COPPE/UFRJ

ALGORITMO PARA RECONHECIMENTO DE CARACTERÍSTICAS FACIAIS BASEADO EM FILTROS DE CORRELAÇÃO

Gabriel Matos Araujo

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Eduardo Antônio Barros da Silva

Rio de Janeiro
Fevereiro de 2010

ALGORITMO PARA RECONHECIMENTO DE CARACTERÍSTICAS FACIAIS BASEADO EM FILTROS DE CORRELAÇÃO

Gabriel Matos Araujo

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Cláudio Rosito Jung, D.Sc.

Prof. Sergio Lima Netto, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2010

Araujo, Gabriel Matos

Algoritmo para reconhecimento de características faciais baseado em filtros de correlação/Gabriel Matos Araujo. – Rio de Janeiro: UFRJ/COPPE, 2010.

XIII, 65 p.: il.; 29, 7cm.

Orientador: Eduardo Antônio Barros da Silva

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2010.

Referências Bibliográficas: p. 57 – 61.

1. Reconhecimento de padrões. 2. Processamento de imagens. 3. Visão computacional. I. Silva, Eduardo Antônio Barros da. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha família.

Agradecimentos

Em primeiro lugar, gostaria de agradecer à minha família, incluindo os Molinas, pelo apoio incondicional em todos os momentos.

Gostaria de agradecer também a todos os meus amigos, pelo apoio e por compreender a minha ausência em muitos momentos.

Aos amigos e colegas do LPS por estarem sempre dispostos em ajudar.

Ao meu orientador, Eduardo Antônio Barros da Silva e a Waldir Sabino da Silva Júnior, pela paciência, incentivo e orientação.

Agradeço ainda à CAPES, pelo apoio financeiro, a COPPE/UFRJ e ao LPS por tornar possível a realização deste trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ALGORITMO PARA RECONHECIMENTO DE CARACTERÍSTICAS FACIAIS BASEADO EM FILTROS DE CORRELAÇÃO

Gabriel Matos Araujo

Fevereiro/2010

Orientador: Eduardo Antônio Barros da Silva

Programa: Engenharia Elétrica

A detecção de características faciais é um problema que tem sido muito investigado nos últimos anos. Atualmente existem algumas soluções, inclusive comerciais, para este tipo de problema. Isto se deve, principalmente, ao fato de que pontos de controle definidos na face (também conhecidos como *landmarks* ou pontos fiduciais) podem ser utilizados em etapas iniciais de muitos outros sistemas, como, por exemplo, sistemas de rastreamento, sistemas de segurança, de modelagem 3D, dentre outros.

Neste trabalho propomos um nova metodologia para a detecção de pontos fiduciais em faces. A principal diferença entre o sistema proposto e os outros é a utilização do DPI (Detector por Produto Interno) na etapa de classificação. O DPI é uma nova classe de detectores baseados em filtros de correlação, cuja principal vantagem é a fácil incorporação da estatística do conjunto de treinamento no projeto do detector.

O resultado é um sistema capaz de reconhecer pontos fiduciais com baixo custo computacional e com taxas de acerto competitivas, principalmente para os pontos situados na região dos olhos e do nariz.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

FACIAL FEATURE RECOGNITION ALGORITHM BASED ON CORRELARION FILTERS

Gabriel Matos Araujo

February/2010

Advisor: Eduardo Antônio Barros da Silva

Department: Electrical Engineering

The detection of facial features is a problem that has been given a lot of attention in recent years. There are several solutions to this problem in use today. This is so because control points defined in the face (also known as landmarks or fiducial points) can be used in initial steps in many other systems, such as tracking systems, security systems, 3D modeling, among others.

In this work we propose a new methodology for fiducial point detection on faces. Our main contribution is the use of the DPI (from portuguese *Detector por Produto Interno*) in the classification step. The DPI is a new class of detectors based on correlation filters, whose main advantage is the easy incorporation of the training set statistics in the design of the detector.

The result is a system able to recognize fiducial points with low computational cost and competitive hit rates, especially for points around the eyes and nose.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas

AAM	*Active Appearance Model*, p. 22
ASM	*Active Shape Model*, p. 21, 22
AdaBoost	*Adaptative Boosting*, p. 3
CFA	*Class-dependence Feature Analysis*, p. 1
DFT	*Discrete Fourier Transform*, p. 2
DoG	*Diference of Gaussian*, p. 26
IDFT	*Inverse Discrete Fourrier Transform*, p. 2
LDA	*Linear Discriminant Analysis*, p. 8
MACE *filter*	*Minimum Average Correlation Energy Filter*, p. 33
RBF	*Radial Basis Function*, p. 18
SVM	*Suport Vector Machine*, p. 2

Capítulo 1

Introdução

Características faciais são um conjunto de informações que caracterizam uma face humana. É possível definir inúmeras características faciais, dentre as quais a largura da boca, espaço entre os olhos ou tamanho do nariz. As posições relativa de pontos de controle (também conhecidos como *landmarks* ou pontos fiduciais) sobre regiões que definem estas características são informações muito úteis em diversos tipos de sistemas, como sistemas de segurança, de reconhecimento de expressões, de rastreamento e de modelagem 3D, dentre outros.

Na literatura, é possível encontrar diversas abordagens empregadas na detecção de pontos fiduciais na face. Existem duas vertentes principais. A primeira delas utiliza uma cascata de classificadores *Adaboost* [1], [2] e [3], enquanto a segunda utiliza variações de uma técnica conhecida como AAM (*Active Appearence Model*), que utiliza busca o melhor casamento com um modelo combinado de textura e forma [4], [5], [6] e [7].

Neste trabalho é proposto um novo sistema de reconhecimento de pontos fiduciais capaz de detectar um conjuntos de treze pontos fiduciais, que estão ilustrados na Figura1.1. Estes pontos foram escolhidos por estarem localizados em saliências da face (o que torna mais fácil a detecção) e por serem utilizados em outros trabalhos, como [1], [2] e [4]. A principal diferença entre o sistema proposto e os demais é a utilização de um novo tipo detector, baseado em filtro de correlação, na etapa de classificação. Nos referimos a este detector como o DPI (Detetor por Produto Interno). A seguir, ele é descrito com mais detalhes.

A correlação pode ser considerada uma medida de similaridade entre dois sinais. Esta é uma das razões que tornam a filtragem de correlação e suas variações, como o CFA (*Class-dependence Feature Analysis*), muito utilizadas em sistemas de reconhecimento de padrões. Dentre as aplicações é possível destacar reconhecimento de objetos em imagens.

A ideia central da filtragem de correlação é a filtragem casada no domínio da frequência. Ela é obtida através da correlação cruzada entre o conjugado invertido de

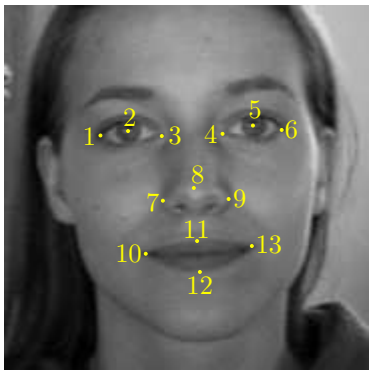


Figura 1.1: Pontos fiduciais utilizados neste trabalho

um padrão que se deseja detectar e um sinal desconhecido no domínio da frequência. Isto é feito com o auxílio da DFT (*Discrete Fourier Transform*). Se a IDFT (*Inverse Discrete Fourrier Transform*) da correlação cruzada apresenta um pico discernível, então o padrão é localizado.

No DPI um detector $\mathbf{h}$ é obtido a partir das amostras de treinamento e a detecção é feita pelo produto interno entre o detector e uma amostra desconhecida $\mathbf{X}$:

$$\mathbf{h}^t\mathbf{X} = C, \tag{1.1}$$

onde, idealmente, $C = 1$ se a amostra é o padrão que se deseja reconhecer e $C = 0$ caso contrário.

O sistema desenvolvido é composto por uma associação de classificadores em série conhecida como cascata de classificadores. Como a saída do DPI é um valor real, um classificador é utilizado na saída do DPI no intuito de obter automaticamente um limiar que determina se a amostra em questão é ponto fiducial. Neste trabalho foram propostas duas versões, uma com o discriminante de Fisher [8] e outra com o *AdaBoost* [9] como forma de obter este limiar. Com isto cada estágio da cascata é composto de um DPI e um segundo classificador.

Para comparação com o método proposto, foram utilizados dois outros métodos. O primeiro deles é composto por uma cascata de classificadores SVM (*Support Vector Machine*). Neste caso, foi utilizada uma biblioteca disponível em [10]. O segundo é um método baseado em cascata de classificadores *AdaBoost* [2]. Com relação a este último, foi utilizado um algoritmo de teste, disponibilizado pelos autores em [11].

Para o treino e avaliação do sistema proposto, bem como dos métodos utilizados para a comparação (o algoritmo proposto por [2] permite apenas a detecção usando um classificador previamente treinado) foi utilizada uma base de dados gratuita, a BioID, disponível em [12]. Esta base apresenta imagens de face humanas com grande variedade de poses, condições de iluminação e escala. Para este trabalho foi selecionado um subconjunto da BioID com 503 imagens frontais e sem oclusão do

padrão, ou seja, imagens com indivíduos sem óculos, sem barba e sem bigode.

1.1 Organização da dissertação

No Capítulo 2 são apresentados alguns conceitos básicos de reconhecimento de padrões e de aprendizado de maquina. São descritos também os métodos clássicos de reconhecimento de padrões utilizados neste trabalho: o discriminante de Fisher, o AdaBoost (*Adaptative Boosting*) e suas principais variações, e o SVM. São apresentados ainda alguns métodos estado-da-arte: o detector de objetos Viola-Jones [13] e alguns sistemas de reconhecimento de características faciais recentes [1], [2] e [4]. Estas técnicas apresentadas foram utilizadas para compor o método proposto ou para comparar os resultados obtidos.

No Capítulo 3, é proposto um sistema de reconhecimento de características faciais como uma aplicação do DPI. A versão complexa do DPI, que ajuda a fornecer uma importante interpretação para o método, é descrita no Apêndice A.

No Capítulo 4 a metodologia utilizada é descrita e em seguida são apresentados os resultados obtidos. São descritas ainda as bases de dados utilizadas neste trabalho, bem como o esquema de validação e a forma utilizada para avaliar o desempenho dos sistemas descritos. As simulações indicam que os resultados obtidos são satisfatórios, principalmente para os pontos situados na região dos olhos e do nariz.

Por fim, no Capítulo 5 são feitas as considerações finais sobre os resultados obtidos além de apresentar perspectivas para trabalhos futuros.

Capítulo 2

Fundamentos teóricos

2.1 Reconhecimento de padrões

2.1.1 Sistemas de reconhecimento de padrões

O Reconhecimento de Padrões faz parte da área de Aprendizado de Máquina, e seu objetivo é separar objetos (ou padrões) em categorias (ou classes). Os objetos podem ser qualquer conjunto de medidas que necessite ser classificado, como por exemplo, os pixeis de uma imagem [14].

Em geral reconhecer um padrão não é uma tarefa fácil e envolve várias etapas. De acordo com [14] um sistema de reconhecimento de padrões típico pode ser dividido em cinco sub-sistemas básicos: sensoriamento, segmentação, extração de características, classificação e pós-processamento. O diagrama em blocos de um sistema típico pode ser visto na Figura 2.1.

Na etapa de sensoriamento, um sensor ou transdutor converte um fenômeno físico em um conjunto de dados, que pode ser uma imagem ou um sinal de fala, por exemplo. Estes dados são compostos de objetos e plano de fundo, geralmente possuindo algum tipo de ruído. Nas etapas seguintes é necessário que os objetos estejam devidamente separados. Os objetos são, portanto, segmentados. Em seguida são extraídas as características dos objetos que são úteis na classificação. O classificador utiliza estas características para determinar a qual classe o objeto em questão pertence. No pós-processamento alguma técnica pode ponderar as saídas de diferentes classificadores e tomar a decisão final. Vale ressaltar que o fluxo dos dados não necessariamente possui sentido único e pode haver retro-alimentação entre estes sub-sistemas [14].

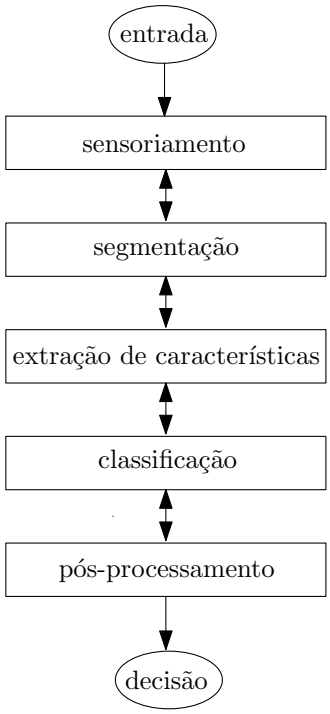


Figura 2.1: Etapas de um sistema de reconhecimento de padrões típico.

2.1.2 Ciclo de projeto

De acordo com [14], o projeto de um sistema de reconhecimento de padrões envolve a repetição das seguintes etapas: coleta dos dados, seleção de características, seleção do modelo, treinamento e avaliação. O diagrama de blocos do ciclo de projeto de um sistema de reconhecimento de padrões está ilustrado na Figura 2.2.

A coleta de dados consiste em selecionar um conjunto de amostras representativas do fenômeno físico que se deseja classificar. A quantidade deve ser grande o suficiente para representar os conjuntos de treino e de teste.

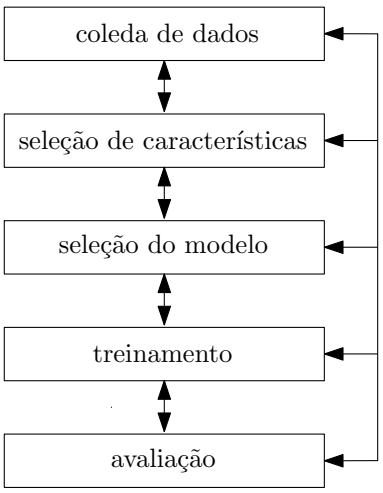


Figura 2.2: Ciclo de projeto de um sistema de reconhecimento de padrões.

Em geral, uma grande quantidade de características pode ser extraída de um determinado objeto. Contudo, pode ser que apenas algumas possuam poder discriminatório significativo. Dentre as diversas características, devem ser selecionadas aquelas que sejam insensíveis a variações irrelevantes do objeto, como por exemplo o tamanho da face em um sistema de reconhecimento de faces. Na seleção do modelo, uma representação do fenômeno é escolhida. O tipo de classificador a ser utilizado está intimamente relacionado com o modelo escolhido. Na etapa de treinamento, os dados de treino são utilizados para ajustar o classificador a fim de obter um desempenho satisfatório. Na etapa de avaliação o desempenho do classificador é medido sobre um novo conjunto de dados, o conjunto de teste. Esta é uma etapa fundamental, pois pode apontar alguma deficiência em outra etapa do projeto.

2.1.3 Aprendizado e adaptação

Aprendizado é o processo de treinamento do classificador. Dado o modelo para o problema e um classificador adequado, o conjunto de treinamento é utilizado para estimar os parâmetros do classificador. De acordo com [14] e [15], a aprendizagem é a aplicação de um algoritmo que reduz o erro do conjunto de treinamento. Basicamente existem dois tipos de aprendizado: o supervisionado e o não supervisionado.

No aprendizado supervisionado, um rótulo para cada amostra do conjunto de treino é utilizado para ajustar os parâmetros de classificação e então reduzir o erro de treinamento. No aprendizado não supervisionado, também conhecido como agrupamento ou "*clusterização*", nenhuma informação sobre a classe do conjunto de treinamento é utilizada. Geralmente a quantidade de classes, bem como suas respectivas condições iniciais, são passadas ao algoritmo de treino, que procura por grupos representativos no conjunto de dados.

Na literatura é possível encontrar outros tipos de aprendizado, como por exemplo o aprendizado por reforço [14] e o aprendizado semi-supervisionado [15]. No aprendizado por reforço nenhum rótulo de classe é utilizado, embora a informação sobre o acerto ou erro de classificação de uma amostra de treinamento seja usada para ajustar o classificador. No aprendizado semi-supervisionado, parte das amostras de treinamento não possuem rótulo de classe. Com isso a parte do conjunto de treino sem rótulo é utilizada para melhorar o classificador final de acordo com algum algoritmo de agrupamento.

2.2 Combinação de classificadores

A justificativa para a combinação de um conjunto de classificadores é a obtenção de um desempenho melhor do que cada classificador individualmente pode atingir.

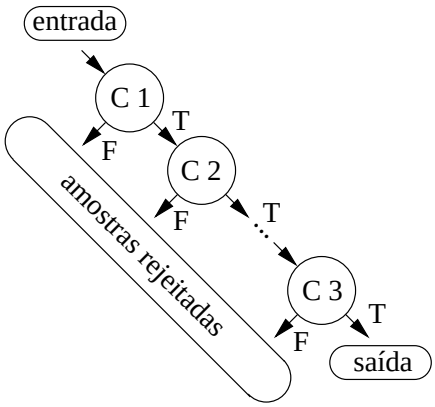


Figura 2.3: Combinação em cascata. Nesta estrutura, cada classificador é ajustado para obter uma alta taxa de acerto e consequentemente alta taxa de falsos positivos. Com o aumento do número de estágios é possível reduzir a taxa de falsos positivos, embora a taxa de acerto também seja reduzida. Cada classificador da combinação é especializado em uma porção do espaço de características.

A ideia é especializar cada classificador constituinte em uma região do espaço de características. Em uma combinação os classificadores constituintes podem ser do mesmo tipo ou não. Existem diversas formas de combinar classificadores, mas não existe uma regra que determina em qual situação deve-se empregar um determinado tipo de combinação. De uma maneira geral a estratégia de combinação envolve alguma heurística direcionada ao problema em questão.

Neste trabalho busca-se resolver um problema de duas classes, onde cada amostra deve ser classificada como objeto de interesse ou não. Para tanto foi adotada uma combinação de classificadores no formato de árvore degenerada. Este tipo de estrutura, também conhecido como "cascata", foi originalmente proposto por [13] para resolver o problema de detecção de faces. Nesta combinação, somente as amostras classificadas como positivas em um estágio são apresentadas ao classificador subsequente. Cada classificador constituinte é ajustado para obter um alta taxa de acerto e consequentemente uma alta taxa de falsos positivos. Com o aumento do número de classificadores na cascata, é possível reduzir a taxa de falsos positivos tanto quanto se queira, embora a taxa de acerto tenda a diminuir. Um esquema de uma combinação em cascata está ilustrado na Figura 2.3.

2.3 Métodos clássicos

Nesta seção serão apresentados os métodos de classificação de padrões utilizados neste trabalho. Na Seção 2.3.1 é mostrado um método de classificação linear muito utilizado para tratar casos simples, o método de Fisher. Na Seção 2.3.2, é apresentado o AdaBoost (*Adaptive Boosting*), uma meta-heurística que reutiliza as amostras de treinamento ponderando suas probabilidades de ocorrência. Por fim, na Seção

2.3.3 é apresentado o SVM (*Support Vectors Machine*), um método cujo objetivo é encontrar um plano de separação que maximiza a distância entre as amostras mais próximas de classes distintas.

2.3.1 Discriminante linear de Fisher

A Análise de Discriminantes Lineares é uma classe de métodos de aprendizado de máquina que utilizam combinações lineares das amostras de treinamento para dividir os espaço de características através de hiperplanos [16]. Um dos métodos de discriminação linear mais utilizados é o Discriminante Linear de Fisher, também conhecido como Fisher LDA (*Fisher Linear Discriminant Analysis*).

O Discriminante de Fisher foi originalmente proposto por R. A. Fisher para resolver um problema de taxonomia [8]. Neste método, nenhuma suposição sobre a distribuição das classes é feita. Ele encontra as direções de projeção de um conjunto de dados que melhor discriminam as classes. O método de Fisher pode ser utilizado também para redução de dimensionalidade ou como pós-processamento de classificadores mais complexos [16].

Para descrever o método, seja $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n, \cdots, \mathbf{x}_N\}$ um conjunto d-dimensional composto por N amostras. Esta amostras são divididas em dois conjuntos, $\mathbf{X}_1$, com n_1 amostras pertencentes à classe C_1 e $\mathbf{X}_2$, com n_2 amostras pertencentes à classe C_2. Sejam ainda $\boldsymbol{\mu}_i$, a média, e $\boldsymbol{\Sigma}_i$, a matriz de covariância, das amostras da classe i, onde $i = 1, 2$.

A partição do espaço de características é feita através de um hiperplano do tipo:

$$f(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + b, \tag{2.1}$$

onde $\mathbf{w}$ é a direção normal ao hiperplano e $|b|$ é a distancia do hiperplano à origem ($\|\mathbf{w}\| = 1$).

Projetando cada elemento $\mathbf{x}_n$ no hiperplano definido por $\mathbf{w}$, um novo conjunto $Y = \{y_1, \cdots, y_n, \cdots, y_N\}$ é gerado:

$$y_n = \mathbf{w}^t\mathbf{x}_n. \tag{2.2}$$

A ideia central é encontrar a direção que melhor classifica a projeção do conjunto de dados. Para tanto, dada a média $\boldsymbol{\mu}$ do conjunto de dados, são definidas a matriz de espalhamento intra-classe,

$$\mathbf{S}_\mathbf{W} = \sum_{i=1}^{2} \frac{n_i}{n}\boldsymbol{\Sigma}_i, \tag{2.3}$$

e a matriz de espalhamento entre-classes,

$$\mathbf{S_B} = \sum_{i=1}^{2} \frac{n_i}{n}(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t. \tag{2.4}$$

Com estas matrizes define-se a seguinte função objetivo:

$$\mathbf{J}(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S_B} \mathbf{w}}{\mathbf{w}^t \mathbf{S_W} \mathbf{w}}. \tag{2.5}$$

No caso de $\mathbf{S_W}$ ser inversível, a solução desta função é a direção cuja projeção dos dados apresenta o menor espalhamento intra-classe e o maior espalhamento entre classes:

$$\mathbf{w} = {\mathbf{S_W}}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \tag{2.6}$$

Embora a direção do hiperplano separador possa ser determinada pela Equação (2.6), não existe uma regra geral que determina o parâmetro b. Se a distribuição das classes for gaussiana, então o valor de b pode ser calculado a partir de $\mathbf{w}$ e das probabilidades a priori das classes. Normalmente adota-se algum critério como, por exemplo, a maximização da taxa de acerto, minimização da taxa de falsos positivos ou minimização do erro de classificação. A regra de classificação é dada por:

$$\begin{cases} \mathbf{x}_n \in C_1, \text{ se } \mathbf{w}^t \mathbf{x}_n + b > 0 \\ \mathbf{x}_n \in C_2, \text{ se } \mathbf{w}^t \mathbf{x}_n + b < 0. \end{cases} \tag{2.7}$$

Para o caso de $\mathbf{S_W}$ não ser inversível, ver [16].

2.3.2 AdaBoost

Algumas técnicas de combinação de classificadores utilizam versões reamostradas do conjunto de treino no aprendizado dos classificadores constituintes. Uma técnica bastante utilizada e com muitas variações é o *Boosting*. A ideia é que é possível gerar um classificador forte (*Strong Learner* ou *Strong Classifier*), a partir de um conjunto de classificadores fracos (*Weak Learners* ou *Weak Classifiers*) [17], [18]. Um classificador fraco possui uma probabilidade de acerto um pouco maior que 0.5, enquanto um forte possui uma probabilidade de acerto $1 - \epsilon$, para ϵ tão pequeno quanto se queira [19], [20]. Das diversas variações do *Boosting* uma das mais usadas é o *AdaBoost* (*Adaptive Boosting*). A formulação original, também conhecida como *Discrete AdaBoost* [9], é descrita a seguir.

Seja um conjunto de treinamento T composto por N amostras do tipo $(\mathbf{x}_n, y_n)$, onde $\mathbf{x}_n \in \mathbf{X}$ e $y_n \in Y$. O vetor $\mathbf{x}_n$ pertence ao espaço de características d-dimensional e $y_n \in \{-1, 1\}$ é o seu respectivo rótulo. Seja ainda um classificador

fraco, que é uma função do tipo $h_i : \mathbf{X} \rightarrow Y$. No *AdaBoost*, cada amostra de treinamento está associada a um peso que corresponde à probabilidade desta amostra ser selecionada. Estes pesos formam a distribuição $D_i(n)$.

O algoritmo é inicializado com um número C de classificadores e uma distribuição inicial D_1 uniforme. Para cada iteração $i = 1, \cdots, C$ do algoritmo, as amostras de T são ponderadas de acordo com a distribuição D_i. Esta versão ponderada é utilizada para treinar o classificador $h_i(\mathbf{x}_n)$, ou seja, seus parâmetros são ajustados de forma a obter a melhor taxa de acerto possível com relação às amostras de treinamento. O erro de classificação de h_i é computado de acordo com

$$e_i = E_{D_i}[h_t(\mathbf{x}_n) \neq y_n], \tag{2.8}$$

onde $E_{D_i}[\cdot]$ é o valor esperado com relação a distribuição D_i. Em seguida os pesos α_i são calculados por

$$\alpha_i = \frac{1}{2} \ln \left(\frac{1 - e_i}{e_i} \right). \tag{2.9}$$

Estes pesos são utilizados para ajustar a distribuição D_{i+1} de acordo com a seguinte regra:

$$D_{i+1}(n) = \frac{D_i(n) \exp\left(-\alpha_i y_n h_i(\mathbf{x}_n)\right)}{z_i}, \tag{2.10}$$

onde z_i é um fator de normalização que assegura que $\sum_n D_i(n) = 1$. Ao final do algoritmo, as saídas dos classificadores h_i são ponderadas pelos pesos α_i

$$g(\mathbf{x}_n) = \sum_{i=1}^{C} \alpha_i h_i(\mathbf{x}_n). \tag{2.11}$$

A classificação final H é dada pelo sinal desta ponderação:

$$H(\mathbf{x}_n) = sign(g(\mathbf{x}_n)). \tag{2.12}$$

O *Discrete AdaBoost* está resumido no Algoritmo 1.

Existem ainda outras variações do *Adaboost*. As mais utilizadas são *Real AdaBoost* [21], *Gentle AdaBoost* [22] e *Modest AdaBoost* [23]. A principal diferença entre estes algoritmos é a regra da atualização da distribuição D_i.

No *Real AdaBoost* os classificadores fracos retornam um valor real ao invés de um valor discreto do tipo $\{-1, 1\}$, ou seja, $h_i : \mathbf{X} \rightarrow \mathbb{R}$. Assim, a classificação é dada por $sign\left(h_i(\mathbf{x}_n)\right)$, enquanto $|h_i(\mathbf{x}_n)|$ mede a "confiança" da classificação. Definindo W_b, $b \in \{-1, 1\}$, como sendo:

$$W_b = \sum_{k: y_n h_c(\mathbf{x}_n) = b} D_i\left(k\right), \tag{2.13}$$

Entrada: O conjunto $T = \{(\mathbf{x}_n, y_n)\}$ e os pesos $D_1(N) = \frac{1}{N}$, $n = 1, \cdots, N$

1 **para cada** $i = 1, \cdots, C$ **faça**

2 Estime o classificador fraco h_i a partir do conjunto T, ponderado pela distribuição D_i;

3 Calcule o erro de classificação de h_i por $e_i = E_{D_i}[h_t(\mathbf{x}_n) \neq y_n]$;

4 Atualize os pesos $\alpha_i = \frac{1}{2} \ln\left(\frac{1-e_i}{e_i}\right)$;

5 Atualize a distribuição $D_{i+1}(n) = \frac{D_i(n) \exp\left(-\alpha_i y_n h_i(\mathbf{x}_n)\right)}{z_i}$;

Saída: O classificador $H(\mathbf{x}) = sign\left[\sum_{i=1}^{C} \alpha_i h_i(\mathbf{x})\right]$

Algoritmo 1: *Discrete Adaboost*

a regra de atualização dos pesos no *Real AdaBoost* é:

$$\alpha_i = \frac{1}{2} \ln\left(\frac{W_{+1}}{W_{-1}}\right). \tag{2.14}$$

O *Gentle Adaboost* é uma versão mais robusta e estável do *Real Adaboost*. Esta versão é usada, por exemplo, no detector de objetos Viola-Jones (veja Seção 2.4). A atualização dos pesos é feita de acordo com a seguinte expressão:

$$\alpha_i = \frac{1}{2} \ln\left(\frac{W_{+1} - W_{-1}}{W_{-1} + W_{-1}}\right). \tag{2.15}$$

Por fim, no *Modest Adaboost* uma nova forma da ajuste dos pesos é proposta. Esta modificação acarreta em um menor erro de generalização às custas de, possivelmente, um erro de treinamento mais alto. Sejam as definições de $\overline{W_b}$ e $\overline{D_i}$:

$$\overline{D}_i(k) = (1 - D_i(k))\, \overline{z}_i, \tag{2.16}$$

$$\overline{W}_b = \sum_{k: y_n h_i(\mathbf{x}_n) = b} \overline{D}_i\,(k)\,, \tag{2.17}$$

onde $\overline{z}_i$ é uma fator de normalização que assegura que $\sum_k \overline{D}_i(k) = 1$. A regra de atualização dos pesos é dada por:

$$\alpha_i = W_{+1}\left(1 - \overline{W}_{+1}\right) - W_{-1}\left(1 - \overline{W}_{-1}\right). \tag{2.18}$$

Neste trabalho o *AdaBoost* foi utilizado com o auxílio de uma *toolbox* para *Matlab*, chamada de *GML AdaBoost Matlab Toolbox*. Esta é uma biblioteca gratuita e está disponível em [24]. Ela contém códigos do *Real*, do *Gentle* e do *Modest Adaboost*.

2.3.3 Máquina de vetores suporte (SVM)

O SVM (*Support Vector Machines*), proposto por Vapnik, é uma teoria de aprendizado de máquina baseada em aprendizado estatístico, que tem sido amplamente utilizado para resolver problemas de regressão e de classificação [25], [26], [27], [28]. A principal ideia é que dois conjuntos de dados podem ser separados por um hiperplano que maximiza a distância entre ele e as amostras mais próximas de cada conjunto. Esta distância entre as amostras mais próximas é conhecida como margem. Caso isto não seja possível, os conjuntos são ditos não-linearmente separáveis e duas abordagens podem ser adotadas. A primeira delas é a utilização de uma margem "suave", na qual permite-se que algumas amostras estejam na margem ou estejam contidas na porção do espaço que corresponde à outra classe. A outra abordagem está relacionada com o mapeamento não linear do conjunto de dados em um espaço de dimensão mais elevada, onde a separação por um hiperplano seja viável. A seguir, cada um destas abordagens serão apresentadas.

Caso o conjunto de dados possa ser separado por um hiperplano, a abordagem com margens rígidas pode ser empregada [25]. Para ilustrar o método, seja um conjunto de treinamento T composto por N amostras do tipo $(\mathbf{x}_n, y_n)$, onde $\mathbf{x}_n \in \mathbf{X}$ e $y_n \in Y$. O vetor $\mathbf{x}_n$ pertence ao espaço de características d-dimensional e $y_n \in \{-1, 1\}$ é o seu respectivo rótulo.

Seja ainda um hiperplano $f(\mathbf{x})$, dado por

$$f(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + b, \tag{2.19}$$

onde $\mathbf{w}$ é a direção normal ao hiperplano e $|b|$ é a distancia do hiperplano à origem ($\|\mathbf{w}\| = 1$).

Dado um hiperplano separador, a margem é definida como sendo a distância entre as amostras mais próximas de cada classe e o hiperplano. Estas amostras são denominadas vetores suporte e estão sobre dois hiperplanos paralelos que delimitam a margem. Estes hiperplanos são $H_{+1} : \mathbf{w}^t\mathbf{x} + b = 1$, para as amostras "positivas" e $H_{-1} : \mathbf{w}^t\mathbf{x} + b = -1$, para as amostras "negativas". A Figura 2.4 ilustra estes conceitos para o caso linearmente separável em duas dimensões.

Resta agora encontrar o hiperplano separador que possui a maior margem. Para isto, todas as amostras devem satisfazer as seguintes condições:

$$\begin{cases} \mathbf{w}^t\mathbf{x}_n + b \geq +1, \text{ se } y_n = +1 \\ \mathbf{w}^t\mathbf{x}_n + b \leq -1, \text{ se } y_n = -1. \end{cases} \tag{2.20}$$

Como a distância l entre H_{+1} e H_{-1} é $l = \frac{2}{\|\mathbf{w}\|}$ (veja a Figura 2.4), para encontrar o hiperplano que maximiza a margem, basta minimizar $\|\mathbf{w}\|$, de acordo com as con-

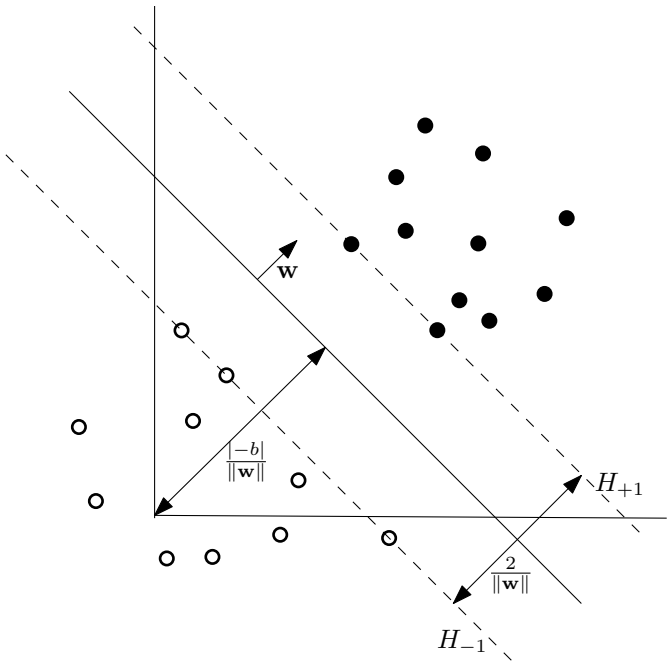


Figura 2.4: Ilustração do caso linearmente separável. A linha sólida representa o hiperplano separador $\mathbf{w}^t\mathbf{x} + b = 0$. A margem é definida como sendo a região delimitada pelos hiperplanos $H_{+1} : \mathbf{w}^t\mathbf{x} + b = 1$ e $H_{-1} : \mathbf{w}^t\mathbf{x} + b = -1$. As 5 amostras que estão postas sobre os hiperplanos H_{+1} e H_{-1} são os vetores suporte.

dições da Equação (2.20). Com isto, chega-se ao seguinte problema de otimização:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2, \tag{2.21}$$

restrito à

$$y_n(\mathbf{w}^t\mathbf{x}_n + b) - 1 \geq 0, \ \forall(\mathbf{x}_n, y_n) \in T. \tag{2.22}$$

Este é um problema de otimização quadrático com restrições lineares. A solução para este problema é mais fácil se a formulação lagrangiana for adotada. Para tanto, sejam os multiplicadores de Lagrange α_n, $n = 1, \ldots, N$. Incorporando a restrição ao problema de otimização chega-se à equação conhecida como forma primal:

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} \alpha_n(y_n\left(\mathbf{w}^t\mathbf{x}_n + b\right) - 1) + \sum_{n=1}^{N} \alpha_n. \tag{2.23}$$

Tomando as derivadas parciais de L_p com relação a $\mathbf{w}$ e b igualando e a zero, obtêm-se, respectivamente:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n, \tag{2.24}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0. \tag{2.25}$$

Substituindo estas duas expressões na Equação (2.23), uma nova formulação, conhecida como forma dual é obtida (os índices foram trocados sem perda de generalidade):

$$L_d = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j, \tag{2.26}$$

$$\alpha_i \geq 0. \tag{2.27}$$

A solução ótima $\boldsymbol{\alpha}^*$ do problema dual deve satisfazer as condições de Karush-Kuhn-Tucker, que incluem as Equações (2.24), (2.25) e (2.27), além da seguinte expressão [29]:

$$\alpha_n \left[y_n(\mathbf{w}^t \mathbf{x}_n + b) - 1 \right] = 0, \quad \forall n. \tag{2.28}$$

Com isto $\alpha_n > 0$ somente para os pontos que estão sobre os hiperplanos H_{+1} e H_{-1} (os vetores suporte). Para os outros pontos, as condições de Karush-Kuhn-Tucker são obedecidas somente se $\alpha_n = 0$. Atendidas as condições e dada a solução do problema dual $\boldsymbol{\alpha}^*$, o hiperplano ótimo, definido por $\mathbf{w}^*$ e b^* pode ser obtido diretamente das Equações (2.24) e (2.28), respectivamente. A regra final de classificação pode ser obtida pelas seguintes expressões:

$$g(\mathbf{x}) = sign\left(\sum_{i \in SV} y_i \alpha_i^* \mathbf{x}_i^t \mathbf{x} + b^* \right), \tag{2.29}$$

$$b^* = \frac{1}{n_{SV}} \sum_{i \in SV} \left(\frac{1}{y_i} - \sum_{j \in SV} \alpha_j^* y_j \mathbf{x}_j^t \mathbf{x}_i \right), \tag{2.30}$$

onde SV é o conjunto formado pelos vetores suporte e n_{SV} é a quantidade de elementos deste conjunto. A expressão da Equação (2.30) foi obtida a partir da Equação (2.28) computando-se a média sobre todos os vetores suporte.

Na prática, dificilmente se obtém um problema linearmente separável. As principais fontes de não linearidade são ruído, *outliers* e a própria natureza não linear do problema. No SVM, uma maneira de lidar com isso é permitindo que alguns pontos possam estar na margem ou, até mesmo, na região da outra classe. Isto é feito introduzindo uma "folga" ξ_n nas condições da Equação (2.20), o que resulta na versão do SVM com margens suaves [30]:

$$\begin{cases} \mathbf{w}^t \mathbf{x}_n + b \geq +1 - \xi_n, \text{ se } y_n = +1 \\ \mathbf{w}^t \mathbf{x}_n + b \leq -1 - \xi_n, \text{ se } y_n = -1, \end{cases} \tag{2.31}$$

onde $\xi_n \geq 0$ para $n = \{1, \cdots, N\}$. A Figura 2.5 ilustra o SVM com margem suave.

Estas condições permitem que alguns pontos fiquem entre as margens, caso $0 \leq \xi_n \leq 1$, ou estejam na região correspondente à outra classe (erro), caso $\xi_n > 1$.

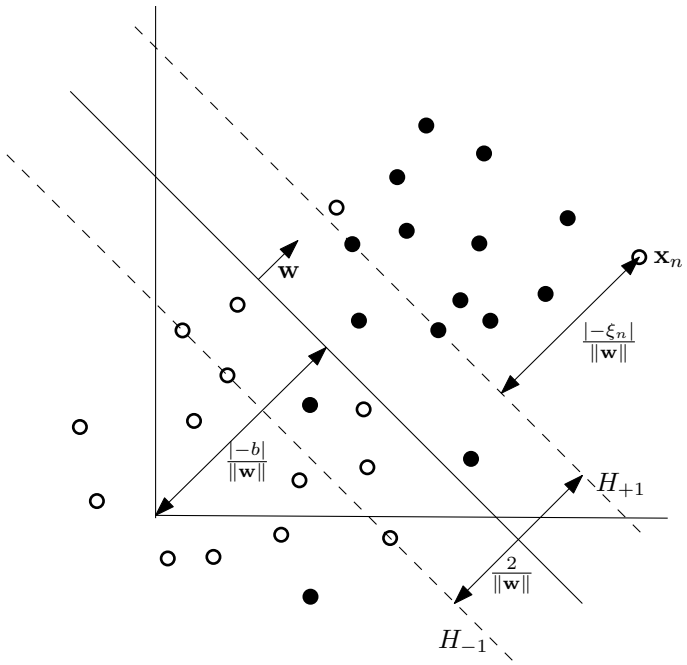


Figura 2.5: Ilustração do caso SVM com margem suave. A linha sólida representa o hiperplano separador e as tracejadas representam os hiperplanos que delimitam a margem. No caso de margens suaves alguns vetores suporte podem estar na margem ou na região da outra classe.

Com isto, o problema de otimização pode ser reformulado de acordo com a seguinte expressão

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{n=1}^{N} \xi_n\right), \tag{2.32}$$

com a restrição dada por

$$y_n(\mathbf{w}^t\mathbf{x}_n + b) - 1 + \xi_n \geq 0, \tag{2.33}$$

onde C é um parâmetro livre que define uma penalidade sobre o erro.

Novamente o problema de otimização é quadrático. Sendo μ_n multiplicadores de Lagrange, a forma primal é:

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} \alpha_n(y_n\left(\mathbf{w}^t\mathbf{x}_n + b\right) - 1 + \xi_n) - \sum_{n=1}^{N} \mu_n\xi_n, \tag{2.34}$$

com a restrição dada por

$$\alpha_n,\ \mu_n \geq 0. \tag{2.35}$$

A solução é obtida através passos similares aos da formulação com margens rígidas. Tomando as derivadas parciais de L_p com relação a $\mathbf{w}$, b e ξ e igualando a

zero, chega-se às seguintes expressões, respectivamente:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n, \tag{2.36}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0, \tag{2.37}$$

$$C - \alpha_n - \mu_n = 0. \tag{2.38}$$

Substituindo estas expressões na Equação (2.34) obtém-se a forma dual

$$L_d = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j, \tag{2.39}$$

com as seguintes restrições:

$$0 \leq \alpha_n \leq C, \tag{2.40}$$

$$\sum_{n=1}^{N} \alpha_n y_n. \tag{2.41}$$

As condições de Karush-Kuhn-Tucker são dadas pelas Equações (2.33), (2.35), (2.36), (2.37), (2.38) e pelas expressões a seguir [29]:

$$\alpha_i (y_i \left(\mathbf{w}^t \mathbf{x}_i + b \right) - 1 + \xi_n) = 0, \tag{2.42}$$

$$\mu_n \xi_n = 0. \tag{2.43}$$

A partir da solução do problema dual $\boldsymbol{\alpha}^*$ é possível obter o $\mathbf{w}^*$ ótimo da Equação (2.36) e o b^* ótimo da condição dada na Equação (2.42).

Vale ressaltar que, assim como no caso anterior, $\alpha_n^* \geq 0$ apenas para os vetores suporte. Contudo, neste caso, distinguem-se 4 tipos de vetores suporte. Se $\alpha_n^* < C$ então, obrigatoriamente $\xi_n^* = 0$ e o respectivo vetor suporte está sobre a margem. Se $\alpha_n^* = C$ há três casos. Erro, caso $\xi_n^* > 0$; está entre as margens, caso $0 < \xi_n^* \leq 1$; está sobre as margens se $\xi_n^* = 0$.

Assim como no caso de margens rígidas, a regra final de classificação pode ser dada por:

$$g(\mathbf{x}) = sign \left(\sum_{i \in SV} y_i \alpha_i^* \mathbf{x}_i^t \mathbf{x} + b^* \right), \tag{2.44}$$

$$b^* = \frac{1}{n_{\tilde{SV}}} \sum_{i \in \tilde{SV}} \left(\frac{1}{y_i} - \sum_{j \in SV} \alpha_j^* y_j \mathbf{x}_j^t \mathbf{x}_i \right), \tag{2.45}$$

na qual $\tilde{SV}$ é o conjunto formado pelos vetores suporte que satisfazem $0 < \alpha_i < C$ e $n_{\tilde{SV}}$ é a quantidade de elementos deste conjunto.

Nos casos em que a abordagem com margens suaves não apresenta resultados satisfatórios, pode-se utilizar a versão não linear do SVM [31]. Esta técnica é baseada no mapeamento não linear do conjunto de dados em um espaço de dimensão suficientemente mais alta, possivelmente infinita, onde os dados podem ser separados por um hiperplano [32]. Este mapeamento Φ é dado por:

$$\Phi : \mathbb{R}^d \rightarrow \mathcal{H}, \tag{2.46}$$

onde $\mathcal{H}$ representa o espaço de dimensão mais alta.

Adotando margens "suaves" e procedimentos similares aos anteriores, a forma dual do problema é:

$$L_d = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \Phi^t(\mathbf{x}_i) \Phi(\mathbf{x}_j), \tag{2.47}$$

com as seguintes restrições

$$0 \leq \alpha_n \leq C, \tag{2.48}$$

$$\sum_{n=1}^{N} \alpha_n y_n. \tag{2.49}$$

A regra de classificação é dada por:

$$g(\mathbf{x}) = sign \left(\sum_{i \in SV} y_i \alpha_i^* \Phi^t(\mathbf{x}_i) \Phi(\mathbf{x}) + b^* \right), \tag{2.50}$$

$$b^* = \frac{1}{n_{\tilde{SV}}} \sum_{i \in \tilde{SV}} \left(\frac{1}{y_i} - \sum_{j \in SV} \alpha_j^* y_j \Phi^t(\mathbf{x}_j) \Phi(\mathbf{x}_i) \right), \tag{2.51}$$

na qual SV é o conjunto formado pelos vetores suporte e n_{SV} é a quantidade de elementos deste conjunto.

Entretanto, efetuar o mapeamento Φ pode ser uma tarefa muito difícil ou até mesmo inviável. Como no treinamento os dados aparecem somente na forma de produtos internos (vide Equações (2.47), (2.50) e (2.51)), basta que se conheça o produto interno no espaço transformado $\Phi^t(\mathbf{x}_i)\Phi(\mathbf{x}_j)$. A este produto interno dá-se

o nome de função *kernel* $K(\mathbf{x}_i, \mathbf{x}_j)$:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi^t(\mathbf{x}_i)\Phi(\mathbf{x}_j). \tag{2.52}$$

Dada uma função *kernel* $K(\mathbf{x}_i, \mathbf{x}_j)$ é necessário garantir que o problema de otimização, apresentado na Equação (2.47), permaneça convexo. Para tanto a função *kernel* deve satisfazer a condição de Mercer [33]: uma função *kernel* $K(\mathbf{x}, \mathbf{y})$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, é um produto interno em um espaço de dimensão mais elevada se e somente se $K(\mathbf{x}, \mathbf{y}) = K(\mathbf{y}, \mathbf{x})$ e

$$\int K(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0, \tag{2.53}$$

onde $f(\mathbf{x})$ é uma função qualquer que satisfaz

$$\int f(\mathbf{x}^2) d\mathbf{x} < \infty. \tag{2.54}$$

Para mais detalhes veja [16], [25] [26], [29] e [34].

Os tipos de *kernel* mais utilizados são os polinomiais, os sigmoidais e os gaussianos, também conhecidos como RBF (*Radial Basis Functions*). Um resumo sobre os principais tipos de *kernel* está apresentado na seguinte tabela:

Tabela 2.1: Tipos de funções *kernel* mais utilizadas

Não-Linearidade	Função $K(\mathbf{x}, \mathbf{y})$
polinomial	$(1 + \mathbf{x}^t\mathbf{y})^d$
sigmoidal	$\tanh(k\mathbf{x}^t\mathbf{y} - \delta)$
gaussiano	$\exp\left(-\lvert\mathbf{x} - \mathbf{y}\rvert^2/\sigma^2\right)$

Neste trabalho o SVM é utilizado através de uma versão para *Matlab* de uma biblioteca gratuita, o SVMlight [35]. Esta versão para *Matlab* está disponível em [10].

2.4 Detecção de faces através do algoritmo Viola-Jones

Detecção de faces é uma etapa inicial em muitos sistemas, incluindo os sistemas de detecção de características faciais. Existem diversas técnicas que podem ser empregadas para tal tarefa, mas o detector de objetos Viola-Jones tem sido muito utilizado recentemente [13]. O Viola-Jones é capaz de detectar faces com precisão, alta taxa de acerto, baixa taxa de falsos positivos e baixo custo computacional. O algoritmo é composto de três partes. A primeira delas é a representação da imagem em um espaço de características baseadas nos filtros de Haar. Isto é feito

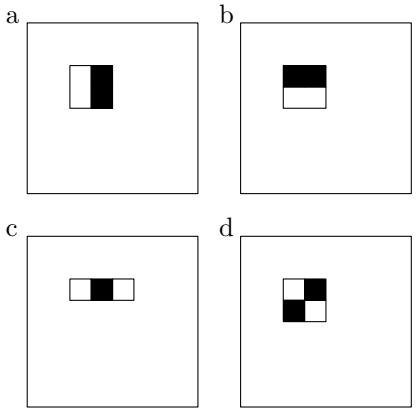


Figura 2.6: Os quatro tipos de características utilizadas na abordagem original do detector de objetos Viola-Jones

com o auxílio da "imagem integral". A segunda é a montagem de um classificador baseado em *Boosting* capaz de selecionar as características mais relevantes. Por fim é feita uma combinação em cascata destes classificadores de modo garantir bom desempenho e velocidade de processamento. A etapa de detecção de faces neste trabalho é feita através do método de Viola-Jones. Por isto este método é descrito com mais detalhes a seguir.

No Viola-Jones, a representação dos dados de treinamento no espaço de características é obtida através da "imagem integral" $I(m, n)$, definida por:

$$I(m,n) = \sum_{m \geq m', n \geq n'} g(m', n'), \tag{2.55}$$

onde $g(m, n)$ é uma imagem de tamanho $L \times C$, $1 \leq m, m' \leq L$ e $1 \leq n, n' \leq C$.

A principal vantagem desta representação é que ela possibilita calcular a soma dos elementos de qualquer retângulo contido na imagem com apenas quatro pontos de $I(m, n)$. Além disso é possível obtê-la com apenas uma varredura na imagem [13].

Um conjunto de características, dado pela diferença entre a soma dos *pixels* de regiões retangulares, é facilmente obtido através da imagem integral. Este tipo de característica é semelhante ao produto interno com as *wavelets* de Haar e por isso são também conhecidas como *Haar-like features*. Na abordagem original de Viola-Jones foram utilizados quatro tipos de características, como ilustrado na Figura 2.6, onde o valor de uma dada característica é a diferença entre a soma dos pixeis da região branca e a soma dos pixeis da região preta.

Em versões mais recentes, um conjunto estendido de características é utilizado [36]. Este novo conjunto inclui um novo tipo de característica e versões rotacionadas das características utilizadas na abordagem original. Além disto, a característica de quatro retângulos não é utilizada. O conjunto extendido está ilustrado na Figura

2.7. Detalhes de como calcular as características rotacionadas são apresentados em [36].

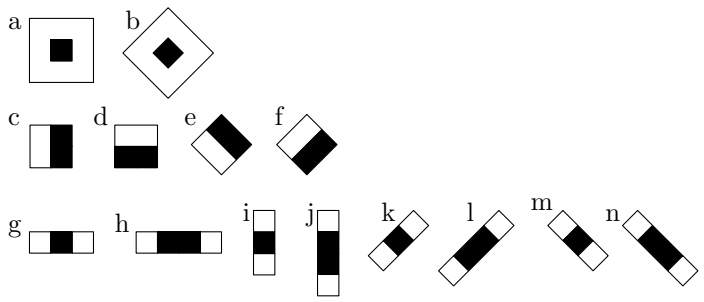


Figura 2.7: Conjunto extendido de características

O conjunto de treinamento do Viola-Jones é formado por amostras do tipo $(\mathbf{x}_n, y_n)$ com N amostras, onde $\mathbf{x}_n$ é uma imagem de dimensão 24×24 e $y_n = \{0, 1\}$ é o rótulo de classe. Neste caso $y_n = 1$ corresponde a uma imagem de face e $y_n = 0$ a uma imagem de não-face. A dimensão das amostras faz com que o número total de características seja maior que $180,000$. Isto torna necessário a seleção das características mais relevantes. Para tanto é utilizada uma versão do *AdaBoost* conhecida como *Gentle Adaboost* (para mais detalhes veja a Seção 2.3.2).

Em cada iteração do *AdaBoost*, um conjunto de classificadores fracos h_j é ajustado para minimizar o erro de classificação. Cada um destes classificadores corresponde a uma característica $f_j(\mathbf{x}_n)$, onde $j = 1, \cdots, J$ e J é o total de características. Dado um limiar θ_j e uma paridade p_j, a regra de classificação pode ser dada por:

$$h_j(\mathbf{x}_n) = \begin{cases} 1, \text{ se } p_j f_j(\mathbf{x}_n) > p_j \theta_j \\ 0, \text{ caso contrário,} \end{cases} \tag{2.56}$$

onde a paridade p_j indica a direção da desigualdade.

Em problemas práticos as taxas alcançadas por esta abordagem não são satisfatórias. Por isso é feita uma combinação de classificadores na forma de uma árvore degenerada, também conhecida como cascata de classificadores. Nesta combinação, cada nó (ou estágio) é invocado sequencialmente e corresponde a um classificador *AdaBoost* ajustado para obter uma taxa de falso negativo próxima a zero. Para reduzir o tempo de processamento o número de características selecionadas em cada estágio é menor que no estágio seguinte. Isto faz com que os estágios sejam sequencialmente mais complexos e o número de amostras diminua rapidamente à medida que eles são invocados. A Figura 2.8 ilustra a classificação em cascata feita pelo Viola-Jones.

Na detecção, como não se sabe a posição nem o tamanho da face na imagem de teste, as características selecionadas no treinamento são escalonadas do tamanho

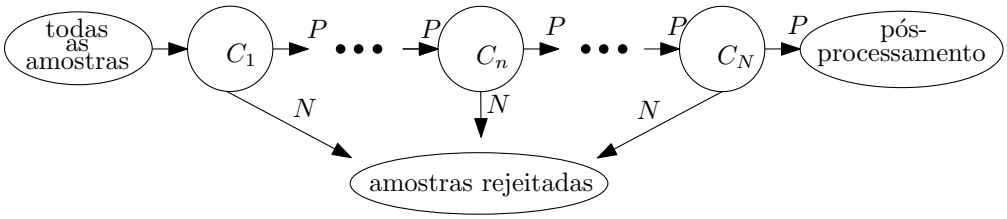


Figura 2.8: Cascata de classificadores utilizada no Viola-Jones. Em cada estágio da cascata, as amostras classificadas negativamente (N) são rejeitadas, enquanto as classificadas positivamente (P) são passadas ao estágio seguinte

mínimo até o tamanho da imagem, nos respectivos estágios da cascata. Estas versões escalonadas são aplicadas em todas as sub-janelas possíveis dentro da imagem de teste. A maioria das sub-janelas são rejeitadas nos primeiros estágios enquanto os últimos são responsáveis por classificar as sub-janelas mais difíceis.

Neste trabalho o Viola-Jones foi utilizado através de uma biblioteca de código aberto, a OpenCV, disponível em [37].

2.5 Métodos recentes de reconhecimento de características faciais

A detecção de faces em imagens é alvo de estudos há vários anos. Contudo, recentemente a detecção de pontos fiduciais em faces passou a ser também investigada. O principal motivo disto é que pontos fiduciais em faces podem ser utilizados em diversos outros sistemas, tais como sistemas de segurança, de reconhecimento de expressões, de rastreamento e de modelagem 3D, dentre outros.

O reconhecimento de pontos fiduciais em faces não é uma tarefa fácil e geralmente é feita em diversas etapas. Algumas abordagens têm sido propostas neste sentido e já existem algumas soluções comerciais, como pode ser visto em [38] [39]. Na literatura é possível encontrar soluções que utilizam diversas combinações de técnicas.

Em [40] é proposta uma técnica que combina o SVM com o ASM (*Active Shape Model*) para encontrar pontos fiduciais. Já em [41] é feita uma restrição do espaço de busca através de gradientes direcionais e os pontos são localizados por *template matching*. Já em [42] as *wavelets* de Gabor são utilizadas para localizar os pontos fiduciais.

Embora existam diversas outras abordagens é possível destacar duas, por serem utilizadas em muitos trabalhos. As que utilizam modelos ativos e as que utilizam cascatas de classificadores baseados em *Boosting*. A seguir estas técnicas são descritas com um pouco mais de detalhes.

2.5.1 Sistemas baseados em modelos ativos

O objetivo das técnicas que utilizam modelos ativos é a abusca pelo melhor casamento entre a imagem e algum modelo para as características. É comum o uso de versões do modelo ativo de forma ou ASM (*Active Shape Model*) e do modelo ativo de aparência ou AAM (*Active Appearance Model*) para detectar pontos fiduciais. O primeiro faz uma busca iterativa na imagem pelo melhor casamento com um dado modelo de forma. O segundo, busca o melhor casamento com um modelo combinado de textura e forma.

Em [40] uma versão do ASM é empregada. Já no trabalhos [4], [5], [6] e [7], é possível encontrar diversas variações do AAM. Em todos os casos o método é utilizado no reconhecimento de pontos fiduciais em faces.

2.5.2 Sistemas baseados em cascata de classificadores *Boosting*

A cascata de classificadores baseados em *Boosting*, geralmente o *AdaBoost*, é outra classe de métodos que tem sido muito empregada no reconhecimento de pontos fiduciais em faces. Nesta técnica, a cascata de classificadores é utilizada para selecionar um conjunto de características extraídas das imagens. É comum a extração destas características através de *wavelets* de Gabor ou *wavelets* de Haar. Técnicas que utilizam esta última, são, em geral, variações do detector Viola-Jones [13] para localização de pontos fiduciais.

Em [1] a cascata de classificadores *AdaBoost* é utilizada para selecionar características obtidas com o auxílio de *wavelets* de Gabor. Já nos trabalhos desenvolvidos por [2] e [3], uma restrição do espaço de busca é feita através de um modelo de mistura de gaussianas e em seguida as características extraídas com o auxílio de filtros de Haar são submetidas à cascata.

Capítulo 3

Método proposto

Características faciais são um conjunto de informações que definem a face. Existem inúmeras características faciais, como por exemplo, largura da boca, espaço entre os olhos ou tamanho do nariz, por exemplo. Este tipo de informação pode ser muito útil em sistemas de segurança, de reconhecimento de expressões, de modelagem 3D, dentre outros.

Neste trabalho é proposto um sistema de detecção de características faciais. Isto é feito através da detecção de pontos de controle dispostos sobre regiões salientes da face, como canto dos olhos, canto da boca ou a borda lateral do nariz. Este pontos são também conhecidos como pontos fiduciais.

O sistema proposto é composto de cinco etapas. A primeira delas é a segmentação da face utilizando o detector de objetos Viola-Jones, apresentado na Seção 2.4. Em seguida, um algoritmo de correção de iluminação é aplicado como pré-processamento. A posição de cada ponto fiducial é considerada independente da posição dos demais e uma restrição da região de busca é feita através da aplicação de um modelo probabilístico para pontos fiduciais. A classificação é feita através de uma combinação de classificadores em cascata que detecta cada ponto fiducial na imagem dada a respectiva região de busca. Cada estágio desta cascata é composto de um conjunto de Detectores por Produto Interno (DPI) e um conjunto de limiares (determinados por classificadores baseados em *Boosting* ou discriminantes lineares de Fisher). Cada ponto fiducial está associado a um detector e um limiar por estágio da cascata. Por fim, um algoritmo de agrupamento é aplicado na saída da cascata como uma etapa de pós-processamento. O sistema proposto está ilustrado na Figura 3.1.

Na Seção 3.1 é apresentado o algoritmo de correção de iluminação utilizado. Na Seção 3.2 é apresentado o modelo probabilístico utilizado na restrição da região de busca dos pontos fiduciais. O DPI está descrito na Seção 3.3. O algoritmo de grupamento usado como pós-processamento está descrito na seção 3.4.

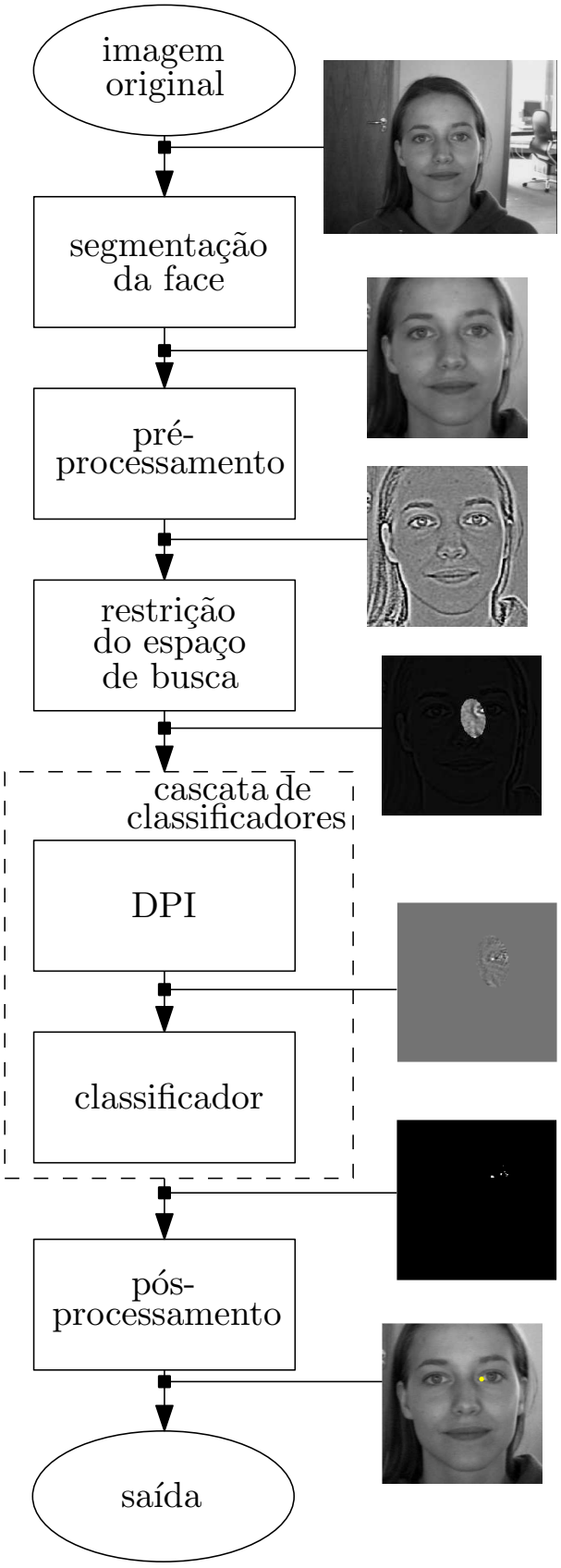


Figura 3.1: Diagrama em blocos do sistema de detecção de características faciais proposto

3.1 Pré-processamento

De uma maneira geral, as faces segmentadas pelo algoritmo Viola-Jones (descrito na Seção 2.4) possuem pequenas rotações, variações de escala e variação de iluminação. É desejado um sistema que possua robustez em relação a estas variantes. Como a variação da inclinação das faces é muito pequena, o desempenho global do sistema não deve ser afetado de uma maneira significativa. Por outro lado, o problema de variação de escala pode ser resolvido simplesmente redimensionando todas as faces de saída do Viola-Jones para o mesmo tamanho. Contudo, não existe uma maneira simples de reduzir os efeitos da iluminação e um pré-processamento adequado deve ser aplicado. Nesta Seção é apresentado o método de correção de iluminação utilizado neste trabalho.

O reconhecimento de faces em imagens com condições não controladas de iluminação não é uma tarefa fácil, principalmente em sistemas práticos [43]. Com o reconhecimento de pontos fiduciais não é diferente. Como pode ser visto em [44], existem duas maneiras básicas de abordar o problema. A primeira, extrai um modelo de variação de iluminação a partir das amostras de treinamento e então generaliza-o para aplicar em novas imagens. A segunda utiliza técnicas convencionais de processamento de imagens que são aplicadas às imagens de treino para transformá-las em versões com uma variação de iluminação menor. Esta segunda abordagem apresenta a vantagem de não precisar de grandes conjuntos de treinamento, além da simplicidade.

Neste trabalho foi adotado um sistema de correção de iluminação, proposto em [44], que segue esta segunda abordagem. O objetivo é reduzir o efeito da variação de iluminação, do sombreamento e do gradiente de iluminação sem destruir as informações necessárias ao reconhecimento. Neste método uma sequência de passos são aplicados às imagens de face antes do treinamento dos classificadores. Estes passos são: correção de gama, filtragem por diferença de gaussianas e a equalização do contraste. O resultado do algoritmo de correção de iluminação para algumas imagens da base BioID está ilustrado na Figura 3.2.

A correção de gama tem por objetivo aumentar a faixa dinâmica da imagem nas regiões mais escuras e diminuir nas mais iluminadas. Dados os níveis de cinza $I(m, n)$ dos pixeis de uma imagem de tamanho $M \times N$ $(0 \leq m \leq M; 0 \leq n \leq N)$, a correção de gama é uma transformação não linear do tipo:

$$I \leftarrow I^{\gamma},\ \gamma > 0, \tag{3.1}$$

onde $\gamma \in [0, 1]$ é um parâmetro definido pelo usuário. Embora a correção de gama tenda a amplificar o ruído nas regiões escuras, um bom compromisso é um valor de γ no intervalo $[0, 0.5]$ [44]. Neste trabalho foi utilizado o valor recomendado por [44]

Figura 3.2: Exemplos de aplicação do algoritmo de correção de iluminação para algumas imagens da base BioID. Na coluna superior algumas faces extraídas das imagens originais. Na coluna inferior o resultado da correção de iluminação para estas imagens.

de $\gamma = 0.2$.

Embora o efeito de sombra seja diminuído com a correção de gama, os gradientes de sombra permanecem. Estes efeitos são gerados por componentes de baixa frequência. Por outro lado, componentes de alta frequência, como o ruído, também são indesejáveis. Uma das formas de aplicar um filtro passa-faixa é através da filtragem por diferença de gaussianas, também conhecido como filtro DoG (*Difference of Gaussians*). Dada uma gaussiana bidimensional $g(m, n)$ de desvio padrão σ:

$$g(m,n) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{m^2+n^2}{2\sigma^2}\right), \tag{3.2}$$

e uma imagem I, a filtragem DoG pode ser obtida da seguinte forma:

$$\begin{aligned} DoG(m,n) &= g_{\sigma_1} * I - g_{\sigma_2} * I \\ &= (g_{\sigma_1} - g_{\sigma_2}) * I \\ &= G * I, \end{aligned} \tag{3.3}$$

onde G é:

$$G(m,n) = \frac{1}{\sqrt{2\pi}}\left(\frac{1}{\sigma_1}exp\left(-\frac{m^2+n^2}{2\sigma_1^2}\right) - \frac{1}{\sigma_2}exp\left(-\frac{m^2+n^2}{2\sigma_2^2}\right)\right). \quad (3.4)$$

A gaussiana com desvio padrão menor é responsável por filtrar os detalhes de alta frequência, enquanto a gaussiana com desvio padrão maior é responsável por filtrar os detalhes de baixa frequência. Neste trabalho foram utilizados os valores recomendados por [44]: $\sigma_1 = 1.0$ e $\sigma_2 = 2.0$.

O passo final da correção de iluminação é a equalização do contraste. Isto é feito com em uma sequência de passos que basicamente re-escala os níveis de cinza da imagem:

$$I(m,n) \leftarrow \frac{I(m,n)}{(media(|I(m',n')|^{\alpha}))^{1/\alpha}} \quad (3.5)$$

$$I(m,n) \leftarrow \frac{I(m,n)}{(media(min(\tau,|I(m',n')|)^{\alpha})^{1/\alpha}} \quad (3.6)$$

$$I(m,n) \leftarrow \tau \tanh(I(m,n)/\tau), \quad (3.7)$$

onde $media(\cdot)$ é significa uma operação de média, α é um parâmetro que reduz a influência dos valores altos de luminância e τ é um limiar que trunca estes valores altos de luminância após o primeiro passo. Por fim a tangente hiperbólica comprime valores extremos e limita I ao intervalo $(-\tau, \tau)$. Os valores utilizados foram $\alpha = 0.1$ e $\tau = 10$ [44].

Na Figura 3.3 pode ser observado um diagrama em blocos do sistema de correção de iluminação utilizado neste trabalho. Um código para *Matlab* do método descrito nesta seção está disponível em [45].

3.2 Modelo probabilístico para pontos fiduciais

Para modelar a distribuição espacial dos pontos fiduciais, as imagens de face provenientes do detector de objetos Viola-Jones são consideradas centralizadas. Adotando o centro destas imagens como a referência e reamostrando-as para que todas tenham as mesmas dimensões, os pontos fiduciais se agrupam em regiões. O modelo gaussiano foi adotado neste trabalho e os parâmetros da distribuição foram obtidos das amostras de treinamento.

Supondo a posição do ponto fiducial uma variável aleatória bidimensional $\mathbf{X}$ com N realizações $\mathbf{x}_n$ referentes a um ponto fiducial, a média é:

$$\boldsymbol{\mu}_{\mathbf{X}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n, \quad (3.8)$$

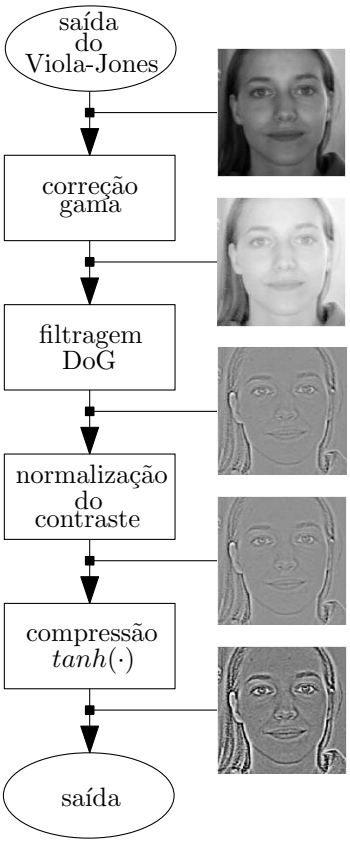


Figura 3.3: Diagrama do sistema de correção de iluminação [44].

e a matriz de covariância é dada por:

$$\mathbf{\Sigma_X} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_\mathbf{X})(\mathbf{x}_n - \boldsymbol{\mu}_\mathbf{X})^t. \tag{3.9}$$

Com isto, a distribuição de probabilidade do ponto fiducial estar na posição $\mathbf{x}$ pode ser obtida a partir da posição $p_\mathbf{X}$ como segue:

$$p_\mathbf{X}(\mathbf{x}) = \frac{1}{2\pi\sqrt{|\mathbf{\Sigma_X}|}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_\mathbf{X})^t\mathbf{\Sigma}_\mathbf{X}^{-1}(\mathbf{x}-\boldsymbol{\mu}_\mathbf{X})\right]. \tag{3.10}$$

A partir da distribuição $p_\mathbf{X}(\mathbf{x})$ é possível selecionar uma região da imagem com uma grande probabilidade de se encontrar um ponto fiducial. Para determinar esta região utilizam-se as coordenadas do ponto fiducial do conjunto de treinamento que maximiza a distância de Mahalanobis [46], dada por

$$r = \sqrt{(\mathbf{x}-\boldsymbol{\mu}_\mathbf{X})^t\,\mathbf{\Sigma}_\mathbf{X}^{-1}\,(\mathbf{x}-\boldsymbol{\mu}_\mathbf{X})}. \tag{3.11}$$

Neste trabalho, uma margem de segurança de 5% foi acrescida à distância de Mahalanobis máxima $r_{\max}$. Este modelo determina uma região elíptica. No restante do

sistema, são considerados candidatos a ponto fiducial apenas os pontos que estiverem dentro desta região. A esta região está associada uma probabilidade de erro que pode ser obtida a partir da função de erro complementar $erfc(\cdot)$:

$$erfc(1,05r_{\max}) = \frac{2}{\sqrt{\pi}} \int_{1,05r_{\max}}^{\infty} e^{-t^2} dt. \tag{3.12}$$

Existem outras maneiras de restringir o espaço de busca. Uma delas seria a utilização de um modelo que leva em consideração medidas antropométricas. Contudo, o objetivo principal deste trabalho é apresentar uma solução para o problema de detecção de pontos fiduciais em faces usando detectores baseados em filtros de correlação. A investigação de técnicas mais eficientes para a restrição do espaço de busca foge ao escopo deste trabalho.

3.3 Detector por produto interno - DPI

A filtragem de correlação é uma técnica de filtragem casada no domínio da frequência, na qual o reconhecimento é feito a partir da correlação cruzada entre o filtro projetado e a representação da amostra de teste no domínio da frequência, obtida a partir da DFT (*Discrete Fourier Transform*). O sinal de saída apresenta um pico caso a amostra seja correlacionada com o filtro [47]. Uma das vantagens desta técnica é a tolerância a pequenas variações do padrão que se deseja detectar. A filtragem de correlação e algumas de suas variações como o CFA (*Class-dependence Feature Analysis*) têm sido utilizada no reconhecimento de objetos, como por exemplo, faces humanas [47], [48], [49] e [50]. Neste trabalho é proposto um método baseado em filtragem de correlação denominado detector por produto interno (DPI). No DPI, o classificador (ou filtro) obtido é um vetor $\mathbf{h}$. Este classificador é ótimo no sentido de minimização do erro quadrático de classificação. Assim como na teoria original de filtragem de correlação a saída é obtida a partir do produto interno da amostra de entrada com o classificador $\mathbf{h}$. Caso a amostra apresentada seja do padrão que se deseja detectar, a saída é grande. Caso contrário ela é pequena.

O DPI está sendo desenvolvido como parte do trabalho de Doutorado do Sr. Waldir Sabino da Silva Júnior, da COPPE/UFRJ. Nas próximas Subseções o DPI é descrito com mais detalhes. Na Subseção 3.3.1 é apresentado o projeto do classificador $\mathbf{h}$ para detecção de uma única classe. Na Subseção 3.3.2 o projeto do classificador é extendido para incorporar a detecção de múltiplas classes. Na Subseção 3.3.3 o projeto do DPI é considerado a partir de uma transformação linear das amostras de entrada e uma interpretação no domínio transformado é feita. Por fim, na Subseção 3.3.4 é apresentado um esquema de normalização das amostras de treinamento que facilita a classificação pelo DPI.

3.3.1 DPI - única classe

Seja uma variável aleatória $\mathbf{X}$, d-dimensional, cujas realizações $\mathbf{x}$ possam ser associadas a uma classe A_i, com $i = \{1, \cdots, n\}$. O objetivo é determinar um classificador $\mathbf{h}_{A_i}$ d-dimensional, ótimo no sentido dos mínimos quadrados, que para o caso ideal, é capaz de detectar um objeto pertencente à classe A_i através da seguinte regra de classificação:

$$\mathbf{h}_{A_i}^t \mathbf{x} = \begin{cases} 1, \text{ se } \mathbf{x} \in A_i \\ 0, \text{ caso contrário.} \end{cases} \tag{3.13}$$

Esta regra[1] pode ser resumida da seguinte forma

$$\mathbf{h}_{A_i}^t \mathbf{X} = C, \tag{3.14}$$

onde $C = 1$ se $\mathbf{x} \in A_i$ e $C = 0$ se $\mathbf{x} \notin A_i$. Um exemplo do uso do DPI no reconhecimento de uma classe pode ser observado na Figura 3.4.

Definindo o erro de classificação e como sendo

$$e = \mathbf{h}_{A_i}^t \mathbf{X} - C, \tag{3.15}$$

o erro quadrático, na sua forma mais geral, fica da seguinte forma[2]:

$$\|e\|^2 = (\mathbf{h}_{A_i}^t \mathbf{X} - C)(\mathbf{h}_{A_i}^t \mathbf{X} - C)^{*t}. \tag{3.16}$$

Considerando $\mathbf{h}_{A_i}$, $\mathbf{X}$ e C reais e dado que $\mathbf{h}_{A_i}^t \mathbf{X}$ e C são escalares, a equação (3.16) pode ser expandida da seguinte forma:

$$\begin{aligned} \|e\|^2 &= (\mathbf{h}_{A_i}^t \mathbf{X} - C)(\mathbf{h}_{A_i}^t \mathbf{X} - C)^t \\ &= (\mathbf{h}_{A_i}^t \mathbf{X})(\mathbf{h}_{A_i}^t \mathbf{X})^t - (\mathbf{h}_{A_i}^t \mathbf{X})(C)^t - (C)(\mathbf{h}_{A_i}^t \mathbf{X})^t + (C)(C)^t \\ &= \mathbf{h}_{A_i}^t \mathbf{X}\mathbf{X}^t \mathbf{h}_{A_i} - \mathbf{h}_{A_i}^t \mathbf{X}C - \mathbf{h}_{A_i}^t \mathbf{X}C + C^2 \\ &= \mathbf{h}_{A_i}^t \mathbf{X}\mathbf{X}^t \mathbf{h}_{A_i} - 2\mathbf{h}_{A_i}^t \mathbf{X}C + C^2. \end{aligned} \tag{3.17}$$

Com isto, o valor esperado do erro quadrático $E\left[\|e\|^2\right]$ fica:

$$E\left[\|e\|^2\right] = \mathbf{h}_{A_i}^t E\left[\mathbf{X}\mathbf{X}^t\right] \mathbf{h}_{A_i} - 2\mathbf{h}_{A_i}^t E\left[\mathbf{X}C\right] + E\left[C^2\right]. \tag{3.18}$$

O vetor $\mathbf{h}_{A_i}$ que minimiza o erro quadrático médio é obtido igualando-se a zero o

[1] Neste trabalho, $\mathbf{a}^t$ indica o transposto do vetor $\mathbf{a}$.
[2] Neste trabalho, $\mathbf{a}^{*t}$ indica o conjugado transposto do vetor $\mathbf{a}$.

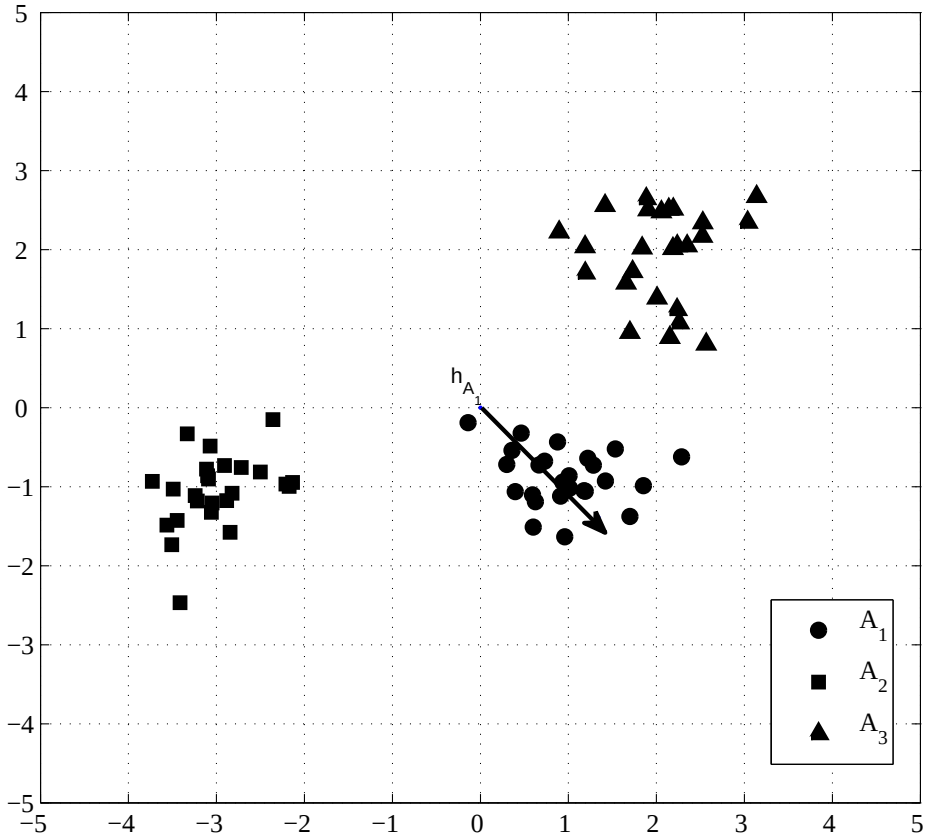


Figura 3.4: DPI empregado no reconhecimento de uma classe. Neste exemplo deseja-se reconhecer os elementos da lasse A_1. O produto interno do vetor $\mathbf{h}_{A_1}$ com amostras da classe A_1 devem possuir valor maior que com amostras das outras classes.

gradiente da Equação (3.18) com relação a $\mathbf{h}_{A_i}$. Utilizando as seguintes propriedades

$$\frac{\partial \mathbf{a}^t \mathbf{B} \mathbf{a}}{\partial \mathbf{a}} = (\mathbf{B} + \mathbf{B}^t)\mathbf{a}, \tag{3.19}$$

$$\frac{\partial \mathbf{a}^t \mathbf{b}}{\partial \mathbf{a}} = \mathbf{b}, \tag{3.20}$$

e supondo que se tem um conjunto de treino com uma quantidade suficiente de amostras distintas, ou seja, que $E[\mathbf{X}\mathbf{X}^t]$ é uma matriz de posto completo (inversível), obtém-se:

$$\begin{aligned}
\frac{\partial \|e\|^2}{\partial \mathbf{h}_{A_i}} &= \frac{\partial}{\partial \mathbf{h}_{A_i}} \left(\mathbf{h}_{A_i}^t E\left[\mathbf{X}\mathbf{X}^t\right] \mathbf{h}_{A_i} - 2\mathbf{h}_{A_i}^t E\left[\mathbf{X}C\right] + E\left[C^2\right] \right) \\
&= \left(E\left[\mathbf{X}\mathbf{X}^t\right] + E\left[\mathbf{X}\mathbf{X}^t\right]^t \right) \mathbf{h}_{A_i} - 2E\left[\mathbf{X}C\right] \\
&= 2E\left[\mathbf{X}\mathbf{X}^t\right] \mathbf{h}_{A_i} - 2E\left[\mathbf{X}C\right] \\
&= 0.
\end{aligned} \tag{3.21}$$

Logo, o vetor $\mathbf{h}_{A_i}$ é dado por:

$$\mathbf{h}_{A_i} = \left(E[\mathbf{X}\mathbf{X}^t]\right)^{-1} E[\mathbf{X}C]. \tag{3.22}$$

Resta agora obter os valores de $E[\mathbf{X}\mathbf{X}^t]$ e $E[\mathbf{X}C]$. O termo $E[\mathbf{X}\mathbf{X}^t]$ pode ser expandido da seguinte forma:

$$E[\mathbf{X}\mathbf{X}^t] = \sum_{j=1}^{n} E[\mathbf{X}\mathbf{X}^t | A_j] p(A_j), \tag{3.23}$$

onde $p(A_i)$ é a probabilidade de uma realização de $\mathbf{X}$ pertencer à classe A_i. Dado que a classe A_i possui L_i amostras de treinamento $\mathbf{x}_{ik}$, $k = \{1, \cdots, L_i\}$, a Equação (3.23) pode ser re-escrita em termos destas amostras como segue:

$$E[\mathbf{X}\mathbf{X}^t] = \sum_{j=1}^{n} p(A_j) \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{x}_{jk} \mathbf{x}_{jk}^t. \tag{3.24}$$

Considerando $\overline{A_i}$ como sendo o complemento de A_i, o termo $E[\mathbf{X}C]$ pode ser expandido em:

$$E[\mathbf{X}C] = E[\mathbf{X}C|A_i]p(A_i) + E[\mathbf{X}C|\overline{A_i}](1 - p(A_i)). \tag{3.25}$$

Como, idealmente, $C = 1$ para o caso em que a realização de $\mathbf{X}$ pertence a classe A_i e $C = 0$ caso contrário, o termo $E[\mathbf{X}C|\overline{A_i}]$ é igual a zero. Levando isto em consideração

e re-escrevendo a equação (3.25) em função das amostras de treinamento obtém-se:

$$E[\mathbf{X}C] = p(A_i)\frac{1}{L_i}\sum_{k=1}^{L_i}\mathbf{x}_{ik}. \tag{3.26}$$

Os resultados das equações (3.24) e (3.26) permitem expressar do vetor $\mathbf{h}_{A_i}$, dado na equação (3.22), em termos das amostras de treinamento:

$$\mathbf{h}_{A_i} = \left(\sum_{j=1}^{n} p(A_j)\frac{1}{L_j}\sum_{k=1}^{L_j}\mathbf{x}_{jk}\mathbf{x}_{jk}^t\right)^{-1} p(A_i)\frac{1}{L_i}\sum_{k=1}^{L_i}\mathbf{x}_{ik}. \tag{3.27}$$

Sendo a matriz de autocorrelação $\mathbf{R}_{A_i}$ e a média $\boldsymbol{\mu}_{A_i}$ da classe A_i dadas, respectivamente, por:

$$\mathbf{R}_{A_i} = \frac{1}{L_i}\sum_{j=1}^{L_i}\mathbf{x}_{ij}\mathbf{x}_{ij}^{*t}, \tag{3.28}$$

$$\boldsymbol{\mu}_{A_i} = \frac{1}{L_i}\sum_{j=1}^{L_i}\mathbf{x}_{ij}, \tag{3.29}$$

a expressão do classificador $\mathbf{h}_{A_i}$ pode ser escrita em termos dos momentos da variável aleatória $\mathbf{X}$:

$$\mathbf{h}_{A_i} = \left(\sum_{j=1}^{n} p(A_j)\mathbf{R}_{A_j}\right)^{-1} p(A_i)\boldsymbol{\mu}_{A_i}. \tag{3.30}$$

Vale ressaltar que a condição de existência da matriz inversa $\left(\sum_{j=1}^{n} p(A_j)\mathbf{R}_{A_j}\right)^{-1}$ é que a dimensão dos vetores de entrada deve ser menor que o número de amostras disponíveis. Caso a quantidade de amostras não seja suficiente, é possível encontrar uma solução aproximada, que pode recair em uma formulação idêntica à do CFA [49], ou ainda à do MACE *filter* [48]. Por fim, o vetor $\mathbf{h}_{A_i}$ é utilizado para classificar uma amostra $\mathbf{x}_n$ de acordo com a regra de classificação apresentada na Equação (3.13).

3.3.2 DPI - múltiplas classes

O classificador $\mathbf{h}_{A_i}$ ($i = \{1, \cdots, n\}$), descrito na seção anterior, é capaz de discriminar as realizações da variável aleatória $\mathbf{X}$ que pertencem à classe A_i das demais. Nesta seção o projeto do classificador é modificado de modo que este passe a discriminar padrões pertencentes a um conjunto de classes $A = \{A_1, \cdots, A_m\}$, $m < n$. Esta abordagem pode ser útil quando já se tem as estatísticas das classes que formam o conjunto A. Quando não, basta definir a classe A como sendo $A \equiv A_1 \cup \cdots \cup A_m$ e tratar o problema como no caso de única classe.

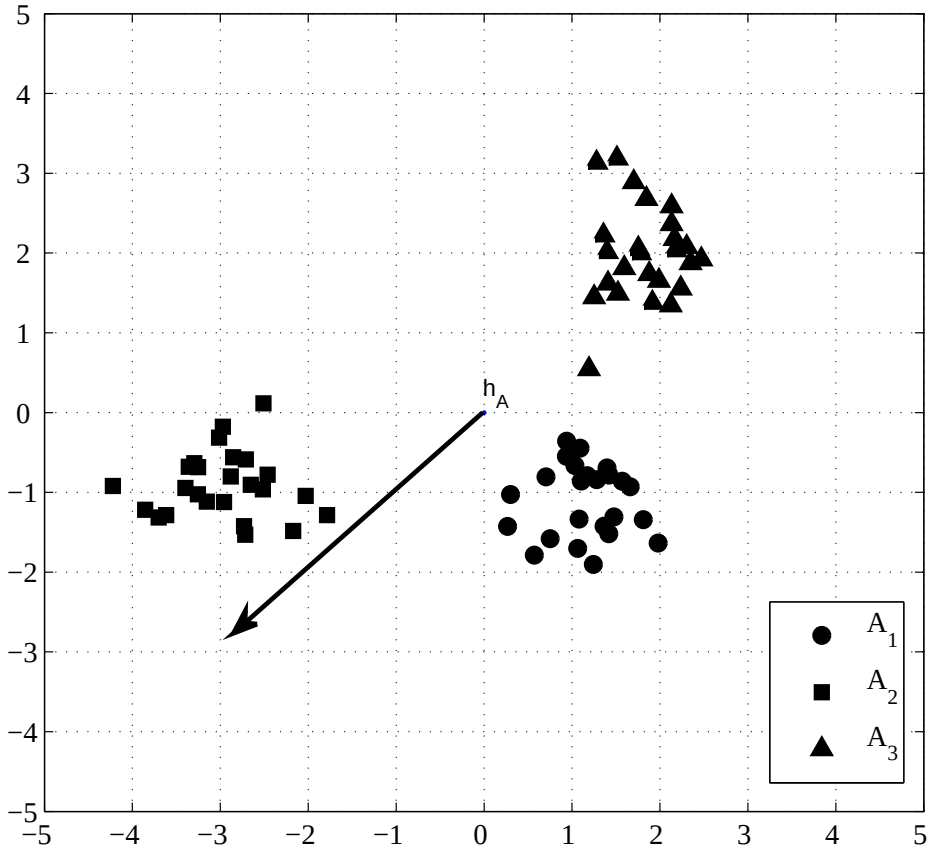


Figura 3.5: DPI empregado no reconhecimento de múltiplas classes. Neste caso, o conjunto A é formado pelas classe A_1 e A_2. O produto interno do vetor $\mathbf{h}_A$ com os elementos da classe A_1 e A_2 deve apresentar um valor maior que com os elementos da classe A_3.

De forma semelhante ao caso de única classe, a classificação é dada pelo produto interno entre o classificador $\mathbf{h}_A$ e uma realização $\mathbf{x}$ da variável aleatória $\mathbf{X}$:

$$\mathbf{h}_A^t \mathbf{x} = \begin{cases} 1, \text{ se } \mathbf{x} \in A \\ 0, \text{ caso contrário,} \end{cases} \tag{3.31}$$

que pode ser resumida da seguinte forma

$$\mathbf{h}_A^t \mathbf{X} = C, \tag{3.32}$$

onde, idealmente, $C = 1$ se $\mathbf{x} \in A$ e $C = 0$ se $\mathbf{x} \notin A$. Um exemplo do uso do DPI no reconhecimento de múltiplas classes pode ser observado na Figura 3.5.

Definindo o erro como sendo $e = \mathbf{h}_A^t \mathbf{X} - C$ e desenvolvendo de forma semelhante

à seção 3.3.1, chega-se à seguinte forma básica do vetor $\mathbf{h}_A$:

$$\mathbf{h}_A = \left(E[\mathbf{X}\mathbf{X}^t]\right)^{-1} E[\mathbf{X}C]. \tag{3.33}$$

O termo $E[\mathbf{X}\mathbf{X}^t]$ é obtido da mesma forma que na Seção anterior. Este termo pode ser escrito em função das amostras de treinamento como segue:

$$E[\mathbf{X}\mathbf{X}^t] = \sum_{j=1}^{n} p(A_j) \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{x}_{jk}\mathbf{x}_{jk}^t. \tag{3.34}$$

Já o termo $E[\mathbf{X}C]$ sofre algumas modificações. Considerando $\overline{A}$ como sendo o complemento do conjunto A o termo $E[\mathbf{X}C]$ fica:

$$E[\mathbf{X}C] = E[\mathbf{X}C|A]p(A) + E[\mathbf{X}C|\overline{A}](1 - p(A)). \tag{3.35}$$

Como, idealmente, $C = 1$ para o caso em que a realização de $\mathbf{X}$ pertence ao conjunto de classes A e $C = 0$ caso contrário, o termo $E[\mathbf{X}C|\overline{A}]$ é igual a zero. Levando isto em consideração é possível desenvolver o termo $E[\mathbf{X}C]$ da seguinte forma:

$$\begin{aligned} E[\mathbf{X}C] &= E[\mathbf{X}C|A]p(A) + E[\mathbf{X}C|\overline{A}](1 - p(A)) \\ &= E[\mathbf{X}C|A]p(A) \\ &= E[\mathbf{X}C|A_1]p(A_1) + \cdots + E[\mathbf{X}C|A_m]p(A_m) \\ &= \sum_{j=1}^{m} p(A_j) \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{x}_{jk}. \end{aligned} \tag{3.36}$$

Com os resultados das Equações (3.34) e (3.36), a expressão do classificador $\mathbf{h}_A$, escrita em termos das amostras de treinamento, é dada por:

$$\mathbf{h}_A = \left(\sum_{j=1}^{n} p(A_j) \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{x}_{jk}\mathbf{x}_{jk}^t \right)^{-1} \sum_{j=1}^{m} p(A_j) \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{x}_{jk}. \tag{3.37}$$

Reconhecendo na expressão acima os momentos da variável aleatória $\mathbf{X}$, apresentados nas Equações (3.28) e (3.29), é possível re-escrever a Equação (3.37) em termos dos momentos:

$$\mathbf{h}_A = \left(\sum_{j=1}^{n} p(A_j)\mathbf{R}_{A_j} \right)^{-1} \sum_{j=1}^{m} p(A_j)\boldsymbol{\mu}_{A_j}. \tag{3.38}$$

3.3.3 DPI com transformação linear e interpretação no domínio transformado

Nas Seções 3.3.1 e 3.3.2 foi visto como projetar um detector por produto interno para discriminar amostras de única classe e de múltiplas classes, respectivamente. Contudo, muitas vezes é conveniente realizar uma transformação linear nas amostras de entrada de forma que as classes possuam uma maior separação entre si no domínio transformado. Nesta seção o projeto do vetor $\mathbf{h}$ será desenvolvido a partir de uma transformação linear dos dados de entrada. Neste caso será adotada a versão complexa do DPI, descrita em detalhes no Apêndice A. Em seguida esta nova abordagem será utilizada para interpretar o DPI no domínio transformado.

Para tanto, seja uma transformação linear complexa, dada por uma matriz quadrada $\mathbf{Q}$ $(d \times d)$, que aplicada à variável aleatória $\mathbf{X}$ retorna:

$$\mathbf{X}' = \mathbf{Q}\mathbf{X}. \tag{3.39}$$

Com isto podemos desenvolver a expressão para o classificador no domínio transformado como segue:

$$\begin{aligned}
\mathbf{h}'_A &= (E[\mathbf{X}'\mathbf{X}'^{*t}])^{-1}E[\mathbf{X}'C^{*t}] \\
&= (E[\mathbf{Q}\mathbf{X}(\mathbf{Q}\mathbf{X})^{*t}])^{-1}E[\mathbf{Q}\mathbf{X}C^{*t}] \\
&= (E[\mathbf{Q}\mathbf{X}\mathbf{X}^{*t}\mathbf{Q}^{*t}])^{-1}E[\mathbf{Q}\mathbf{X}C^{*t}] \\
&= (\mathbf{Q}E[\mathbf{X}\mathbf{X}^{*t}]\mathbf{Q}^{*t})^{-1}\mathbf{Q}E[\mathbf{X}C^{*t}] \\
&= (\mathbf{Q}^{*t})^{-1}(E[\mathbf{X}\mathbf{X}^{*t}])^{-1}(\mathbf{Q})^{-1}\mathbf{Q}E[\mathbf{X}C^{*t}] \\
&= (\mathbf{Q}^{*t})^{-1}(E[\mathbf{X}\mathbf{X}^{*t}])^{-1}E[\mathbf{X}C^{*t}] \\
&= (\mathbf{Q}^{*t})^{-1}\mathbf{h}_A.
\end{aligned} \tag{3.40}$$

Considerando $\mathbf{Q}$ uma transformação ortogonal, o produto interno no domínio transformado pode ser desenvolvido da seguinte forma:

$$\begin{aligned}
\mathbf{h}'^{*t}_A\mathbf{X}' &= ((\mathbf{Q}^{*t})^{-1}\mathbf{h}_A)^{*t}\mathbf{Q}\mathbf{X} \\
&= \mathbf{h}^{*t}_A((\mathbf{Q}^{*t})^{-1})^{*t}\mathbf{Q}\mathbf{X} \\
&= \mathbf{h}^{*t}_A(\mathbf{Q})^{-1}\mathbf{Q}\mathbf{X} \\
&= \mathbf{h}^{*t}_A\mathbf{X}.
\end{aligned} \tag{3.41}$$

Este último resultado indica que o produto interno entre o detector $\mathbf{h}'_A$ e os dados no domínio transformado $\mathbf{X}'$ possuem o mesmo efeito que o produto interno no domínio original $\mathbf{h}^{*t}_A\mathbf{X}$. Com isto, a transformação $\mathbf{Q}$ não apresenta nenhum efeito do ponto de vista da minimização do erro de classificação. Por outro lado, aplicando

uma transformação linear derivada da KLT (*Karhunen-Loève Transform*), também conhecida como *Hotelling Transform*, ou ainda PCA (*Principal Component Analysis*), é possível fazer uma importante interpretação do DPI no domínio transformado. Esta interpretação é dada a seguir.

Reconhecendo o termo $E[\mathbf{X}\mathbf{X}^{*t}]$ como sendo a matriz de autocorrelação $\mathbf{R_X}$ da variável aleatória $\mathbf{X}$, é possível escrever a expressão do vetor $\mathbf{h}_A$ (dado na Equação (3.33)) da seguinte forma:

$$\mathbf{h}_A = (\mathbf{R_X})^{-1} E[\mathbf{X}\mathbf{C}^{*t}]. \tag{3.42}$$

Decompondo a matriz de autocorrelação em função da matriz de autovetores $\mathbf{\Phi}$ e da matriz de autovalores $\mathbf{\Lambda}$, chega-se a:

$$\begin{aligned} \mathbf{\Phi}^{*t}\mathbf{R_X}\mathbf{\Phi} &= \mathbf{\Lambda} \\ \Rightarrow \mathbf{\Phi}(\mathbf{\Phi}^{*t}\mathbf{R_X}\mathbf{\Phi})\mathbf{\Phi}^{*t} &= \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{*t} \\ \Rightarrow \mathbf{R_X} &= \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{*t}, \end{aligned} \tag{3.43}$$

onde as colunas de $\mathbf{\Phi}$ são compostas pelos autovetores ortonormais de $\mathbf{R_X}$ e $\mathbf{\Lambda}$ é uma matriz diagonal na qual cada elemento da diagonal principal é um autovalor correspondente.

Substituindo o resultado da decomposição de $\mathbf{R_X}$ na expressão do vetor $\mathbf{h}_A$ e desenvolvendo, obtêm-se:

$$\begin{aligned} \mathbf{h}_A &= (\mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{*t})^{-1} E[\mathbf{X}\mathbf{C}^{*t}] \\ \Rightarrow \mathbf{h}_A &= \mathbf{\Phi}\mathbf{\Lambda}^{-1}\mathbf{\Phi}^{*t} E[\mathbf{X}\mathbf{C}^{*t}] \\ \Rightarrow \mathbf{\Phi}^{*t}\mathbf{h}_A &= \mathbf{\Phi}^{*t}\mathbf{\Phi}\mathbf{\Lambda}^{-1/2}\mathbf{\Lambda}^{-1/2}\mathbf{\Phi}^{*t} E[\mathbf{X}\mathbf{C}^{*t}] \\ \Rightarrow \mathbf{\Lambda}^{1/2}\mathbf{\Phi}^{*t}\mathbf{h}_A &= E[\mathbf{\Lambda}^{-1/2}\mathbf{\Phi}^{*t}\mathbf{X}\mathbf{C}^{*t}] \\ \Rightarrow ((\mathbf{\Lambda}^{-1/2}\mathbf{\Phi}^{*t})^{*t})^{-1}\mathbf{h}_A &= E[\mathbf{\Lambda}^{-1/2}\mathbf{\Phi}^{*t}\mathbf{X}\mathbf{C}^{*t}]. \end{aligned} \tag{3.44}$$

Fazendo com que a transformação $\mathbf{Q}$ seja dada por $\mathbf{Q} = \mathbf{\Lambda}^{-1/2}\mathbf{\Phi}^{*t}$ e comparando o resultado da Equação (3.44) com o da Equação (3.40), a expressão do classificador no domínio transformado fica:

$$\mathbf{h}'_A = (\mathbf{Q}^{*t})^{-1}\mathbf{h}_A = E[\mathbf{Q}\mathbf{X}\mathbf{C}^{*t}]. \tag{3.45}$$

O termo $\mathbf{QX}$ nada mais é que a KLT de $\mathbf{X}$, dada por $\mathbf{\Phi}^{*t}\mathbf{X}$, multiplicada por $\mathbf{\Lambda}^{-1/2}$. A transformação $\mathbf{\Phi}^{*t}$ aplica uma rotação às realizações de $\mathbf{X}$, tornando-as descorrelacionadas. A multiplicação pela matriz $\mathbf{\Lambda}^{-1/2}$ normaliza as realizações em relação à variância em cada direção, causando uma separação angular máxima entre

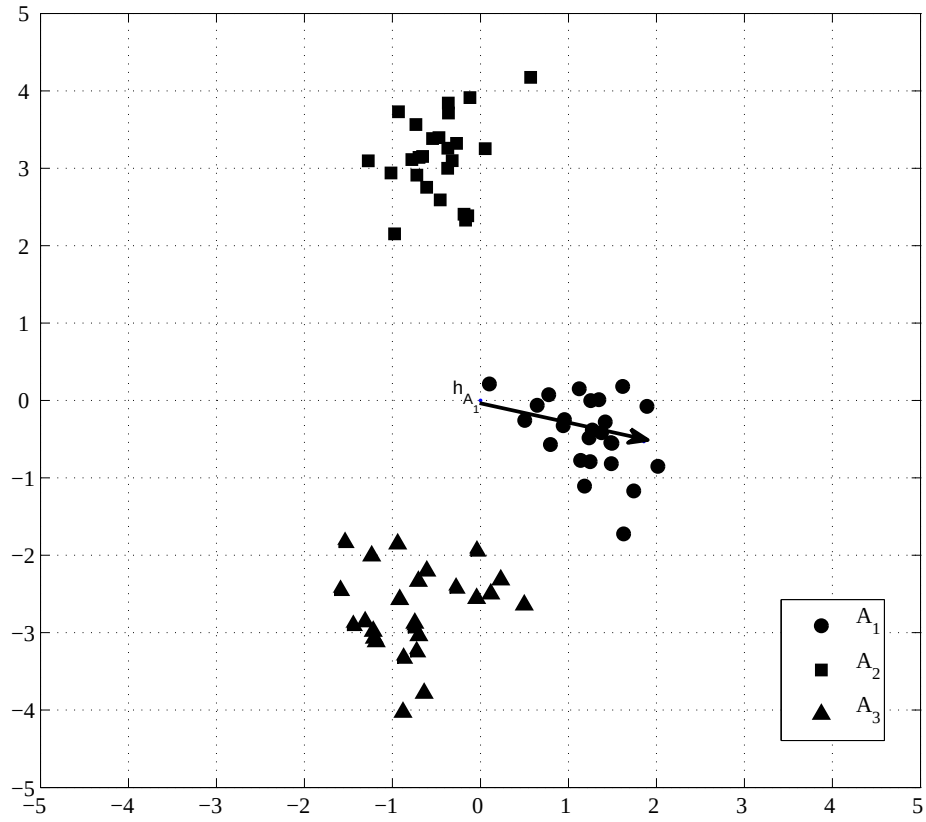


Figura 3.6: A KLT aplica uma rotação ao conjunto de dados, de modo que a projeção dos dados nos eixos possuam variância máxima. As amostras desta figura, assim como o detector, são versões rotacionadas dos apresentados na Figura 3.4.

as amostras. Com isto, a matriz de correlação dos dados transformados fica proporcional à identidade. Esta operação é conhecida na literatura como transformada de branqueamento (em inglês *whitening transform*) [14]. Os efeitos da KLT e do branqueamento no DPI podem ser observados nas Figuras 3.6 e 3.7, respectivamente.

Ainda no domínio transformado, o produto interno $\mathbf{h}_A'^{*t}\mathbf{X}$ pode ser escrito da seguinte forma:

$$\mathbf{h}_A'^{*t}\mathbf{X}' = \sum_{k=1}^{d} X'(k)\mathbf{h}_A'^{*}(k). \tag{3.46}$$

Por outro lado, a convolução linear discreta entre os vetores s_1 e s_2 pode ser desen-

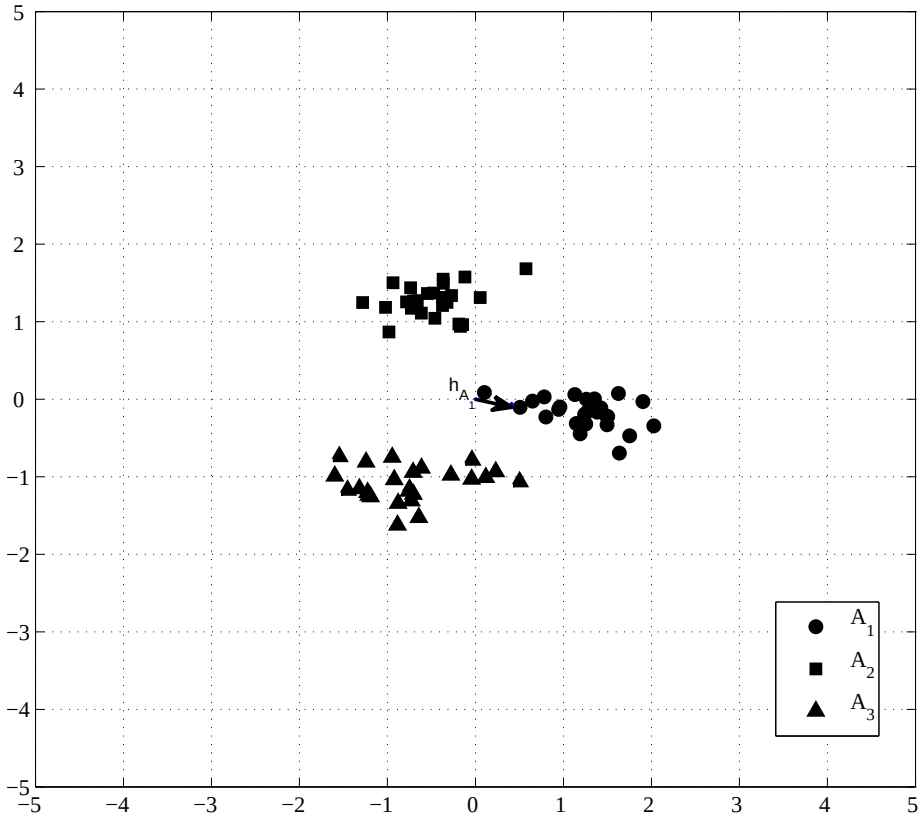


Figura 3.7: O branqueamento causa uma separação angular máxima entre as amostras, fazendo com que a matriz de correlação fique proporcional à matriz identidade. Veja que o detector no domínio branqueado está alinhado com a média da classe que se deseja detectar.

volvida como segue[3]:

$$
\begin{aligned}
s_1(n) \star s_2(n) &= \sum_{k=-\infty}^{\infty} s_1(k) s_2(n-k) \\
s_1(n) \star s_2^*(-n) &= \sum_{k=-\infty}^{\infty} s_1(k) s_2^*(k-n) \\
s_1(n) \star s_2^*(-n)|_{n=0} &= \sum_{k=-\infty}^{\infty} s_1(k) s_2^*(k). \qquad (3.47)
\end{aligned}
$$

Assumindo que $\mathbf{X}'(k) = 0$ e $\mathbf{h}'_A(k) = 0$ se $k < 1$ ou $k > d$, e substituindo o resultado obtido na Equação (3.47), na Equação (3.46) chega-se à seguinte Equação:

$$
\mathbf{h}'^{*t}_A \mathbf{X}' = \mathbf{X}'(n) \star \mathbf{h}'^{*}_A(-n)|_{n=0}. \qquad (3.48)
$$

Logo, no domínio transformado, a classificação é obtida pela convolução entre uma realização de $\mathbf{X}'(n)$ com a versão invertida no domínio transformado do classificador $\mathbf{h}'_A(-n)$ para $n = 0$. Esta é a forma clássica da filtragem casada.

Por fim, usando a relação entre o termo $E[\mathbf{X}\mathbf{C}^{*t}]$ e a média $\boldsymbol{\mu}_{A_i}$ da classe A_i e partindo do resultado da Equação (3.49), pode-se representar o classificador no domínio transformado $\mathbf{h}'_A$ por:

$$
\begin{aligned}
\mathbf{h}'_A &= E[\mathbf{Q}\mathbf{X}\mathbf{C}^{*t}] \\
&= \mathbf{Q}E[\mathbf{X}\mathbf{C}^{*t}] \\
&= \mathbf{Q}\sum_{k=1}^{m} p(A_k)\boldsymbol{\mu}_{A_k} \\
&= \sum_{k=1}^{m} p(A_k)\boldsymbol{\mu}'_{A_k}. \qquad (3.49)
\end{aligned}
$$

Lembrando que o conjunto $A = \{A_1, \cdots, A_m\}$ é composto por m classes, esta é uma importante interpretação do vetor $\mathbf{h}'_A$ no domínio transformado, pois permite concluir que $\mathbf{h}'_A$ é o filtro casado com a média dos elementos do conjunto A no domínio transformado (branqueado).

3.3.4 DPI no espaço de dimensão estendida

O DPI é treinado para apresentar uma saída binária, onde é obtido 1, quando um ponto fiducial é detectado e 0, quando não. Dado que as classes em questão não são necessariamente ortogonais entre si, este tipo de saída não ocorre na prática. A

[3]Em muitos trabalhos a operação de convolução é denotada por um asterisco "$*$". Neste trabalho, para não haver confusão com o asterisco utilizado para representar o conjugado, a convolução é representada por uma "estrela": "$\star$".

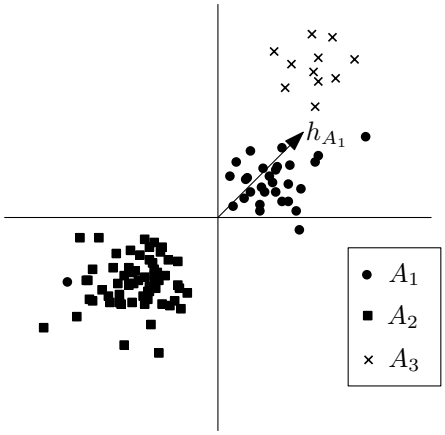


Figura 3.8: Exemplo do DPI no reconhecimento de uma classe. Neste detector $\mathbf{h}_{A_1}$ foi projetado para reconhecer elementos da classe A_1

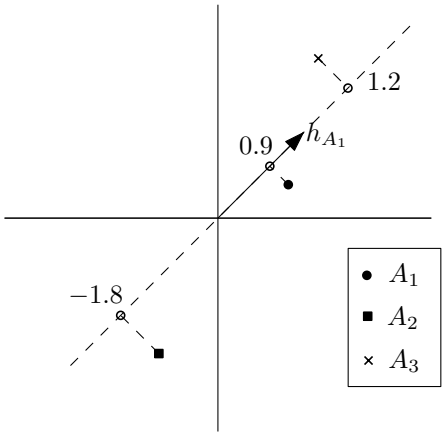


Figura 3.9: Produto interno de amostras das classes A_1, A_2 e A_3 com o classificador $\mathbf{h}_{A_1}$

saída do DPI é um valor real. Além disto, o produto interno entre o vetor $\mathbf{h}$ e uma amostra que não seja o padrão pode apresentar valores negativos ou, até mesmo, valores maiores do que o produto interno com o padrão. Nesta seção este efeitos serão interpretados e uma solução é proposta.

Seja um caso de classificação de 3 classes, A_1, A_2 e A_3, onde deseja-se reconhecer elementos pertencentes à classe A_1, como ilustrado na Figura 3.8.

Interpretando o produto interno como sendo a projeção de um vetor sobre o outro, é possível perceber através da Figura 3.8, que algumas amostras da classe A_3 possuem um valor de produto interno com o classificador $\mathbf{h}_{A_1}$ maior que o de amostras da classe A_1. Por outro lado, as amostras da classe A_2 possuem valores negativos. Isto pode ser observado na Figura 3.9.

Para resolver este problema uma dimensão é acrescentada em cada amostra. O objetivo é normalizar a saída do DPI e fazer com que as amostras da classe que se deseja detectar tenham uma tendência de possuir o valor do produto interno maior

que o das outras classes.

Inicialmente procura-se, dentre as N amostras de treinamento aquela que possua a maior norma E_{max}:

$$E_{max} = max(\|\mathbf{x}_i\|^2),\ i = \{1, \cdots, N\}. \tag{3.50}$$

Em seguida, cada amostra $\mathbf{x}_i$ (d-dimensional) possui sua dimensão aumentada da seguinte forma:

$$\tilde{\mathbf{x}}_i = \left[x_1 \quad \cdots \quad x_d \quad \sqrt{(E_{max} - \|\mathbf{x}_i\|^2)}\right]^t. \tag{3.51}$$

Esta expansão faz com que todas as amostras de treinamento sejam projetadas na casca de uma hiper-esfera de raio E_{max}. Todas as amostras $\tilde{\mathbf{x}}_i$, bem como o detector obtido com elas $\tilde{\mathbf{h}}_{A_i}$, possuem a mesma norma quadrática (E_{max}). A divisão do produto interno no espaço de dimensão $d+1$ por E^2_{max} permite que o resultado do DPI esteja no intervalo $[-1, 1]$. Isto significa que a classificação é dada pelo cosseno do angulo entre o detector $\tilde{\mathbf{h}}_{A_i}$ e uma amostra $\tilde{\mathbf{x}}_i$

3.4 Pós-processamento

A saída do DPI com dimensão extra, descrito na Seção anterior, é um valor real no intervalo $[-1, 1]$. Logo, é necessária a determinação de um limiar que, dada a saída do DPI, separe os pontos fiduciais dos outros pontos. Neste trabalho isto é feito automaticamente através de um classificador Fisher ou *AdaBoost* (descritos nas Seções 2.3.1 e 2.3.2, respectivamente). Cada estágio da cascata é composto por este conjunto formado pelo DPI e o segundo classificador.

Dada uma imagem de teste, a saída da cascata para cada ponto fiducial é um ou mais pontos nos quais o sistema reconhece o padrão. Por outro lado, quando o sistema responde mais de um ponto, estes geralmente estão aglutinados em torno do ponto fiducial ou de regiões muito próximas. A Figura 3.10 ilustra algumas saídas típicas da cascata de classificadores.

A solução adotada para o problema de múltiplas saídas é a utilização de uma heurística que encontra o ponto que melhor representa o conjunto de saída. Este ponto é a saída final do sistema.

Este agrupamento é feito a partir da distância euclidiana entre os pontos de saída do sistema. Se a distância entre estes pontos é menor que um determinado valor, então estes pontos são agrupados e representados pela posição média do grupo. A Figura 3.11 ilustra este procedimento para as saídas típicas do sistema.

Neste trabalho foram utilizadas duas versões do algoritmo de agrupamento. Na primeira foi utilizado um limiar "flexível", que permite mais de um ponto na saída por imagem. Na segunda foi utilizado um limiar "rígido", que obriga o sistema a

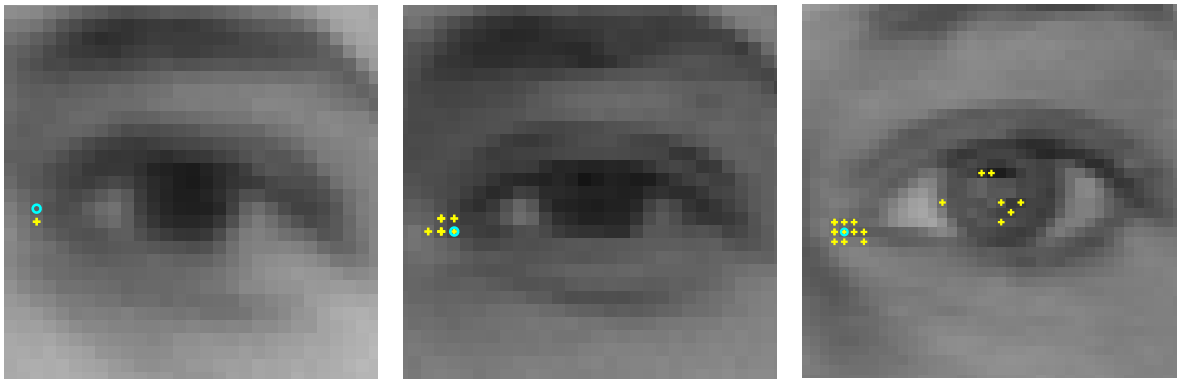


Figura 3.10: Saídas típicas da cascata de classificadores na detecção do canto interno do olho esquerdo. Os pontos marcados com "o" representam a marcação do ponto fiducial e os marcados com "cruz" são saída do sistema.

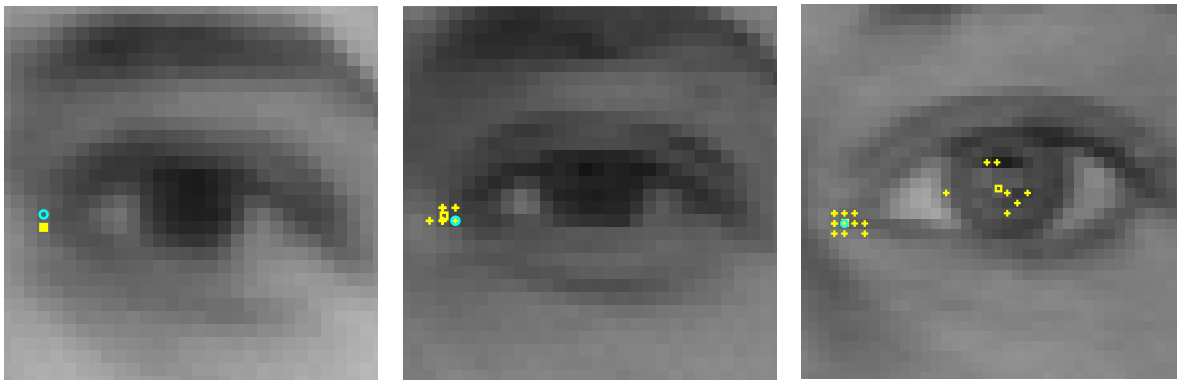


Figura 3.11: Resultados do algoritmo de agrupamento para as saídas típicas da cascata na detecção do canto interno olho esquerdo. O ponto marcado com "o" representa a marcação do ponto fiducial, os em "cruz" representam a saída da cascata de classificadores e os marcados com "quadrados" representam o resultado do algoritmo de agrupamento. O limiar utilizado foi de 5 pixeis.

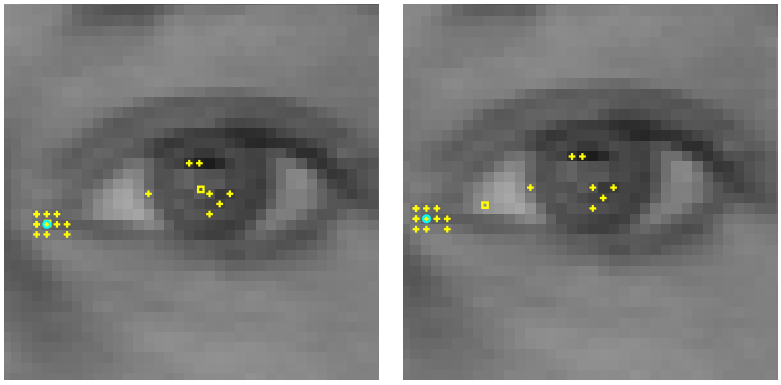


Figura 3.12: Resultado do algoritmo de agrupamento para os casos "flexível" e "rígido". Os pontos marcados com "o" indicam as marcações dos pontos fiduciais, os pontos em "cruz" indicam as saídas da cascata e os marcados com "quadrado" indicam o resultado do algoritmo de agrupamento. Na imagem da direita foi utilizada a abordagem flexível, onde os pontos com distância entre si menor que 5 pixeis foram agrupados. Na imagem da esquerda foi utilizada uma abordagem rígida em que o ponto escolhido é a média de todos os ponto de saída da cascata

exibir apenas uma saída por imagem, se houver algum ponto na saída. Na versão "rígida", a média de todos os pontos da saída da cascata é utilizada como ponto representante. Está ilustrado na Figura 3.12 a saída do agrupamento para os casos "flexível" e "rígido".

Um pseudo-código do pós-processamento utilizado está descrito no Algoritmo 2.

Entrada: Os P pontos de saída da cascata e a distância d de agrupamento dos pontos.

para cada $\mathbf{p}_i, \quad i = \{1, \cdots, P\}$ **faça**
  se $i = 1$ **então**
    Crie um grupo novo e coloque $\mathbf{p}_1$ neste grupo
  senão
    Calcule a distância de $\mathbf{p}_i$ para a média de todos os grupos criados.
    Se a distância entre $\mathbf{p}_i$ e a média de um determinado grupo for menor que d, junte $\mathbf{p}_i$ a este grupo. Senão crie um grupo novo e coloque $\mathbf{p}_i$ neste grupo.

Saída: A média dos elementos de cada grupo.

Algoritmo 2: Pseudo-código do pós-processamento utilizado.

Capítulo 4

Resultados e discussões

Neste capítulo é apresentada metodologia experimental e em seguida os resultados obtidos são apresentados e discutidos. Na Seção 4.1 está descrita a bases de dados utilizada, a base BioID. Na Seção 4.2, está descrito o esquema de validação cruzada usada na avaliação dos métodos. Na Seção 4.3 está apresentada a medida de precisão de detecção utilizada neste trabalho. Por fim, na Seção 4.5 são apresentados os resultados obtidos além de algumas considerações sobre estes resultados.

4.1 Base de dados

As entradas do sistema de reconhecimento de características faciais descrito neste trabalho são imagens contendo faces em pose frontal. Para o treino e para o teste (tanto do método proposto, quanto dos outros utilizados na comparação) foi utilizada uma bases de dados gratuita que contém estas características. Trata-se da base BioID, disponível em [12]. A seguir a base BioID é descrita com mais detalhes.

A BioID é uma base de dados contendo 1.521 imagens em níveis de cinza com uma resolução de 384×286 no formato "PGM". Estas imagens foram retiradas de 23 indivíduos diferentes em várias seções. As imagens possuem variação de escala, iluminação e *background*. Além disto, há imagens cujos indivíduos estão usando óculos, barba e bigode. Acompanhando o conjunto de imagens, está disponível também uma anotação manual de 20 pontos na face e uma anotação separada contendo a posição dos olhos. A Figura 4.1 contém algumas imagens da BioID.

Neste trabalho foi utilizado um subconjunto da BioID composto por 503 imagens. Deste subconjunto, estão excluídas as imagens cujos indivíduos estão portando óculos, possuem barba ou bigode e as imagens que possuem grande rotações. Embora estejam disponíveis as anotações manuais de vinte pontos de todas as imagens da base em [12], neste trabalho foi feita a anotação manual dos treze pontos das 503 imagens utilizadas.

Figura 4.1: Exemplo de imagens da base BioID.

4.2 Validação cruzada

Validação cruzada é uma técnica que permite uma avaliação estatística do desempenho de um modelo, quando este é aplicado em um conjunto de dados independente. Nas estratégias mais comuns de validação cruzada, o conjunto de dados é dividido em duas partes. Um delas, utilizada para a análise, é conhecido como conjunto de treino. Com a outra, conhecida como conjunto de teste, é feita a validação.

Existem diversos tipos de validação cruzada. Os mais utilizados são: sorteio, *leave-one-out* e *k-fold*. Cada uma delas é descrita a seguir:

- No **sorteio**, uma quantidade fixa de amostras é sorteada e estes dados são utilizados no treino. o restante das amostras é utilizada no teste.
- No ***leave-one-out***, dado um conjunto de N amostras, $N - 1$ são utilizadas para compor o conjunto de treino e a restante é utilizada no teste. Este procedimento é repetido N vezes, para que o teste seja feito com todas as amostras.
- No k***-fold***, as amostras são divididas em k partes. $k - 1$ formam o conjunto de treino e uma parte forma o de teste. este procedimento é repetido k vezes, de modo que todas as partes sejam utilizadas no teste.

A validação cruzada por *k-fold* tem a vantagem de possuir mais amostras de validação do que o método de sorteio. Além disto, ele envolve menos etapas de treino (que no nosso caso é uma etapa que consome tempo e processamento) do que o método *leave-one-out*. Neste trabalho a validação cruzada é feita por *k-fold*, na qual foram utilizada 7 partições ($k = 7$).

4.3 Precisão da localização dos pontos fiduciais

Em alguns sistemas de localização de faces e de olhos, a precisão da detecção tem sido avaliada através de uma medida de erro entre as posições estimadas e reais das pupilas [51], [52]. Considerando a marcação das pupilas esquerdas C_e e direitas C_d como sendo as posições reais e sendo D_e a distancia entre a posição estimada e a real da pupila esquerda e D_d a da pupila direita, a medida relativa de erro d_{olho}, é dada por:

$$d_{olho} = \frac{max(D_e, D_d)}{\|C_e - C_d\|}. \tag{4.1}$$

Na literatura, um valor considerado razoável para uma estimação da posição é $d_{olho} < 0.25$.

Neste trabalho, esta medida é generalizada para ser aplicada em pontos fiduciais. Supondo as faces centralizadas, com mesma resolução e considerando verdadeiras as marcações dos olhos, a medida relativa de erro adotada d_{pf} é:

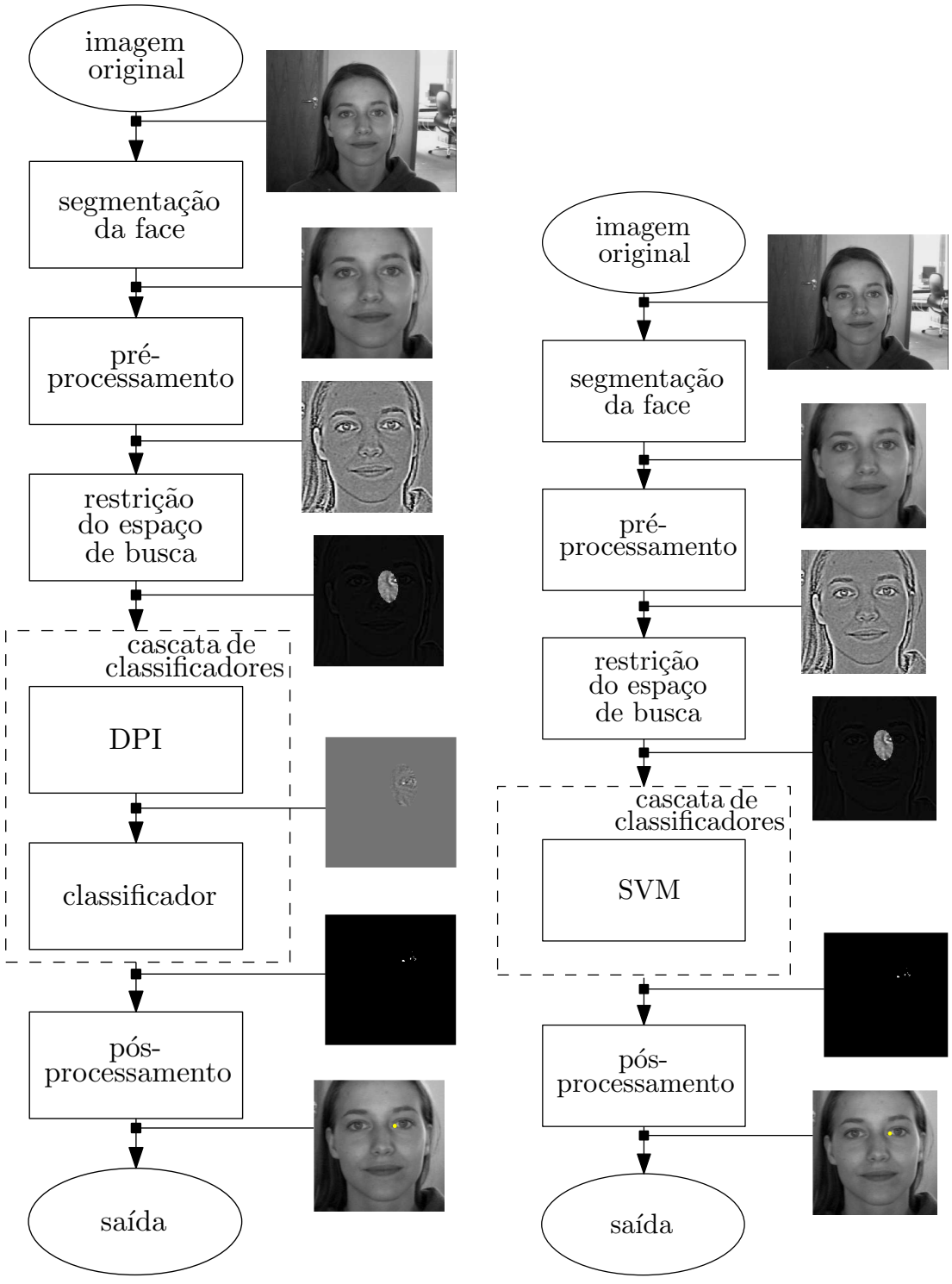
$$d_{pf} = \frac{\|D_{pf} - D'_{pf}\|}{\|C_e - C_d\|}, \tag{4.2}$$

onde D_{pf} é a posição de um ponto fiducial dada pela marcação e D'_{pf} é a posição estimada. Neste trabalho, foi considerado acerto, apenas os casos em que $d_{pf} < 0.1$. Em outras palavras, uma estimativa é considerada certa, somente se estiver a uma distância menor que 10% da distância intra-ocular do ponto fornecido pela marcação.

4.4 Comparação com o SVM

Um dos métodos utilizados na comparação com o proposto é um sistema usando o SVM como classificador principal. Este sistema é semelhante ao proposto, com a diferença de possuir o SVM ao invés do conjunto formado pelo DPI e *AdaBoost* (ou discriminante de Fisher). Neste caso foi empregado o SVM linear com margem suave. Para tanto foi utilizada a biblioteca SVMlight, disponível em [35]. A Figura 4.2 apresenta dois diagramas de blocos, um do sistema proposto e outro do sistema baseado em SVM. Note que a única diferença entre estes métodos é a utilização do SVM em cada estágio da cascata no lugar do DPI e o segundo classificador.

Nas duas versões do sistema proposto (uma com *AdaBoost* e outra com o discriminante de Fisher) foi adotado um problema de três classes. As amostras da classe A_1, a que se quer reconhecer, são formadas por blocos 21×21 centrados no ponto fiducial em questão. As outras classes são: A_2, cujas amostras são os blocos 21×21 centrados nos vizinhos de 1 a 10 pixeis do ponto fiducial; A_3, cujas amostras são blocos centrados nos outros pontos da região de busca.



(a) Sistema proposto.

(b) Sistema com SVM.

Figura 4.2: Comparação entre os diagramas de blocos dos sistemas proposto e com SVM.

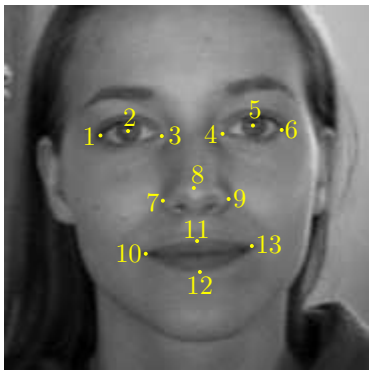


Figura 4.3: Pontos fiduciais utilizados neste trabalho

No caso do SVM são consideradas amostras positivas somente os blocos 21×21 centrados no ponto fiducial. Os outros blocos da região de busca são considerados amostras negativas.

4.5 Resultados

Nesta seção são apresentados os resultados obtidos. Estes resultados estão contidos nas Tabelas 4.1 4.2, 4.3 e 4.4. Os valores apresentados nas tabelas são as médias das taxas obtidas nos conjuntos de teste de cada *fold*. Está associado a estes valores um intervalo de confiança de 95%. Em todas as tabelas, VP indica as taxas de acerto (verdadeiro positivo) e FP as taxas de falsos positivos. Nestas tabelas, as colunas referentes ao método proposto recebem o nome **DPI+Adaboost**, no caso do *AdaBoost* (Lembrando que foi utilizado o *Gentle AdaBoost* [22]) como segundo classificador, e **DPI+Fisher**, no caso do discriminante Linear de Fisher. As colunas com o título **SVM** referem-se ao sistema que utiliza o SVM como classificador principal. Nas colunas intituladas **nface** estão apresentados os resultados referentes ao método proposto em [3]. Este último método está presente apenas na tabela 4.3 (que apresenta resultados para um pós processamento rígido), porque o *nface* sempre apresenta 9 pontos na saída para cada face detectada.

Por conveniência, uma imagem da BioID com a numeração dos pontos fiduciais sobreposta é repetida na Figura 4.3.

Nas duas versões do método proposto e no SVM, foram utilizadas cascatas de classificadores. Para cada par definido por ponto fiducial e método, a cascata pode possuir um número diferente de estágios, que pode ir de 1 a 8. O critério utilizado para determinar este número foi a adição de estágios até a taxa de acerto (verdadeiros positivos) começar a diminuir. A Tabela 4.1 contém os números de estágios dos métodos utilizados para cada ponto fiducial. Com relação às versões do método proposto, os números de estágios para cada um dos pontos são os mesmos. Note

Tabela 4.1: Número de estágios por ponto fiducial de cada método.

ponto fiducial	DPI+Adaboost	DPI+Fisher	SVM
1	3	3	2
2	3	3	2
3	4	4	2
4	3	3	2
5	3	3	2
6	3	3	2
7	4	4	2
8	6	6	2
9	4	4	2
10	8	8	2
11	8	8	2
12	8	8	2
13	8	8	2

que o maior número de estágios (8) ocorreu para os pontos ao redor de boca. Isto é reflexo da grande variação de deslocamento e forma dos padrões referentes a estes pontos, tornando-os mais difíceis de serem reconhecidos. O ponto 8 (centro do nariz) também apresentou um número de estágios elevado (6). Isto provavelmente ocorreu porque o padrão referente a este ponto não possui uma forma muito bem definida, sendo parecido com pontos da bochecha, por exemplo. Dois estágios foram suficientes para todos os pontos no caso do SVM.

Antes de apresentar os resultados finais, é importante apresentar os resultados sem o uso do agrupamento como pós-processamento. Estes resultados estão na Tabela 4.2. Apesar das taxas de acerto estarem altas em todos os métodos (com exceção dos pontos 8, 11 e 12 para o SVM), na prática estes resultados não são satisfatórios. Como a quantidade de blocos processados por $fold$ varia de dezenas de milhares a centenas de milhares, as taxas de falsos positivos estão muito altas, principalmente no caso do SVM. Os resultados indicam que o SVM não está funcionando para os pontos 8, 11, 12. Dado que a saída da cascata tende a se agrupar em regiões, é natural o uso de algum algoritmo de agrupamento.

A Tabela 4.3 contém os resultados utilizando a versão rígida do pós-processamento descrito na seção 3.4. Esta abordagem permite apenas uma saída por ponto fiducial, se houver. Analisando esta tabela, é possível perceber uma queda significativa das taxas de acerto para os pontos ao redor da boca (pontos 10, 11, 12 e 13) em todos os métodos. Isto reflete a variabilidade destes pontos, tornando-os difíceis de serem detectados. Os resultados das duas versões do método proposto possuem resultados parecidos em todos os pontos e, como é esperado, as taxas para os pontos simétricos são compatíveis.

Apesar não ser possível fazer uma comparação direta entre o método nface e os demais, o nfaces apresentou um bom resultado em todos os pontos, com exceção dos pontos 8 e 10. No caso do ponto 8, uma possível explicação para uma taxa tão baixa

Tabela 4.2: Resultados obtidos sem pós-processamento.

ponto	DPI+Adaboost		DPI+Fisher		SVM	
fiducial	VP	FP	VP	FP	VP	FP
1	**95,23±4,50**	0,18±0,10	**93,64±8,46**	0,12±0,08	92,25±31,62	8,03±22,26
2	**97,42±5,87**	0,19±0,09	**99,60±1,36**	0,39±0,21	**99,60±2,1**	1,28±0,38
3	**95,63±4,06**	0,39±0,06	**99,80±1,05**	1,03±0,47	93,04±28,64	2,60±1,88
4	**97,81±5,52**	0,66±0,20	**99,40±1,49**	0,99±0,26	**99,80±1,06**	2,27±0,36
5	**98,41±3,74**	0,31±0,14	**99,40±3,35**	0,61±0,18	**97,20±8,44**	4,40±11,88
6	**93,65±6,38**	0,17±0,06	**91,06±6,56**	0,15±0,03	**98,81±6,63**	3,70±2,06
7	**93,24±6,30**	0,20±0,10	**93,04±4,52**	0,26±0,08	**99,60±2,1**	2,32±3,46
8	**88,67±6,48**	0,18±0,06	**88,87±6,18**	0,22±0,07	47,13±1,96	0,14±0,12
9	**90,66±7,10**	0,29±0,12	**93,65±6,17**	0,38±0,13	**95,43±7,80**	1,12±0,36
10	87,28±6,272	0,20±0,10	88,87±8,31	0,36±0,13	**96,83±11,08**	3,77±10,62
11	**80,71±7,29**	0,40±0,18	74,36±9,01	0,40±0,16	63,60±43,58	1,25±2,52
12	**84,50±9,11**	0,37±0,13	74,36±10,34	0,16±0,05	46,24±68,02	2,70±4,32
13	86,08±12,87	0,23±0,10	76,34±21,23	0,24±0,07	**98,61±3,92**	2,01±1,60

Tabela 4.3: Taxas de acerto para um pós-processamento rígido.

ponto	DPI+Adaboost	DPI+Fisher	SVM	nface
1	**87,67±4,42**	**88,27±11,45**	52,84±37,76	96,93±1,94
2	**96,42±5,99**	**98,01±3,56**	**97,21±4,84**	N/A
3	**88,07±7,66**	**89,26±8,89**	78,12±35,66	91,65±1,83
4	**92,04±9,60**	**92,63±9,16**	**97,01±6,34**	92,86±1,86
5	**98,02±3,53**	**98,41±3,38**	**91,64±10,30**	N/A
6	**85,70±10,80**	**84,31±9,79**	60,44±10,44	98,35±1,97
7	**82,50±4,85**	**81,51±5,58**	85,07±17,42	98,36±1,97
8	**64,62±6,91**	**63,63±8,92**	39,57±11,40	40,22±0,80
9	**74,34±10,19**	**75,93±11,02**	**79,92±8,60**	95,3±1,91
10	**64,01±6,32**	**63,23±8,61**	65,22±21,72	63,76±1,28
11	52,68±14,17	50,69±14,55	44,53±21,56	N/A
12	30,83±12,04	36,18±13,61	11,71±19,42	N/A
13	33,19±13,57	35,59±15,08	24,84±32,58	85,57±1,71

é a diferença entre a marcação utilizada no treinamento dos dois métodos. Algumas saídas do método *nfaces* podem ser observados na Figura 4.5.

A Tabela 4.4 contém os resultados utilizando um pós-processamento flexível (que permite mais de uma saída por ponto fiducial). O critério utilizado para determinar a distância no qual os pontos da saída da cascata foram agrupados é semelhante ao adotado na determinação do número de estágios. A distância usada em cada método foi a que apresentou a menor taxa de falsos positivos sem que houvesse uma queda significativa na taxa de acerto.

Como é esperado, as taxas de acerto obtidas com o uso de um agrupamento flexível são maiores que as taxas com agrupamento rígido. Contudo, isto ocorre às custas de um aumento nas taxas de falsos positivos. Novamente, os resultados obtidos para as duas versões propostas são parecidos e os pontos simétricos apresentaram taxas compatíveis. Assim como nos casos anteriores, os pontos ao redor da boca apresentaram um resultado inferior aos outros pontos (e a explicação para isto é a mesma, ou seja, estes pontos possuem uma variabilidade maior que os outros).

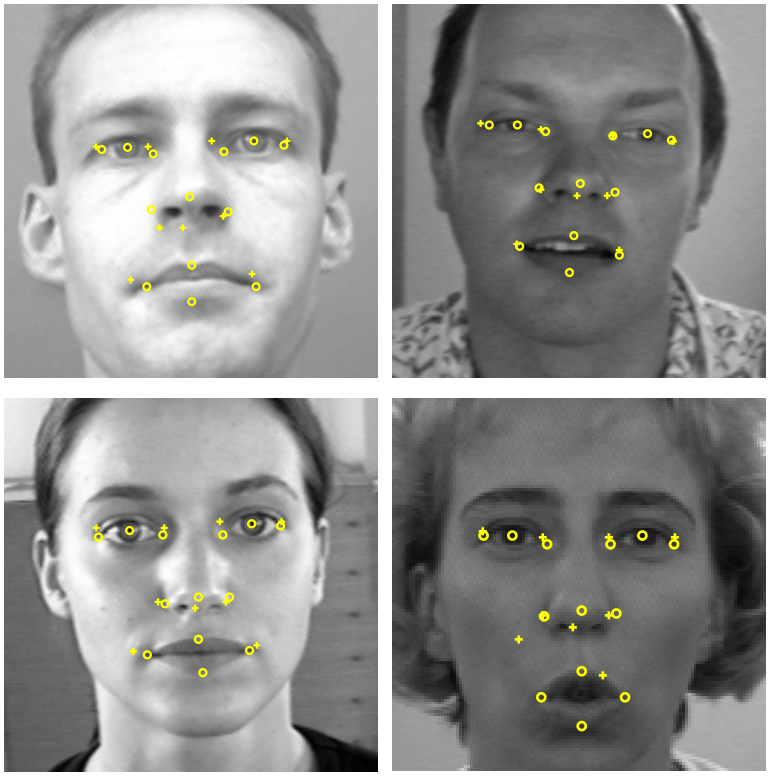


Figura 4.4: Saídas do método *nfaces* [3]. Os pontos marcados com "o" representam a marcação da base e os pontos marcados com "cruz" são a saída do método

Tabela 4.4: Resultados obtidos usando um agrupamento flexível.

ponto	DPI+Adaboost		DPI+Fisher		SVM	
fiducial	VP	FP	VP	FP	VP	FP
1	**92,64±4,45**	0,01±0,01	**92,05±9,97**	0,01±0,01	80,53±25,48	0,21±0,36
2	**97,42±5,87**	0,01±0,01	**99,4±1,48**	0,01±0,01	**99,60±3,90**	0,02±0,01
3	**89,86±8,73**	0,02±0,01	**88,87±9,19**	0,02±0,01	85,09±27,96	0,08±0,02
4	**92,25±9,06**	0,01±0,01	**92,83±8,83**	0,01±0,01	**98,60±4,84**	0,03±0,02
5	**98,41±3,74**	0,01±0,01	**98,61±3,21**	0,02±0,01	**95,22±8,70**	0,05±0,10
6	**91,86±7,91**	0,02±0,01	**89,67±5,24**	0,01±0,01	**90,26±6,20**	0,12±0,02
7	**91,65±5,32**	0,01±0,01	**92,05±6,56**	0,02±0,01	**92,23±11,22**	0,06±0,06
8	**86,68±6,93**	0,02±0,01	**87,08±6,53**	0,02±0,01	47,13±19,72	0,03±0,02
9	**89,86±6,73**	0,02±0,01	**93,05±5,99**	0,02±0,01	**95,24±7,50**	0,07±0,02
10	**75,94±5,30**	0,02±0,01	**74,96±7,95**	0,02±0,01	88,47±16,18	0,14±0,32
11	**64,81±8,28**	0,05±0,01	61,64±12,21	0,04±0,01	61,02±40,56	0,13±0,16
12	63,03±12,47	0,07±0,02	60,85±16,96	0,04±0,01	39,30±58,26	0,22±0,26
13	70,97±13,43	0,03±0,01	64,42±17,00	0,02±0,01	68,50±20,60	0,06±0,02

Embora as duas versões do sistema proposto possuam classificadores diferentes na saída do DPI, seus resultados foram parecidos. Como a saída do DPI é um valor real, o *AdaBoost* é capaz de discriminar intervalos de valores enquanto o discriminante de Fisher consegue apenas usar um limiar. Contudo, com o uso da cascata, esta limitação do Fisher desaparece, pois cada estágio determina um limiar e no fim é possível selecionar intervalos de valores, assim como no *AdaBoost*. Por outro lado o Fisher é muito mais fácil e rápido de ajustar do que o *AdaBoost*, uma vez que este último possui alguns parâmetros livres. Considerando ainda que a versão com discriminante de Fisher possui uma complexidade computacional menor que a versão com *AdaBoost* (que por sua vez possui uma complexidade menor que a do SVM), a versão com Fisher é a melhor opção.

Os resultados obtidos indicam ainda que é possível obter taxas melhores, principalmente se as características de deformação e de deslocamento dos pontos ao redor da boca forem incorporados ao projeto do DPI. Isto pode ser feito através de um reconhecimento multi-classes, onde a boca fechada representa uma das classes que se quer reconhecer e a boca fechada representa a outra. Ainda pode ser explorado o uso de alguma técnica de pós-processamento mais sofisticada que leve em consideração a saída dos classificadores dos outros pontos na tomada de decisão final.

Algumas imagens de saída do método proposto com o Fisher como classificador e agrupamento rígido como pós-processamento podem ser observadas na Figura 4.5.

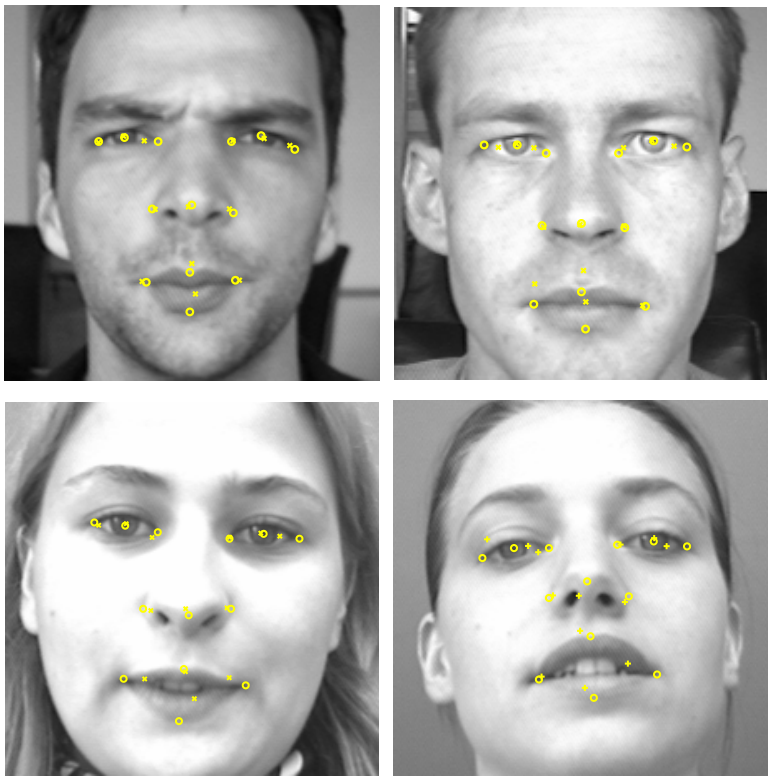


Figura 4.5: Saídas do método proposto. Nestes casos foi utilizado o discriminante de Fisher como segundo classificador e um agrupamento rígido na saída. Os pontos marcados com “o” representam a marcação da base e os pontos marcados com “cruz” são a saída do método

Capítulo 5

Conclusões

Neste trabalho é proposto um novo sistema de reconhecimento de características faciais. Este sistema é composto de um conjunto de etapas listadas a seguir:

- Reconhecimento da face, através do algoritmo Viola-Jones.
- Pré-processamento da imagem de face, com o auxílio de uma sistema de correção de iluminação proposto por [44].
- Restrição da região de busca com um modelo probabilístico para pontos fiduciais.
- A cascata de classificadores, onde cada estágio é composto pelo DPI (Detector por Produto Interno) e um segundo classificador. Foi proposta uma versão com o discriminante de Fisher e outra com o *AdaBoost*.
- Pós-processamento, no qual a saída da cascata é agrupada para fornecer a saída do sistema. Foi utilizada neste caso uma versão flexível, que permite mais de uma saída por ponto fiducial e por imagem e uma versão rígida, que permite apenas uma saída.

A principal etapa deste sistema é a cascata de classificadores. Cada estágio desta cascata contém um novo tipo de detector baseado em filtros de correlação, denominado DPI. O DPI foi desenvolvidas em conjunto com o Sr. Waldir Sabino da Silva Júnior em seu trabalho de Doutorado. Embora o DPI possua uma forte restrição com relação à ortogonalidade das classes, uma solução foi proposta no sentido de contornar este problema. Trata-se de um esquema de normalização onde os vetores de características (utilizados no treino e no teste) são mapeados em um espaço de dimensão $d + 1$. Isto faz com que a saída do DPI seja um valor real no intervalo $[-1, 1]$. Um segundo classificador é utilizado na saída do DPI com o objetivo de encontrar um limiar automático. Caso o valor do produto interno entre

o detector e a amostra em questão seja maior que este limiar, o reconhecimento é positivo.

O sistema proposto foi comparado com dois outros métodos. O primeiro deles é um que utiliza o SVM (*Support Vector Machine*) como classificador principal. Neste caso, o treinamento e teste foram feitos nas mesmas condições que o sistema proposto. A outra comparação foi feita com um método estado-da-arte proposto em [2]. Como neste caso o algoritmo disponível apenas aplica um detector previamente treinado, não foi possível efetuar um treinamento sob as mesmas condições dos métodos citados anteriormente. Contudo, o teste foi feito nas mesmas condições, permitindo, ao menos, uma comparação qualitativa.

Outra contribuição deste trabalho foram as anotações manuais dos treze pontos fiduciais de todas as 503 imagens utilizadas. Foram anotadas as imagens frontais, sem óculos, sem barba e sem bigode da base BioID.

5.1 Trabalhos futuros

Nesta seção são apresentadas algumas sugestões para a continuação deste trabalho:

- A utilização de outras bases de dados como a FERET [53], está sendo feita no momento. Contudo, isto não foi concluído ainda e portanto segue como trabalho futuro.

- Embora um estudo sobre a complexidade computacional do sistema proposto não tenha sido feita, a detecção de características faciais é rápida o suficiente para ser empregada em aplicações de tempo real. Uma das possíveis continuações deste trabalho é o desenvolvimento do sistema de reconhecimento de características faciais em vídeo.

- A utilização do DPI em outros tipos de problemas, como reconhecimento de faces, de placas de automóvel, reconhecimento de caracteres óticos etc.

- Outra alternativa seria o uso do PCA para encontrar as direções de maior energia. Um estudo sobre a separabilidade das classes nestas direções pode apontar qual delas apresentam maior poder de classificação. Um detector DPI poderia ser obtido para cada uma destas direções em particular.

- A utilização de outra técnica de pós-processamento. Na versão atual do pós-processamento com abordagem rígida, a decisão final é a média de todos os pontos classificados positivamente. Uma alternativa seria a utilização do ponto mais provável como decisão final. Isto pode ser feito escolhendo-se o ponto que possui a menor distância de Mahalanobis.

Referências Bibliográficas

[1] VUKADINOVIC, D., PANTIC, M. "Fully automatic facial feature point detection using Gabor feature based boosted classifiers". In: *IEEE Intl. Conf. on Systems, Man and Cybernetics 2005*, pp. 1692–1698, September 2005.

[2] SIVIC, J., EVERINGHAM, M., ZISSERMAN, A. "Who are you? Learning person specific classifiers from video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[3] EVERINGHAM, M., SIVIC, J., ZISSERMAN, A. "Hello! My name is... Buffy. Automatic naming of characters in TV video". In: *Proceedings of the Brittish Machine Vision Conference*, 2006.

[4] CRISTINACCE, D., COOTES, T. "Automatic feature localisation with constrained local models", *Pattern Recognition*, v. 41, n. 10, pp. 3054–3067, October 2008.

[5] COOTES, T. F., EDWARDS, G. J., TAYLOR, C. J. "Active appearance models", *Proceedings of the European Conference on Computer Vision*, v. 2, pp. 484–498, 1998.

[6] CRISTINACCE, D., COOTES, T. "A comparison of shape constrained facial feature detectors". In: 6^{th} *International Conference on Automatic Face and Gesture Recognition 2004, Seoul, Korea*, pp. 375–380, 2004.

[7] CRISTINACCE, D., COOTES, T. "Facial feature detection and tracking with automatic template selection". In: *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pp. 429–434, April 2006.

[8] FISHER, R. A. "The use of multiple measurements in taxonomic problems", *Annals of Eugenics*, v. 7, pp. 179–188, 1936.

[9] FREUND, Y., SCHAPIRE, R. E. "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Sciences*, v. 55, n. 1, pp. 119–139, 1997.

[10] BRIGGS, T. "SMATLAB interface for SVM-Light". http://sourceforge.net /projects/mex-svm/, 2010. [último acesso em Fevereiro de 2010].

[11] SIVIC, J., EVERINGHAM, M., ZISSERMAN, A. "Automatic naming of characters in TV video". http://www.robots.ox.ac.uk/∼vgg/research/nface /index.html, 2010. [último acesso em Fevereiro de 2010].

[12] AG, B. "BIOID". http://www.bioid.com, 2010. [último acesso em Fevereiro de 2010].

[13] VIOLA, P., JONES, M. "Rapid object detection using a boosted cascade of simple features". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, v. 1, pp. I–511–I–518 vol.1, 2001.

[14] DUDA, R. O., HART, P. E., STORK, D. G. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[15] THEODORIDIS, S., KOUTROUMBAS, K. *Pattern Recognition*. 4 ed. San Diego, California, USA, Academic Press, 2009.

[16] WEBB, A. R. *Statistical Pattern Recognition*. John Wiley & Sons, 2002.

[17] SCHAPIRE, R. E. "The strength of weak learnability", *Machine Learning*, v. 5, n. 2, pp. 197–227, June 1990.

[18] FREUND, Y. "Boosting a weak learning algorithm by majority", *Inform. Comput.*, v. 121, n. 2, pp. 256–285, 1995.

[19] KEARNS, M., VALIANT, L. G. *Learning boolean formulae or finite automata is as hard as factoring*. Relatório Técnico TR 14-88, Harvard University Aiken Computation Laboratory, 1988.

[20] KEARNS, M., VALIANT, L. "Cryptographic limitations on learning boolean formulae and finite automata", *Journal of the ACM*, v. 41, pp. 433–444, 1994.

[21] SCHAPIRE, R. E., SINGER, Y. "Improved boosting algorithms using confidence-rated predictions", *Machine Learning*, v. 37, n. 3, pp. 297–336, December 1999.

[22] FRIEDMAN, J., HASTIE, T., TIBSHIRANI, R. "Additive logistic regression: a statistical view of boosting", *Annals of Statistics*, v. 28, 2000.

[23] VEZHNEVETS, A., VEZHNEVETS, V. "Modest AdaBoost. Teaching AdaBoost to generalize better". In: *Graphicon*, 2005.

[24] GRAPHICS, LAB, M. "GML AdaBoost Matlab toolbox". http://graphics.cs.msu.ru/en/science/research/machinelearning/adaboosttoolbox. [último acesso em Fevereiro de 2010].

[25] VAPNIK, V. N. *The Nature of Statistical Learning Theory*. New York, NY, USA, Springer-Verlag New York, Inc., 1995.

[26] VAPNIK, V. N. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[27] VAPNIK, V. N., CHERVONENKIS, Y. A. "On the uniform convergence of relative frequencies of events to their probabilities", *Theory of Probability and its Applications*, v. 16, n. 2, pp. 264–280, 1971.

[28] VAPNIK, V. N. *Estimation of Dependences Based on Empirical Data*. USSR, Nauka, 1979. [in Russian].

[29] BURGES, C. J. C. "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, v. 2, pp. 121–167, 1998.

[30] CORTES, C., VAPNIK, V. "Support-vector networks". In: *Machine Learning*, pp. 273–297, 1995.

[31] BOSER, B. E., GUYON, I. M., VAPNIK, V. N. "A training algorithm for optimal margin classifiers". In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152. ACM Press, 1992.

[32] AIZERMAN, A., BRAVERMAN, E. M., ROZONER, L. I. "Theoretical foundations of the potential function method in pattern recognition learning", *Automation and Remote Control*, v. 25, pp. 821–837, 1964.

[33] MERCER, J. "Functions of positive and negative type and their connection with the theory of integral equations", *Philosophical Transactions of the Royal Society*, 1909.

[34] COURANT, R., HILBERT, D. *Methods of Mathematical Physics*. Interscience, 1953.

[35] JOACHIMS, T. "SVMlight". http://svmlight.joachims.sorg/, 2010. [último acesso em Fevereiro de 2010].

[36] LIENHART, R., MAYDT, J. "An extended set of Haar-like features for rapid object detection". In: *IEEE ICIP 2002*, v. 1, pp. I–900–I–903 vol.1, 2002.

[37] OPENCV. "Open Computer Vision Library". http://sourceforge.net/projects/opencvlibrary/, 2010. [último acesso em Fevereiro de 2010].

[38] PITTSBURGH PATTERN RECOGNITION, I. "Pittpatt". http://www.pittpatt.com, 2010. [último acesso em Fevereiro de 2010].

[39] LUXAND, I. "Luxand". http://www.luxand.com, 2010. [último acesso em Fevereiro de 2010].

[40] DU, C., WU, Q., YANG, J., et al. "SVM based ASM for facial landmarks location". pp. 321 –326, july 2008.

[41] NAGAMALLA, S., DHARA, B. "A novel face recognition method using facial landmarks". pp. 445 –448, feb. 2009.

[42] JAHANBIN, S., BOVIK, A., CHOI, H. "Automated facial feature detection from portrait and range images". pp. 25 –28, 2008.

[43] PHILLIPS, P., FLYNN, P., SCRUGGS, T., et al. "Overview of the face recognition grand challenge". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, v. 1, pp. 947–954 vol. 1, June 2005.

[44] TAN, X., TRIGGS, B. "Enhanced local texture feature sets for face recognition under difficult lighting conditions". In: *AMFG*, pp. 168–182, 2007.

[45] TAN, X., TRIGGS, B. "Bill Triggs". http://lear.inrialpes.fr/people/triggs/src/, 2007. [último acesso em Fevereiro de 2010].

[46] MAHALANOBIS, P. C. "On the generalised distance in statistics". In: *Proceedings National Institute of Science, India*, v. 2, pp. 49–55, 1936.

[47] KUMAR, B. V. K. V., MAHALANOBIS, A., JUDAY, R. D. *Correlation Pattern Recognition*. New York, NY, USA, Cambridge University Press, 2005.

[48] MAHALANOBIS, A., KUMAR, B. V. K. V., CASASENT, D. "Minimum average correlation energy filters", *Applied Optics*, v. 26, n. 17, pp. 3633–3640, 1987.

[49] XIE, C., SAVVIDES, M., KUMAR, B. V. "Redundant class-dependence feature analysis based on correlation filters using FRGC2.0 data", *Computer Vision and Pattern Recognition Workshop*, v. 0, pp. 153, 2005.

[50] LAI, H., RAMANATHAN, V., WECHSLER, H. "Reliable face recognition using adaptive and robust correlation filters", *Computer Vision and Image Understanding*, v. 111, n. 3, pp. 329 – 350, 2008.

[51] JESORSKY, O., KIRCHBERG, K. J., FRISCHHOLZ, R. "Robust face detection using the Hausdorff distance". In: *AVBPA '01: Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 90–95, London, UK, 2001. Springer-Verlag.

[52] MAIA, J. G. R., GOMES, F. D. C., DE SOUZA, O. "Automatic eye localization in color images". In: *Computer Graphics and Image Processing, 2007. SIBGRAPI 2007. XX Brazilian Symposium on*, pp. 195–204, Oct. 2007.

[53] OF STANDARDS, N. I., TECHNOLOGY. "The facial recognition technology (FERET) database". . http://www.itl.nist.gov/iad/humanid/feret/, 2010. [último acesso em Fevereiro de 2010].

Apêndice A

DPI - caso complexo

Nas Seções 3.3.1 e 3.3.2 o classificador $\mathbf{h}$ foi obtido levando-se em consideração que $\mathbf{X}$ é uma variável aleatória real. Caso $\mathbf{X}$ seja uma variável aleatória complexa, o projeto do classificador sofre algumas alterações. Este Apêndice descreve o DPI para o caso complexo. Para tanto, será adotado o caso multi-classes, por se tratar uma generalização do caso de única classe.

Seja uma variável aleatória complexa $\mathbf{X}$, cujas realizações $\mathbf{x}$ podem ser associadas a uma classe A_i, $i = 1, \cdots, n$. Deseja-se um vetor $\mathbf{h}_A$, complexo, capaz de discriminar amostras associadas a um conjunto de classes $A = \{A_1, \cdots, A_m\}$, $m \leq n$, das demais. Para tanto a regra de classificação deve ser:

$$\mathbf{h}_A^{*t}\mathbf{x} = \begin{cases} 1, \text{ se } \mathbf{x} \in A \\ 0, \text{ caso contrário,} \end{cases} \tag{A.1}$$

que pode ser resumida em:

$$\mathbf{h}_A^{*t}\mathbf{X} = C, \tag{A.2}$$

onde $C = 1$ se $\mathbf{x} \in A$ e $C = 0$ se $\mathbf{x} \notin A$.

Como $\mathbf{h}_A$ deve ser ótimo no sentido dos mínimos quadrados, definindo o erro como sendo $e = \mathbf{h}_A^{*t}\mathbf{x} - C$, o erro quadrático pode ser desenvolvido como segue:

$$\begin{aligned} \|e\|^2 &= (\mathbf{h}_A^{*t}\mathbf{X} - C)(\mathbf{h}_A^{*t}\mathbf{X} - C)^{*t} \\ &= (\mathbf{h}_A^{*t}\mathbf{X})(\mathbf{h}_A^{*t}\mathbf{X})^{*t} - (\mathbf{h}_A^{*t}\mathbf{X})(C)^{*t} - (C)(\mathbf{h}_A^{*t}\mathbf{X})^{*t} + (C)(C)^{*t} \\ &= \mathbf{h}_A^{*t}\mathbf{X}\mathbf{X}^{*t}\mathbf{h}_A - \mathbf{h}_A^{*t}\mathbf{X}C^{*t} - C\mathbf{X}^{*t}\mathbf{h}_A + CC^{*t}. \end{aligned} \tag{A.3}$$

Dado que $\mathbf{h}_A$ é um vetor complexo é possível expressá-lo como uma soma de suas componentes real $\mathbf{h}_R$ e imaginária $\mathbf{h}_I$. Escrevendo $\mathbf{h}_A = \mathbf{h}_R + j\mathbf{h}_I$, o erro quadrático

pode ser expandido da seguinte forma:

$$\begin{aligned}
\|e\|^2 &= (\mathbf{h}_R + j\mathbf{h}_I)^{*t}\mathbf{X}\mathbf{X}^{*t}(\mathbf{h}_R + j\mathbf{h}_I) - (\mathbf{h}_R + j\mathbf{h}_I)^{*t}\mathbf{X}C^{*t} \\
&\quad -C\mathbf{X}^{*t}(\mathbf{h}_R + j\mathbf{h}_I) + CC^{*t} \\
&= (\mathbf{h}_R^t - j\mathbf{h}_I^t)\mathbf{X}\mathbf{X}^{*t}(\mathbf{h}_R + j\mathbf{h}_I) - (\mathbf{h}_R^t - j\mathbf{h}_I^t)\mathbf{X}C^{*t} \\
&\quad -C\mathbf{X}^{*t}(\mathbf{h}_R + j\mathbf{h}_I) + CC^{*t} \\
&= \mathbf{h}_R^t\mathbf{X}\mathbf{X}^{*t}\mathbf{h}_R + j\mathbf{h}_R^t\mathbf{X}\mathbf{X}^{*t}\mathbf{h}_I - j\mathbf{h}_I^t\mathbf{X}\mathbf{X}^{*t}\mathbf{h}_R + \mathbf{h}_I^t\mathbf{X}\mathbf{X}^{*t}\mathbf{h}_I \\
&\quad -\mathbf{h}_R^t\mathbf{X}C^{*t} + j\mathbf{h}_I^t\mathbf{X}C^{*t} - C\mathbf{X}^{*t}\mathbf{h}_R - jC\mathbf{X}^{*t}\mathbf{h}_I + CC^{*t}. \qquad \text{(A.4)}
\end{aligned}$$

O valor esperado do erro quadrático $E[\|e\|^2]$ fica:

$$\begin{aligned}
E[\|e\|^2] &= \mathbf{h}_R^t E[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_R + j\mathbf{h}_R^t E[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_I - j\mathbf{h}_I^t E[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_R \\
&\quad +\mathbf{h}_I^t E[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_I - \mathbf{h}_R^t E[\mathbf{X}C^{*t}] + j\mathbf{h}_I^t E[\mathbf{X}C^{*t}] \\
&\quad -E[C\mathbf{X}^{*t}]\mathbf{h}_R - jE[C\mathbf{X}^{*t}]\mathbf{h}_I + E[CC^{*t}]. \qquad \text{(A.5)}
\end{aligned}$$

Para encontrar o vetor $\mathbf{h}_A$ que minimiza o erro quadrático médio, deve-se tomar as derivadas parciais das partes real e imaginária e igualar a zero como segue:

$$\begin{aligned}
\frac{\partial\|e\|^2}{\partial\mathbf{h}_R} &= (E[\mathbf{X}\mathbf{X}^{*t}] + E[\mathbf{X}\mathbf{X}^{*t}]^{*t})\mathbf{h}_R + jE[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_I - jE[\mathbf{X}\mathbf{X}^{*t}]^{*t}\mathbf{h}_I \\
&\quad -E[\mathbf{X}C^{*t}] - E[C\mathbf{X}^{*t}]^{*t} \\
&= 0, \qquad \text{(A.6)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\|e\|^2}{\partial\mathbf{h}_I} &= jE[\mathbf{X}\mathbf{X}^{*t}]^{*t}\mathbf{h}_R - jE[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_R + (E[\mathbf{X}\mathbf{X}^{*t}] + E[\mathbf{X}\mathbf{X}^{*t}]^{*t})\mathbf{h}_I \\
&\quad +jE[\mathbf{X}C^{*t}] - jE[C\mathbf{X}^{*t}]^{*t} \\
&= 0. \qquad \text{(A.7)}
\end{aligned}$$

Multiplicando a Equação (A.7) por j e somando com a Equação(A.6), obtém-se:

$$\begin{aligned}
&\frac{\partial\|e\|^2}{\partial\mathbf{h}_R} + j\frac{\partial\|e\|^2}{\partial\mathbf{h}_I} = \\
= \;& (E[\mathbf{X}\mathbf{X}^{*t}] + E[\mathbf{X}\mathbf{X}^{*t}]^{*t})\mathbf{h}_R + jE[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_I - jE[\mathbf{X}\mathbf{X}^{*t}]^{*t}\mathbf{h}_I \\
& -E[\mathbf{X}C^{*t}] - E[C\mathbf{X}^{*t}]^{*t} - E[\mathbf{X}\mathbf{X}^{*t}]^{*t}\mathbf{h}_R \\
& +E[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_R + j(E[\mathbf{X}\mathbf{X}^{*t}] + E[\mathbf{X}\mathbf{X}^{*t}]^{*t})\mathbf{h}_I - E[\mathbf{X}C^{*t}] \\
& +E[C\mathbf{X}^{*t}]^{*t} \\
= \;& (E[\mathbf{X}\mathbf{X}^{*t}] + E[\mathbf{X}\mathbf{X}^{*t}]^{*t})(\mathbf{h}_R + j\mathbf{h}_I) + E[\mathbf{X}\mathbf{X}^{*t}](\mathbf{h}_R + j\mathbf{h}_I) \\
& -E[\mathbf{X}\mathbf{X}^{*t}]^{*t}(\mathbf{h}_R + j\mathbf{h}_I) - 2E[\mathbf{X}C^{*t}] \\
= \;& 2E[\mathbf{X}\mathbf{X}^{*t}](\mathbf{h}_R + j\mathbf{h}_I) - 2E[\mathbf{X}C^{*t}] \\
= \;& 2E[\mathbf{X}\mathbf{X}^{*t}]\mathbf{h}_A - 2E[\mathbf{X}C^{*t}] \\
= \;& 0. \qquad\qquad (A.8)
\end{aligned}$$

Com isto, a expressão para o vetor $\mathbf{h}_A$ fica:

$$\mathbf{h}_A = \left(E[\mathbf{X}\mathbf{X}^{*t}]\right)^{-1} E[\mathbf{X}C^{*t}]. \qquad (A.9)$$

Para obter a expressão final do vetor $\mathbf{h}_A$, basta desenvolver os termos $E[\mathbf{X}C^{*t}]$ e $E[\mathbf{X}\mathbf{X}^{*t}]$. Considerando $\bar{A}$ como sendo o complemento do conjunto A e $p(A_i)$ a probabilidade da realização de uma amostra pertencente à classe A_i, o termo $E[\mathbf{X}C^{*t}]$ pode ser expandido da seguinte forma:

$$E[\mathbf{X}C^{*t}] = E[\mathbf{X}C^{*t}|A_i]p(A_i) + E[\mathbf{X}C^{*t}|\bar{A}_i](1 - p(A_i)). \qquad (A.10)$$

Como $C = \{0, 1\}$ e considerando que A_i possui L_i amostras de treinamento $\mathbf{x}_{ik}$, $k = \{1, \cdots, L_i\}$, a Equação (A.10) pode ser expressa em termos das amostras de treinamento:

$$E[\mathbf{X}C^{*t}] = p(A_i)\frac{1}{L_i}\sum_{k=1}^{L_i}\mathbf{x}_{ik}, \qquad (A.11)$$

que é o mesmo resultado obtido para o caso real.

Já o termo $E[\mathbf{X}\mathbf{X}^{*t}]$ possui uma pequena modificação. Re-escrevendo em função das amostras de treinamento:

$$E[\mathbf{X}\mathbf{X}^{*t}] = \sum_{k=1}^{n} p(A_k)\frac{1}{L_k}\sum_{l=1}^{L_k}\mathbf{x}_{kl}\mathbf{x}_{kl}^{*t}. \qquad (A.12)$$

Este últimos resultados permitem escrever o vetor $\mathbf{h}_A$ em função da amostras de

treinamento:

$$\mathbf{h}_A = \left(\sum_{k=1}^{n} p(A_k) \frac{1}{L_k} \sum_{l=1}^{L_k} \mathbf{x}_{kl} \mathbf{x}_{kl}^{*t} \right)^{-1} \sum_{k=1}^{m} p(A_k) \frac{1}{L_k} \sum_{l=1}^{L_j} \mathbf{x}_{kl}. \tag{A.13}$$

Para o caso complexo, a matriz de correlação $\mathbf{R}_{A_i}$ e a média $\boldsymbol{\mu}_{A_i}$ são:

$$\mathbf{R}_{A_i} = \frac{1}{L_i} \sum_{j=1}^{L_i} \mathbf{x}_{ij} \mathbf{x}_{ij}^{*t}, \tag{A.14}$$

$$\boldsymbol{\mu}_{A_i} = \frac{1}{L_i} \sum_{j=1}^{L_i} \mathbf{x}_{ij}. \tag{A.15}$$

Com isto, o vetor $\mathbf{h}_A$ pode, finalmente, ser expresso em termos dos momentos amostrais:

$$\mathbf{h}_A = \left(\sum_{j=1}^{n} p(A_j) \mathbf{R}_{A_j} \right)^{-1} \sum_{j=1}^{m} p(A_j) \boldsymbol{\mu}_{A_j}. \tag{A.16}$$

Este vetor pode ser utilizado para discriminar realizações da variável aleatória $\mathbf{X}$ associadas ao conjunto A. A condição de existência do vetor $\mathbf{h}_A$ é que o termo $\left(\sum_{j=1}^{n} p(A_j) \mathbf{R}_{A_j} \right)$ possua uma inversa. Para tanto a dimensão dos vetores de entrada deve ser menor que o número de amostras de treinamento.