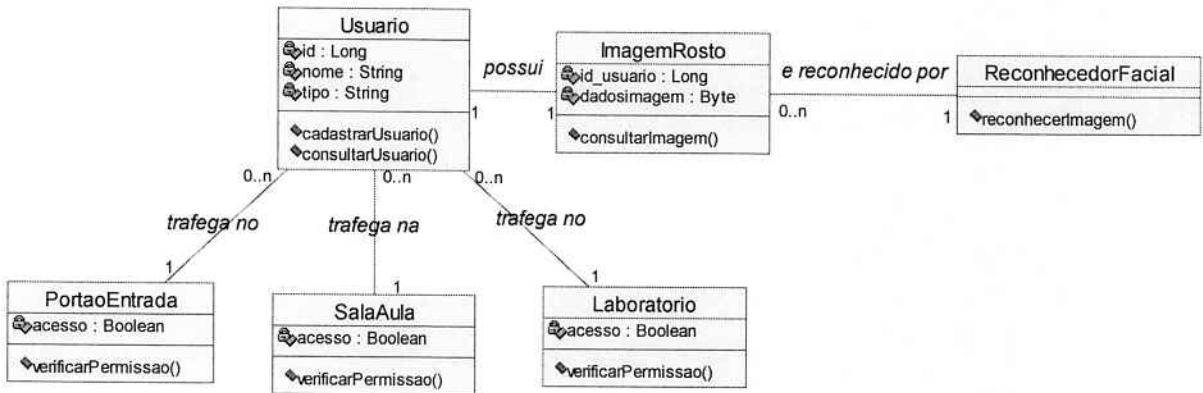


Figura 5: Diagrama de classes

2.9 – Diagramas de Seqüência

Diagrama de Seqüência de Cadastro de Usuário

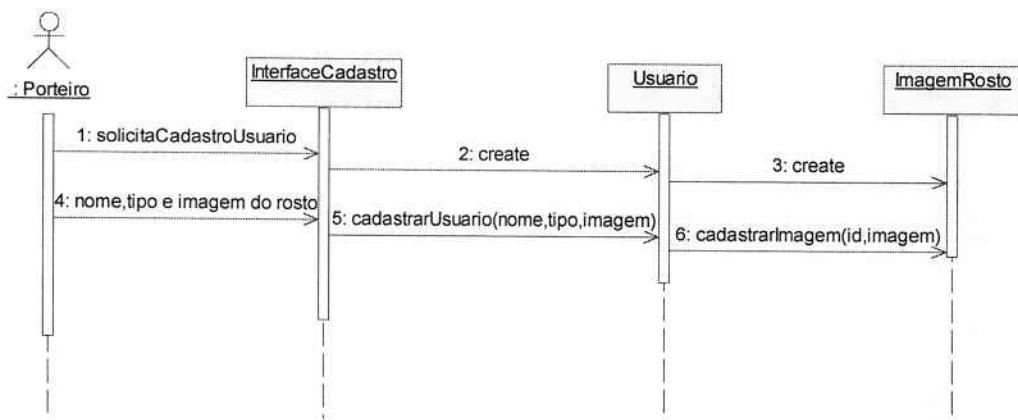


Figura 6: Diagrama de seqüência de cadastro de usuário

Diagrama de Seqüência de Acesso ao Portão de Entrada

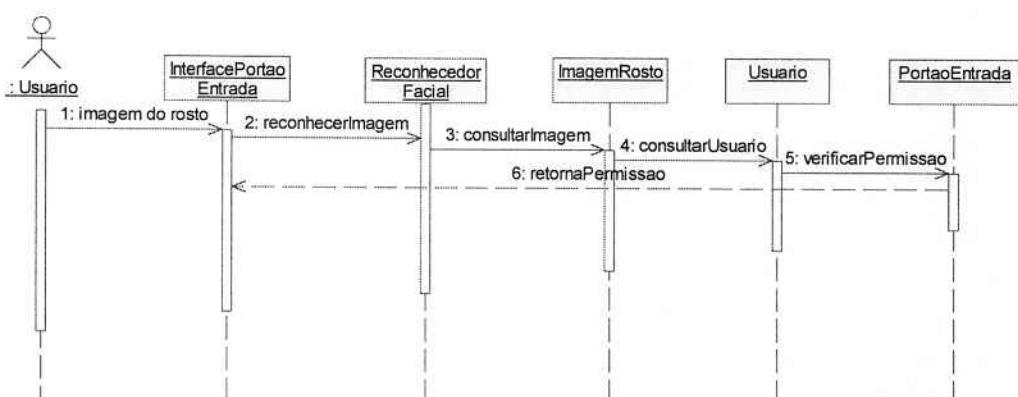


Figura 7: Diagrama de seqüência do acesso ao portão de entrada

Diagrama de Seqüência de Acesso à Sala de Aula

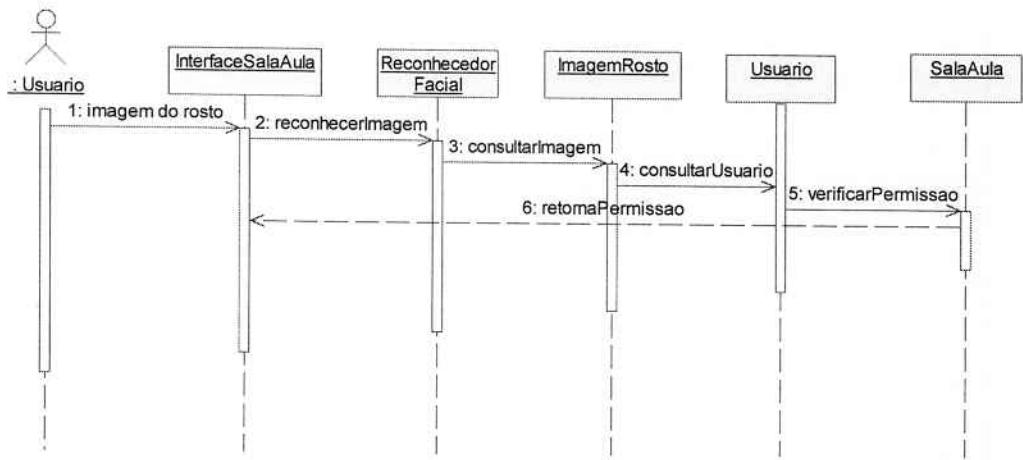


Figura 8: Diagrama de seqüência do acesso à sala de aula

Diagrama de Seqüência de Acesso ao Laboratório

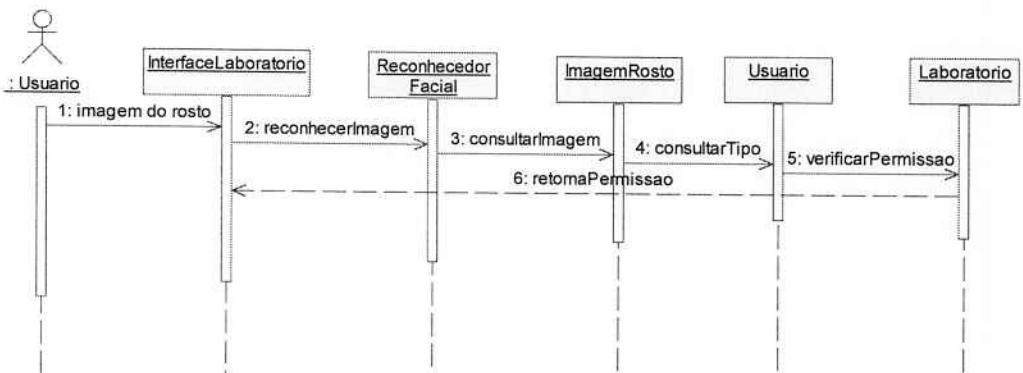


Figura 9: Diagrama de seqüência do acesso ao laboratório

3 – Arquitetura

3.1 – Arquitetura em camadas do software

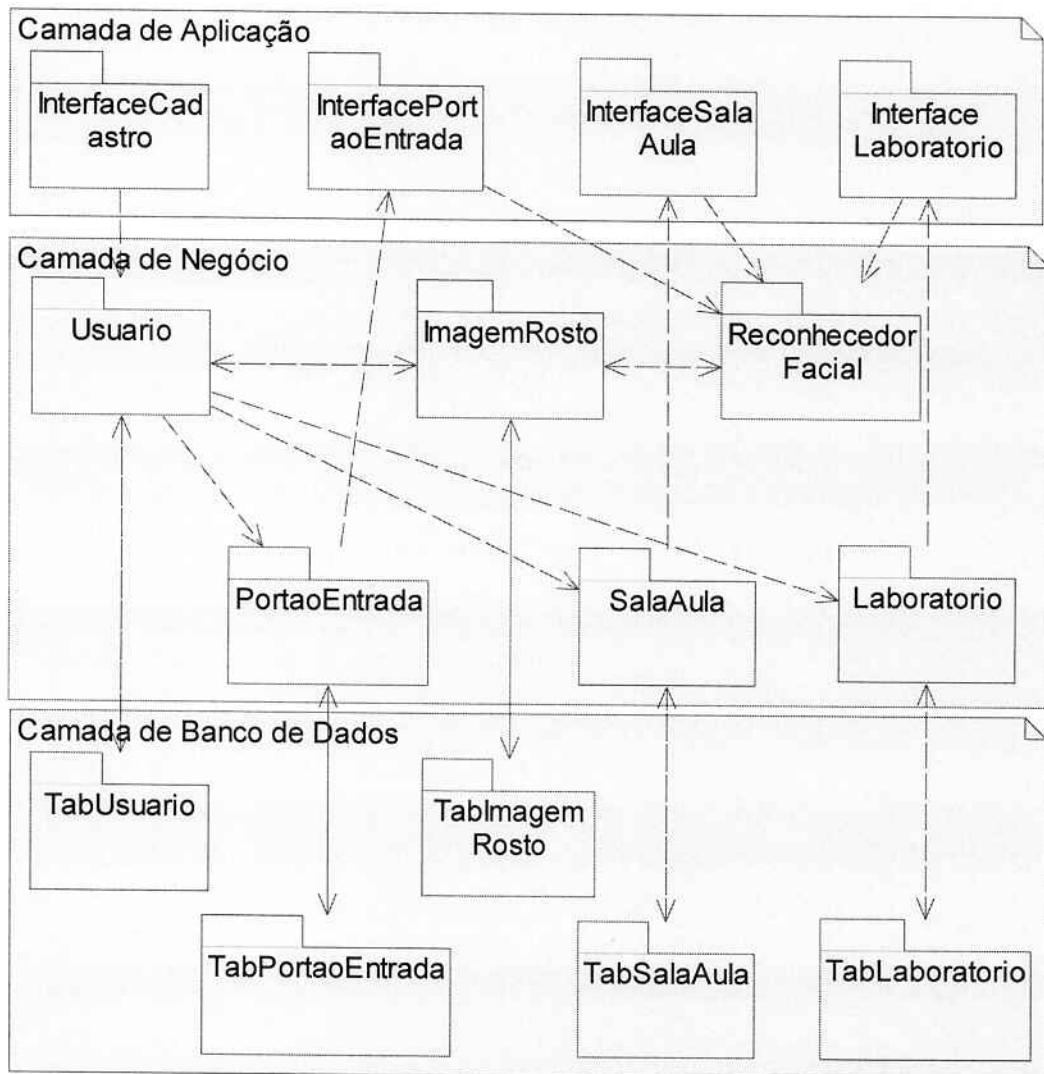


Figura 10: Arquitetura em camadas do software

3.2 – Descrição do conteúdo dos pacotes

Camada de Aplicação

- **InterfaceCadastro:** Pacote relacionado com a interação do funcionário da portaria com o formulário de cadastro de novo usuário do sistema;
- **InterfacePortaoEntrada:** Pacote relacionado com a interação de um ator qualquer com o sistema no portão de entrada inicial do prédio de Engenharia Elétrica (captura de imagem do rosto de um ator qualquer no portão de entrada);
- **InterfaceSalaAula:** Pacote relacionado com a interação de um usuário cadastrado com o sistema na entrada de uma sala de aula do prédio de Engenharia Elétrica (captura de imagem do rosto de um usuário na entrada da sala de aula);
- **InterfaceLaboratorio:** Pacote relacionado com a interação de um usuário cadastrado com o sistema na entrada de um laboratório do prédio de Engenharia Elétrica (captura de imagem do rosto de um usuário na entrada do laboratório).

Camada de Negócio

- **Usuario:** Pacote relacionado à classe de negócio (entidade) de usuários cadastrados no sistema;
- **ImagenRosto:** Pacote relacionado à classe de negócio (entidade) de imagens de rosto cadastradas no sistema;
- **PortaoEntrada:** Pacote relacionado à classe de negócio (entidade) de permissões de acesso de usuários no portão de entrada inicial do prédio;
- **SalaAula:** Pacote relacionado à classe de negócio (entidade) de permissões de acesso de usuários na entrada da sala de aula do prédio;
- **Laboratorio:** Pacote relacionado à classe de negócio (entidade) de permissões de acesso de usuários na entrada do laboratório do prédio;
- **ReconhecedorFacial:** Pacote contendo o algoritmo de reconhecimento facial e que utilizará um software livre para processar os dados de imagem vindos da camada de aplicação, acessar a tabela de imagens cadastradas no sistema na camada de banco de dados e conferir se a imagem de entrada está cadastrada e a qual usuário

pertence. Um exemplo de software que pode ser utilizado é o **A4Vision**, cujo diagrama de hardware e de blocos (que são proprietários) podem ser vistos a seguir:

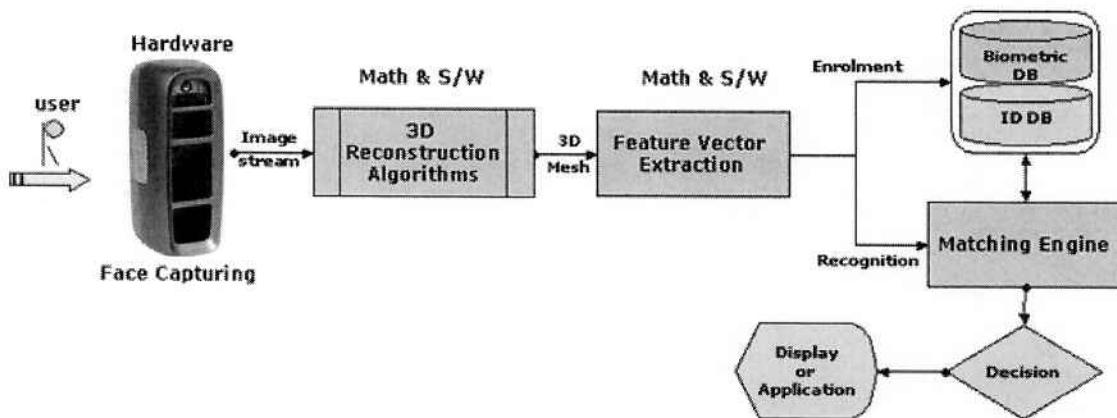


Figura 11: Arquitetura do reconhecedor facial A4Vision [1]

Camada de Banco de Dados

- **TabUsuario:** Tabela do banco de dados de usuários cadastrados no sistema;
- **TabImagenRosto:** Tabela do banco de dados de imagens de rosto cadastradas no sistema;
- **TabPortaoEntrada:** Tabela do banco de dados de permissões de acesso de usuários no portão de entrada inicial do prédio;
- **TabSalaAula:** Tabela do banco de dados de permissões de acesso de usuários na entrada da sala de aula do prédio;
- **TabLaboratorio:** Tabela do banco de dados de permissões de acesso de usuários na entrada do laboratório do prédio.

3.3 – Arquitetura lógica do sistema

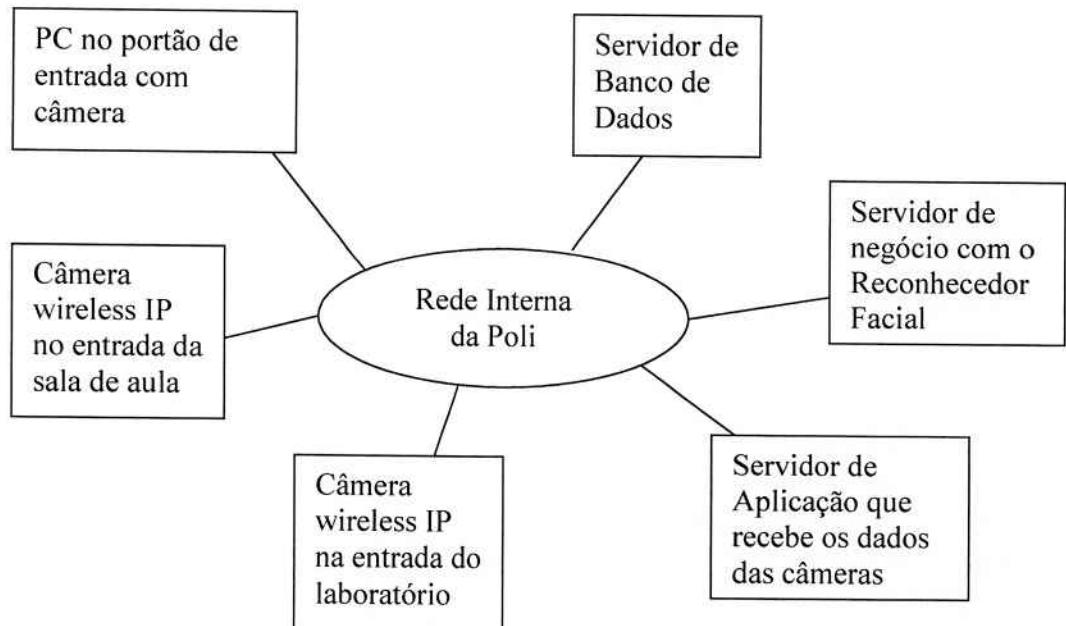


Figura 12: Arquitetura lógica do sistema

4 – Visão Geral do Reconhecimento Facial

4.1 – Análise das alternativas

Para o desenvolvimento do projeto será necessário escolher uma tecnologia que atenda a algumas restrições e necessidades.

As escolhas a serem feitas são basicamente de qual banco de dados a ser utilizado e em qual linguagem serão programadas as camadas de negócio e de aplicação.

Para a escolha da linguagem de programação deve levar-se em consideração aspectos como: familiaridade dos membros do grupo com a linguagem, nível e diferença de complexidade de se resolver os problemas do projeto nas diferentes linguagens e, principalmente, se existe uma API/biblioteca de reconhecimento facial disponível para uso integrado com a linguagem em questão.

A partir do conhecimento da equipe e do suporte da linguagem às necessidades do projeto, concluímos que tanto C++/C# quanto Java são linguagens que nos serviriam bem, contanto que consigamos uma API/biblioteca para essas linguagens que nos de suporte ao reconhecimento facial.

Estamos estudando e pesquisando estas tecnologias para definir qual delas nos atende melhor, porém já foi possível reconhecer alguns pontos positivos e negativos em cada uma delas.

O que pode ser citado como vantagens da linguagem Java é o fato de os membros do grupo estarem mais familiarizados com a mesma em relação a C++/C#, e que através do *Java Media Framework* (JMF) pareceu simples a tarefa de se capturar a imagem de uma webcam para uma aplicação.

A principal vantagem do C++/C# até o momento é o fato de APIs/bibliotecas de reconhecimento facial estarem disponíveis, em sua maioria, para uso com essa linguagem.

O principal ponto negativo da linguagem Java até o momento é a quantidade limitada, em relação à C++/C#, de APIs/bibliotecas para reconhecimento facial.

No caso de C++/C#, as desvantagens são a falta de familiaridade da equipe com a linguagem, além de se mostrar consideravelmente mais complexo capturar a imagem de uma *Webcam* em relação à linguagem Java.

As pesquisas de softwares de reconhecimento facial buscaram aqueles de licença gratuita devido a questões de custo e tempo para possibilitar testes de diferentes implementações de reconhecedores faciais e poder compará-los entre si em relação às funcionalidades oferecidas e requisitos não-funcionais como confiabilidade e desempenho.

Entre as diversas implementações encontradas na pesquisa, o grupo optou por selecionar aquelas com suporte à plataforma Windows devido a uma maior familiaridade e facilidade de desenvolvimento pelos integrantes. Os softwares testados foram:

Open Biometry

O software consiste de uma biblioteca de reconhecimento facial e algoritmos de suporte baseado num método chamado “eigenface” e é implementado na linguagem C. Em uma versão pré-alfa, o software oferece um pré-processador, que exigiu uma biblioteca chamada *Devil*, para possibilitar todo o processamento, carregamento e manipulação de imagens, e um detector facial que usa as funcionalidades de um software da Intel chamado *OpenCv's Haar*.

FaceVACS

O FaceVACS consiste de um SDK (Software Development Kit) comercial da empresa Cognitec que implementa o módulo de reconhecimento facial.

Para que isso fosse possível, foi utilizada uma licença temporária fornecida pela empresa para fins de pesquisa e desenvolvimento.

O FaceVACS-SDK é uma ferramenta de desenvolvimento que fornece o algoritmo nativo do reconhecimento facial para integradores de sistemas, desenvolvedores e eventuais clientes interessados. Este SDK inclui todas as funcionalidades básicas de enrollment, verificação e identificação para fotos e vídeo ao vivo.

As interfaces estão disponíveis em diversas linguagens de programação e para os sistemas operacionais mais comumente utilizados, sem requerimentos obrigatórios, como banco de dados específicos, ou ambientes de programação.

O conjunto de ferramentas e documentação permite um desenvolvimento eficiente, e caso seja necessária a reconstrução de alguma característica da aplicação, será necessário apenas um pequeno conjunto de bibliotecas binárias.

4.2 – Aspectos atuais do reconhecimento facial

O estudo do reconhecimento facial tem recebido grande atenção por parte da comunidade de computação gráfica. As pessoas detectam e identificam rostos em uma cena com pequeno ou nenhum esforço. Entretanto, construir um sistema automatizado para realizar esta tarefa introduz certas dificuldades.

A necessidade por um sistema de reconhecimento facial confiável tem aumentado bastante devido a uma demanda crescente por sistemas de monitoramento e identificação criminal. Além disso, há muitos investimentos em infra-estrutura de segurança por parte do governo e de corporações da justiça. Ao contrário de outras técnicas biométricas como reconhecimento por íris ou por impressão digital, o reconhecimento facial é capaz de atuar em tempo real, não necessitando de cooperação ou aviso por parte do indivíduo.

Portanto, o reconhecimento facial é adequado para checagens antiintrusivas em locais públicos de alto risco como aeroportos, assim como a monitoração e identificação passivas de criminosos e terroristas conhecidos.

O principal desafio dos softwares de reconhecimento facial é identificar ou verificar uma ou mais pessoas em um cenário usando um banco de dados de imagens de rostos.

Os principais assuntos relacionados à pesquisa na área são:

- Reconhecimento de imagens ao ar livre;
- Reconhecimento de imagens faciais não frontais;
- Reconhecimento com baixas taxas de alarme (ou aceitação) falsas;
- Explicações sobre o reconhecimento mais fácil de homens em relação a mulheres;
- Maior entendimento dos efeitos de fatores demográficos no desempenho;
- Desenvolvimento de melhores métodos estatísticos para compreender o desempenho dos sistemas;
- Desenvolver melhores modelos para prever o desempenho das identificações faciais em grandes galerias de imagens;
- Efeito do treinamento do sistema e dos algoritmos com desempenhos variados;
- Integração de modelos morfológicos no desempenho do reconhecimento facial.

4.3 – Algoritmos de reconhecimento facial

Algoritmos de reconhecimento facial baseado em imagem

- PCA (*Principal Component Analysis*)

Dada uma representação de um vetor de dimensão s de cada face em um conjunto de imagens para treinamento, o PCA tenta encontrar um subespaço de dimensão t , cujos vetores base correspondem à máxima variação de direção do espaço original da imagem. Este novo subespaço é normalmente de dimensão menor ($t << s$). Se os elementos da imagem são considerados como variáveis aleatórias, os vetores base do PCA são definidos como autovetores da matriz de dispersão.

- ICA (*Independent Component Analysis*)

O ICA minimiza tanto as dependências de segunda ou maior ordem nos dados de entrada e tenta encontrar a base, nos quais os dados são estaticamente independentes.

- LDA (*Linear Discriminant Analysis*)

O LDA encontra os vetores no espaço base que melhor se diferenciam entre as classes. Para todas as amostras de todas as classes a matriz de dispersão de classe intermediária S_B e a matriz de dispersão de classe interna S_W são definidas. O objetivo é maximizar S_B enquanto minimiza-se S_W . A relação é maximizada quando os vetores coluna da matriz de projeção são os autovetores de $(S_W^{-1} \times S_B)$.

- EP (*Evolutionary Pursuit*)

O EP busca o melhor conjunto de projeções dos eixos para maximizar uma função de adaptação, medindo simultaneamente a precisão de classificação e a capacidade de generalização do sistema. Devido à dimensão do espaço de soluções deste problema ser muito grande, ele é resolvido usando um tipo específico de algoritmo genético chamado Busca Evolucionária.

- EBGM (*Elastic Bunch Graph Matching*)

Todas as faces humanas compartilham de uma estrutura topológica similar. As faces são representadas como gráficos, com os nós posicionados em pontos fiduciais e em bordas rotuladas com os vetores em duas dimensões da distância. Cada nó contém um conjunto de 40 coeficientes complexos em escalas diferentes e em orientações

(fase, amplitude). São chamados “jatos”. O reconhecimento é baseado em gráficos rotulados. Um gráfico rotulado é um conjunto dos nós conectados por arestas, nós são rotulados com jatos, arestas são rotuladas com distâncias.

- *Kernel Methods*

O distribuidor da face no subespaço não necessita ser linear. Os métodos do *kernel* são uma generalização de métodos lineares. Os esquemas múltiplos não lineares diretos são explorados para aprender este distribuidor não linear.

- *Trace Transform*

O *Trace Transform*, uma generalização do *Radom Transform*, é uma ferramenta nova para processar uma imagem e que pode ser usado reconhecendo objetos sob transformações, por exemplo, rotação, translação e escalar. Para produzir o *Trace Transform* é computada uma funcionalidade ao longo das linhas de traçado de uma imagem. *Trace Transforms* diferentes podem ser produzidos de uma imagem usando funcionalidades diferentes do traçado.

- AAM

Um modelo ativo da aparência (AAM) é um modelo estatístico integrado que combine um modelo da variação da forma com um modelo das variações da aparência em um frame com formas normalizadas. Um AAM contém um modelo estatístico se a forma e o nível de obscuridade do objeto de interesse pode ser generalizado a quase todo o exemplo válido. Combinar a uma imagem envolve encontrar os parâmetros do modelo que minimizam a diferença entre a imagem e um exemplo de modelo sintetizado projetados na imagem.

- *3-D Morphable Model*

A face humana é uma superfície que se encontra no espaço em três dimensões intrinsecamente. Conseqüentemente o modelo em três dimensões deve ser melhor para representar as faces, especialmente para assegurar variações faciais, tais como a pose e a iluminação. Alguns pesquisadores propuseram um método baseado em um modelo morfológico em três dimensões da face que codificasse a forma e a textura baseadas nos parâmetros modelo, e no algoritmo que recupera estes parâmetros de uma única imagem de uma face.

- *3-D Face Recognition*

A principal novidade desta aproximação é a habilidade de comparar as superfícies independentes de deformações naturais resultantes das expressões faciais. Primeiramente, a escala da imagem e a textura da face são adquiridas. Em seguida, a escala da imagem é pré-processada removendo determinadas partes tais como o cabelo, que pode complicar o processo do reconhecimento. Finalmente, um formulário canônico da superfície facial é computado. Tal representação é insensível às orientações da cabeça e às expressões faciais, assim significativamente simplificando o procedimento de reconhecimento. O reconhecimento propriamente dito é executado nas superfícies canônicas.

- *Bayesian Framework*

Uma medida probabilística da similaridade baseada na opinião *Bayesiana* de que as diferenças da intensidade da imagem são características de variações típicas na aparência de um indivíduo. Duas classes de variações faciais da imagem são definidas: variações internas e variações externas. A similaridade entre as faces são medidas usando as regras *Bayesianas*.

- SVM

Dado um conjunto de pontos que pertencem a duas classes, uma máquina vetorial de sustentação (SVM) encontra o plano superior que separa o maior número possível frações de pontos da mesma classe no mesmo lado, maximizando a distância de uma ou outra classe ao plano superior. PCA é usado primeiramente para extrair características de imagens da face e as funções de discriminação entre cada par de imagens são aprendidas então por SVMs.

- HMM

Os modelos ocultados de *Markov* (HMM) são um conjunto de modelos estatísticos usados para caracterizar as propriedades estatísticas de um sinal. HMM consiste em dois processos relacionados: uma cadeia subjacente, medidas não observáveis de *Markov* com um número finito dos estados, uma matriz da probabilidade da transição do estado e uma distribuição da probabilidade do estado inicial e um conjunto de funções da densidade da probabilidade associados com cada estado.

- *Boosting & Ensemble Solutions*

A idéia por trás do algoritmo é empregar seqüencialmente um aprendizado fraco em uma versão tornada mais pesada dado um conjunto de amostras de treinamento para generalizar um conjunto de classificadores de seu tipo. Embora qualquer classificador individual possa executar ligeiramente melhor do que o aleatório, o grupo formado pode fornecer um classificador muito exato (forte). Dois pesquisadores (Viola e Jones) construíram o primeiro sistema em tempo real da detecção de face usando *AdaBoost*, que é considerado uma descoberta dramática na pesquisa da detecção de face.

Algoritmo de reconhecimento facial baseado em vídeo

Durante os últimos anos mais e mais pesquisas foram feitas na área de reconhecimento da face de imagens seqüenciais. Reconhecer seres humanos de vídeos reais é difícil por causa da qualidade baixa das imagens e porque as imagens da face são pequenas. Ainda assim, muitas melhorias foram feitas nessa área.

4.4 – Bases de dados para testes

Ao testar um algoritmo, é recomendável usar uma série de dados padronizados de teste para que os pesquisadores possam comparar diretamente os resultados. Enquanto houver muitas bases de dados em uso, a escolha de uma base de dados apropriada a ser usada deve ser feita baseado na tarefa dada (envelhecimento, expressões, iluminação). Uma outra maneira é escolher uma série de dados específicos à propriedade a ser testada (por exemplo, como o algoritmo se comporta quando determinadas imagens com iluminação mudam ou imagens com expressões faciais diferentes).

Alguns dos exemplos de bases de dados de face usadas freqüentemente por pesquisadores são:

The Color FERET Database

O programa de FERET se ajustou para estabelecer uma base de dados grande de imagens facial que fosse recolhida independentemente dos colaboradores do algoritmo. O Dr. Harry Wechsler da universidade George Mason foi selecionado para dirigir a coleção desta base de dados. A coleção da base de dados era um esforço colaborativo entre o Dr.

Wechsler e o Dr. Phillips. As imagens foram coletadas em um ambiente controlado parcialmente. Para manter um grau de consistência durante toda a base de dados, a mesma instalação física foi usada em cada sessão da fotografia. Porque o equipamento teve que ser remontado para cada sessão, havia alguma variação menor nas imagens coletadas em datas diferentes. A base de dados de FERET foi coletada em 15 sessões entre agosto 1993 e julho 1996. A base de dados contém 1564 conjuntos das imagens para um total de 14.126 imagens que incluem 1199 indivíduos e 365 conjuntos duplicados das imagens. O conjunto duplicado é um segundo conjunto de imagens de uma pessoa já existente na base de dados e feitas geralmente em um dia diferente. Para alguns indivíduos, dois anos se passaram entre a realização das primeiras e últimas imagens, sendo que alguns indivíduos foram fotografados múltiplas vezes. Este lapso de tempo é importante porque permite aos pesquisadores estudarem, pela primeira vez, as mudanças na aparência de um indivíduo que ocorrem ao longo de um ano.

The Yale Face Database

A base de dados de faces de Yale contém 165 imagens em um formato GIF de 15 indivíduos. Há 11 imagens por pessoa, um por expressão facial diferente ou por configuração: iluminação de centro, com óculos, feliz, iluminação à esquerda, sem óculos, normal, iluminação à direita, triste, com sono, surpreso e piscando.

The Yale Face Database B

Contém 5760 imagens com fontes de luz simples de 10 pessoas cada uma vista sob 576 condições de observação (9 poses x 64 condições da iluminação). Para cada pessoa em uma pose particular, uma imagem com iluminação ambiental (de fundo) foi capturada também.

PIE Database, CMU

Uma base de dados de 41.368 imagens de 68 pessoas, cada pessoa sob 13 poses diferentes, 43 condições diferentes da iluminação, e com 4 expressões diferentes.

Project – Face In Action (FIA) Face Video Database, AMP, CMU

O cenário de capturação imita as aplicações do mundo real, por exemplo, quando uma pessoa está atravessando o ponto de checagens do aeroporto. Seis câmeras capturam os rostos humanos de três ângulos diferentes. Três fora das seis câmeras têm o comprimento menor do foco, e os outros três têm o comprimento maior do foco. Planear capturar 200 assuntos em 3 sessões em períodos de tempo diferentes. Para uma sessão, tanto interna quanto externa, o cenário será capturado. Poses dependentes de usuário e variações da expressão esperam-se das seqüências de vídeo.

AT&T “The Database of Faces” (formerly “The ORL Database of Faces”)

Dez imagens diferentes de cada uma das 40 pessoas diferentes. Para algumas pessoas, as imagens foram tiradas em instantes diferentes, variando a iluminação, as expressões faciais (olhos abertos/fechados, sorrindo/não sorrindo) e os detalhes faciais (com/sem óculos). Todas as imagens foram tiradas contra um fundo homogêneo escuro com as pessoas em uma posição ereta, frontal (com tolerância para algum movimento lateral).

Cohn-Kanade AU Coded Facial Expression Database

As pessoas na parte livre da base de dados Facial AU-Codificada Cohn-Kanade Expression são 100 estudantes da universidade, com idade entre 18 a 30 anos. Sessenta e cinto por cento eram mulheres, 15 por cento eram Afro-Americanos, e três por cento eram Asiáticos ou Latinos. As pessoas foram instruídas por um experimentador a executar uma série de 23 exposições facial que incluíram unidades da ação simples e combinações de unidades da ação. As seqüências de imagem do neutro à exposição do alvo foram digitadas em 640 por 480 ou 490 disposições do pixel com precisão de 8 bits para valores de escala entre o preto e o branco. Incluídos com os arquivos da imagem estão a “seqüência” de arquivos; estes são pequenos arquivos texto que descrevem a ordem em que as imagens deveriam ser lidas.

MIT-CBCL Face Recognition Database

A base de dados MIT-CBCL Face Recognition contém imagens do rosto de 10 pessoas. Fornece dois conjuntos de treinamento: 1. Imagens de alta resolução, incluindo a vista frontal, de meio-perfil e de perfil; 2. Imagens sintetizadas (324/pessoa) renderizadas dos modelos 3D da cabeça das 10 pessoas. Os modelos de cabeça foram gerados ajustando um modelo morfológico às imagens de alta resolução do treinamento. Os modelos 3D não são incluídos na base de dados. O conjunto de testes consiste de 200 imagens por pessoa. Nós variamos a iluminação, a pose (até aproximadamente 30 graus de rotação em profundidade) e o fundo.

Image Database of Facial Actions and Expressions – Expression Image Database

24 pessoas são representadas nesta base de dados, gerando aproximadamente 6 a 18 exemplos das 150 requisições de ações diferentes. Assim, aproximadamente 7.000 imagens coloridas são incluídas na base de dados, e cada uma tem uma imagem dentro da escala entre o preto e o branco usada na análise de rede neural.

Face Recognition Data, University of Essex, UK

395 indivíduos (entre homens e mulheres), 20 imagens por indivíduo. Contém imagens das pessoas de várias origens raciais, principalmente de estudantes do primeiro ano de graduação, assim a maioria dos indivíduos está entre 18 e 20 anos de idade, mas alguns indivíduos mais velhos também estão presentes. Alguns indivíduos estão usando óculos e barbas.

NIST Mugshot Identification Database

Há imagens de 1.573 indivíduos (casos), 1495 homens e 78 mulheres. A base de dados contém vistas frontais e de lado (perfil) quando disponíveis. Separando vistas frontais e de perfil, há 131 casos com duas ou mais vistas frontais e 1418 com somente uma vista frontal. Os perfis têm 89 casos com dois ou mais perfis e 1268 com somente um perfil. Os casos com ambas as partes frontais e de perfil têm 89 casos com dois ou mais de

ambas as partes frontais e de perfis, 27 com dois ou mais partes frontais e um perfil, e 1217 com somente um frontal e um perfil.

NLPR Face Database

450 imagens de rosto. 896 x 592 pixels. Formato JPEG. 27 ou apenas pessoas com iluminação/expressões/fundos diferentes.

M2VTS Multimodal Face Database (Release 1.00)

A base de dados é composta de 37 rostos diferentes e fornece 5 fotos para cada pessoa. Estas fotos foram tiradas em intervalos de uma semana ou quando as mudanças drásticas do rosto ocorreram no período. Durante cada disparo de foto, as pessoas foram solicitadas para contar de “0” a “9” em sua língua nativa (a maioria das pessoas são franceses falando), giram a cabeça de 0 a -90 graus, retornando a 0, então a +90 e de volta novamente a 0 graus. Além disso, solicitou-se às pessoas para girar mais uma vez a cabeça sem óculos se estivessem usando algum.

The Extended M2VTS Database, University of Surrey, UK

Contém quatro imagens de 295 pessoas tiradas sobre um período de quatro meses. Cada imagem contém uma foto da cabeça conversando e uma foto da cabeça girando. Os conjuntos de dados obtidos desta base de dados estão disponíveis incluindo imagens coloridas de alta qualidade, arquivos de som de 32 KHz 16 bits, seqüências de vídeo e um modelo 3D.

The AR Face Database, Purdue University, USA

4.000 imagens coloridas que correspondem a 126 rostos de pessoas (70 homens e 56 mulheres). As imagens caracterizam as vistas frontais dos rostos com diferentes expressões faciais, condições de iluminação e oclusões (óculos de sol e manta).

The University of Oulu Physics-Based Face Database

Contém 125 rostos diferentes cada um em 16 diferentes calibrações da câmera e condições de iluminação, mais 16 se a pessoa estiver usando óculos. Rostos nas posições frontais capturados sob o horizonte, sob luz incandescente, sob luz fluorescente e à luz do

dia. Inclui 3 refletâncias espectrais de pele por pessoa medidos tanto da bochecha quanto da testa. Contém a resposta espectral RGB da câmera usada e a distribuição da energia espectral de iluminação.

CAS-PEAL Face Database

A base de dados CAS-PEAL Face foi construída sob os patrocinadores do programa Nacional Hi-Tech e ISVISION. Os objetivos da criação da base de dados do PEAL Face incluem: fornecer aos pesquisadores de todo o mundo da comunidade do FR uma base de dados facial chinesa em grande escala para o treinamento e validação de seus algoritmos; facilitar o desenvolvimento do FR fornecendo imagens faciais em grande escala com diferentes fontes de variações, especialmente da pose, da expressão, dos acessórios, e da iluminação (PEAL); avanço das tecnologias de reconhecimento facial que visam aplicações práticas especialmente a oriental.

Japanese Female Facial Expression (JAFFE) Database

A base de dados contém 213 imagens de 7 expressões faciais (6 expressões faciais básicas + 1 neutra) posadas por 10 modelos japonesas. Cada imagem foi avaliada em 6 adjetivos emotivos por 60 pessoas japonesas.

BioID Face DB – HumanScan AG, Switzerland

O conjunto de dados consiste em 1521 imagens do nível preto ao branco com uma resolução de 384x286 pixels. Cada um mostra a vista frontal de um rosto de uma das 23 pessoas diferentes do teste. Por razões da comparação o conjunto contém também posições ajustadas manualmente do olho.

Psychological Image Collection at Stirling (PICS)

Este é um conjunto de imagens úteis para a pesquisa em psicologia, tal como conjuntos de rostos e objetos. As imagens na base de dados são organizadas em conjuntos, sendo cada conjunto representando freqüentemente um estudo experimental separado.

The UMIST Face Database

Consiste em 564 imagens de 20 pessoas. Cada uma abrangendo uma escala de poses de perfil até vistas frontais. As pessoas abrangem uma escala de raça/sexo/aparência. Cada pessoa pertence a seu próprio diretório etiquetado 1^a, 1b,... 1t e as fotos são numeradas consecutivamente, assim que são tiradas. Os arquivos estão todos no formato PGM, aproximadamente 220 x 220 pixels em 256 máscaras do cinza.

Caltech Faces

450 imagens de rosto. 896 x 592 pixels. Formato JPEG. 27 ou apenas pessoas sob diferentes condições de iluminação/expressão/fundo.

EQUINOX HID Face Database

A identificação humana a partir de características faciais foi estudada primeiramente usando as imagens das câmeras de vídeo visíveis. Os sensores térmicos da imagem são uma das maiores inovações emergentes das tecnologias no mercado. Buscando sempre abaixar os custos e melhorar a sensibilidade e resolução, os sensores fornecem novas maneiras de identificação biométrica. O Equinox está coletando uma base de dados grande de imagens de rosto nas seguintes modalidades: co-registradas amplamente-visíveis/LWIR (8-12 mícrons), MWIR (3-5 mícrons), SWIR (0.9-1.7 mícrons). Este levantamento de dados torna-se disponível para a experimentação e avaliações estatísticas de desempenho.

VALID Database

Com o objetivo de facilitar o desenvolvimento de sistemas de reconhecimento de pessoas, sonoro, de rosto e de diversos outros modos (áudio-visuais) mais robustos, a base de dados VALID foi adquirida em um cenário do “mundo real” sem nenhum controle na iluminação ou no ruído acústico. A base de dados consiste em cinco sessões de imagens de 106 pessoas sobre o período de um mês. A sessão é feita em um estúdio com iluminação controlada e nenhum ruído de fundo, as outras 4 sessões são feitas em cenários de escritório. A base de dados contém imagens JPEG não-comprimidas em resolução 720x576 pixels.

The UCD Colour Face Image Database for Face Detection

A base de dados tem duas partes. A primeira parte contém retratos coloridos de rostos com alta variabilidade na escala, posição, orientação, pose, expressão facial e condições de iluminação, enquanto a segunda parte tem resultados segmentados manualmente para cada uma das imagens da primeira parte da base de dados. Estas imagens são adquiridas de uma grande variedade de fontes tais como câmeras digitais, imagens escaneadas usando scanner, outras bases de dados faciais e a Internet. A base de dados é destinada para distribuição aos pesquisadores nesta área.

Georgia Tech Face Database

A base de dados contém imagens de 50 pessoas e é armazenada no formato JPEG. Para cada indivíduo, há 15 imagens coloridas capturadas entre 01/06/99 e 15/11/99. A maioria das imagens foi tirada em duas sessões diferentes para contabilizar a variação nas condições de iluminação, na expressão facial e na aparência. Além disto, as faces foram capturadas em diferentes escalas e orientações.

5 – Reconhecimento

5.1 – Reconhecimento utilizando eigenfaces

O reconhecedor facial *Open Biometry* tem a tarefa de discriminar sinais de entrada (dados de imagem) em diversas classes (pessoas). Os sinais de entrada são altamente ruidosos (o ruído é causado por diferentes condições de luminosidade, posição, etc), além disso, as dados de entrada (imagens) não são completamente aleatórias e apesar das suas diferenças não há padrões de dados que ocorrem em qualquer sinal de entrada. Esses padrões, que podem ser observados em todo sinal poderiam ser, no campo de reconhecimento facial, a presença de alguns objetos (olhos, nariz, boca) em qualquer rosto bem como a distância relativa entre esses objetos. Essas características são chamadas de “eigenfaces” no domínio do reconhecimento facial (ou componentes principais, de maneira geral). Eles podem ser obtidos dos dados da imagem original por meio de ferramentas matemáticas chamadas de *Principal Component Analysis* (PCA), citado no item 18.

Através do PCA, pode-se transformar cada imagem original do conjunto de imagens de treinamento para a correspondente eigenface. Uma característica importante do PCA é que se pode reconstruir qualquer imagem original do conjunto de treinamento combinando eigenfaces. Logo, poderia dizer-se que a imagem original da face pode ser reconstruída de eigenfaces se somar todas as eigenfaces (características) na proporção certa. Cada eigenface representa apenas uma parte das características do rosto, que pode ou não estar presente na imagem original. Se a característica estiver presente na imagem original em um nível mais alto, a parte da correspondente eigenface no “somatório” de eigenfaces deveria ser maior. Se, ao contrário, uma característica particular não estiver presente na imagem original, então a correspondente eigenface deveria contribuir com uma parte menor no “somatório” de eigenfaces. Então, para reconstruir a imagem original a partir das eigenfaces, deve-se adotar um tipo de somatório ponderado de todas as eigenfaces. Ou seja, a imagem original reconstruída é igual ao somatório de todas as eigenfaces, com cada eigenface tendo um certo peso. Este peso especifica em qual nível a característica específica (eigenface) está presente na imagem original.

Se alguém usa todas as eigenfaces extraídas da imagem original, pode-se reconstruir a imagem original de forma exata a partir das eigenfaces. Porém, pode-se também usar

somente uma parte das eigenfaces. Então a reconstrução da imagem é uma aproximação da imagem original. Entretanto, pode-se garantir que as perdas devido à omissão de algumas eigenfaces podem ser minimizadas. Isto pode ser feito escolhendo somente as eigenfaces mais importantes.

Visão geral do algoritmo

A omissão de eigenfaces é necessária devido à escassez de recursos computacionais. Como isto se relaciona com o reconhecimento facial? A dica é que é possível não somente extrair a face das eigenfaces juntamente com o conjunto de pesos, mas também caminhar na direção oposta. Esta direção oposta seria a extração de pesos das eigenfaces e o rosto a ser reconhecido. Esses pesos dizem nada menos que o valor cuja face em questão difere das faces “típicas” representadas pelas eigenfaces. Portanto, através do uso desses pesos pode-se determinar duas diferentes coisas:

1. Determina, se a imagem em questão é o rosto como um todo. Caso os pesos da imagem diferirem muito dos pesos das imagens faciais (isto é, imagens que se conhece com certeza que são rostos), a imagem provavelmente não é um rosto.
2. Rostos similares (imagens) possuem características (eigenfaces) similares em nível (pesos). Se extraímos pesos de todas as imagens disponíveis, as imagens poderiam ser agrupadas em “clusters”. Ou seja, todas as imagens com pesos similares são prováveis de serem rostos similares.

O algoritmo de reconhecimento facial usando eigenfaces pode ser descrito na figura abaixo:

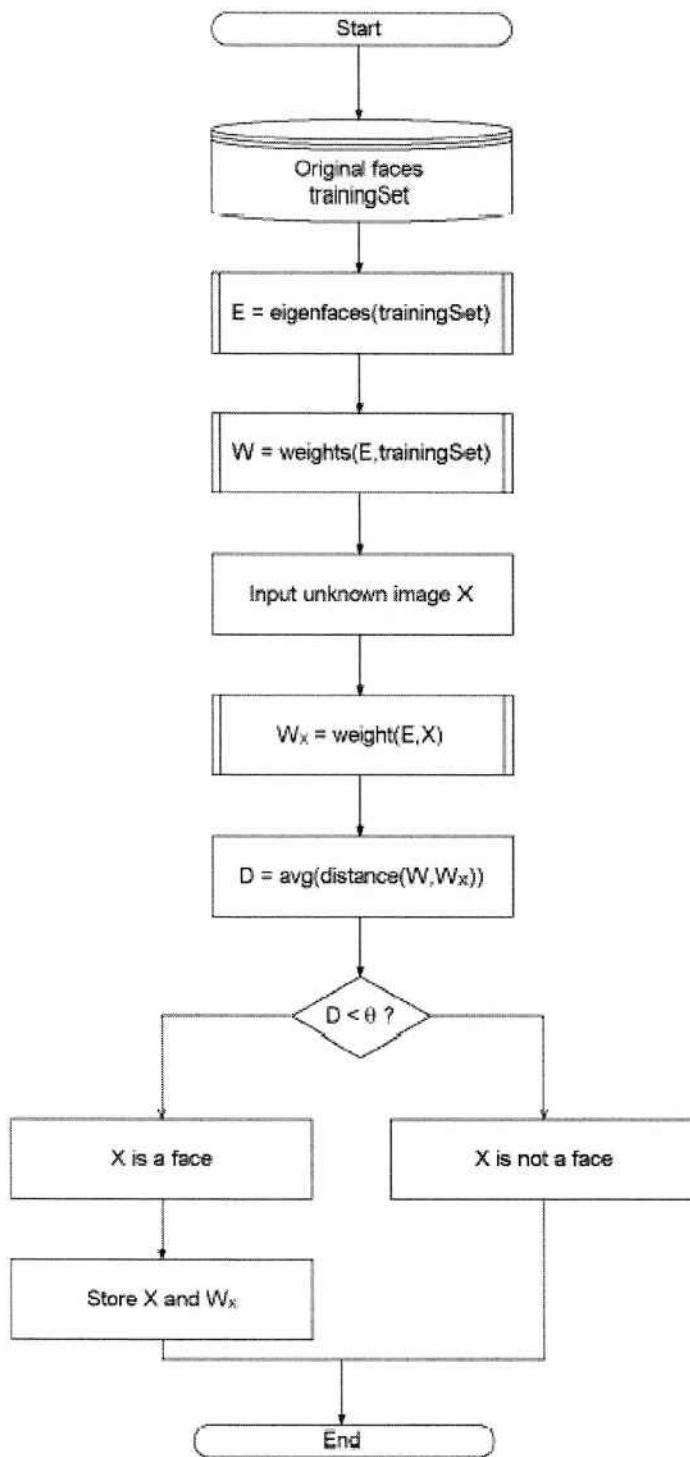


Figura 13: Visão geral do funcionamento do algoritmo de reconhecimento facial baseado em eigenface.

[12]

Primeiramente, a imagem original do conjunto de treinamento é transformada em um conjunto de eigenfaces E. Depois, os pesos são calculados para cada imagem do conjunto de treinamento e são armazenados no conjunto W.

Observando uma imagem desconhecida X, os pesos são calculados para cada imagem particular e armazenados no vetor W_x . Depois, W_x é comparado com os pesos das imagens, a partir do qual pode-se saber com certeza, que são rostos (os pesos do conjunto de treinamento W). Uma maneira para fazer isto seria estimando cada peso do vetor como um ponto do espaço e calcular uma distância média D entre os vetores de peso de W_x e o vetor de peso da imagem desconhecida X (Uma medida para isto poderia ser a distância Euclidiana descrita abaixo). Se a distância média exceder um valor limite θ , então o vetor de peso da imagem desconhecida X estará bastante “separado” dos pesos dos rostos. Neste caso, o desconhecido X não é considerado um rosto. Caso contrário, seu vetor de pesos W_x é armazenado para posterior classificação. O valor limite ótimo para θ é determinado empiricamente.

Autovetores e Autovalores

Um autovetor de uma matriz é um vetor tal que, se multiplicado com a matriz, o resultado é sempre um múltiplo inteiro deste vetor. Este valor inteiro é o correspondente autovalor do autovetor. Esta relação pode ser descrita pela equação $M \times u = \lambda \times u$, onde u é um autovetor da matriz M e λ é o correspondente autovalor.

Autovetores possuem as seguintes propriedades:

- Podem ser calculados somente a partir de matrizes quadradas;
- Existem n autovetores (e os correspondentes autovalores) em uma matriz n x n;
- Todos os autovetores são perpendiculares, isto é com um ângulo determinado entre si;

Cálculo de eigenfaces com PCA

1. Preparar os dados

Nesta etapa, as faces constituintes do conjunto de treinamento (Γ_i) devem ser preparadas para processamento.

2. Deduzir a média

A matriz média Ψ deve ser calculada, e então deduzida das faces originais (Γ_i), e o resultado armazenado na variável Φ_i :

$$\begin{aligned}\Psi &= \frac{1}{M} \sum_{n=1}^M \Gamma_n \\ \Phi_i &= \Gamma_i - \Psi\end{aligned}$$

3. Calcular a matriz de covariância

Nesta etapa a matriz de covariância é calculada de acordo com:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T$$

4. Calcular os autovetores e autovalores da matriz de covariância

Nesta etapa, os autovetores (eigenfaces) u_i e os correspondentes autovalores λ_i devem ser calculados. Os autovetores (eigenfaces) devem ser normalizados para que eles se tornem vetores unitários, isto é, de comprimento 1. A descrição do algoritmo exato para determinação dos autovetores e autovalores não é mostrado, já que pertence ao arsenal padrão da maioria das bibliotecas matemáticas de programação.

5. Escolher os componentes principais

Dos M autovetores (eigenfaces) u_i , somente M' devem ser escolhidos, aqueles que tem os maiores autovalores. Quanto maior o autovalor, mais características de um rosto o particular autovetor possui. Eigenfaces com baixos autovalores podem ser omitidos, já que eles descrevem somente uma pequena parte das características de um rosto. Depois que M' eigenfaces u_i são determinadas, a fase de treinamento do algoritmo está acabada.

Melhorias no algoritmo original

Há um problema no algoritmo descrito na seção anterior. A matriz de covariância C calculada na etapa 3 tem uma dimensão de $N^2 \times N^2$, ou seja, haveria N^2 autovetores e autovalores. Para uma imagem de tamanho 256 x 256, deve-se computar uma matriz 65.536 x 65.536 e calcular 65.536 eigenfaces. Computacionalmente, isto não é muito eficiente,

assim como essas eigenfaces não são muito úteis para a tarefa do reconhecedor. Por isso, as etapas 3 e 4 da seção anterior são substituídas pelo seguinte esquema:

$$\begin{aligned} C &= \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \\ L &= A^T A \quad L_{n,m} = \Phi_m^T \Phi_n \\ u_l &= \sum_{k=1}^M v_{lk} \Phi_k \quad l = 1, \dots, M \end{aligned}$$

onde, L é uma matriz $M \times M$, v são M autovetores de L e u são eigenfaces. A matriz de covariância é calculada usando a fórmula $C = AA^T$, a fórmula original (ineficiente) é fornecida somente para fins de explicação de A. A vantagem deste método é que se deve fornecer somente M valores e não N^2 . Em geral, $M \ll N^2$, ou seja, poucos componentes principais (eigenfaces) serão relevantes. A quantidade de cálculos a serem feitos é reduzida do número de pixels ($N^2 \times N^2$) para o número de imagens do conjunto de treinamento (M). Na etapa 5, os autovalores associados permitem ordenar as eigenfaces de acordo com a sua utilidade. Em geral, usamos apenas um subconjunto de M eigenfaces, as M' eigenfaces com os maiores autovalores.

Classificação das faces

O processo de classificação de uma nova (desconhecida) face Γ_{new} para uma das classes (faces conhecidas) ocorre em duas etapas.

Primeiramente, a nova imagem é transformada em seus componentes (eigenfaces). Os pesos resultantes formam o vetor de pesos Ω_{new}^T :

$$\begin{aligned} \omega_k &= u_k^T (\Gamma_{\text{new}} - \Psi) \quad k = 1 \dots M' \\ \Omega_{\text{new}}^T &= [\omega_1 \quad \omega_2 \quad \dots \quad \omega_{M'}] \end{aligned}$$

A distância Euclidiana entre dois vetores de peso $d(\Omega_i, \Omega_j)$ fornece uma medida da similaridade entre as correspondentes imagens i e j. Se a distância Euclidiana entre Γ_{new} e outras faces exceder, em média, algum valor limite θ , pode-se assumir que Γ_{new} não é uma face. $d(\Omega_i, \Omega_j)$ também permite construir “clusters” de faces tal que faces similares são designadas a um cluster.

Distância Euclidiana

Uma instância arbitrária x pode ser descrita pelo vetor característico abaixo:

$$x = [a_1(x), a_2(x), \dots, a_n(x)]$$

onde $a_r(x)$ é o valor do r-ésimo atributo da instância x. Então a distância entre duas instâncias x_i e x_j são definidas por:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2}$$

5.2 – Reconhecimento no software FaceVACS

Processamento da Intensidade da Imagem

No FaceVACS, as imagens são processadas da seguinte maneira:

- Localização do rosto: a imagem é analisada para determinar a posição e o tamanho de um ou mais rostos (em todas as etapas a seguir, é assumido que somente um rosto é encontrado na imagem).
- Localização dos olhos: as posições dos centros dos olhos dentro do rosto são determinadas.
- Verificação da qualidade da imagem: a qualidade da imagem de um rosto é verificada para ver se são suficientes para as etapas seguintes.
- Normalização: o rosto é extraído da imagem e é dimensionado e rotacionado de maneira que o resultado é uma imagem de tamanho fixo, com os centros dos olhos em posições fixas dentro da imagem.
- Pré-processamento: a imagem normalizada é pré-processada com técnicas padronizadas como a equalização de histogramas, normalização de luminosidade e outras.
- Extração das características: na imagem pré-processada, são extraídas as características que são relevantes para distinguir entre uma pessoa e outra.
- Construção do conjunto de referência: durante o registro das características faciais de várias imagens de uma pessoa são extraídas e combinadas para um conjunto de referência, também chamado de “modelo biométrico”.
- Comparação: Para verificação, o conjunto de características extraídas é comparado com o conjunto de referência da pessoa cuja imagem processada

declarou ser; para identificação, o conjunto de características é comparado com todos os conjuntos de referência armazenados, e a pessoa com o maior valor de comparação é selecionada; em ambos os casos o reconhecimento é considerado bem sucedido se o maior valor obtido exceder um certo valor limite.

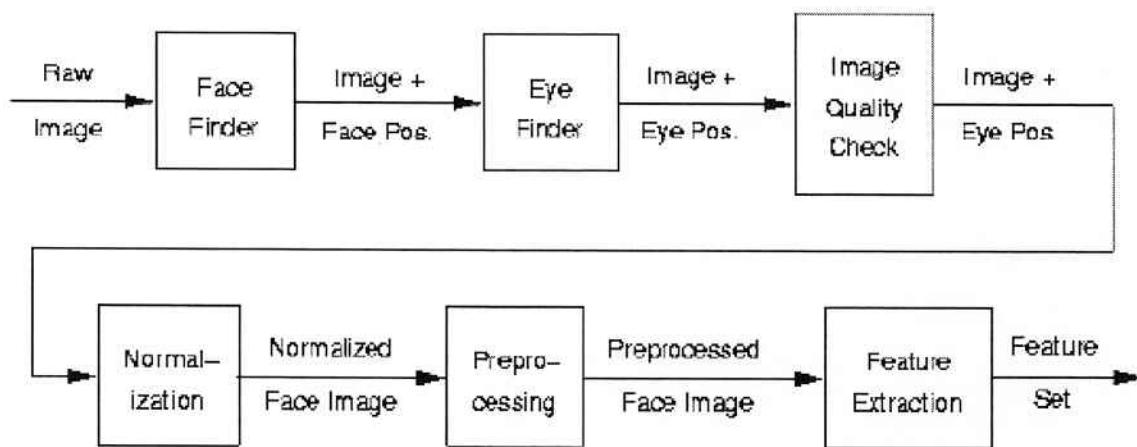


Figura 14: Arquitetura do FaceVACS: criação do conjunto de características [6]

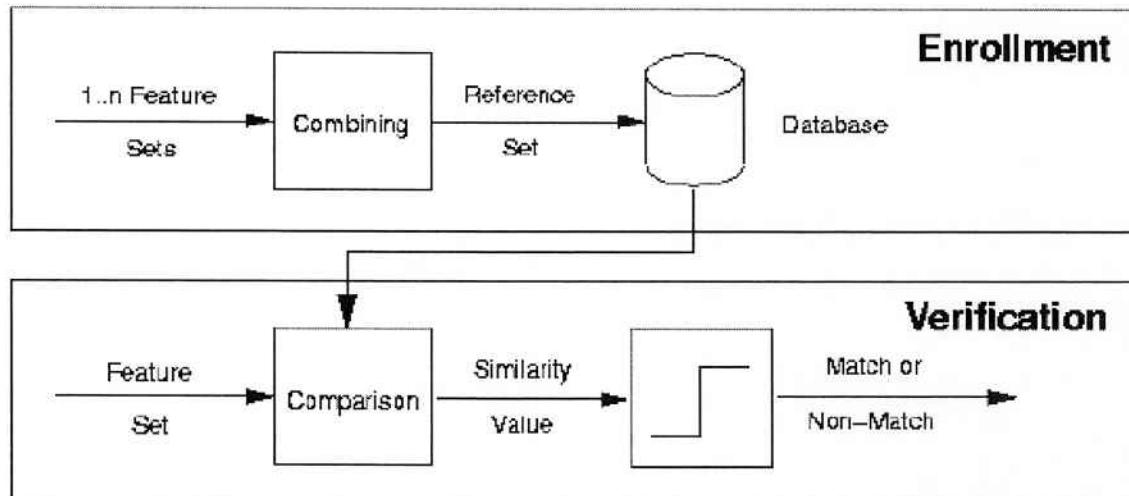


Figura 15: Arquitetura do FaceVACS: registro e verificação [6]

Além disso, o FaceVACS tem uma “verificação ao vivo”, uma facilitação para garantir que o rosto em frente à câmera é verdadeiro e não apenas uma fotografia. Para isto, as mudanças nas aparências durante o movimento de um rosto (rotações em torno do eixo vertical, em particular) são aproveitadas. Devido à estrutura em três dimensões de um rosto real, estas mudanças são muito diferentes para um rosto real comparadas com as mudanças em uma foto. Portanto, se um usuário quer passar pela “verificação ao vivo”, ele deve girar levemente a sua cabeça para trás e para frente. Uma outra maneira de se obter informação sobre a estrutura em três dimensões é utilizar duas ou mais câmeras, obtendo-se diferentes visões do rosto.

Nas seções seguintes, serão fornecidos mais detalhes das etapas individuais. Um exemplo de imagem é usado para mostrar o efeito de cada etapa de processamento.

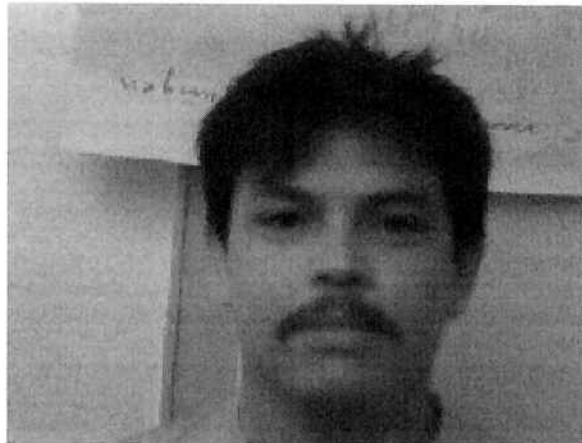


Figura 16: Exemplo de imagem [6]

Localização do rosto e dos olhos

Para localizar o rosto, a denominada imagem pirâmide é formada da imagem original. Uma imagem pirâmide é um conjunto de cópias da imagem original em diferentes escalas, representando um conjunto com diferentes resoluções de imagem. Uma máscara é movimentada de um pixel para outro sobre cada imagem na pirâmide e em cada posição a porção da imagem abaixo da máscara é passada para uma função que estima a similaridade da porção da imagem para um rosto. Se o valor resultante é alto o suficiente, a presença de

um rosto naquela posição e resolução é assumida como verdade. Daquela posição e resolução, a posição e o tamanho de um rosto na imagem original podem ser calculados.

Da posição de um rosto, a primeira estimativa das posições dos olhos podem ser deduzidas. De acordo com estas posições estimadas e de sua vizinhança, a busca pelas posições exatas dos olhos é iniciada. Esta busca é muito similar à busca pela posição do rosto, a principal diferença é que a resolução das imagens na pirâmide é maior do que a resolução na qual o rosto foi previamente encontrado. As posições que produzirem os maiores valores são tomadas como estimativas finais das posições dos olhos.

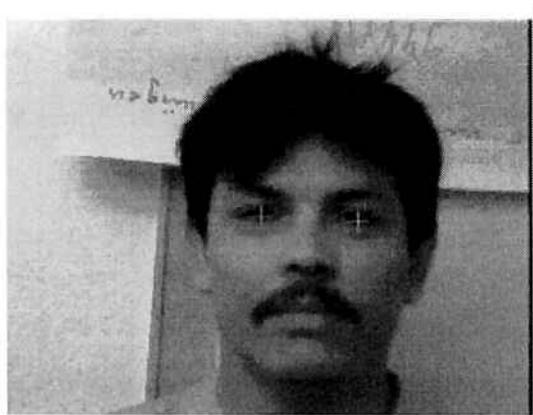


Figura 17: Posições dos olhos encontradas pelo algoritmo [6]

Verificação da Qualidade da Imagem

Para a usabilidade das etapas seguintes, a parte da imagem ocupada pelo rosto deve atender a certos requisitos de qualidade, isto é, não deve ser muito ruidosa ou borrosa. A qualidade é medida por meio de um conjunto de funções aplicadas à imagem. Se a qualidade for considerada muito baixa, a imagem é rejeitada.

Normalização e Pré-Processamento

Na etapa de normalização, o rosto é extraído, rotacionado e dimensionado de maneira que os centros dos olhos estejam em posições pré-definidas. Mais precisamente, eles são posicionados para estarem situados no pixel de mesma linha horizontal para que o ponto médio desta linha esteja alinhado com o ponto médio entre os centros dos olhos.

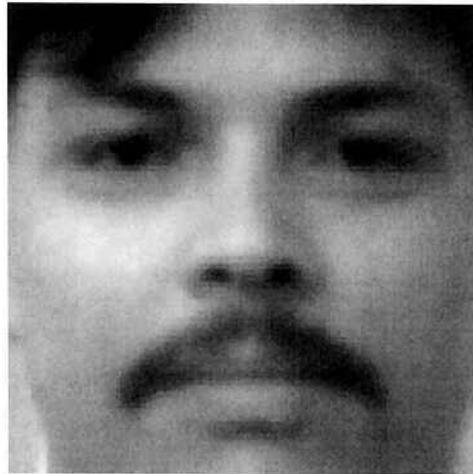


Figura 18: Depois da normalização [6]

A etapa de pré-processamento inclui, entre outras transformações, a eliminação de freqüências espaciais muito altas e muito baixas e a normalização de contraste.

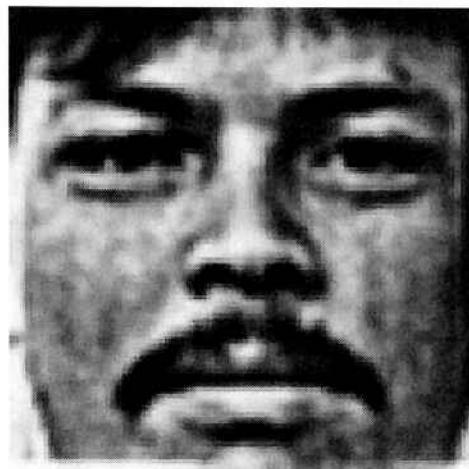


Figura 19: Depois do pré-processamento [6]

Extração das características e criação e comparação do conjunto de referência

A extração das características começa com transformações da imagem local que são aplicadas em locais fixos da imagem. Estas transformações capturam informações locais relevantes para distinguir pessoas, isto é, as amplitudes em certas freqüências espaciais em uma área local. Os resultados são representados em um vetor.

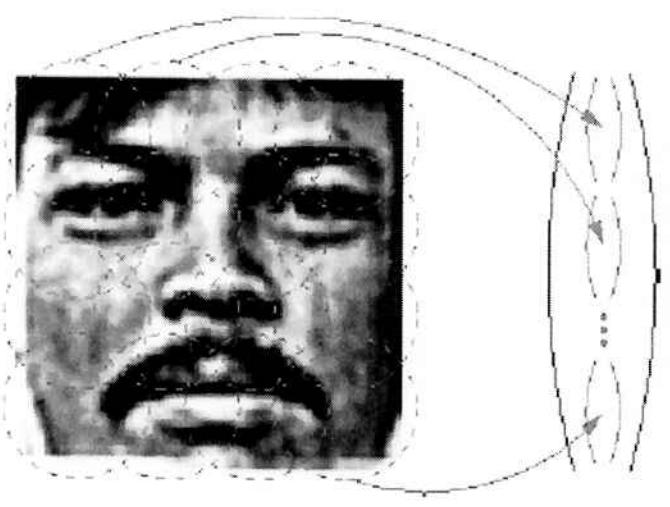


Figura 20: Extração de características locais [6]

Uma transformação global é então aplicada a este vetor. Utilizando uma grande base de dados de imagens faciais, os parâmetros dessa transformação são escolhidos para maximizar a relação entre a variância entre pessoas e a variância dentro de uma pessoa no espaço dos vetores transformados, isto é, as distâncias entre vetores correspondendo a imagens de diferentes pessoas devem ser grandes se comparadas às distâncias entre vetores correspondendo a imagens da mesma pessoa. O resultado dessa transformação é outro vetor que representa o conjunto de características da imagem de rosto processada.

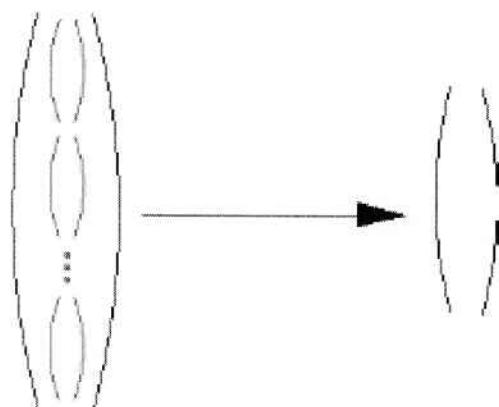


Figura 21: Transformação global, produzindo o conjunto de características da imagem do rosto [6]

Para a criação do conjunto de referência, várias imagens de uma mesma pessoa são utilizadas durante o registro para melhor cobrir o intervalo de possíveis aparências do rosto daquela pessoa. O conjunto de referência gerado para uma pessoa consiste de pelo menos cinco conjuntos de características, que são os centros de um grupo obtidos através de um processo de agrupamento dos conjuntos de características criados dessas imagens.

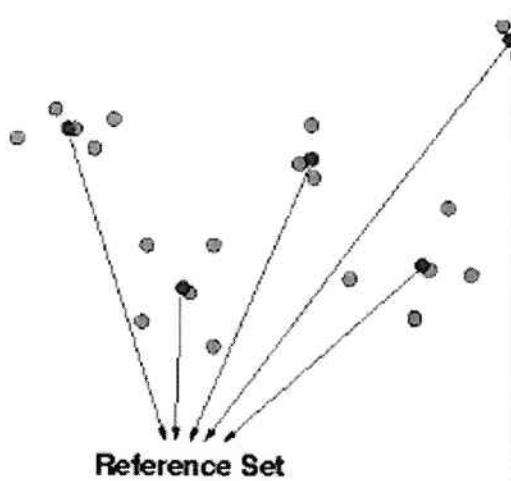


Figura 22: Combinando centros de grupos (em vermelho) para um conjunto de referência (pontos verdes são conjuntos de características criados das imagens) [6]

A função que é usada para comparar o conjunto de características com o conjunto de referência é simples e pode ser processada rapidamente. Isto possibilita a identificação ser feita em segundos, mesmo se milhares de conjuntos de referência tiverem que ser comparados.

Combinação do formato e intensidade para o processamento da imagem

Seqüência de processamento de amostras de dados (dados faciais)

Observando a intensidade da imagem, amostras de dados contendo tanto intensidades de imagens quanto informações sobre os formatos são processadas da mesma maneira que o descrito acima. A posição dos olhos obtidas nesta etapa são importantes para o posterior processamento dos dados de formato da imagem.

A comparação das amostras de dados é baseada numa etapa de fusão que mistura os resultados do reconhecimento da intensidade e do formato.

Toda a seqüência de processamento da amostra de dados acontece da seguinte maneira:

- Processamento da intensidade da imagem conforme descrito acima.
- Pré-processamento dos dados de forma: dependendo do tipo de sensor e das condições de aquisição, os dados de formato retornados pelos sensores em três dimensões são freqüentemente ruidosos e incompletos. Antes dos dados de formato serem passados às etapas padrão de reconhecimento, eles devem ser pré-processados e estabilizados de alguma maneira.
- Normalização: similarmente ao que é feito com a intensidade de imagens, os formatos dos rostos são dimensionados para um tamanho padrão e organizados no espaço de forma a minimizar as variações devido a translações, rotações e escalamento.
- Extração das características: a partir do formato pré-processado e normalizado, as características de formato do rosto são extraídas de tal forma a serem relevantes para descrever similaridades e diferenças entre rostos.
- Construção do conjunto de referência de amostras: durante o registro, tanto a intensidade quanto o formato baseados nas características do rosto de uma ou mais amostras de uma pessoa são extraídas e combinadas para um conjunto de referência, também chamado de “modelo biométrico”.
- Comparação: para comparar um conjunto de características com o conjunto de referência, uma classificação ou nota é determinada considerando tanto as suas características de intensidade quanto de formato.

Processamento da nota para o conjunto de referência da amostra

1. Processamento da nota parcial das características baseadas na intensidade.
2. Processamento da nota parcial das características baseadas no formato
3. Processamento da nota final baseado em um algoritmo de fusão.

Pré-processamento dos dados de formato

Em geral, os dados fornecidos por sensores em três dimensões contêm ruídos e dados vazios (lacunas) e outliers, isto é, pequenos grupos de posições muito distantes do formato do rosto. Além disso, dependendo do princípio do sensor em três dimensões, pode haver muitas partes do formato do rosto faltando se estas partes do rosto estão obstruídas.

As figuras abaixo mostram exemplos para todas estas falhas:



Figura 23: Dados de formato ruidosos vindos do sensor [6]



Figura 24: Imagem do sensor, embora mais aplainada que a de cima, contém dados vazios (lacunas) e outliers [6]



Figura 25: Vistas não-frontais resultam em obstruções e buracos no formato do rosto. Esta visão do formato do rosto esconde as lacunas [6]



Figura 26: Rotacionando o formato, as porções faltantes do formato tornam-se evidentes [6]

Dos diversos algoritmos qualificados a lidar com esses problemas, apenas aqueles cujo tempo de execução é compatível com o algoritmo de reconhecimento facial em um mundo real podem ser empregados.

As principais etapas requeridas para obter os dados de formato coerentes com a extração de características de formato além dos dados originais do sensor são:

1. Remoção do Outlier
2. Redução do Ruído
3. Preenchimento das lacunas

Remoção do Outlier

A remoção da camada é o primeiro passo da seqüência de pré-processamento, já que outliers no sensor de dados pode causar distúrbios nas subseqüentes operações de alisamento nos dados de formato.

O problema com a remoção de outliers é detectar o que é um “outlier” e o que é um dado normal. Como os dados do sensor podem conter lacunas e saltos, uma definição ingênuca como “um outlier é aquilo que não é uma continuação do alisamento da superfície de um rosto” falhará em muitos casos. Uma aproximação para fazer esta distinção é fazer processamento de estatísticas locais da superfície do rosto e eliminar todos os vértices que estão muito distantes do local médio.

Exemplo: Remoção de Outlier baseado em estatísticas locais



Figura 27: Remoção de outlier [6]

Redução do Ruído

Generalizações de operações de processamento de imagens em duas dimensões bem conhecidas como ordenamento por níveis e filtragem média dos dados de formato, em geral produzem resultados satisfatórios.

Além do alisamento dos dados, o ordenamento por níveis também contribui para a remoção do outlier para alguma extensão.

Veja abaixo alguns exemplos demonstrando a redução do ruído do formato por média e filtro médio.

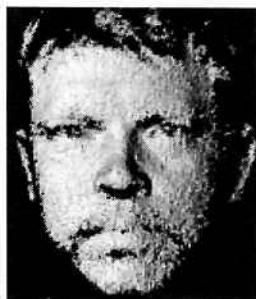


Figura 28: Dados originais ruidosos [6]



Figura 29: Resultado após combinar média e filtro médio (a estrutura em malha nessas imagens é um artefato de renderização) [6]

Preenchimento das lacunas

Dados faltantes podem ser reconstruídos a partir de aproximações de superfícies locais. Embora a reconstrução de remendos de superfícies quase planas é, na maioria das vezes apropriada, a aplicabilidade desses métodos em regiões com alta curvatura é limitada.



Figura 30: Exemplo de preenchimento de lacunas [6]

Normalização

A identidade de informação contida no formato de um rosto é “intrínseca” a este formato, isto é, não é afetado por translações ou rotações do formato.

Por outro lado, os dados de formato obtidos dos sensores podem ter orientações arbitrárias no espaço. Para eliminar as diferenças entre os conjuntos de dados de formato devido a orientações espaciais, o passo de normalização é aplicado depois do pré-processamento onde os rostos estarão centralizados e alinhados à vista frontal.

Exemplo de Normalização:



Figura 31: Diferentes vistas de um rosto obtidas do sensor [6]



Figura 32: Vista padrão depois da normalização [6]

Extração das características

A extração das características dos dados de formato é um processo similar àquele aplicado aos dados de intensidade da imagem. Em um conjunto de localizações espaciais fixas definidas relativamente às posições dos olhos, as descrições de formato são obtidas e coletadas em um vetor.

Assim como nos dados de intensidade da imagem, este vetor é transformado para uma representação que distingui formatos de rostos de diferentes pessoas.

Fusão

A fusão da intensidade da imagem e o processamento do formato da imagem são melhorados em nível de nota, isto é, a nota obtida da comparação do conjunto de características de intensidade é combinada com a nota obtida da comparação do conjunto de características de formato, resultando em uma nota única. A função de fusão leva em consideração os diferentes níveis de confiabilidade com o qual uma nota de intensidade ou formato reflete a probabilidade de duas respectivas imagens mostrarem a mesma pessoa.

Qualidade da Amostra

A qualidade das imagens que são processadas tem influência significativa no desempenho biométrico do software FaceVACS nestas imagens. Ao invés de comparar imagens de alta qualidade entre si, comparar imagens de baixa qualidade com imagens de alta ou baixa qualidade leva a uma maior taxa de falso negativo, e sob certas circunstâncias, comparar imagens de baixa qualidade entre si pode causar também uma maior taxa de falso positivo.

O software FaceVACS pode ser configurado para processar notas para a qualidade e usar isto para rejeitar imagens de baixa qualidade para registro ou para reconhecimento. Processar as notas para a qualidade de uma imagem inclui a determinação de certas propriedades encontradas na imagem para correlacionar com o desempenho do reconhecimento pelo FaceVACS. A figura abaixo mostra alguns exemplos de imagens com qualidades baixas por diferentes motivos. Uma nota maior indica uma qualidade maior da imagem.

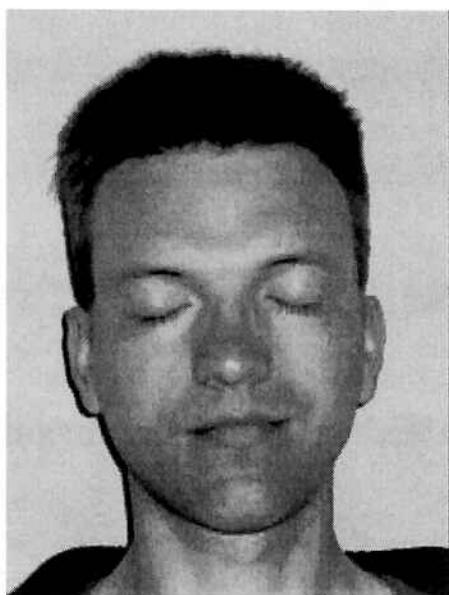


Figura 33: Qualidade baixa devido aos olhos estarem fechados [7]



Figura 34: Qualidade baixa devido à posição não-frontal [7]

Teoricamente, a escala de valores de notas para a qualidade está no intervalo [0,1], mas geralmente somente valores acima de 0.9 ocorrem. A figura abaixo mostra a distribuição das notas para a qualidade em um conjunto de 1000 fotos típicas de passaporte. Nota-se que 2,3% das imagens tem notas para a qualidade abaixo do limite padrão de 0.975.

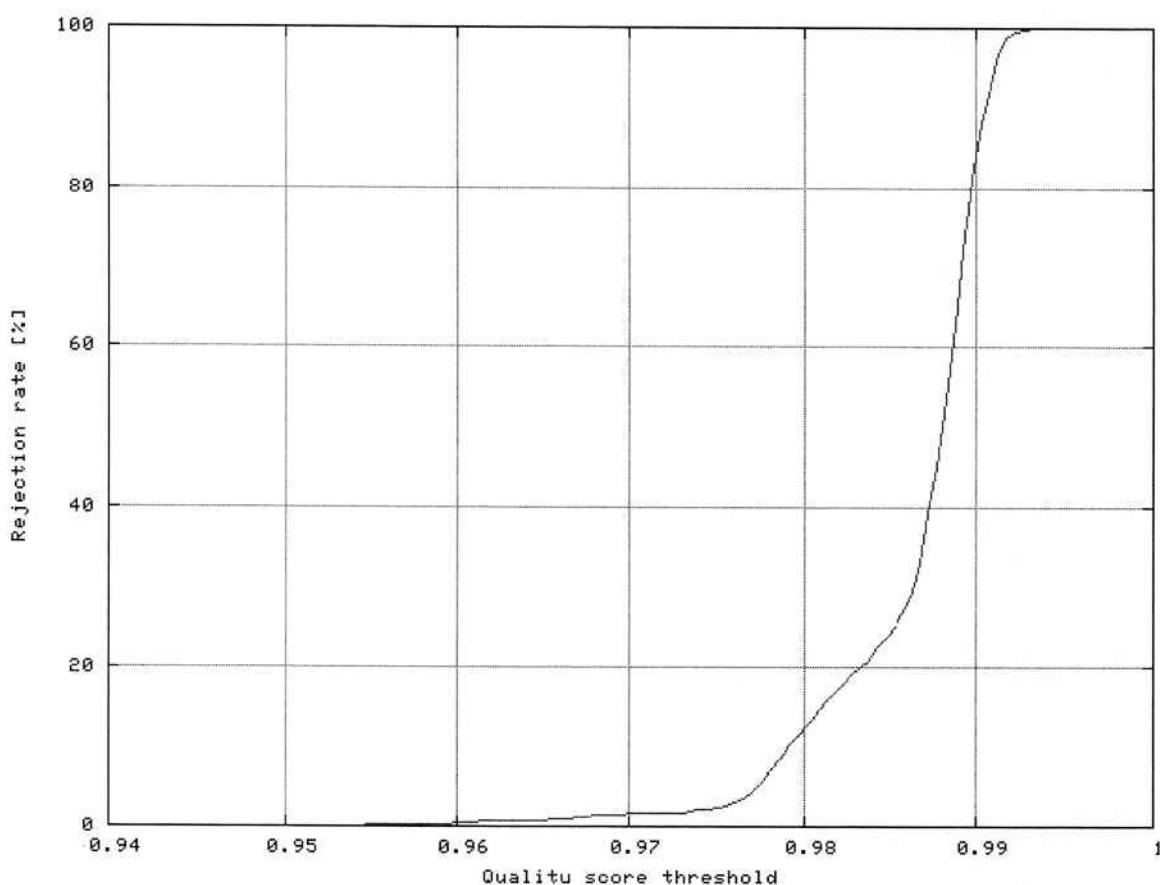


Figura 35: Distribuição das notas para a qualidade em um conjunto de 1000 fotos típicas de passaporte. A taxa de rejeição é a porcentagem de imagens cuja nota para a qualidade está abaixo de um dado limite, de modo que eles seriam rejeitados devido à baixa qualidade da amostra. [7]

A figura a seguir mostra o efeito no desempenho ao rejeitar imagens com baixa qualidade. Para quase todos os valores de falsos positivos, a taxa de verificação aumenta quando o registro de imagens de baixa qualidade é descartado. O valor limite para as notas

de qualidade usado no teste foi o padrão (0.975), e o conjunto de imagens para o registro é aquele cuja distribuição de notas de qualidade é dado na figura acima.

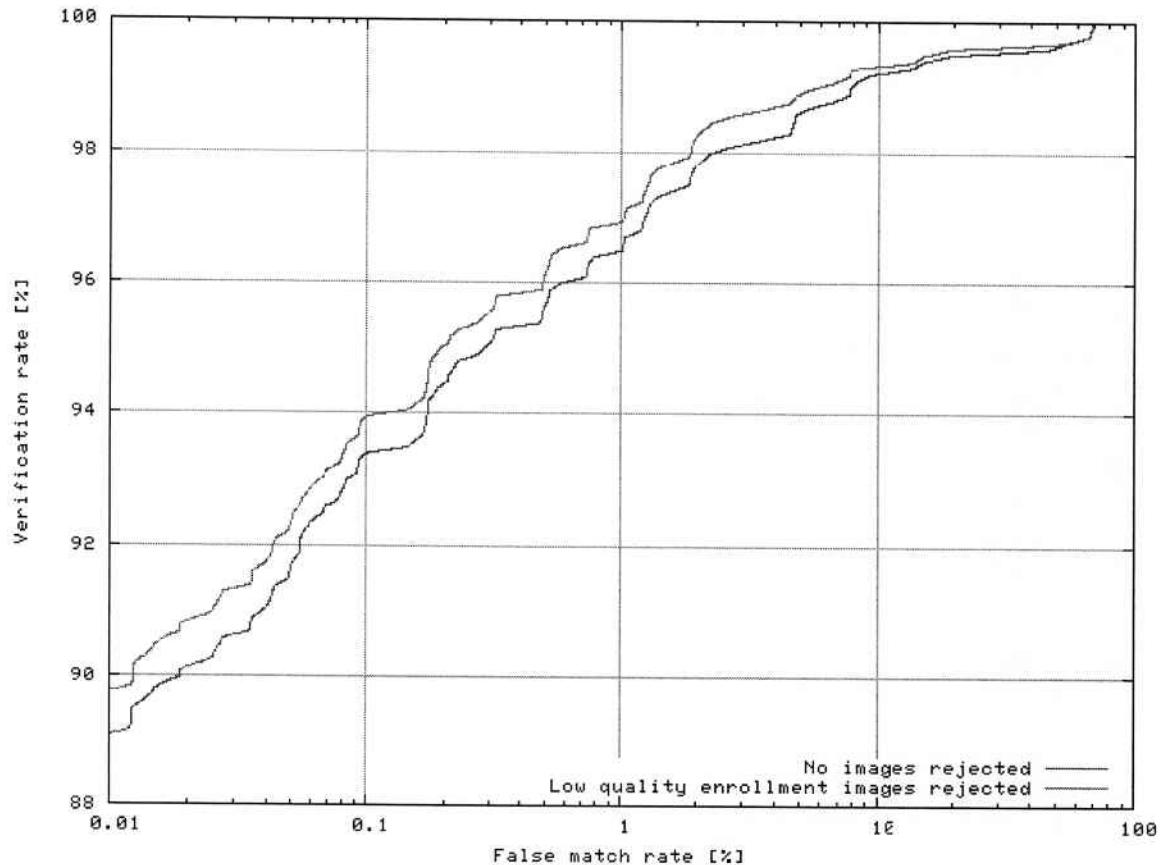


Figura 36: Curvas que mostram o efeito de rejeitar o registro de imagens com baixa qualidade. Os resultados foram obtidos com a máquina B2T6 em um conjunto de 2000 fotos típicas de passaporte (1000 imagens para registro e 1000 imagens para probe) [7]

Em geral, não é necessário rejeitar imagens com baixa qualidade no momento do reconhecimento, garantindo-se qualidade suficiente das imagens de registro. Isto ocorre devido a taxa de falsos negativos generalizada, isto é, a variante da taxa de falsos negativos que leva em consideração a taxa de falha na aquisição, para a qual as rejeições baseadas em qualidade contribuem, deve ser algo mais baixo no caso de aceitar imagens com baixa qualidade para o reconhecimento do que no caso de rejeitá-los, enquanto a mudança na taxa de falsos negativos generalizada deve ser negligenciada.

Explicando de outra maneira, rejeitar uma imagem para o reconhecimento baseado na qualidade, em geral constitui um falso negativo de qualquer maneira, enquanto aceitar a imagem ainda fornece ao software da FaceVACS a oportunidade de um resultado correto, enquanto o perigo de um falso positivo para a imagem analisada não é mais alto do que o normal (geralmente o perigo ainda é menor).

Como a função de comparação é simétrica, a conversa também é verdadeira: em geral, não é necessário rejeitar imagens de baixa qualidade para o registro, já que se pode garantir qualidade suficiente das imagens a serem usadas para o reconhecimento. Isto é interessante particularmente em casos onde não é possível influenciar a qualidade das imagens para registro.

6 – Implementação e Testes

O software usado para implementação do módulo de reconhecimento facial foi o *Face Recognition*, que consiste de um conjunto de códigos-fonte implementados em Java. O algoritmo é baseado no método eigenface, explicado anteriormente.

6.1 – Implementação

A técnica de Eigenface pertence a uma família de modelos que processa um conjunto de determinados valores para cada imagem. Estes valores da imagem desejados são comparados com os valores de imagens do banco de dados codificados da mesma maneira. Um dos problemas ignorados até então eram quais aspectos do rosto são importantes para a identificação, pois a codificação e decodificação de imagens faciais usando características locais e globais do rosto como o nariz, olhos e orelhas podem ou não estar relacionadas com o reconhecimento facial. Uma aproximação para obter tais informações relevantes do rosto é capturar a variação em um conjunto de imagens faciais, independente de qualquer julgamento da característica. Em termos matemáticos, o algoritmo encontra os principais componentes da distribuição de rostos, ou os autovetores da matriz de covariância do conjunto de imagens faciais. Estes autovetores podem ser vistos como um conjunto de características que juntos caracterizam a variação entre as imagens faciais. Cada localização da imagem contribui mais ou menos com cada autovetor como uma mistura de rostos “fantasmas” denominados eigenfaces.



Figura 37: Eigenface [11]

Além disso, as imagens faciais também podem ser reconstruídas a partir de somatórios de pesos de um conjunto de características do rosto. Uma forma eficiente de aprendizado e reconhecimento poderia ser a partir da construção de características de imagens faciais conhecidas e reconhecer rostos específicos comparando os pesos necessários das características para reconstruir os rostos com pesos associados com as pessoas conhecidas.

Isto foi exatamente o que foi implementado no framework do reconhecedor facial em Java.

Há três passos no algoritmo:

- Inicialização das imagens de treinamento, cálculo das “eigenfaces” e de seus respectivos pesos;
- Comparação com a imagem de entrada;
- Verificação da distância entre a imagem de entrada e algum rosto do espaço de rostos.

Inicialização

Há oito passos na construção do espaço de rostos:

- Construir um vetor de rostos com as imagens de treinamento;
- Calcular o rosto médio;
- Normalizar as imagens de treinamento;
- Processar os autovetores e autovalores;
- Extrair a variável MAGICNR (um número qualquer, que nesta implementação é onze) que representa os autovalores mais significativos (e seus respectivos autovetores);
- Projetar os autovetores sobre o vetor de rostos, resultando no espaço de rostos;
- Normalizar o vetor de rostos;
- Calcular o conjunto de pesos associados com cada imagem de treinamento.

O algoritmo assume o seguinte:

- Todas as imagens têm as mesmas dimensões;

- Cada imagem é representada como um componente de um vetor. As dimensões do vetor são $1 \times N$, onde $N = \text{largura} \times \text{altura}$.

Imagens de treinamento

Neste passo é construído um vetor de rostos que consiste de dezesseis modelos de treinamento (imagens). Uma matriz de dimensão $16 \times N$ é construída (N é o produto largura x altura da imagem), onde cada linha é uma imagem e cada coluna é um valor de intensidade da imagem.

O código abaixo ilustra como a imagem (representada como um vetor) é copiada para o array $16 \times N$:

```
double[][] face_v = new double[16][width*height];

for (i = 0; i < files.length; i++) {
    face_v[i] = files[i].getDouble();
}
```

onde “files” é um objeto JPGFile (ou PPMFile) e “getDouble()” retorna a representação da imagem no formato RGB como um vetor.

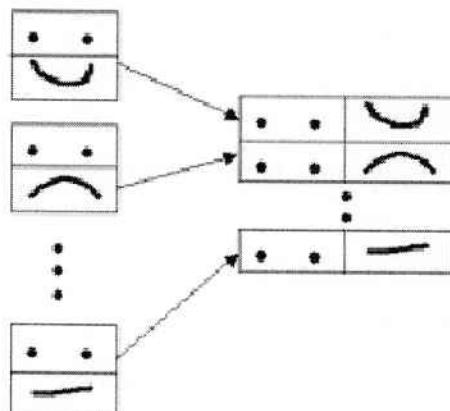


Figura 38: Como as imagens são construídas em um vetor de rostos [11]

Rosto médio

O rosto médio que é processado é um vetor no qual cada coluna determina a intensidade média das dezesseis (M) imagens.

O rosto médio é obtido pela seguinte fórmula:

$$A(j) = \frac{1}{M} \sum_{i=0}^M I_i(j)$$

onde M é dezesseis, I é uma das M imagens e A é o vetor do rosto médio.

O código abaixo ilustra este passo:

```
double[] avgF = new double[length];
for (pix = 0; pix < length; pix++) {
    temp = 0;
    for (image = 0; image < nrfaces; image++) {
        temp += face_v[image][pix];
    }
    avgF[pix] = temp / nrfaces;
}
```

onde “length” é N (largura*altura da imagem) e “nrfaces” também é N.

Normalização de imagens

O rosto médio é subtraído do vetor de rostos (de dimensão M x N, com N imagens) para normalizar as imagens usando:

$$I_i(j) = I_i(j) - A(j)$$

onde:

$$0 \leq j < N; 0 \leq i < M$$

O código abaixo ilustra este passo:

```
for (image = 0; image < nrfaces; image++) {
    for (pix = 0; pix < length; pix++) {
        face_v[image][pix] = face_v[image][pix] - avgF[pix];
    }
}
```

Autovalores e Autovetores

Este passo processa os determinados valores citados no início, denominados autovetores e autovalores. Os autovetores e autovalores são valores característicos de uma matriz, e podem ser definidos da seguinte maneira:

Cada autovalor pode ser imaginado como um valor que, quando subtraído de cada elemento da diagonal, torna a matriz singular. Autovetores são vetores característicos da matriz. Cada autovetor corresponde a um dos autovalores.

$$Av = \lambda v$$

onde v são os autovetores e λ são os autovalores.

Por exemplo, supondo:

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

$$A \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Neste caso, 3 é o autovalor da matriz A e o autovetor é

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Processar estes valores é uma tarefa pesada para imagens de tamanho típico. Felizmente, podem-se determinar estes autovalores e autovetores resolvendo uma matriz menor $M \times M$ e pegando a combinação linear do resultado.

$$C = A^T A$$

The diagram shows a vertical rectangle labeled A^T above a horizontal rectangle labeled A . To the right of the multiplication symbol (*) is a vertical rectangle labeled C , with an equals sign (=) preceding it. This visualizes the equation $C = A^T A$.

Figura 39: Matriz de covariância [11]

Os autovalores possuem uma distribuição aleatória após o cálculo. Para encontrar valores extremos para a variável MAGICNR é necessário mesclar os autovalores e seus correspondentes autovetores. Isto é feito usando um método que reposiciona os autovetores nas suas colunas apropriadas (dependendo dos autovalores misturados).

O código a seguir ilustra esta etapa:

```
int index = new int[nrfaces];
double tempVector = new double[nrfaces][nrfaces];
/* Temporary new eigVector */

for ( i = 0; i < nrfaces; i++) /* Enumerate all the entries */
    index[i] = i;

doubleQuickSort(eigValue, index,0,nrfaces-1);
// Put the index in inverse
int tempV = new int[nrfaces];
for ( j = 0; j < nrfaces; j++)
    tempV[nrfaces-1-j] = index[j];
index = tempV;
/*
 * Put the sorted eigenvalues in the appropriate columns.
 */
for ( col = nrfaces-1; col >= 0; col --) {
    for ( rows = 0; rows < nrfaces; rows++ ){
        tempVector[rows][col] = eigVector[rows][index[col]];
    }
}
eigVector = tempVector;
```

Projeção sobre o espaço de rostos

Multiplicando o autovetor mesclado com o vetor de rostos resulta no vetor de espaço de rostos. Isto fornece o mesmo resultado do que se tivesse processado as eigenfaces sobre vetor de rostos em si e não sobre a matriz de covariância.

Normalização do espaço de rostos

A normalização do espaço de rostos é um procedimento relativamente simples. O valor máximo do espaço de rostos é dividido por cada pixel do espaço de rostos.

$$Q(i) = \frac{P(i)}{\max(P)}$$

O código abaixo ilustra este passo:

```
Matrix eigVectorM = new Matrix(eigVector, nrfaces,nrfaces);
double[][] faceSpace = eigVectorM.times(faceM).getArray();
eigVector = null;
for ( image = 0; image < nrfaces; image++) {
    temp = max(faceSpace[image]); // Our max
    for ( pix = 0; pix < faceSpace[0].length; pix++)
        faceSpace[image][pix] = Math.abs( faceSpace[image][pix] / temp);
}
```

Pesos

O último passo é calcular o conjunto de pesos associados com o espaço de rostos. Cada peso é um vetor de dimensão 1 x MAGICNR. Os pesos são o resultado da multiplicação da transposta de cada linha do vetor de espaço de rostos com as imagens de treinamento normalizadas.

A implementação calcula M pesos: cada imagem de treinamento tem um vetor de pesos associado de tamanho MAGICNR com cada uma das outras imagens. Então, cada vetor de pesos da imagem de treinamento representa um vetor de dimensão MAGICNR. Cada valor no vetor de pesos representa a “similaridade” com as outras M imagens de treinamento.

O código abaixo ilustra este passo:

```
double[][] wk = new double[nrfaces][MAGIC_NR]; // M rows, 11 columns
for (image = 0; image < nrfaces; image++) {
    for (j = 0; j < MAGIC_NR; j++) {
        temp = 0.0;
        for ( pix=0; pix< length; pix++)
            temp += faceSpace[j][pix] * faces[image][pix];
        wk[image][j] = Math.abs( temp );
    }
}
```

Reconhecimento

O reconhecimento de uma imagem é ainda mais simples comparado às etapas anteriores.

Há quatro etapas no reconhecimento:

- Transformar a imagem de entrada em componentes eigenfaces (projetá-la sobre o espaço de rostos);

- Calcular os pesos da imagem de entrada;
- Determinar a distância Euclidiana dos pesos da imagem de entrada para os pesos do conjunto de imagens do espaço de rostos;
- Determinar, baseado na distância Euclidiana e no valor mínimo, se a imagem de entrada é encontrada entre as imagens do banco de dados.

Transformação

A projeção da imagem de entrada sobre o espaço de rostos inclui:

- Subtrair o rosto médio do rosto de entrada (para normalizar a imagem);
- Projetar a imagem normalizada sobre espaço de rostos.

Para normalizar o rosto de entrada, usa-se o vetor do rosto médio (calculado anteriormente) para subtrair cada valor de intensidade da imagem:

$$I(j) = I(j) - A(j)$$

onde:

$$0 \leq j < N$$

Projetar a imagem normalizada sobre o espaço de rostos consiste em multiplicar a imagem do espaço de rostos com a imagem de entrada normalizada. Este também é o passo no qual se determinam os pesos da imagem.

O código a seguir ilustra este passo:

```
double[] input_vk = new double[MAGIC_NR];
double temp = 0;
for (j = 0; j < MAGIC_NR; j++) {
    temp = 0.0;
    for (pix=0; pix < length; pix++)
        temp += faceSpace[j][pix] * inputFace[pix];

    input_vk[j] = Math.abs( temp );
}
```

onde “inputFace” é o vetor representando a imagem.

Distância Euclidiana

A distância Euclidiana é a soma das diferenças de cada índice em um vetor. Calcula-se a distância sobre os pesos da imagem de entrada e sobre os pesos das imagens de treinamento.

O código abaixo ilustra este passo:

```
double[] distance = new double[MAGIC_NR];
double[] minDistance = new double[MAGIC_NR];
idx = 0;
for (image = 0; image < nrfaces; image++) {
    for (i = 0; i < MAGIC_NR; i++) {
        distance[i] = Math.abs(input_wk[i] - wk[image][i]);
    }
    ...
}
```

Limitação

O último passo é determinar se a distância normalizada é menor do que o valor limite. A imagem é considerada reconhecida se o valor da distância é menor do que o valor limite.

O código a seguir ilustra este passo:

```
.....
if (image == 0)
    System.arraycopy(distance,0,minDistance,0,MAGIC_NR);
if (sum(minDistance) > sum(distance)) {
    this.idx = image;
    System.arraycopy(distance,0,minDistance,0,MAGIC_NR);
}
if (max(minDistance) > 0.0)
    divide(minDistance, max(minDistance));
minD = sum(minDistance);
```

baseado em minD e no valor limite global, é determinado se a imagem de entrada pertence ao conjunto de imagens de treinamento.

Foi usado o Eclipse para a implementação do módulo de interface, sua integração com o módulo de reconhecimento facial, detalhado acima e o módulo de banco de dados de imagens faciais, implementado em MySql.

6.2 – Testes dos softwares de reconhecimento disponíveis

Teste do Software OpenBiometry

O teste do software foi feito inicialmente sem as bibliotecas exigidas, rodando o projeto faces (dentro do arquivo zip do software) no ambiente Visual Studio 2005 e verificando que os erros ocorridos exigiam arquivos das bibliotecas Devil e OpenCv.

Após a inclusão dos arquivos dessas bibliotecas um a um no include no Visual Studio, ainda assim ocorreu um erro numa chamada de função ‘cvHaarDetectObjects’. Por estar em uma versão pré-alfa, a implementação deste software ainda possui alguns problemas (como o erro encontrado), e que os integrantes do grupo não conseguiram solucionar.

Teste do Software FaceVACS

O teste do software FaceVACS não foi realizado em tempo devido a dificuldades na configuração do ambiente de desenvolvimento.

Teste do Software Face Recognition

Em um primeiro teste, o software Face Recognition mostrou um alto grau de imprecisão no reconhecimento facial. Com duas imagens faciais de uma mesma pessoa com condições de distância, angular e iluminação similar, o software apontou para diferentes imagens da base de comparação como sendo a mais parecida com a imagem facial de entrada. Com uma base de comparação de 17 imagens, o software acertou 11 imagens faciais, com uma precisão de aproximadamente 65%, o que é inviável para uma implementação prática em um cenário do mundo real, como por exemplo, dentro do prédio de Engenharia Elétrica da Universidade de São Paulo.

Após uma segunda análise da documentação do software, percebeu-se que ele tinha como pré-requisito o cadastro de pelo menos 16 imagens de uma mesma pessoa na base de dados para o treinamento do reconhecedor, com condições de iluminação e fundo uniformes. Com estes pré-requisitos atendidos, o software teve a sua precisão aumentada para mais de 80%.

A implementação foi feita de tal forma que no cadastro de uma pessoa as 16 fotos do rosto da pessoa sejam tiradas em seqüência e salvas com o nome da pessoa como prefixo.

6.3 – Exemplo de tela de captura e verificação

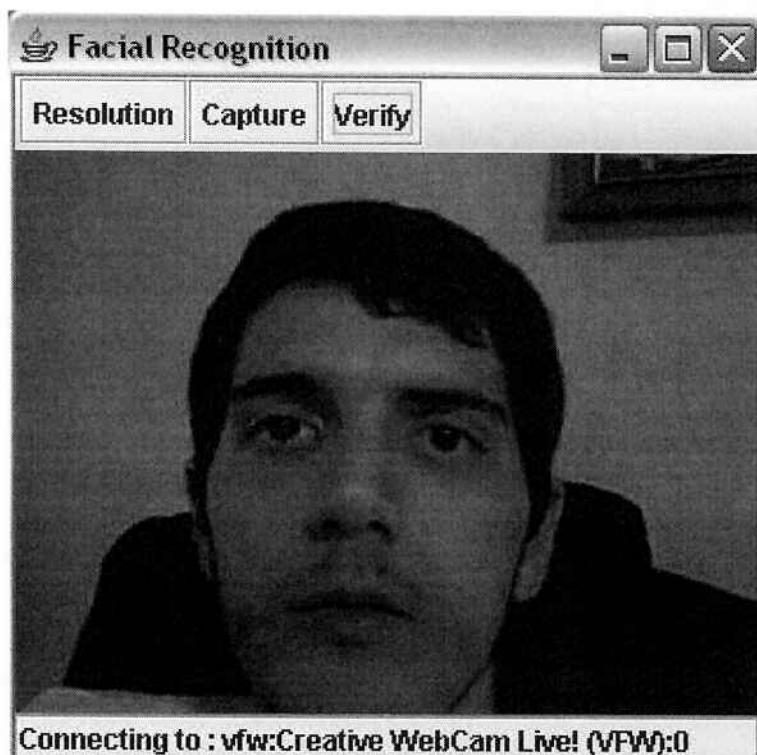


Figura 40: Exemplo de tela de captura e verificação

7 – Considerações finais

O projeto foi de fundamental importância para o entendimento dos conceitos e metodologias de reconhecimento facial, sua integração com bases de dados de imagens faciais, as áreas de pesquisa existentes, casos comerciais de sucesso em empresas de segurança e casos de algoritmos de reconhecimento facial de código aberto (desenvolvidos por estudantes de pós-graduação em Universidades conhecidas pelo mundo) que ainda possuem um alto grau de imprecisão, como é o caso da maioria, sendo que as aplicações que atingem um nível de precisão satisfatório acabam se tornando comerciais.

A implementação da arquitetura física e lógica do cenário proposto no projeto acabou sendo simplificada para uma interface de cadastro de dados de usuário, um módulo de reconhecimento facial e um banco de dados de imagens faciais com uma saída permitindo ou negando o acesso de um indivíduo, devido a fatores limitadores do tempo de implementação disponível.

Entre os fatores limitadores do tempo de implementação podemos destacar as pesquisas e os testes com as alternativas encontradas de reconhecedores faciais, o que tomou um tempo muito grande na análise de suas respectivas viabilidades de implantação em ambientes de execução disponíveis assim como a viabilidade da integração com os módulos de interface e banco de dados.

Outro fator limitador do tempo de implementação está relacionado ao funcionamento do algoritmo de reconhecimento facial escolhido, já que o seu grau de imprecisão estava relativamente elevado, e isso exigiu testes exaustivos para possibilitar a análise de condições necessárias para que o software funcionasse com um nível de precisão satisfatório.

A implementação completa da arquitetura física e lógica proposta pelo grupo fica como proposta para aquelas pessoas que se interessarem pelo projeto e que acreditam nas vantagens econômicas e sociais que um cenário de controle de acesso de usuários baseado em biometria como aquele mostrado neste documento pode trazer.

Anexo I – Cronograma do Projeto

As etapas do projeto foram divididas da seguinte maneira:

- De 09/01/2006 a 06/03/2006 – Etapa do projeto que consiste na especificação dos requisitos principais funcionais e não-funcionais do sistema, objetivo do projeto e determinação do nível de complexidade ou escopo do projeto;
- De 07/03/2006 a 27/03/2006 – Etapa do projeto que consiste na especificação da arquitetura do sistema e dos recursos necessários para a implementação do projeto;
- De 28/03/2006 a 17/04/2006 – Etapa do projeto que consiste na implementação de um protótipo inicial do sistema, possivelmente incluindo uma das funcionalidades do sistema (por exemplo, o cadastro de um ator no sistema).
- De 17/04/2006 a 25/08/2006 – Etapa do projeto que consiste na pesquisa de softwares de reconhecimento facial e equipamentos necessários à implementação e início da digitação da monografia.
- De 25/08/2006 a 04/12/2006 – Etapa do projeto que consiste na implementação e testes finais do software de reconhecimento facial, implementação dos módulos de interface de cadastro, banco de dados de imagens e integração entre os módulos, assim como a digitação de todo o conteúdo da monografia.

Lista de Referências

- [1] A4VISION. Estados Unidos. Disponível em:
http://www.a4vision.com/2_3dfacial.html. Acesso em: 15 mar. 2006
- [2] FACE RECOGNITION HOME PAGE. Croácia. Disponível em: <http://www.face-rec.org/>. Acesso em: 15 mar. 2006
- [3] OPEN SOURCE COMPUTER VISION LIBRARY. Estados Unidos. Disponível em:
<http://www.intel.com/research/mrl/research/opencv/>. Acesso em: 12 abr. 2006
- [4] COGNITEC SYSTEMS. Alemanha. Disponível em: <http://www.cognitec-systems.de/>. Acesso em: 22 mai. 2006
- [5] COGNITEC SYSTEMS BRASIL. Rio de Janeiro. Disponível em:
<http://www.cognitec.com.br>. Acesso em: 05 set. 2006
- [6] FACEVACS Algorithms White Paper. [S. l.: s.n.], jul. 2005
- [7] FACEVACS Algorithms White Paper. [S.l.: s.n.], may 2006
- [8] SMITH, L. I. A tutorial on principal components analysis, feb. 2002. Disponível em:
http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.
- [9] TURK, M.; PENTLAND, A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1),1991a. Disponível em:
<http://www.cs.ucsb.edu/~mturk/Papers/jcn.pdf>.
- [10] TURK, M. A.; PENTLAND, A. P. Face recognition using eigenfaces. In *Proc. of Computer Vision and Pattern Recognition*, p. 586-591. IEEE, jun. 1991b. Disponível em:
<http://www.cs.wisc.edu/~dyer/cs540/handouts/mturk-CVPR91.pdf>.
- [11] RZESZUTEK, K. Dynamic Scalable Distributed Face Recognition System Security Framework. [S.l.: s.n.], apr. 2002.

[12] PISSARENKO, D. Eigenface-based facial recognition. [S.l.: s.n.], dez. 2002