



COPPE/UFRJ

RECONHECIMENTO DE PADRÕES UTILIZANDO FILTROS DE
CORRELAÇÃO COM ANÁLISE DE COMPONENTES PRINCIPAIS

Waldir Sabino da Silva Júnior

Tese de Doutorado 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 Doutor em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da
Silva
Siome Klein Goldenstein

Rio de Janeiro
Julho de 2010

RECONHECIMENTO DE PADRÕES UTILIZANDO FILTROS DE
CORRELAÇÃO COM ANÁLISE DE COMPONENTES PRINCIPAIS

Waldir Sabino da Silva Júnior

TESE 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 DOUTOR
EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

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

Prof. Siome Klein Goldenstein, Ph.D.

Prof. Carlos Eduardo Pedreira, Ph.D.

Prof. Cláudio Rosito Jung, D.Sc.

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Prof. José Ricardo de Almeida Torreão, Ph.D.

RIO DE JANEIRO, RJ – BRASIL

JULHO DE 2010

Silva Júnior, Waldir Sabino da

Reconhecimento de Padrões utilizando Filtros de Correlação com Análise de Componentes Principais/Waldir Sabino da Silva Júnior. – Rio de Janeiro: UFRJ/COPPE, 2010.

XX, 204 p.: il.; 29,7cm.

Orientadores: Eduardo Antônio Barros da Silva

Siome Klein Goldenstein

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2010.

Referências Bibliográficas: p. 186 – 201.

1. Filtragem Discriminativa. 2. Detectores por Produto Interno. 3. Análise de Componentes Principais. I. Barros da Silva, Eduardo Antônio *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

- Primeiramente à minha Mãe Sônia Ely e ao meu pai Waldir pelo apoio em todos os momentos de minha vida.
- À minha esposa Thyrsiana e ao meu filho Vitor.
- À Universidade Federal do Amazonas (UFAM) por permitir a realização desta Tese.
- À Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) pelo apoio financeiro.
- A meus orientadores Prof. Eduardo e Prof. Siome. Em especial ao prof. Eduardo por ser um profissional exemplar e sempre me incentivar nos momentos difíceis.
- Aos amigos professores da UFAM, principalmente Prof. Vicente, Prof. Cícero, Prof^a. Marly e Prof. Edgar.
- A todos os amigos do Laboratório de Processamento de Sinais (LPS) pela ajuda nesta jornada.

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

RECONHECIMENTO DE PADRÕES UTILIZANDO FILTROS DE
CORRELAÇÃO COM ANÁLISE DE COMPONENTES PRINCIPAIS

Waldir Sabino da Silva Júnior

Julho/2010

Orientadores: Eduardo Antônio Barros da Silva

Siome Klein Goldenstein

Programa: Engenharia Elétrica

Neste trabalho investigamos algoritmos para detecção de padrões. Duas contribuições foram feitas. Primeiramente, modificamos a técnica por Filtragem Discriminativa (DF), que efetua a detecção de padrões utilizando filtragem bidimensional, para obter filtros discriminativos robustos. Isto é feito através do projeto de filtros para cada uma das componentes principais com maiores energias. A seguir, desenvolvemos um método para detecção de padrões denominado Detector por Produto Interno (IPD), que será ótimo no sentido de minimizar o erro médio quadrático de classificação. O IPD utiliza o produto interno para determinar se um candidato é ou não o padrão de interesse. Demonstramos que a Filtragem Discriminativa e os Filtros de Correlação são casos particulares do IPD. Por último, apresentamos como projetar IPDs robustos utilizando as componentes principais. Os desempenhos dos métodos propostos são avaliados em um contexto de detecção de pontos salientes em faces humanas, utilizando-se validação cruzada. Isto é feito para duas bases de dados (*BioID* e *Feret*) com, respectivamente, 503 e 2004 imagens rotuladas. Para comparação, desenvolvemos métodos similares utilizando classificadores SVM (linear e não-linear). Os métodos propostos proporcionam resultados competitivos quando comparados aos dos baseados em SVM.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

PATTERN RECOGNITION USING CORRELATION FILTERS WITH
PRINCIPAL COMPONENTS ANALYSIS

Waldir Sabino da Silva Júnior

July/2010

Advisors: Eduardo Antônio Barros da Silva

Siome Klein Goldenstein

Department: Electrical Engineering

In this work, we investigate pattern detection algorithms. We have made two contributions. First, we modified the discriminative filtering (DF) technique, that detects patterns using two-dimensional filtering, in order to obtain robust discriminative filters. This is achieved by designing filters for the highest energy principal components of the patterns. Then, we developed a method for pattern detection referred to as inner product detector (IPD), that is optimum in the sense of minimizing the mean-squared detection error. The IPD uses the inner product for determining if a candidate is a pattern of interest. We demonstrate that the discriminative filtering and the correlation filters are particular cases of the IPD. We also demonstrate how to design robust IPDs using principal components. The performance of the proposed methods is evaluated in the context of fiducial points detection in human faces using cross validation. This is performed for two face databases (*BioID* and *Feret*), using, respectively, 503 and 2004 labeled images. For comparison, we develop similar methods using linear and nonlinear SVM classifiers. The proposed methods provide competitive results when compared with the results of SVM-based methods.

Sumário

Lista de Abreviaturas	xix
1 Introdução	1
1.1 Organização da Tese	3
2 Fundamentos Teóricos	5
2.1 Reconhecimento de Padrões	5
2.1.1 Conceitos Iniciais	5
2.1.2 Classificação supervisionada, não-supervisionada e semi-supervisionada	6
2.1.3 Aprendizado e Operação	7
2.1.4 Técnicas de Classificação	8
2.2 Técnicas para Detecção de Pontos Fiduciais	15
3 Filtragem Discriminativa	18
3.1 Método por Restauração do Impulso	19
3.2 Restauração do Impulso Bidimensional: Aplicação em Imagens	22
3.3 Filtro OU	26
3.4 Simulações	28
3.5 Considerações Finais	32
4 Filtragem Discriminativa utilizando Análise de Componentes Principais (DF-B-PCA)	33
4.1 Análise de Componentes Principais	33
4.2 <i>Eigenfaces</i>	35
4.3 <i>Eigenpoints</i>	36

4.4	Filtragem Discriminativa utilizando Análise de Componentes Principais	37
4.5	Considerações Finais	40
5	Filtragem Discriminativa utilizando Análise de Componentes Principais (DF-B-PCA): Procedimentos Experimentais e Simulações	41
5.1	Introdução	41
5.2	Rótulos Manuais e Automáticos	42
5.3	Bases de Dados	42
5.4	Métrica de Desempenho	43
5.4.1	Desempenho do Classificador	44
5.5	Procedimentos para Pré-processamento, Treinamento e Teste	46
5.5.1	Pré-processamento das Imagens	46
5.5.2	Treinamento (Fase de Aprendizagem)	47
5.5.3	Teste (Fase de Operação)	54
5.6	Influência da Quantidade de Componentes Principais	59
5.6.1	Análise dos Resultados	72
5.7	Comparação entre DF-B-PCA, DF e DF-B	75
5.8	Comparação entre DF-B-PCA e SVM	78
5.9	Estratégias para Pós-processamento para o Método DF-B-PCA	81
5.10	Estratégias para Pós-processamento: Comparação entre DF-B-PCA e SVM	89
5.11	Considerações Finais	97
6	Detectors por Produto Interno com Minimização do Erro Quadrático Médio (IPD)	98
6.1	Detectors por Produto Interno com Minimização do Erro Quadrático Médio para uma Única Classe de Padrões	99
6.2	Detectors por Produto Interno com Minimização do Erro Quadrático Médio para Múltiplas Classes de Padrões	103
6.3	Detectors por Produto Interno Complexo com Minimização do Erro Quadrático Médio	106
6.3.1	Detectors por Produto Interno Complexo com Transformação Linear	109

6.3.2	Detectores por Produto Interno Complexo com Transformação Linear - Interpretação no Domínio Transformado	111
6.4	Detectores por Produto Interno e Filtragem Discriminativa	116
6.4.1	Detectores por Produto Interno e Filtragem Discriminativa: Caso para Único Padrão	116
6.4.2	Detectores por Produto Interno e Filtro OU: caso para Múltiplos Padrões	121
6.5	Detectores por Produto Interno para um Pequeno Número de Imagens (IPD-hd)	127
6.5.1	Detectores por Produto Interno para um Pequeno Número de Imagens (IPD-hd) com Transformação Linear	129
6.6	Detector por Produto Interno com Dimensão Extra	132
6.7	Detector por Produto Interno utilizando Análise de Componentes Principais (IPD-B-PCA)	134
6.8	Considerações Finais	137
7	Detectores por Produto Interno (IPD): Procedimentos Experimentais e Simulações	138
7.1	Introdução	138
7.2	Procedimentos para Pré-processamento, Treinamento e Teste	139
7.2.1	Pré-processamento das Imagens	139
7.2.2	Treinamento (Fase de Aprendizagem)	139
7.2.3	Teste (Fase de Operação)	143
7.3	Influência da Quantidade de Componentes Principais	145
7.3.1	Análise dos Resultados	158
7.4	Comparação entre IPD-B-PCA and IPD-B	160
7.5	Comparação entre IPD-B-PCA, DF-B-PCA e SVM	162
7.6	Estratégias para Pós-processamento para o Método IPD-B-PCA	165
7.7	Estratégias para Pós-processamento: Comparação entre IPD-B-PCA, DF-B-PCA e SVM	173
7.8	Considerações Finais	181
8	Conclusões	182

8.1	Considerações Finais da Tese	182
8.2	Propostas para Trabalhos Futuros	183
	Referências Bibliográficas	186
A	Imagens bases de dados <i>BioID</i> e <i>Feret</i>	202
B	Artigos Desenvolvidos nesta Tese	204

Lista de Figuras

2.1	Problema de classificação em termos de espaços vetoriais	6
2.2	Diagrama em blocos de um sistema de classificação	7
3.1	Sinal de saída $\mathbf{C} = \Theta * \mathbf{U}$	22
3.2	Detector \mathbf{A} para um padrão quadriculado, com dimensões 3×3	23
3.3	Detecção em uma imagem de teste da letra “T”.	31
3.4	Detecção em uma imagem de teste das letras “T” e “I”.	32
4.1	Representação da Análise de Componentes Principais	34
4.2	Exemplos pontos fiduciais e <i>eigenpoints</i>	37
4.3	Projeto de S filtros discriminativos para S componentes principais	38
5.1	Exemplo da margem usada para determinarmos a taxa de FP	45
5.2	Exemplo Taxas de TP e FP	45
5.3	Exemplo para distribuição de acerto média acumulada <i>versus</i> distância inter-ocular.	46
5.4	Pré-processamento das imagens	47
5.5	Modelo gaussiano aplicado em uma imagem	48
5.6	Exemplo de blocos para a classe de negativos e para a classe de positivos	49
5.7	Diagrama em blocos para os procedimentos iniciais para treinamento	49
5.8	Fase de Aprendizagem para o método DF-B-PCA	52
5.9	Diagramas em blocos para os métodos DF-B, DF e SVM.	53
5.10	Diagrama em blocos para os procedimentos iniciais para teste	54
5.11	Exemplo de rótulos automáticos agrupados.	55
5.12	Diagrama em blocos para pós-processamento na etapa de teste.	55
5.13	Diagrama em blocos para teste do método DF-B-PCA	57
5.14	Diagrama em blocos para teste dos métodos DF-B, DF e SVM.	58

5.15	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 00 e 05. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	60
5.16	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 00 e 05. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . .	61
5.17	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 02 e 03. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	62
5.18	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 02 e 03. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . .	63
5.19	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 01 e 04. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	64
5.20	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 01 e 04. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . .	65
5.21	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 06 e 08. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	66
5.22	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 06 e 08. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . .	67
5.23	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para o ponto fiducial 07. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	68

5.24	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para o ponto fiducial 07. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.	69
5.25	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 09 e 10. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	70
5.26	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 09 e 10. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . .	71
5.27	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 00 e 05.	83
5.28	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 02 e 03.	84
5.29	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 01 e 04.	85
5.30	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 06 e 08.	86
5.31	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para o ponto fiducial 07.	87
5.32	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 09 e 10.	88
5.33	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA, SVM com <i>kernel</i> polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 00 e 05. .	91
5.34	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA, SVM com <i>kernel</i> polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 02 e 03. .	92
5.35	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA, SVM com <i>kernel</i> polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 01 e 04. .	93

5.36	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA, SVM com <i>kernel</i> polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 06 e 08.	94
5.37	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA, SVM com <i>kernel</i> polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para o ponto fiducial 07.	95
5.38	Bases de dados <i>BioID</i> e <i>Feret</i> : resultados para o método DF-B-PCA, SVM com <i>kernel</i> polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 09 e 10.	96
6.1	Representação das classes para $n = 4$	99
6.2	Detector por produto interno para uma única classe	103
6.3	Detector por produto interno para múltiplas classes	106
6.4	Exemplo para o branqueamento de um conjunto de dados	113
6.5	Exemplo para Interpretação por Filtro Casado	115
6.6	Projeto para S detectores por produto interno considerando-se S componentes principais	134
6.7	Exemplo para determinar os deslocamentos lineares $\phi_i^{(n,m)}$ utilizando-se a componente principal $\tilde{\phi}_i$	136
7.1	Fase de Aprendizagem para o método IPD-B	141
7.2	Fase de Aprendizagem para o método IPD-B-PCA	142
7.3	Diagrama em blocos para teste do método IPD-B	144
7.4	Diagrama em blocos para teste do método IPD-B-PCA	144
7.5	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 00 e 05. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	146
7.6	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 00 e 05. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.	147

7.7	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 02 e 03. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	148
7.8	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 02 e 03. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . .	149
7.9	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 01 e 04. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	150
7.10	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 01 e 04. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . .	151
7.11	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 06 e 08. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	152
7.12	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 06 e 08. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . .	153
7.13	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para o ponto fiducial 07. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	154
7.14	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para o ponto fiducial 07. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.	155
7.15	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 09 e 10. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.	156

7.16	Base de dados <i>BioID</i> e <i>Feret</i> : resultados ao variarmos o número de componentes principais para os pontos fiduciais 09 e 10. Cada coordenada indica as taxas de TP e FP obtidas por cada componente. . . .	157
7.17	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 00 e 05.	167
7.18	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 02 e 03.	168
7.19	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 01 e 04.	169
7.20	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 06 e 08.	170
7.21	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para o ponto fiducial 07. . . .	171
7.22	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 09 e 10.	172
7.23	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 00 e 05. . . .	175
7.24	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04. . . .	176
7.25	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04. . . .	177
7.26	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04. . . .	178
7.27	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04. . . .	179

7.28	Base de dados <i>BioID</i> e <i>Feret</i> : resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04.	180
A.1	Quatro exemplos de imagens da base de dados <i>BioID</i>	202
A.2	Quatro exemplos de imagens da base de dados <i>Feret</i>	203

Lista de Tabelas

2.1	Funções <i>kernels</i> tipicamente utilizadas	15
2.2	Síntese de técnicas para detecção de pontos fiduciais	17
5.1	Numerações, descrições e acrônimos dos pontos fiduciais das bases de dados <i>BioID</i> e <i>Feret</i>	43
5.2	Estratégias para obtermos somente um rótulo automático.	55
5.3	Bases de dados <i>BioID</i> e <i>Feret</i> : síntese dos melhores desempenhos, em termos da quantidade de componentes principais, para o método DF-B-PCA	74
5.4	Bases de dados <i>BioID</i> e <i>Feret</i> : comparação entre os métodos DF-B-PCA, DF, e DF-B	77
5.5	Bases de dados <i>BioID</i> e <i>Feret</i> : comparação entre os métodos DF-B-PCA, SVM-L e SVM-Pol	80
7.1	Base de dados <i>BioID</i> e <i>Feret</i> : síntese dos melhores desempenhos, em termos da quantidade de componentes principais, para o método IPD-PCA	159
7.2	Base de dados <i>BioID</i> : comparação entre os métodos IPD-B-PCA e IPD	161
7.3	Base de dados <i>BioID</i> : comparação entre os métodos IPD-B-PCA, DF-B-PCA, SVM-L e SVM-Pol	163
7.4	Base de dados <i>Feret</i> : comparação entre os métodos IPD-B-PCA, DF-B-PCA, SVM-L e SVM-Pol	164

Lista de Abreviaturas

AAM	<i>Active Appearance Models.</i> , p. 15
ASM	<i>Active Shape Model.</i> , p. 15
A	Técnica de pós-processamento para selecionar um rótulo automático como saída do método. O critério para seleção é a média de todos os rótulos automáticos., p. 54
AdaBoost	<i>Adaptative Boosting</i> , p. 8
CF	<i>Correlation Filters.</i> , p. 2
CPR	Reconhecimento de Padrões por Correlação., p. 1
DF-B-PCA	Filtragem Discriminativa utilizando Análise de Componentes Principais., p. 38
DFT	Transformada Discreta de <i>Fourier.</i> , p. 126
DF	Filtragem Discriminativa., p. 2, 18
EBGM	<i>Elastic Bunch Graph Matching</i> , p. 172
EXM	<i>Expansion Matching.</i> , p. 2
GML	Técnica de pós-processamento para selecionar um rótulo automático como saída do método. Neste agrupa-se os rótulos de P pixels e seleciona-se o rótulo mais provável., p. 54
IPD-B-PCA	Detector por Produto Interno utilizando Análise de Componentes Principais., p. 127
IPD-B	Detector por Produto Interno., p. 4, 92

KLT	Transformada <i>Karhunen-Loève.</i> , p. 33
LMMSE	<i>Linear Minimum Mean Square Error.</i> , p. 20
MACE	<i>Minimum Average Correlation Energy.</i> , p. 126
ML	Técnica de pós-processamento para selecionar um rótulo automático como saída do método. O critério para seleção é o rótulo mais provável., p. 54
NA	Técnica de pós-processamento que indica que todos os rótulos automáticos serão considerados., p. 54
OTF	<i>Optimal Tradeoff Filter.</i> , p. 126
PCA	Análise de Componentes Principais., p. 33
RBF	<i>Radial Basis Function</i> , p. 14
SVD	Decomposição em Valores Singulares., p. 34
SVM	<i>Support Vector Machines</i> , p. 3, 11, 50

Capítulo 1

Introdução

Reconhecimento de Padrões é uma área de grande abrangência, sendo potencialmente aplicável em diversos campos de pesquisa, como medicina, economia, processamento de sinais e imagens, entre outros. Dentre suas aplicações de grande relevância podemos citar: sistemas de detecção e reconhecimento de faces humanas [1–6]; detectores de objetos [7–9]; detectores de quinas [10–14] e inúmeras outras [15–18]. Embora diversas abordagens robustas e eficientes tenham sido desenvolvidas nos últimos anos, verificamos que a detecção e o reconhecimento de padrões é um problema em aberto.

As áreas de Reconhecimento de Padrões e Visão Computacional estão relacionadas. A Visão Computacional tem por objetivo a extração de informações em imagens ou vídeo de forma automática, para auxílio à tomada de decisão [19, 20]. Nessa, em geral, utilizam-se os conhecimentos de campos de pesquisa como processamento de sinais e imagens e inteligência artificial. Como exemplo desta ligação entre as áreas podemos citar a detecção e o reconhecimento de padrões que é um passo fundamental em sistemas de visão e acompanhamento, extração de características e sistemas de detecção e reconhecimento de objetos.

Recentemente, vem crescendo o interesse da comunidade científica na detecção/reconhecimento de padrões através de filtragem. Os métodos desenvolvidos, os quais deram origem à subárea de Reconhecimento de Padrões por Correlação (CPR), utilizam como medida de similaridade a correlação entre o padrão o qual desejamos detectar/reconhecer e um padrão candidato. Assim, a correlação é uma importante ferramenta para detecção/reconhecimento de um padrão. Representan-

tes desta subárea, utilizados para o reconhecimento de padrões unidimensionais e bidimensionais, são os métodos por Filtragem Discriminativa [14, 21–28].

O método por Filtragem Discriminativa (DF) foi inicialmente desenvolvido por J. Ben-Arie e K.R. Rao (também chamada de *Expansion Matching* - EXM) [10–13] e o seu objetivo concentrava-se no reconhecimento de padrões simples (quinas por exemplo). Idealmente, a proposta deste método é obter um filtro Θ que, ao ser convoluído com o padrão de interesse \mathbf{U} , resulte em um sinal de saída \mathbf{C} maximamente concentrado em apenas uma de suas amostras. Em [14], a Filtragem Discriminativa (DF) foi utilizada no reconhecimento de Letras do Alfabeto e os resultados apresentados foram satisfatórios. Porém, percebemos que a DF foi pouco explorada e pode ser potencialmente utilizada em outras aplicações.

Uma possibilidade, sendo uma das contribuições desta Tese, é projetar filtros discriminativos capazes de detectar um padrão e suas variações, como por exemplo, leves rotações e deformações. Logo, estas leves variações no padrão de interesse não afetariam de forma significativa o poder discriminatório dos filtros. Note que podemos incorporar robustez à Filtragem Discriminativa original. Para tal, diferentes filtros discriminativos são projetados para cada uma das componentes principais [29, 30]. Esta abordagem fornece um interessante modelo para obtermos a matriz de covariância utilizada no projeto dos filtros discriminativos. Nesta, podemos considerar que os padrões que não são de interesse irão se localizar em um subespaço ortogonal ao formado pela componente principal de interesse ϕ . Dessa forma, o ruído associado a ϕ será considerado como a projeção de padrões típicos no subespaço ortogonal à ϕ .

Conforme mencionamos, é crescente o interesse da comunidade científica por métodos que utilizam filtros para a detecção/reconhecimento de padrões. Um popular método denominado Filtros de Correlação (em inglês, *Correlation Filters* - CF), que utiliza filtros projetados através da Transformada de *Fourier* Discreta (DFT), tem sido empregado com sucesso para reconhecimento de faces humanas [31–35], detecção de íris [36] e biometria [37–39]. Em nossa investigação, concebemos a segunda contribuição desta Tese na forma de uma nova classe de detectores denominados Detectores por Produto Interno (IPD). Os detectores IPD podem ser projetados levando-se em consideração a importância atribuída para os padrões envolvidos (ou

seja, os de interesse e os que não são de interesse), proporcionando flexibilidade no projeto deste tipo de detector. Mostramos que o IPD é uma generalização da Filtragem Discriminativa e dos filtros de correlação obtidos através da DFT. A seguir, mostramos como projetar detectores IPD robustos. Isto é feito projetando-se detectores IPD para as componentes principais. Dessa forma, os padrões que não são de interesse têm interpretação semelhante à exposta no parágrafo anterior. A ortogonalidade das componentes principais simplifica o projeto dos detectores IPD.

Os desempenhos dos métodos propostos foram avaliados no contexto de um problema de detecção de pontos salientes em faces humanas (por exemplo a ponta do nariz) denominados por pontos fiduciais, que vêm sendo utilizados de forma crescente em sistemas de Visão Computacional. A detecção destes pontos pode ser usada em aplicações envolvendo faces humanas, tais como: reconhecimento [40,41], análise de expressões [42], estimação de pose [41,43], além de outras [44–46]. As simulações foram realizadas para onze pontos fiduciais de faces, utilizando-se validação cruzada, para duas bases de dados denominadas *BioID* e *Feret* com, respectivamente, 503 e 2004 imagens. Com o propósito de comparação empregamos classificadores *Support Vector Machines* (SVM) [47] que compreendem um dos métodos estado da arte em Reconhecimento de Padrões. A análise dos resultados demonstra que os métodos propostos são eficazes. Em vários casos, os resultados obtidos são comparáveis ou mesmo chegam a superar os resultados alcançados por classificadores SVM não lineares.

1.1 Organização da Tese

A Tese está organizada da seguinte maneira:

- No Capítulo 2, são apresentados os fundamentos teóricos utilizados na Tese. Primeiramente, verificamos os conceitos gerais e os métodos relacionados à área de Reconhecimento de Padrões. Ao final deste capítulo, revemos várias técnicas para detecção de pontos fiduciais.
- O Capítulo 3, apresenta a Teoria de Filtragem Discriminativa. Neste, verificamos os conceitos fundamentais da Filtragem Discriminativa unidimensional e bidimensional. A seguir, mostramos a formulação para detecção de múlti-

plos padrões. Por fim, realizamos algumas simulações didáticas dos métodos desenvolvidos no capítulo.

- No Capítulo 4, mostramos uma das contribuições desta Tese. O método proposto, denominado Filtragem Discriminativa utilizando Análise de Componentes Principais, é desenvolvido.
- No Capítulo 6, desenvolvemos uma nova classe de detectores denominados Detectores por Produto Interno (IPD). Neste capítulo, iremos demonstrar a relação existente entre os métodos baseados nos Filtros de Correlação e a Filtragem Discriminativa. Por fim, utilizamos a Análise de Componentes Principais para obter IPD robustos.
- Nos Capítulos 5 e 7, temos os procedimentos experimentais e as simulações realizadas para os métodos baseados em, respectivamente, Filtragem Discriminativa (DF) e Detectores por Produto Interno (IPD), utilizando-se validação cruzada para onze pontos fiduciais de duas bases de dados (*BioID* e *Feret*), com respectivamente, 503 e 2004 imagens.
- O Capítulo 8, apresenta as conclusões desta Tese bem como sugestões para trabalhos futuros.
- O Anexo A, apresenta alguns exemplos de imagens das bases de dados *BioID* e *Feret* utilizadas neste trabalho.
- No Anexo B, apresentamos as publicações obtidas até o término desta Tese.

Capítulo 2

Fundamentos Teóricos

Neste capítulo temos uma revisão dos conceitos e métodos utilizados nesta Tese. A organização é feita da seguinte forma. Primeiramente, mostramos uma revisão dos conceitos relacionados à área de Reconhecimento de Padrões. Finalizamos o capítulo apresentando técnicas para detecção de pontos fiduciais.

2.1 Reconhecimento de Padrões

Nesta seção, iremos apresentar um resumo dos conceitos e métodos da área de Reconhecimento de Padrões. Primeiramente, iremos abordar os conceitos e fundamentos da área. A seguir, apresentamos respectivamente, os tipos de aprendizagem, os sistemas de classificação e os tipos de classificadores.

2.1.1 Conceitos Iniciais

O termo *padrão* pode ser definido "como oposto de caos; uma entidade, vagamente definida, à qual pode ser dado um nome" [48,49]. Dessa forma, uma letra do alfabeto ou uma imagem do olho humano podem ser exemplos de padrões. Usamos o termo *vetor de características* para nos referirmos ao conjunto de variáveis¹ ordenadas utilizadas para representar o padrão, sendo usualmente suas medidas ou observações. O vocábulo *classe* refere-se ao conjunto de padrões com características similares, de acordo com algum critério². Por fim, supondo um padrão, temos que

¹As variáveis podem ser numéricas (contínuas ou discretas) e/ou simbólicas.

²Geralmente, uma classe possui uma denominação simbólica ou numérica chamada *rótulo* (em inglês, *label*).

a decisão por uma classe ou outra será realizada através de um *classificador*.

Matematicamente, podemos interpretar o problema de classificação utilizando uma notação por espaços vetoriais, como na Figura 2.1. Neste contexto, o classificador será um conjunto de funções f_i que mapeia o espaço de características \mathcal{X} , formado por padrões \mathbf{x}_j representados por vetores de características d -dimensionais, no espaço de classes \mathcal{Y} , formado por rótulos y_i unidimensionais. Assim, um classificador $f_i: \mathcal{X} \rightarrow \mathcal{Y}$, é tal que $f_i(\mathbf{x}_j) = y_{ij}$. Os espaços ∞ e \mathcal{P} definem, respectivamente: o meio físico que possui dimensionalidade infinita, e o espaço dos padrões com dimensionalidade p . A função \tilde{f} mapeia o espaço ∞ em \mathcal{P} e \hat{f} é conhecida na literatura pelo termo *redução de dimensionalidade* [29].

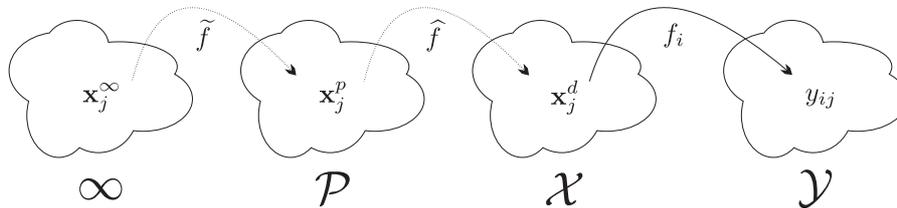


Figura 2.1: Na figura, temos a interpretação do problema de classificação de padrões em termos de espaços vetoriais. Os super-escritos no vetor \mathbf{x} indicam sua dimensão.

2.1.2 Classificação supervisionada, não-supervisionada e semi-supervisionada

Geralmente, o classificador utilizado nas técnicas de Reconhecimento de Padrões aprende como mapear o espaço de características \mathcal{X} , através do uso de um subconjunto de treinamento composto por N vetores \mathbf{x}_j que podem ou não indicar a qual classe \mathbf{x}_j pertence. Dessa forma, podemos distinguir as técnicas de classificação, quanto ao tipo de aprendizado, em três formas [48, 50], são elas: classificação supervisionada, classificação não-supervisionada, classificação semi-supervisionada.

Na classificação supervisionada, todos os padrões utilizados para treinamento pertencem a uma classe pré-definida. Dessa forma, os métodos de classificação supervisionada procuram mapear os padrões de treino em suas respectivas classes para tentar prever qual será a classe de um padrão de teste.

Os métodos de classificação não-supervisionada utilizam padrões que não possuem rótulos de classe e seu objetivo principal é obter agrupamentos (em inglês,

clustering) de padrões semelhantes segundo algum critério de similaridade.

Na classificação semi-supervisionada temos, para treinamento, tanto padrões com rótulos de classe como padrões que não possuem rótulos. Assim, o projeto do classificador utiliza tanto o agrupamento quando o mapeamento dos padrões de treinamento.

2.1.3 Aprendizado e Operação

Um típico sistema de classificação pode ser dividido em duas fases, são elas: fase de operação (teste) e fase de aprendizado (treino).

A fase de operação pode ser dividida em seis etapas. A primeira consiste na extração das informações relevantes do padrão, originando um vetor \mathbf{x}^p , p -dimensional. A seguir, podemos realizar algum tipo de pré-processamento em \mathbf{x}^p tais como, filtragem, remoção de ruído e/ou *background*. Em seguida, fazemos uma seleção do conjunto mínimo de variáveis do vetor \mathbf{x}^p que são mais relevantes, originando o vetor \mathbf{x}^d , d -dimensional, onde $d \leq p$. A seguir, o classificador, obtido na fase de aprendizado, é utilizado para categorizar \mathbf{x}^d em alguma classe y_i e por fim, podemos aproveitar a composição do problema para realizar algum pós-processamento.

A fase de aprendizado pode ser dividida em quatro etapas. Nesta fase, temos um conjunto de padrões \mathbf{x} que serão usados para treinamento do modelo de aprendizado. As etapas de pré-processamento e extração de características são idênticas às da fase de operação. Na terceira etapa, utilizamos os vetores \mathbf{x}^d para treinarmos algum modelo de classificação utilizando um algoritmo apropriado. O modelo será capaz de categorizar os vetores \mathbf{x}^d nas suas respectivas classes.

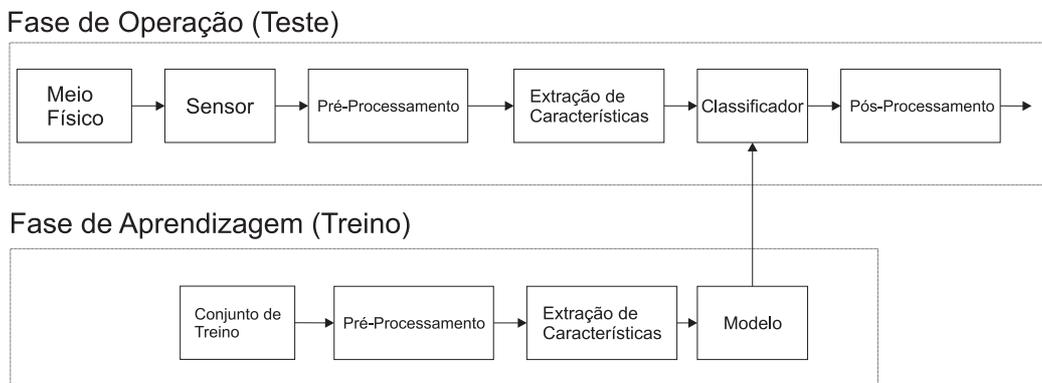


Figura 2.2: Diagrama em blocos de um sistema de classificação

2.1.4 Técnicas de Classificação

Nesta seção, apresentamos uma revisão de algumas técnicas de classificação que foram utilizadas nesta Tese. Nossa intenção será proporcionar um roteiro breve das técnicas bem como apresentar os algoritmos e suas referências.

2.1.4.1 AdaBoost

A técnica de aprendizado denominada por *boosting* utiliza o princípio de que podemos construir classificadores fortes a partir de uma combinação de classificadores fracos formando, assim, um comitê de classificadores. A idéia central do *boosting* é construir classificadores de forma sucessiva de tal modo que padrões classificados incorretamente sejam melhor classificados por classificadores sucessores utilizando um conjunto de treinamento repetidamente modificado. Esta alteração no conjunto de treinamento fundamenta-se em atribuir pesos aos padrões conforme uma distribuição de probabilidade. Os pesos serão reduzidos caso o padrão tenha sido classificado corretamente e elevados caso o padrão tenha sido classificado incorretamente. Isto permite que o algoritmo focalize em padrões classificados incorretamente. A técnica de *boosting* mais utilizada é o AdaBoost, proposto em [51]. Outras versões do AdaBoost, denominadas, *Real AdaBoost* e *Gentle AdaBoost* foram propostas [52]. A principal diferença está no fato de estas versões utilizarem uma distribuição de probabilidade diferente da distribuição do AdaBoost original.

O Algoritmo 1, referente à técnica AdaBoost, usa como entrada M pares (\mathbf{x}_m, y_m) , onde cada \mathbf{x}_m representa um vetor de características com rótulo y_m que assume os valores -1 e 1. Os pares (\mathbf{x}_m, y_m) estão associados a uma distribuição de probabilidade que será atualizada em cada iteração $t = \{1, \dots, T\}$ do algoritmo. A designação $D_t(i)$ será usada para indicar a probabilidade associada ao par (\mathbf{x}_i, y_i) na iteração t . Quando $t = 1$ temos uma distribuição de probabilidade uniforme. Em cada iteração t usa-se o classificador h_t para categorizar os vetores \mathbf{x}_m . O erro obtido pelo classificador h_t será:

$$e_t = \sum_{m|h_t(\mathbf{x}_m) \neq y_m} D_t(m). \quad (2.1)$$

Note que, na Equação (2.1), o erro é determinado considerando-se as probabi-

lidades dos vetores \mathbf{x}_m classificados incorretamente por $h_t(\mathbf{x}_m)$. A seguir, utilizamos o erro e_t para determinar o escalar α_t que será utilizado para atualizar a distribuição de probabilidades D_{t+1} . O termo z_t é usado para normalizar D_{t+1} .

O classificador final $H(\mathbf{x})$ irá ponderar os escalares α_t e os classificadores h_t obtidos para classificar um vetor \mathbf{x} qualquer. Lembre que $\text{sgn}(a) = 1$ para $a > 0$ e $\text{sgn}(a) = -1$ para $a < 0$.

Para a técnica *Gentle* AdaBoost podemos utilizar um algoritmo semelhante ao Algoritmo 1. A modificação será feita no cálculo dos escalares α_t , para tal considere dois escalares w_{+1} e w_{-1} calculados a partir da distribuição D_t da seguinte maneira:

$$\begin{aligned} w_{+1} &= \sum_{m|y_m h_t(\mathbf{x}_m)=+1} D_t(m), \\ w_{-1} &= \sum_{m|y_m h_t(\mathbf{x}_m)=-1} D_t(m). \end{aligned}$$

Assim, w_{+1} e w_{-1} são medidas de acerto e erro dos rótulos $+1$ e -1 . O valor de α_t será:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{w_{+1} - w_{-1}}{w_{-1} + w_{+1}} \right). \quad (2.2)$$

Por fim, nas implementações utilizamos a *toolbox* denominada GML AdaBoost Matlab [53], desenvolvida em ambiente Matlab. Verificamos, no conjunto de treinamento, que os desempenhos dos algoritmos que utilizaram esta *toolbox* foram estabilizados para um número de iterações (T) superior a 600.

Entrada: M pares (\mathbf{x}_m, y_m) .

Saída: Classificador \mathbf{H} formado por h_t e α_t .

início

Fazer $D_1(m) = 1/m$, onde $m = \{1, \dots, M\}$

para $t = 1, \dots, T$ **faça**

Obtenha o classificador h_t utilizando D_t

Calcule o erro e_t do classificador h_t . O erro será igual a:

$$e_t = \sum_{m|h_t(\mathbf{x}_m) \neq y_m} D_t(m).$$

Calcule o escalar α_t :

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - e_t}{e_t} \right).$$

Calcule D_{t+1} :

$$D_{t+1} = \frac{D_t(m)}{z_t} = \begin{cases} \exp(-\alpha_t), & \text{se } h_t(\mathbf{x}_m) = y_m. \\ \exp(\alpha_t), & \text{se } h_t(\mathbf{x}_m) \neq y_m. \end{cases}$$

i.e., $D_{t+1} = \frac{D_t(m) \exp(-\alpha_t y_m h_t(\mathbf{x}_m))}{z_t}$.

onde z_t é o termo de normalização de tal forma que D_{t+1} seja uma distribuição de probabilidade.

fim

Construa o classificador final $\mathbf{H}(\mathbf{x})$:

$$\mathbf{H}(\mathbf{x}) = \text{sgn} \left[\sum_{i=1}^T \alpha_i h_i(\mathbf{x}) \right].$$

onde sgn é a função sinal.

fim

Algoritmo 1: AdaBoost [51].

2.1.4.2 *Support Vector Machines (SVM)*

Nesta seção, apresentamos a técnica SVM. Nosso objetivo será mostrar os aspectos relevantes para utilizarmos este método em nossos experimentos.

A técnica de aprendizado SVM (em inglês, *Support Vector Machines*) desenvolvida em [47] está entre os métodos estado da arte em reconhecimento de padrões devido a seu excelente desempenho frente a problemas em diversos segmentos (várias referências podem ser encontradas [54, 55]).

Podemos dividir o emprego do método SVM em três contextos, supondo-se um problema à duas classes. No primeiro, denominado por SVM com margens rígidas, leva-se em consideração que as classes são linearmente separáveis. No segundo, modifica-se as SVMs com margens rígidas para lidar com dados com maior generalidade, ou seja, temos exemplares com separação aproximadamente linear. Este método denomina-se SVM com margens suaves. Por último, temos a técnica SVM não linear onde os classificadores obtidos possuem uma fronteira de decisão não linear. A seguir, apresentamos cada um dos métodos.

SVM com Margens Rígidas

O método SVM com margens rígidas pode ser obtido com segue. Suponha um conjunto de treinamento linearmente separável com M vetores \mathbf{x}_i e rótulos de classe y_i iguais a -1 e $+1$. O objetivo é encontrar o hiperplano f que separa as classes com máxima margem³. Vamos considerar que o hiperplano f é dado por:

$$f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle + b = \mathbf{w}^t \mathbf{x} + b, \quad (2.3)$$

onde $\langle \mathbf{x}, \mathbf{w} \rangle$ indica o produto interno entre os vetores \mathbf{x} e \mathbf{w} .

As regiões que delimitam as classes podem ser obtidas através da inequação [55]:

$$(\mathbf{w}^t \mathbf{x}_i + b)y_i - 1 \geq 0, \text{ onde } i = \{1, \dots, M\}. \quad (2.4)$$

Para a equação acima separar as classes, então \mathbf{w} deve ser tal que a margem entre os vetores suporte das classes seja igual a $\frac{2}{\|\mathbf{w}\|}$. A maximização da margem pode

³Margem consiste na distância entre os exemplares \mathbf{x}_i que estão mais próximos do hiperplano que separa as classes. Estes exemplares mais próximos são denominados por vetores suporte.

ser obtida através da minimização de $\|\mathbf{w}\|$ [55]. Assim, nosso problema de otimização será:

$$\begin{aligned} &\text{Minimizar} \quad \frac{1}{2}\|\mathbf{w}\|^2. \\ &\text{Restrito à} \quad (\mathbf{w}^t \mathbf{x}_i + b)y_i - 1 \geq 0, \text{ para } i = \{1, \dots, M\}. \end{aligned} \quad (2.5)$$

O problema de otimização definido através das Equações em (2.5) pode ser solucionado utilizando-se multiplicadores de Lagrange α_i 's da seguinte forma:

$$L = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^M \alpha_i [(\mathbf{w}^t \mathbf{x}_i + b)y_i - 1]. \quad (2.6)$$

A minimização da função L , exposta na Equação (2.6), nos permite modificar nosso problema de otimização para:

$$\begin{aligned} &\text{Minimizar} \quad \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^t \mathbf{x}_j). \\ &\text{Restrito à} \quad \alpha_i \geq 0, \text{ para } i = \{1, \dots, M\}, \\ &\quad \quad \quad \sum_{i=1}^M \alpha_i y_i = 0. \end{aligned} \quad (2.7)$$

O problema de otimização definido nas Equações em (2.7) é conhecido por forma dual. Uma das vantagens da forma dual é permitir que o problema de otimização seja representado em termos de produtos internos dos elementos de treinamento. As soluções para \mathbf{w} e b serão da forma:

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^M \alpha_i y_i \mathbf{x}_i, \\ b &= \frac{1}{n_S} \sum_{\mathbf{x}_j \in \mathcal{S}} \frac{1}{y_j} - \mathbf{w}^t \mathbf{x}_j. \end{aligned} \quad (2.8)$$

onde os α_i 's são constantes não negativas determinadas numericamente, os y_i 's são os rótulos de classe, n_S é o número de vetores suporte e \mathcal{S} representa o conjunto formado por vetores suporte. Os valores dos α_i 's são iguais a zero exceto para os casos em que eles pertencem a um dos hiperplanos de fronteira das classes (casos em que a Equação (2.4) é igual a zero) [56].

O classificador f será escrito como:

$$\begin{aligned}
 f(\mathbf{x}) &= \langle \mathbf{x}, \mathbf{w} \rangle + b, \\
 &= \mathbf{w}^t \mathbf{x} + b, \\
 &= \sum_{i=1}^M \alpha_i y_i \mathbf{x}_i^t \mathbf{x} + b.
 \end{aligned} \tag{2.9}$$

A função sinal pode ser utilizada para classificar um candidato \mathbf{x} . Assim, o classificador, na forma da função g , é escrito como:

$$\begin{aligned}
 g(\mathbf{x}) &= \text{sgn}(f(\mathbf{x})), \\
 &= \text{sgn} \left[\sum_{i=1}^M \alpha_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b \right], \\
 &= \text{sgn} \left[\sum_{i=1}^M \alpha_i y_i \mathbf{x}_i^t \mathbf{x} + b \right].
 \end{aligned} \tag{2.10}$$

SVM com Margens Suaves

Nós sabemos que, em muitos casos reais, as classes não são linearmente separáveis. Assim, podemos modificar as restrições do nosso problema de otimização de tal forma que se possa ter um compromisso entre erro de classificação e minimização de $\frac{1}{2} \|\mathbf{w}\|^2$. Dessa forma, a técnica por SVM com margens rígidas foi generalizada para lidar com classes mais reais.

Nosso problema de otimização modificado leva em consideração variáveis ξ_i 's que definem um relaxamento das restrições de otimização. Dessa forma, o problema de otimização é definido como:

$$\begin{aligned}
 \text{Minimizar} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \left(\sum_{i=1}^M \xi_i \right). \\
 \text{Restrito à} \quad & (\mathbf{w}^t \mathbf{x}_i + b) y_i - 1 + \xi_i \geq 0, \text{ para } i = \{1, \dots, M\}.
 \end{aligned} \tag{2.11}$$

O problema de otimização definido através das Equações em (2.11) terá como solução o mesmo formato obtido para o SVM com margens rígidas. Logo, a solução para \mathbf{w} será:

$$\mathbf{w} = \sum_{i=1}^M \alpha_i y_i \mathbf{x}_i, \tag{2.12}$$

onde novamente, os α_i 's são determinados numericamente. Então, o classificador f será escrito como:

$$f(\mathbf{x}) = \sum_{i=1}^M \alpha_i y_i \mathbf{x}_i^t \mathbf{x} + b. \quad (2.13)$$

Note que o parâmetro C é fundamental importância para sucesso de desempenho para o método SVM com margens suaves. Na literatura, é comum treinarmos o SVM com margens suaves variando-se este parâmetro em uma faixa que compreende os valores $2^0, 2^1, 2^2, \dots, 2^N$, do menor para o maior. Para nossos experimentos, consideramos $N = 7$.

SVM Não Linear

O método SVM não linear é aplicado nos casos em que as classes não podem ser separadas por um hiperplano. Neste caso, faz-se um mapeamento, utilizando uma função não linear Φ , dos exemplares de treinamento \mathbf{x}_i para um novo espaço de dimensão superior. O mapeamento Φ deve ser determinado de tal forma que os exemplares mapeados $\Phi(\mathbf{x}_i)$ possam ser separados por um classificador SVM linear. O problema de otimização para o método SVM não linear, utilizando margens suaves, é semelhante ao definido nas Equações em (2.11). A diferença é a restrição que leva em consideração o mapeamento Φ . Dessa forma, a restrição, considerando-se o problema de otimização dual, é igual a:

$$\text{Minimizar } \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j (\Phi^t(\mathbf{x}_i) \Phi(\mathbf{x}_j)). \quad (2.14)$$

A solução do problema de otimização acima é semelhante ao exposto na Equação (2.12). O classificador f será escrito como:

$$f(\mathbf{x}) = \sum_{i=1}^M \alpha_i y_i \Phi^t(\mathbf{x}_i) \Phi(\mathbf{x}) + b. \quad (2.15)$$

O termo $\Phi^t(\mathbf{x}_i) \Phi(\mathbf{x})$, expresso nas Equações (2.14) e (2.15), pode ser calculado através do uso de funções *kernel*. Uma função *kernel* é definida como:

$$k(\mathbf{b}_1, \mathbf{b}_2) = \Phi^t(\mathbf{b}_1) \Phi(\mathbf{b}_2). \quad (2.16)$$

As funções *kernel* tipicamente usadas são as polinomiais, gaussianas (ou RBF, do inglês, *Radial Basis Function*) e sigmoidais. Elas estão listadas na Tabela 2.1.

Tabela 2.1: Funções *kernels* tipicamente utilizadas.

Tipo de <i>kernel</i>	Função $k(\mathbf{b}_1, \mathbf{b}_2)$	Parâmetros
polinomial	$(a\mathbf{b}_1^t\mathbf{b}_2 + c)^d$	a, c e d
gaussiana	$\exp(-\sigma\ \mathbf{b}_1 - \mathbf{b}_2\ ^2)$	σ
sigmoidal	$\tanh(a\mathbf{b}_1^t\mathbf{b}_2 + c)$	a e c

Neste trabalho, utilizamos o *kernel* polinomial. Em nossos experimentos empregamos a *toolbox SVM-light* [57]. Quando aos parâmetros, variamos o grau do polinômio d entre 2 e 3 e o valor de C está entre 2^0 e 2^7 .

Na próxima seção, apresentamos uma revisão de alguns métodos utilizados para detectar pontos salientes em faces humanas.

2.2 Técnicas para Detecção de Pontos Fiduciais

Os métodos desenvolvidos neste trabalho foram validados no contexto de um problema de detecção de pontos fiduciais em faces humanas⁴. Podemos citar que estes pontos se caracterizam por determinar pontos salientes como os cantos da boca e ponta do nariz. A detecção de pontos fiduciais pode ser utilizada em diversas aplicações envolvendo faces humanas, tais como: reconhecimento [40,41], análise de expressões [42], estimação de pose [41,43], modelagem em 3D [44] e *tracking* [45,46].

As técnicas para detecção de pontos fiduciais podem ser divididas em duas categorias principais denominadas por métodos globais e locais. Os métodos globais utilizam *templates* deformáveis. A maioria destes métodos [58–60] ou utilizam o algoritmo base do método ASM [61] (em inglês, *Active Shape Model*) ou o do AAM [62] (em inglês, *Active Appearance Models*). Esta categoria tem a característica de proporcionar uma detecção simultânea de diversos pontos fiduciais. Nos métodos locais, os algoritmos detectam o ponto fiducial, como a ponta do nariz, sem utilizar informações de outras partes da face. Estes métodos são capazes de detectar os pontos fiduciais rapidamente. Exemplos de métodos locais podem ser encontrados em [41, 54, 63–65].

⁴Pode-se encontrar outras denominações, em *inglês*, tais como *facial landmarks* e *facial feature points*.

Na Tabela 2.2, apresentamos doze técnicas para detecção de pontos fiduciais em faces humanas, suas categorias e informações sobre disponibilidade ou não de código fonte dos algoritmos. Nesta tabela temos seis algoritmos globais, quatro locais e dois softwares proprietários. Note que em apenas três casos temos disponibilidade de código dos algoritmos, sendo que em um deles o código disponível apresentou erros de execução. Por fim, nas últimas duas linhas da tabela apresentamos implementações proprietárias para detecção de pontos fiduciais. Nestes casos, encontramos apenas um software de demonstração (demo) que pode ser utilizado para detectar pontos fiduciais em imagens fornecidas pelo usuário.

Tabela 2.2: Síntese de técnicas para detecção de pontos fiduciais. Nesta tabela, os métodos são agrupados nas categorias global e local. Além disso, informa-se na terceira coluna, se existe implementação disponível. As duas últimas linhas, indicam métodos proprietários.

#	Método	Categoria	Código Disponível?
1	<i>Active Shape Model</i> [61]	Global	Não
2	<i>Active Appearance Models</i> [62]	Global	Não
3	<i>Extended Active Shape Model</i> [58]	Global	Sim
4	<i>Facial Feature Detection and Tracking with Automatic Template Selection</i> [59]	Global	Não
5	<i>Automatic Feature Localization with Constrained Local Models</i> [60]	Global	Não
6	<i>Robust Pose Invariant Facial Feature Detection and Tracking in Real-Time</i> [66]	Global	Não
7	<i>An Efficient Method to Detect Facial Fiducial Points for Face Recognition</i> [64]	Local	Não
8	<i>Learning Person Specific Classifiers from Video</i> [41]	Local	Sim
9	<i>SVM based ASM for Facial Landmarks Location</i> [54]	Local	Não
10	<i>Fully Automatic Facial Feature Point Detection using Gabor Feature Based Boosted Classifiers</i> [65]	Local	Sim*
11	<i>Luxand</i> [67]	Proprietário	Sim (demo**)
12	<i>PittPatt</i> [68]	Proprietário	Não

* Não funciona corretamente.

** Software para demonstração.

Capítulo 3

Filtragem Discriminativa

A Teoria de Filtragem Discriminativa (DF) foi iniciada através do desenvolvimento dos classificadores denominados *Expansion Matching* (EXM) originalmente publicados em [10,11] (outras referências em [12,13,69,70]). Os classificadores EXM possuem uma formulação matemática para padrões unidimensionais¹ e seu principal objetivo consiste na detecção de quinas e junções. Recentemente em [14] (outras referências em [21–28]), temos contribuições importantes para a Teoria de Filtragem Discriminativa sendo que duas são de particular interesse para o nosso trabalho. A primeira consiste na abordagem da Filtragem Discriminativa como um problema da Teoria de Restauração do Impulso [71,72] e a segunda é a formulação da Filtragem Discriminativa para detecção de múltiplos padrões denominada Filtro OU.

Sendo assim, neste capítulo iremos: apresentar os conceitos da Teoria de Filtragem Discriminativa abordada como um problema da Teoria de Restauração do Impulso. A seguir, apresentaremos a formulação para detecção de múltiplos padrões denominada Filtro OU. Por fim, mostraremos algumas simulações para os métodos.

¹Os classificadores EXM podem ser utilizados para classificar padrões bidimensionais (imagem por exemplo), ordenando-os em ordem lexicográfica.

3.1 Método por Restauração do Impulso

Nesta seção vamos apresentar o método para detecção de objetos denominado Restauração do Impulso (em inglês, *Impulse Restoration*) [71,72]. A principal característica deste método reside no fato do seu detector (denominado estimador linear \mathbf{A}) restaurar um impulso na posição exata onde ocorre o padrão o qual desejamos detectar. A formulação deste método utiliza versões unidimensionais dos sinais envolvidos podendo ser aplicado a sinais bidimensionais através da concatenação de suas linhas transpostas.

Para apresentar o problema da Restauração do Impulso vamos considerar uma seqüência $g(k)$ com dimensões $N \times 1$ de tal forma que um padrão conhecido $f(k)$ encontra-se em uma posição desconhecida k_0 de $g(k)$. O objetivo do método será encontrar a localização k_0 de $f(k)$ na seqüência $g(k)$. Assim, utilizando o método por Restauração do Impulso iremos obter, após o processamento de $g(k)$, um impulso exatamente onde o padrão ocorrer. Para iniciar a formulação da Teoria por Restauração do Impulso vamos considerar que $g(k)$ pode ser representada da seguinte forma:

$$g(k) = f(k - k_0) + b(k), \quad (3.1)$$

onde $f(k - k_0)$ representa o padrão conhecido $f(k)$ localizado na posição desconhecida k_0 e $b(k)$ representa o restante da seqüência, que pode ser interpretado como ruído aditivo.

Conforme [30,73], podemos relacionar um impulso $\delta(k)$, um sinal $x(k)$ e uma posição (k_0) da seguinte maneira:

$$x(k - k_0) = x(k) * \delta(k - k_0), \quad (3.2)$$

onde $*$ representa a convolução discreta. Substituindo a Equação (3.2) na Equação (3.1) encontramos:

$$g(k) = f(k) * \delta(k - k_0) + b(k). \quad (3.3)$$

Podemos representar a Equação (3.3) na forma matricial como:

$$\mathbf{g} = \mathbf{F}\boldsymbol{\delta} + \mathbf{b}, \quad (3.4)$$

onde os vetores coluna \mathbf{g} , $\boldsymbol{\delta}$ e \mathbf{b} , com dimensões $N \times 1$, representam respectivamente, $g(k)$, $\delta(k)$ e $b(k)$. A matriz \mathbf{F} , com dimensões $N \times N$, é formada por deslocamentos circulares de $f(k)$ de tal forma que o termo $\mathbf{F}\boldsymbol{\delta}$ seja igual a convolução circular dada por $f(k) * \delta(k - k_0)$, dessa forma \mathbf{F} será igual a:

$$\mathbf{F} = \begin{bmatrix} f(0) & f(N-1) & f(N-2) & \dots & f(1) \\ f(1) & f(0) & f(N-1) & \dots & f(2) \\ f(2) & f(1) & f(0) & \dots & f(3) \\ \dots & \dots & \dots & \dots & \dots \\ f(N-1) & f(N-2) & f(N-3) & \dots & f(0) \end{bmatrix}_{N \times N} \quad (3.5)$$

Finalmente, utilizando a Equação (3.4) como base, podemos apresentar o problema da Restauração do Impulso como segue:

Supondo que seja fornecido o vetor \mathbf{g} , a matriz \mathbf{F} e a estatística do ruído \mathbf{b} (a priori desconhecido) como pode-se realizar a estimativa do vetor $\boldsymbol{\delta}$?

Para encontrarmos uma solução para o problema da restauração do impulso podemos começar supondo que o vetor representativo do ruído \mathbf{b} seja Gaussiano, tenha média igual a zero e matriz de covariância igual a $\frac{1}{N}\mathbf{C}_b$ com dimensões $N \times N$, ou seja:

$$\boldsymbol{\mu} = E[\mathbf{b}] = 0, \quad (3.6)$$

$$\begin{aligned} \boldsymbol{\Sigma} &= E[(\mathbf{b} - \boldsymbol{\mu})(\mathbf{b} - \boldsymbol{\mu})^t] \\ \boldsymbol{\Sigma} &= E[\mathbf{b}\mathbf{b}^t] = \frac{1}{N}\mathbf{C}_b. \end{aligned} \quad (3.7)$$

A probabilidade de um impulso estar em uma posição qualquer do vetor $\boldsymbol{\delta}$ será a mesma. Dessa forma, a sua matriz de autocorrelação (por definição em [29, 73]) será igual a:

$$\mathbf{R} = E[\boldsymbol{\delta}\boldsymbol{\delta}^t] = \frac{1}{N}\mathbf{I}, \quad (3.8)$$

onde \mathbf{I} é a matriz identidade com dimensões $N \times N$.

Assim, podemos estimar o vetor $\boldsymbol{\delta}$, denominado por $\widehat{\boldsymbol{\delta}}$, utilizando a formulação LMMSE [71, 72] (*Linear Minimum Mean Square Error*). Nesta, a melhor

aproximação linear em relação ao vetor \mathbf{g} , dada por $\widehat{\boldsymbol{\delta}} = \mathbf{A}\mathbf{g}$, que minimiza o erro médio quadrático igual à $E \left[\|\boldsymbol{\delta} - \widehat{\boldsymbol{\delta}}\|^2 \right]$, é a que faz o erro tornar-se decorrelacionado de \mathbf{g} , ou seja:

$$E \left[\left(\boldsymbol{\delta} - \widehat{\boldsymbol{\delta}} \right) \mathbf{g}^t \right] = 0. \quad (3.9)$$

A Equação (3.9) pode ser manipulada da seguinte forma:

$$\begin{aligned} E \left[\left(\boldsymbol{\delta} - \widehat{\boldsymbol{\delta}} \right) \mathbf{g}^t \right] &= \\ &= E \left[\left(\boldsymbol{\delta} - \mathbf{A}\mathbf{g} \right) \mathbf{g}^t \right] \\ &= E \left[\left(\boldsymbol{\delta} - \mathbf{A} \left(\mathbf{F}\boldsymbol{\delta} + \mathbf{b} \right) \right) \left(\mathbf{F}\boldsymbol{\delta} + \mathbf{b} \right)^t \right] \\ &= E \left[\left(\boldsymbol{\delta} - \mathbf{A}\mathbf{F}\boldsymbol{\delta} - \mathbf{A}\mathbf{b} \right) \left(\boldsymbol{\delta}^t \mathbf{F}^t + \mathbf{b}^t \right) \right] \\ &= E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \mathbf{F}^t + \boldsymbol{\delta} \mathbf{b}^t - \mathbf{A}\mathbf{F} \boldsymbol{\delta} \boldsymbol{\delta}^t \mathbf{F}^t - \mathbf{A}\mathbf{F} \boldsymbol{\delta} \mathbf{b}^t - \mathbf{A}\mathbf{b} \boldsymbol{\delta}^t \mathbf{F}^t - \mathbf{A}\mathbf{b} \mathbf{b}^t \right] \\ &= E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] \mathbf{F}^t + E \left[\boldsymbol{\delta} \mathbf{b}^t \right] - \mathbf{A}\mathbf{F} E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] \mathbf{F}^t - \mathbf{A}\mathbf{F} E \left[\boldsymbol{\delta} \mathbf{b}^t \right] \\ &\quad - \mathbf{A} E \left[\mathbf{b} \boldsymbol{\delta}^t \right] \mathbf{F}^t - \mathbf{A} E \left[\mathbf{b} \mathbf{b}^t \right]. \end{aligned} \quad (3.10)$$

Substituindo a Equação (3.7) na Equação (3.10) e considerando que $\boldsymbol{\delta}$ e \mathbf{b} são decorrelacionados, ou seja, $E \left[\boldsymbol{\delta} \mathbf{b}^t \right] = 0$, encontramos:

$$E \left[\left(\boldsymbol{\delta} - \widehat{\boldsymbol{\delta}} \right) \mathbf{g}^t \right] = E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] \mathbf{F}^t - \mathbf{A}\mathbf{F} E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] \mathbf{F}^t - \frac{1}{N} \mathbf{A}\mathbf{C}_{\mathbf{b}}. \quad (3.11)$$

Substituindo a Equação (3.9) na Equação (3.11) encontramos:

$$\begin{aligned} \mathbf{F}^t - \mathbf{A}\mathbf{F}\mathbf{F}^t - \mathbf{A}\mathbf{C}_{\mathbf{b}} &= 0 \\ \mathbf{A} \left[\mathbf{F}\mathbf{F}^t + \mathbf{C}_{\mathbf{b}} \right] &= \mathbf{F}^t \\ \mathbf{A} &= \mathbf{F}^t \left[\mathbf{F}\mathbf{F}^t + \mathbf{C}_{\mathbf{b}} \right]^{-1}. \end{aligned} \quad (3.12)$$

Sendo assim, a melhor aproximação linear de \mathbf{g} será igual a:

$$\begin{aligned} \widehat{\boldsymbol{\delta}} &= \mathbf{A}\mathbf{g} \\ \widehat{\boldsymbol{\delta}} &= \mathbf{F}^t \left[\mathbf{F}\mathbf{F}^t + \mathbf{C}_{\mathbf{b}} \right]^{-1} \mathbf{g}. \end{aligned} \quad (3.13)$$

As Equações em (3.13) mostram como podemos determinar o estimador linear \mathbf{A} de tal forma que possamos restaurar um impulso na posição em que ocorre o

padrão o qual desejamos encontrar no vetor \mathbf{g} .

3.2 Restauração do Impulso Bidimensional: Aplicação em Imagens

A Restauração do Impulso pode ser aplicada em sinais bidimensionais através da filtragem com operadores por convolução circular [14,25,26]. Neste caso, podemos obter um filtro bidimensional $\Theta_{N \times N}$, denominado pelo termo *filtro discriminativo*, que irá discriminar um padrão desejado $\mathbf{U}_{N \times N}$. O filtro Θ maximiza a energia do sinal de saída \mathbf{C} , em sua coordenada central, quando a entrada for igual a \mathbf{U} (observe a Figura 3.1). Isto será equivalente ao filtro discriminativo Θ restaurar um impulso quando a entrada for igual a \mathbf{U} .

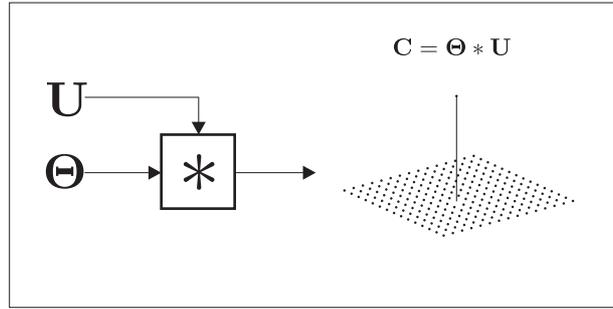


Figura 3.1: Sinal de saída \mathbf{C} obtido através da filtragem de Θ com o padrão \mathbf{U} o qual desejamos detectar. Observe que a filtragem, igual a convolução circular de Θ com \mathbf{U} , restaura um impulso na coordenada central de \mathbf{C} .

A métrica utilizada para avaliar se Θ restaura um impulso denomina-se DSNR_2 [25] sendo definida da seguinte forma:

$$(\text{DSNR}_2)_{(0,0)} = \frac{c(0,0)^2}{\sum_{i=1}^N \sum_{j=1}^N c(i,j)^2 - c(0,0)^2}, \quad (3.14)$$

onde $c(m,n)$ é a convolução circular entre o filtro Θ e o padrão \mathbf{U} . Quando o sinal filtrado se aproxima de um impulso a DSNR_2 se aproxima de infinito.

Para apresentar o problema da Restauração do Impulso bidimensional vamos considerar uma imagem $g(m,n)$ com dimensões $N \times N$ de tal forma que um padrão conhecido $u(m,n)$ encontra-se em uma posição desconhecida (m_0, n_0) de $g(m,n)$. O

objetivo será encontrar a localização (m_0, n_0) de $u(m, n)$ na imagem $g(m, n)$ (observe a Figura 3.2). Vamos considerar que a imagem $g(m, n)$ pode ser representada da seguinte forma:

$$g(m, n) = u(m - m_0, n - n_0) + b(m, n), \quad (3.15)$$

onde $u(m - m_0, n - n_0)$ representa o padrão conhecido $u(m, n)$ localizado na posição desconhecida (m_0, n_0) e $b(m, n)$ representa o restante da imagem, que pode ser interpretado como ruído aditivo.

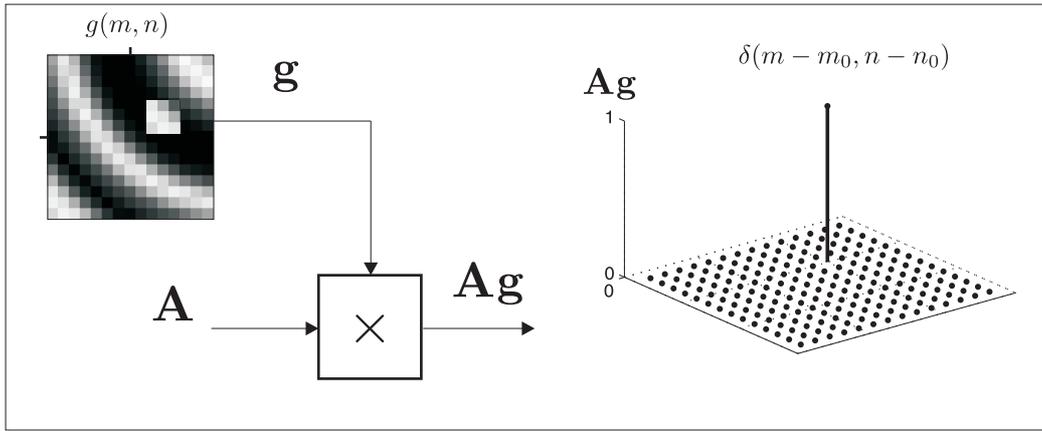


Figura 3.2: O detector para o padrão quadrado , com dimensões 3×3 , denominado estimador linear \mathbf{A} , obtido através do método por Restauração do Impulso, irá restaurar um impulso na posição m_0, n_0 (localização do padrão o qual desejamos encontrar) quando realizarmos o produto entre \mathbf{A} e \mathbf{g} (versão unidimensional da imagem $g(m, n)$).

Podemos expressar a Equação (3.15) em uma forma unidimensional usando ordenação lexicográfica [14]. Para tal, $g(m, n)$, $u(m - m_0, n - n_0)$ e $b(m, n)$ serão representados por $g(k)$, $u(k - k_0)$ e $b(k)$ obtidos através da concatenação de suas linhas transpostas. Dessa forma, a versão unidimensional da Equação (3.15) será:

$$g(k) = u(k - k_0) + b(k). \quad (3.16)$$

No caso bidimensional, considerando $N = 2T + 1$, temos que os vetores \mathbf{g} , δ

e \mathbf{b} (dimensões $N^2 \times 1$) da Equação (3.4), serão indexados da seguinte forma [14]:

$$\mathbf{g} = \begin{bmatrix} g(-T, -T) \\ g(-T, -T+1) \\ \vdots \\ g(-T, T) \\ \vdots \\ g(T, T) \end{bmatrix}, \quad \boldsymbol{\delta} = \begin{bmatrix} \delta(-T, -T) \\ \delta(-T, -T+1) \\ \vdots \\ \delta(-T, T) \\ \vdots \\ \delta(T, T) \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} b(-T, -T) \\ b(-T, -T+1) \\ \vdots \\ b(-T, T) \\ \vdots \\ b(T, T) \end{bmatrix}. \quad (3.17)$$

As matrizes \mathbf{F} e \mathbf{A} , com dimensões $N^2 \times N^2$, serão circulares por bloco [14].

Por exemplo, considerando que $\mathbf{U}_{3 \times 3}$ e $\boldsymbol{\Theta}_{3 \times 3}$ são iguais a:

$$\mathbf{U} = \begin{bmatrix} u(-1, -1) & u(-1, 0) & u(-1, 1) \\ u(0, -1) & u(0, 0) & u(0, 1) \\ u(1, -1) & u(1, 0) & u(1, 1) \end{bmatrix}, \quad \boldsymbol{\Theta} = \begin{bmatrix} \theta(-1, -1) & \theta(-1, 0) & \theta(-1, 1) \\ \theta(0, -1) & \theta(0, 0) & \theta(0, 1) \\ \theta(1, -1) & \theta(1, 0) & \theta(1, 1) \end{bmatrix},$$

então, a estrutura das matrizes $\mathbf{F}_{9 \times 9}$ e $\mathbf{A}_{9 \times 9}$ será:

$$\mathbf{F} = \begin{bmatrix} \begin{array}{ccc|ccc|cc} u(0,0) & u(0,-1) & u(0,1) & u(-1,0) & u(-1,-1) & u(-1,1) & u(1,0) & u(1,-1) & u(1,1) \\ u(0,1) & u(0,0) & u(0,-1) & u(-1,1) & u(-1,0) & u(-1,-1) & u(1,1) & u(1,0) & u(1,-1) \\ u(0,-1) & u(0,1) & u(0,0) & u(-1,-1) & u(-1,1) & u(-1,0) & u(1,-1) & u(1,1) & u(1,0) \end{array} \\ \begin{array}{ccc|ccc|cc} u(1,0) & u(1,-1) & u(1,1) & u(0,0) & u(0,-1) & u(0,1) & u(-1,0) & u(-1,-1) & u(-1,1) \\ u(1,1) & u(1,0) & u(1,-1) & u(0,1) & u(0,0) & u(0,-1) & u(-1,1) & u(-1,0) & u(-1,-1) \\ u(1,-1) & u(1,1) & u(1,0) & u(0,-1) & u(0,1) & u(0,0) & u(-1,-1) & u(-1,1) & u(-1,0) \end{array} \\ \begin{array}{ccc|ccc|cc} u(-1,0) & u(-1,-1) & u(-1,1) & u(1,0) & u(1,-1) & u(1,1) & u(0,0) & u(0,-1) & u(0,1) \\ u(-1,1) & u(-1,0) & u(-1,-1) & u(1,1) & u(1,0) & u(1,-1) & u(0,1) & u(0,0) & u(0,-1) \\ u(-1,-1) & u(-1,1) & u(-1,0) & u(1,-1) & u(1,1) & u(1,0) & u(0,-1) & u(0,1) & u(0,0) \end{array} \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} \begin{array}{ccc|ccc|cc} \theta(0,0) & \theta(0,-1) & \theta(0,1) & \theta(-1,0) & \theta(-1,-1) & \theta(-1,1) & \theta(1,0) & \theta(1,-1) & \theta(1,1) \\ \theta(0,1) & \theta(0,0) & \theta(0,-1) & \theta(-1,1) & \theta(-1,0) & \theta(-1,-1) & \theta(1,1) & \theta(1,0) & \theta(1,-1) \\ \theta(0,-1) & \theta(0,1) & \theta(0,0) & \theta(-1,-1) & \theta(-1,1) & \theta(-1,0) & \theta(1,-1) & \theta(1,1) & \theta(1,0) \end{array} \\ \begin{array}{ccc|ccc|cc} \theta(1,0) & \theta(1,-1) & \theta(1,1) & \theta(0,0) & \theta(0,-1) & \theta(0,1) & \theta(-1,0) & \theta(-1,-1) & \theta(-1,1) \\ \theta(1,1) & \theta(1,0) & \theta(1,-1) & \theta(0,1) & \theta(0,0) & \theta(0,-1) & \theta(-1,1) & \theta(-1,0) & \theta(-1,-1) \\ \theta(1,-1) & \theta(1,1) & \theta(1,0) & \theta(0,-1) & \theta(0,1) & \theta(0,0) & \theta(-1,-1) & \theta(-1,1) & \theta(-1,0) \end{array} \\ \begin{array}{ccc|ccc|cc} \theta(-1,0) & \theta(-1,-1) & \theta(-1,1) & \theta(1,0) & \theta(1,-1) & \theta(1,1) & \theta(0,0) & \theta(0,-1) & \theta(0,1) \\ \theta(-1,1) & \theta(-1,0) & \theta(-1,-1) & \theta(1,1) & \theta(1,0) & \theta(1,-1) & \theta(0,1) & \theta(0,0) & \theta(0,-1) \\ \theta(-1,-1) & \theta(-1,1) & \theta(-1,0) & \theta(1,-1) & \theta(1,1) & \theta(1,0) & \theta(0,-1) & \theta(0,1) & \theta(0,0) \end{array} \end{bmatrix}.$$

Para representar as matrizes \mathbf{F} e \mathbf{A} em uma notação por blocos [14] podemos utilizar o operador $H_r(\mathbf{V})$ que transforma uma linha r de uma matriz $\mathbf{V}_{N \times N} = \{v(m, n)\}$, em uma matriz circulante com dimensões $N \times N$.

$$\mathbf{H}_r(\mathbf{V}) = \begin{bmatrix} v(r,0) & v(r,-1) & v(r,-2) & \dots & v(r,2) & v(r,1) \\ v(r,1) & v(r,0) & v(r,-1) & \dots & v(r,3) & v(r,2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ v(r,N) & v(r,N-1) & v(r,N-2) & \dots & v(r,1-N) & v(r,-N) \\ v(r,-N) & v(r,N) & v(r,N-1) & \dots & v(r,2-N) & v(r,1-N) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ v(r,-1) & v(r,-2) & v(r,-3) & \dots & v(r,1) & v(r,0) \end{bmatrix}_{N \times N}, \quad (3.18)$$

Dessa forma, utilizando o operador definido na Equação (3.18), temos que as matrizes \mathbf{F} e \mathbf{A} serão iguais a:

$$\mathbf{F} = \begin{bmatrix} H_0(\mathbf{U}) & H_{-1}(\mathbf{U}) & \dots & H_{-T}(\mathbf{U}) & H_T(\mathbf{U}) & \dots & H_1(\mathbf{U}) \\ H_1(\mathbf{U}) & H_0(\mathbf{U}) & \dots & H_{-T+1}(\mathbf{U}) & H_{-T}(\mathbf{U}) & \dots & H_2(\mathbf{U}) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_T(\mathbf{U}) & H_{T-1}(\mathbf{U}) & \dots & H_0(\mathbf{U}) & H_{-1}(\mathbf{U}) & \dots & H_{-T}(\mathbf{U}) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_{-1}(\mathbf{U}) & H_{-2}(\mathbf{U}) & \dots & H_T(\mathbf{U}) & H_{T-1}(\mathbf{U}) & \dots & H_0(\mathbf{U}) \end{bmatrix}_{N^2 \times N^2} \quad (3.19)$$

$$\mathbf{A} = \begin{bmatrix} H_0(\Theta) & H_{-1}(\Theta) & \dots & H_{-T}(\Theta) & H_T(\Theta) & \dots & H_1(\Theta) \\ H_1(\Theta) & H_0(\Theta) & \dots & H_{-T+1}(\Theta) & H_{-T}(\Theta) & \dots & H_2(\Theta) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_T(\Theta) & H_{T-1}(\Theta) & \dots & H_0(\Theta) & H_{-1}(\Theta) & \dots & H_{-T}(\Theta) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_{-1}(\Theta) & H_{-2}(\Theta) & \dots & H_T(\Theta) & H_{T-1}(\Theta) & \dots & H_0(\Theta) \end{bmatrix}_{N^2 \times N^2} \quad (3.20)$$

Por fim, lembramos que \mathbf{A} é obtida através da Equação (3.13) e que o filtro discriminativo bidimensional Θ pode ser obtido inspecionando-se a Equação (3.20).

3.3 Filtro OU

Até o momento, apresentamos um método que realiza a discriminação de um único padrão. Nesta seção, mostraremos como projetar um filtro bidimensional, denominado Filtro OU, capaz de realizar a detecção de múltiplos padrões, baseando-se na Filtragem Discriminativa por Restauração do Impulso. Dessa forma, supondo um conjunto de m padrões, teremos que o filtro bidimensional será capaz de reconhecer um grupo de padrões pertencente ao conjunto original. A seguir, veremos como este filtro pode ser obtido. Antes de apresentarmos a solução para este filtro vamos supor:

- Desejamos fazer a detecção de m padrões.
- Um vetor \mathbf{g} formado pela concatenação das linhas transpostas de uma imagem.
- \mathbf{F}_i matrizes associadas a cada padrão $i = \{1, 2, \dots, m\}$ (conforme definição da Equação (3.19)).
- Um vetor \mathbf{b} que representa o ruído aditivo cuja matriz de covariância é igual a $\Sigma = E[\mathbf{b}\mathbf{b}^t] = \frac{1}{N}\mathbf{C}_b$, onde \mathbf{C}_b possui dimensões $N^2 \times N^2$.
- Um vetor $\boldsymbol{\delta}$, que representa o impulso, cuja matriz de autocorrelação é igual a $\mathbf{R} = E[\boldsymbol{\delta}\boldsymbol{\delta}^t] = \frac{1}{N}\mathbf{I}$, onde \mathbf{I} é uma matriz identidade com dimensões $N \times N$.

Assim, nosso objetivo será encontrar a melhor estimativa para $\boldsymbol{\delta}$, denominada por $\hat{\boldsymbol{\delta}}$, que minimiza o erro médio quadrático, que é igual a $E[\|\boldsymbol{\delta} - \hat{\boldsymbol{\delta}}\|^2]$. A solução para tal, como apresentado na Seção 3.1, será fazer o erro tornar-se decorrelacionado de \mathbf{g} , como mostrado na Equação (3.9) apresentada abaixo por comodidade.

$$E\left[\left(\boldsymbol{\delta} - \hat{\boldsymbol{\delta}}\right)\mathbf{g}^t\right] = 0. \quad (3.21)$$

Supondo que cada padrão i possui uma probabilidade p_i de ocorrência então, podemos reescrever a Equação (3.21) como:

$$\begin{aligned}
E \left[\left(\boldsymbol{\delta} - \widehat{\boldsymbol{\delta}} \right) \mathbf{g}^t \right] &= E \left[\left(\boldsymbol{\delta} - \mathbf{A} \mathbf{g} \right) \mathbf{g}^t \right] \\
&= \sum_{i=1}^m p_i \left\{ E \left[\left(\boldsymbol{\delta} - \mathbf{A} \left(\mathbf{F}_i \boldsymbol{\delta} + \mathbf{b} \right) \right) \left(\mathbf{F}_i \boldsymbol{\delta} + \mathbf{b} \right)^t \right] \right\} \\
&= \sum_{i=1}^m p_i \left\{ E \left[\left(\boldsymbol{\delta} - \mathbf{A} \mathbf{F}_i \boldsymbol{\delta} - \mathbf{A} \mathbf{b} \right) \left(\boldsymbol{\delta}^t \mathbf{F}_i^t + \mathbf{b}^t \right) \right] \right\} \\
&= \sum_{i=1}^m p_i E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \mathbf{F}_i^t + \boldsymbol{\delta} \mathbf{b}^t - \mathbf{A} \mathbf{F}_i \boldsymbol{\delta} \boldsymbol{\delta}^t \mathbf{F}_i^t - \mathbf{A} \mathbf{F}_i \boldsymbol{\delta} \mathbf{b}^t - \mathbf{A} \mathbf{b} \boldsymbol{\delta}^t \mathbf{F}_i^t - \mathbf{A} \mathbf{b} \mathbf{b}^t \right] \\
&= \sum_{i=1}^m p_i \left\{ E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] \mathbf{F}_i^t + E \left[\boldsymbol{\delta} \mathbf{b}^t \right] - \mathbf{A} \mathbf{F}_i E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] \mathbf{F}_i^t - \right. \\
&\quad \left. \mathbf{A} \mathbf{F}_i E \left[\boldsymbol{\delta} \mathbf{b}^t \right] - \mathbf{A} E \left[\mathbf{b} \boldsymbol{\delta}^t \right] \mathbf{F}_i^t - \mathbf{A} E \left[\mathbf{b} \mathbf{b}^t \right] \right\}. \tag{3.22}
\end{aligned}$$

Usando as considerações, postas por conveniência abaixo, para a matriz de covariância de \mathbf{b} , para a matriz de autocorrelação de $\boldsymbol{\delta}$ e a hipótese de $\boldsymbol{\delta}$ e \mathbf{b} serem descorrelacionados, ou seja:

$$\begin{aligned}
E \left[\boldsymbol{\delta} \mathbf{b}^t \right] &= 0 \\
\boldsymbol{\Sigma} &= E \left[\mathbf{b} \mathbf{b}^t \right] = \frac{1}{N} \mathbf{C}_b \\
\mathbf{R} &= E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] = \frac{1}{N} \mathbf{I},
\end{aligned}$$

dessa forma:

$$\begin{aligned}
\sum_{i=1}^m p_i \left\{ E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] \mathbf{F}_i^t - \mathbf{A} \mathbf{F}_i E \left[\boldsymbol{\delta} \boldsymbol{\delta}^t \right] \mathbf{F}_i^t - \frac{1}{N} \mathbf{A} \mathbf{C}_b \right\} &= E \left[\left(\boldsymbol{\delta} - \widehat{\boldsymbol{\delta}} \right) \mathbf{g}^t \right] = 0 \\
\sum_{i=1}^m p_i \left\{ \mathbf{F}_i^t - \mathbf{A} \mathbf{F}_i \mathbf{F}_i^t - \mathbf{A} \mathbf{C}_b \right\} &= 0 \\
\sum_{i=1}^m p_i \left\{ \mathbf{A} \left[\mathbf{F}_i \mathbf{F}_i^t + \mathbf{C}_b \right] \right\} &= \sum_{i=1}^m p_i \mathbf{F}_i^t \\
\mathbf{A} &= \sum_{i=1}^m \left\{ p_i \mathbf{F}_i^t \right\} \left\{ \mathbf{C}_b + \sum_{i=1}^m p_i \left[\mathbf{F}_i \mathbf{F}_i^t \right] \right\}^{-1}. \tag{3.23}
\end{aligned}$$

Sendo assim, a melhor aproximação linear de \mathbf{g} será igual a:

$$\begin{aligned}\hat{\boldsymbol{\delta}} &= \mathbf{A}\mathbf{g} \\ \hat{\boldsymbol{\delta}} &= \sum_{i=1}^m \{p_i \mathbf{F}_i^t\} \left\{ \mathbf{C}_b + \sum_{i=1}^m p_i [\mathbf{F}_i \mathbf{F}_i^t] \right\}^{-1} \mathbf{g}.\end{aligned}\quad (3.24)$$

Portanto, a Equação (3.24) representa o Filtro OU sendo utilizada, conforme comentamos no início desta seção, para detectar um grupo de m padrões pertencentes a um conjunto original.

3.4 Simulações

Nesta seção, apresentaremos três exemplos didáticos de projetos de filtros discriminativos organizados da seguinte maneira: no primeiro exemplo, utilizamos o desenvolvimento realizado na Seção 3.2 para obtermos um filtro que detecta um padrão do tipo quina. No segundo, ainda utilizando o desenvolvimento da Seção 3.2, projetamos um filtro para detectar da letra “T” (fonte *Times*), ao longo de uma imagem de teste. Por último, utilizamos o desenvolvimento feito na Seção 3.3 para projetamos um filtro que detecta as letras “T” (fonte *Times*) e “I” (fonte *Currier*) em uma imagem de teste.

Exemplo: Detecção de uma quina

Vamos considerar que desejamos detectar um padrão \mathbf{U} igual a:

$$\mathbf{U} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{3 \times 3} . \quad (3.25)$$

A matriz em (3.25) define um padrão quina com dimensões 3×3 . Perceba que \mathbf{U} possui valores iguais a um somente na terceira coluna e na última linha. Nosso objetivo, neste exemplo, consiste em determinar o filtro discriminativo bidimensional

Θ de tal forma que:

$$\mathbf{U} * \Theta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.26)$$

Assim, segundo a Equação (3.26), o filtro discriminativo Θ deverá restaurar um impulso bidimensional quando convoluído circularmente com \mathbf{U} .

Para encontrarmos Θ devemos determinar, primeiramente, a matriz \mathbf{F} conforme a Equação (3.19). Por comodidade, vamos reescrever sua estrutura:

$$\mathbf{F} = \begin{bmatrix} \begin{array}{ccc|ccc|ccc} u(0,0) & u(0,-1) & u(0,1) & u(-1,0) & u(-1,-1) & u(-1,1) & u(1,0) & u(1,-1) & u(1,1) \\ u(0,1) & u(0,0) & u(0,-1) & u(-1,1) & u(-1,0) & u(-1,-1) & u(1,1) & u(1,0) & u(1,-1) \\ u(0,-1) & u(0,1) & u(0,0) & u(-1,-1) & u(-1,1) & u(-1,0) & u(1,-1) & u(1,1) & u(1,0) \\ \hline u(1,0) & u(1,-1) & u(1,1) & u(0,0) & u(0,-1) & u(0,1) & u(-1,0) & u(-1,-1) & u(-1,1) \\ u(1,1) & u(1,0) & u(1,-1) & u(0,1) & u(0,0) & u(0,-1) & u(-1,1) & u(-1,0) & u(-1,-1) \\ u(1,-1) & u(1,1) & u(1,0) & u(0,-1) & u(0,1) & u(0,0) & u(-1,-1) & u(-1,1) & u(-1,0) \\ \hline u(-1,0) & u(-1,-1) & u(-1,1) & u(1,0) & u(1,-1) & u(1,1) & u(0,0) & u(0,-1) & u(0,1) \\ u(-1,1) & u(-1,0) & u(-1,-1) & u(1,1) & u(1,0) & u(1,-1) & u(0,1) & u(0,0) & u(0,-1) \\ u(-1,-1) & u(-1,1) & u(-1,0) & u(1,-1) & u(1,1) & u(1,0) & u(0,-1) & u(0,1) & u(0,0) \end{array} \end{bmatrix}.$$

Dessa forma, a matriz \mathbf{F} será igual a:

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}_{9 \times 9}. \quad (3.27)$$

Utilizando a Equação (3.13) (repetida abaixo por comodidade) com \mathbf{C}_b igual a identidade, obtemos a matriz \mathbf{A} .

$$\mathbf{A} = \mathbf{F}^t [\mathbf{F}\mathbf{F}^t + \mathbf{C}_b]^{-1}.$$

Dessa forma, a matriz \mathbf{A} será igual a:

$$\mathbf{A} = \begin{bmatrix} -0,2 & +0,3 & -0,2 & +0,3 & -0,2 & +0,3 & -0,2 & +0,3 & -0,2 \\ -0,2 & -0,2 & +0,3 & +0,3 & +0,3 & -0,2 & -0,2 & -0,2 & +0,3 \\ +0,3 & -0,2 & -0,2 & -0,2 & +0,3 & +0,3 & +0,3 & -0,2 & -0,2 \\ -0,2 & +0,3 & -0,2 & -0,2 & +0,3 & -0,2 & +0,3 & -0,2 & +0,3 \\ -0,2 & -0,2 & +0,3 & -0,2 & -0,2 & +0,3 & +0,3 & +0,3 & -0,2 \\ +0,3 & -0,2 & -0,2 & +0,3 & -0,2 & -0,2 & -0,2 & +0,3 & +0,3 \\ +0,3 & -0,2 & +0,3 & -0,2 & +0,3 & -0,2 & -0,2 & +0,3 & -0,2 \\ +0,3 & +0,3 & -0,2 & -0,2 & -0,2 & +0,3 & -0,2 & -0,2 & +0,3 \\ -0,2 & +0,3 & +0,3 & +0,3 & -0,2 & -0,2 & +0,3 & -0,2 & -0,2 \end{bmatrix}_{9 \times 9}. \quad (3.28)$$

Utilizando a Equação (3.20) determinamos o filtro discriminativo bidimensional Θ por inspeção. Por comodidade, reescrevemos a estrutura da matriz \mathbf{A} :

$$\mathbf{A} = \begin{bmatrix} \theta(0,0) & \theta(0,-1) & \theta(0,1) & | & \theta(-1,0) & \theta(-1,-1) & \theta(-1,1) & | & \theta(1,0) & \theta(1,-1) & \theta(1,1) \\ \theta(0,1) & \theta(0,0) & \theta(0,-1) & | & \theta(-1,1) & \theta(-1,0) & \theta(-1,-1) & | & \theta(1,1) & \theta(1,0) & \theta(1,-1) \\ \theta(0,-1) & \theta(0,1) & \theta(0,0) & | & \theta(-1,-1) & \theta(-1,1) & \theta(-1,0) & | & \theta(1,-1) & \theta(1,1) & \theta(1,0) \\ \hline \theta(1,0) & \theta(1,-1) & \theta(1,1) & | & \theta(0,0) & \theta(0,-1) & \theta(0,1) & | & \theta(-1,0) & \theta(-1,-1) & \theta(-1,1) \\ \theta(1,1) & \theta(1,0) & \theta(1,-1) & | & \theta(0,1) & \theta(0,0) & \theta(0,-1) & | & \theta(-1,1) & \theta(-1,0) & \theta(-1,-1) \\ \theta(1,-1) & \theta(1,1) & \theta(1,0) & | & \theta(0,-1) & \theta(0,1) & \theta(0,0) & | & \theta(-1,-1) & \theta(-1,1) & \theta(-1,0) \\ \hline \theta(-1,0) & \theta(-1,-1) & \theta(-1,1) & | & \theta(1,0) & \theta(1,-1) & \theta(1,1) & | & \theta(0,0) & \theta(0,-1) & \theta(0,1) \\ \theta(-1,1) & \theta(-1,0) & \theta(-1,-1) & | & \theta(1,1) & \theta(1,0) & \theta(1,-1) & | & \theta(0,1) & \theta(0,0) & \theta(0,-1) \\ \theta(-1,-1) & \theta(-1,1) & \theta(-1,0) & | & \theta(1,-1) & \theta(1,1) & \theta(1,0) & | & \theta(0,-1) & \theta(0,1) & \theta(0,0) \end{bmatrix}.$$

Assim, o filtro discriminativo será:

$$\Theta = \begin{bmatrix} -0,2 & 0,3 & 0,3 \\ 0,3 & -0,2 & -0,2 \\ 0,3 & -0,2 & -0,2 \end{bmatrix}_{3 \times 3}. \quad (3.29)$$

Por fim, podemos verificar que a convolução circular de Θ com \mathbf{U} é igual a um impulso bidimensional.

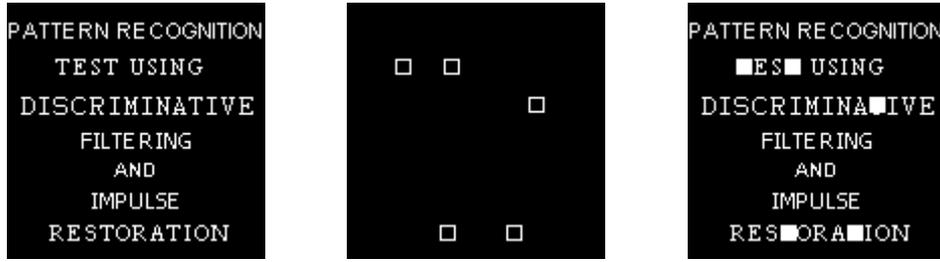
Exemplo: Detecção da letra “T” em uma imagem de teste

Vamos considerar que desejamos detectar o padrão \mathbf{U} igual a:

$$\mathbf{U} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}_{9 \times 9}. \quad (3.30)$$

A Equação (3.30) define um padrão com dimensões 9×9 semelhante à letra “T” do nosso alfabeto no formato *Times*. O filtro discriminativo Θ pode ser encontrado conforme o primeiro exemplo. Utilizando Θ podemos realizar a detecção da letra “T” em uma imagem de teste². Neste caso, considere a imagem de teste apresentada na Figura 3.3(a).

Efetuando o deslocamento pixel a pixel de uma janela J com dimensões 9×9 (mesma de \mathbf{U}) na imagem de teste, poderemos localizar a posição onde a letra “T” ocorre. Para tal, devemos avaliar se a convolução circular entre J e Θ é igual a um impulso bidimensional³. A Figura 3.3(b) apresenta os resultados da detecção, perceba que a letra “T” no formato *Times* foi detectada em todos os casos. Na Figura 3.3(c) temos a sobreposição das detecções na imagem original.



(a) Imagem de teste (b) Detecção para “T” (c) Sobreposição

Figura 3.3: Detecção em uma imagem de teste da letra “T”.

Exemplo: Detecção das letras “T” e “I” em uma imagem de teste

Vamos considerar que desejamos detectar os padrões \mathbf{U}_1 e \mathbf{U}_2 iguais a:

$$\mathbf{U}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}_{9 \times 9}, \quad \mathbf{U}_2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}_{9 \times 9}. \quad (3.31)$$

²Lembrando que $\mathbf{U} * \Theta$ é igual a um impulso bidimensional com $*$ sendo a convolução circular.

³Na prática, utilizamos a métrica denominada DSNR_2 definida na Equação (3.14).

As Equações em (3.31) definem padrões semelhantes as letras “T” (formato *Times*) e “I” (formato *Currier*) com dimensões 9×9 . A detecção foi realizada efetuando-se o deslocamento pixel a pixel de uma janela J na imagem apresentada na Figura 3.4(a) (mesma do exemplo anterior). As detecções das letras “T” e “I” são apresentadas na Figura 3.4(b) e a Figura 3.4(c) mostra a sobreposição das detecções com a imagem original.

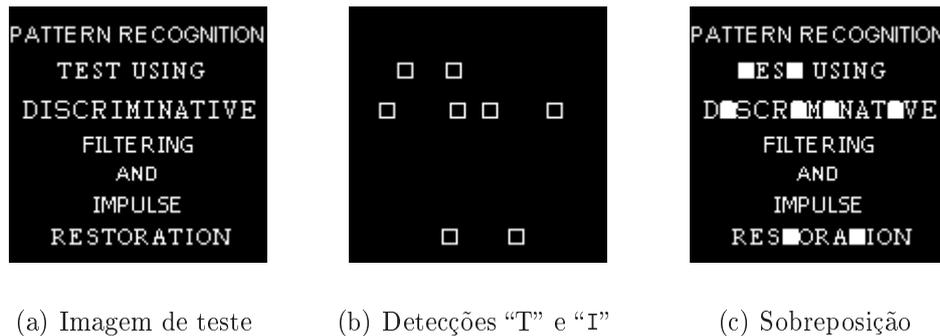


Figura 3.4: Detecção em uma imagem de teste das letras “T” e “I”.

3.5 Considerações Finais

Neste capítulo apresentamos a Teoria de Filtragem Discriminativa como um problema de Restauração do Impulso unidimensional e bidimensional. A idéia principal do método é obter um operador Θ que, ao ser convoluído com o padrão de interesse \mathbf{U} , resulte em um sinal de saída \mathbf{C} (com $\mathbf{C} = \Theta * \mathbf{U}$) cuja energia fique concentrada na amostra central do sinal filtrado. A seguir, apresentamos um filtro discriminativo, denominado Filtro OU, capaz de detectar múltiplos padrões de interesse. Por fim, apresentamos alguns exemplos com o objetivo de demonstrar como é feito o projeto dos filtros discriminativos. Nestes exemplos, os filtros discriminativos foram projetados para detecção de letras do Alfabeto. Foram utilizadas as formulações para único e múltiplos padrões.

No próximo capítulo, apresentamos uma das contribuições da Tese. O objetivo será obtermos filtros discriminativos robustos através do projeto de filtros para as componentes principais de maior energia.

Capítulo 4

Filtragem Discriminativa utilizando Análise de Componentes Principais (DF-B-PCA)

Neste capítulo apresentamos uma das contribuições deste trabalho. O objetivo é mostrar como utilizar a Teoria de Filtragem Discriminativa em conjunto com a Análise de Componentes Principais para realizar a detecção de objetos. A organização das seções será feita como segue. Primeiramente, mostramos conceitos relacionados à Análise de Componentes Principais. A seguir, apresentamos um popular algoritmo para reconhecimento de faces que utiliza componentes principais denominado *Eigenfaces*. Em seguida, verificamos um método análogo denominado *Eigenpoints*. Por fim, mostramos o método proposto denominado Filtragem Discriminativa utilizando Análise de Componentes Principais.

4.1 Análise de Componentes Principais

A Análise de Componentes Principais (PCA), também conhecida como Transformada *Karhunen-Loève* (KLT) [29,30], é uma técnica largamente utilizada sendo aplicada em problemas de redução de dimensionalidade [74,75], compressão de dados [76,77], extração de características (ou *features*) [78,79], detecção e reconhecimento de faces [6,80–82]. A PCA é baseada na extração de um número desejado de componentes principais (*principal components*) de um vetor multidimensional e

pode ser definida como uma projeção ortogonal deste vetor em um subespaço linear cuja dimensão é inferior à dimensão original (conhecido como *subespaço principal*) de tal forma que a variância do vetor projetado seja maximizada [20]. Uma definição alternativa é baseada na minimização da soma quadrática do erro de projeção. Podemos visualizar estas definições na Figura 4.1, onde temos um vetor bidimensional $\mathbf{x} = (x_1, x_2)$ indicado pelos quadrados na cor escura e $\bar{\mathbf{x}} = (\bar{x}_1)$ sendo sua projeção (quadrados claros) em um subespaço de dimensão menor (linha escura). Perceba que a projeção ortogonal de \mathbf{x} é feita de tal forma que sua variância seja maximizada. A definição em termos do erro de projeção está representada pela linha pontilhada.

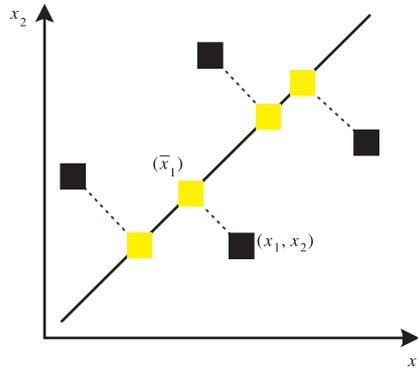


Figura 4.1: Representação da Análise de Componentes Principais

A formulação matemática para PCA pode ser obtida do seguinte modo. Seja $\mathcal{X}_{N \times 1}$ uma variável aleatória com M realizações iguais aos vetores \mathbf{x}_i . A base $\Phi = [\phi_1, \dots, \phi_N]$ do subespaço que maximiza a variância é obtida através da solução do problema dos autovalores Λ , conforme a expressão:

$$\Lambda = \Phi^{*t} \Sigma_{\mathcal{X}} \Phi, \quad (4.1)$$

onde o super-escrito *t indica o transposto conjugado da matriz e $\Sigma_{\mathcal{X}}$ é a matriz de covariância de \mathcal{X} com média igual a zero.

Dessa forma, podemos verificar na literatura [20,83,84] que $\Phi = [\phi_1, \dots, \phi_N]$ é a matriz de autovetores ortogonais de $\Sigma_{\mathcal{X}}$ e a matriz Λ será diagonal e seus elementos da diagonal principal serão os autovalores $\lambda_1 \geq \dots \geq \lambda_N$ de $\Sigma_{\mathcal{X}}$ correspondentes a cada autovetor.

Podemos encontrar na literatura uma abordagem mais eficiente para obter-

mos a PCA. Ela pode ser feita utilizando-se a Decomposição em Valores Singulares (SVD) [85–88] da seguinte forma. Supondo-se uma matriz \mathbf{Y} com dimensão $M \times N$ (com $M \geq N$) então a SVD será:

$$\mathbf{Y} = \mathbf{U}\mathbf{D}\mathbf{V}^{*t}, \quad (4.2)$$

onde $\mathbf{U}_{M \times N}$ e $\mathbf{V}_{N \times N}$ são matrizes com colunas ortonormais e $\mathbf{D}_{N \times N}$ será uma matriz diagonal com elementos da diagonal principal sendo os valores singulares de \mathbf{Y} . Assim, de acordo com [83, 85], a matriz de autovetores Φ de $\Sigma_{\mathcal{X}}$ será igual à matriz \mathbf{V} e o quadrado da matriz \mathbf{D} será igual à matriz de autovalores se considerarmos $\mathbf{Y} = \frac{1}{M-1}\mathbf{X}^{*t}$. As colunas da matriz $\mathbf{X}_{N \times M}$ serão formadas utilizando-se M realizações \mathbf{x}_i de $\mathcal{X}_{N \times 1}$ (média zero), ou seja:

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_M], \quad (4.3)$$

dessa forma, não é necessário calcularmos a matriz de covariância $\Sigma_{\mathcal{X}}$ da variável aleatória \mathcal{X} .

4.2 *Eigenfaces*

O estudo de algoritmos para reconhecimento de faces humanas que utilizam análise de componentes principais foi iniciado através dos trabalhos desenvolvidos em [1, 2]; outros algoritmos podem ser encontrados em [3, 4]. A principal motivação para este método reside no fato de que as faces de um conjunto de faces não se encontram distribuídas de forma aleatória em um espaço de alta dimensão, sendo assim, elas podem ser representadas em subespaços com dimensão inferior. Dessa forma, podemos encontrar as componentes principais para um conjunto de faces. O termo *eigenface* deve-se ao fato da representação das faces através dos autovetores do conjunto de faces.

A formulação matemática para obtermos as *eigenfaces* pode ser obtida da seguinte forma: suponha uma variável aleatória $\mathcal{X}_{N \times 1}$ com M realizações iguais a \mathbf{x}_i . Cada vetor \mathbf{x}_i será formado a partir da concatenação das colunas da face γ_i , onde $i = \{1, \dots, M\}$. Defina \mathbf{X} como uma matriz formada por M vetores, com

$M > N$ da seguinte forma:

$$\mathbf{X} = [\mathbf{x}_1 - \boldsymbol{\mu}_{\mathcal{X}}, \dots, \mathbf{x}_M - \boldsymbol{\mu}_{\mathcal{X}}], \quad (4.4)$$

onde $\boldsymbol{\mu}_{\mathcal{X}}$ é igual a:

$$\boldsymbol{\mu}_{\mathcal{X}} = \frac{1}{M} \sum_{i=1}^M \mathbf{x}_i. \quad (4.5)$$

Seja $\boldsymbol{\Sigma}_{\mathcal{X}}$ a matriz de covariância de \mathcal{X} dada por:

$$\boldsymbol{\Sigma}_{\mathcal{X}} = \frac{1}{M-1} \mathbf{X}\mathbf{X}^{*t}. \quad (4.6)$$

A base $\boldsymbol{\Phi}$ ótima, composta por N direções principais descorrelacionadas, é obtida através da diagonalização de $\boldsymbol{\Sigma}_{\mathcal{X}}$, utilizando a solução do problema de autovalores (detalhes em [20] e [83]). Dessa forma:

$$\boldsymbol{\Lambda} = \boldsymbol{\Phi}^{*t} \boldsymbol{\Sigma}_{\mathcal{X}} \boldsymbol{\Phi}, \quad (4.7)$$

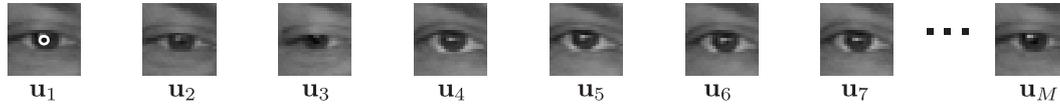
onde $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_N]$ é a matriz de autovetores ortogonais de $\boldsymbol{\Sigma}_{\mathcal{X}}$ e a matriz $\boldsymbol{\Lambda}$ será diagonal e seus elementos da diagonal principal serão os autovalores $\lambda_1 \geq \dots \geq \lambda_N$ de $\boldsymbol{\Sigma}_{\mathcal{X}}$ correspondentes a cada autovetor.

4.3 Eigenpoints

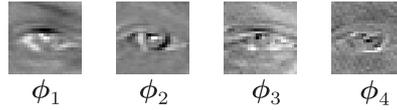
Considerando-se o contexto de faces humanas, podemos usar, ao invés de um conjunto de M faces $\boldsymbol{\gamma}_i$, um conjunto genérico de M blocos \mathbf{u}_i . Neste caso, o centro de cada bloco \mathbf{u}_i irá representar uma característica saliente da face humana denominada ponto fiducial (em inglês, *fiducial point*¹). Analogamente à Seção 4.2, podemos obter os *eigenpoints* através da Equação (4.7). Na Figura 4.2(a), apresentamos um exemplo considerando blocos \mathbf{u}_i com dimensões 27×27 com centros no ponto fiducial localizado no olho direito (observe a marcação em \mathbf{u}_1) de i faces humanas, onde $i = \{1, 2, \dots, M\}$. Efetuando-se as operações definidas nas Equações (4.4) até (4.7) (as realizações de \mathcal{X} serão iguais a \mathbf{u}_i ao invés de \mathbf{x}_i) podemos determinar os

¹O termo em inglês, *facial landmarks* também é comum.

eigenpoints. Na Figura 4.2(b), temos a representação das quatro *eigenpoints* mais significativas (autovetores Φ).



(a) Blocos \mathbf{u}_i de i faces humanas.



(b) Representação dos quatro *eigenpoints* ϕ_i mais significativos.

Figura 4.2: Exemplo de blocos com centros no ponto fiducial localizado no centro do olho direito (observe a marcação do ponto fiducial no bloco \mathbf{u}_1) de algumas faces humanas e a representação dos quatro *eigenpoints* em ordem decrescente de seus respectivos autovalores λ_i .

Podemos utilizar *eigenpoints* em problemas de semi-occlusão de objetos, por exemplo com o rosto coberto ou com disfarce [6, 89]. Na literatura, podemos encontrar definições similares aos *eigenpoints* e suas aplicações possuem ramificações em diversas áreas. Elas incluem *eigeneyes*, *eigennoses* e *eigenmouths* que são usadas para representação, detecção e reconhecimento de marcas faciais [6, 90]; *eigenexpression* usadas para análise de expressão [5] e *eigentracking* [18] para aplicações de marcação de rotas ou trilhas (*tracking*).

4.4 Filtragem Discriminativa utilizando Análise de Componentes Principais

Nas Figuras 4.2(a) e 4.2(b), conforme mencionado, temos um conjunto de blocos \mathbf{u}_i que estão centrados no ponto fiducial localizado no olho direito e quatro *eigenpoints*. Perceba que pequenas mudanças em \mathbf{u}_i aparentemente não impactam significativamente a forma dos *eigenpoints*. Assim, as características comuns aos padrões de interesse estão presentes nos *eigenpoints*, ou seja, os *eigenpoints* são uma boa representação das características invariantes do padrão o qual desejamos detectar. Esta percepção desperta nosso interesse pela seguinte pergunta:

Como podemos utilizar as características comuns aos padrões de interesse para projetarmos filtros discriminativos mais eficientes?

A resposta à pergunta acima motivou a origem do método denominado Filtragem Discriminativa utilizando Análise de Componentes Principais. Com este método podemos incrementar a robustez da filtragem discriminativa através do projeto de filtros discriminativos para as componentes principais de maior variância. Assim, iremos projetar diferentes filtros discriminativos para cada uma das direções de maior variância (observe a Figura 4.3).

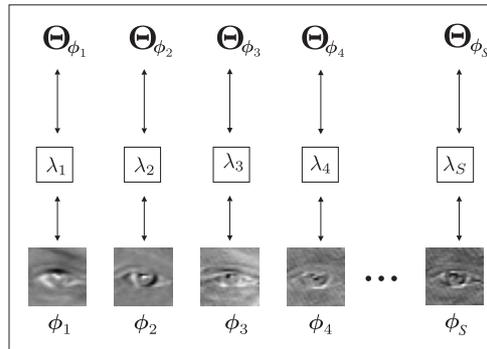


Figura 4.3: Projeto de S filtros discriminativos para S componentes principais. A base Φ possui um total de N componentes. O índice i das componentes principais ϕ_i indica a ordem decrescente de seus valores de variância (autovalores λ_i), dessa forma, ϕ_1 será a componente de maior variância com valor igual a λ_1 ao passo que ϕ_S será a componente de menor variância.

A formulação matemática do método DF-B-PCA pode ser obtida da seguinte maneira. Suponha uma variável aleatória \mathcal{U} com realizações iguais aos vetores $\mathbf{u}_1, \dots, \mathbf{u}_M$. Cada vetor \mathbf{u}_i será obtido concatenando-se as colunas do bloco i cujo centro é o ponto fiducial de interesse. Os *eigenpoints* $\Phi = [\phi_1, \dots, \phi_N]$ e seus respectivos autovalores $\lambda_1, \dots, \lambda_N$ (diagonal principal da matriz Λ) podem ser obtidos efetuando-se as operações definidas nas Equações (4.4) até (4.7) que escreveremos a seguir por conveniência, adequando-se a notação utilizada:

$$\boldsymbol{\mu}_{\mathcal{U}} = \frac{1}{M} \sum_{i=1}^M \mathbf{u}_i,$$

$$\mathbf{U} = [\mathbf{u}_1 - \boldsymbol{\mu}_{\mathcal{U}}, \dots, \mathbf{u}_M - \boldsymbol{\mu}_{\mathcal{U}}],$$

$$\boldsymbol{\Sigma}_{\mathcal{U}} = \frac{1}{M-1} \mathbf{U} \mathbf{U}^{*t},$$

$$\mathbf{\Lambda} = \mathbf{\Phi}^{*t} \mathbf{\Sigma}_{\mathcal{U}} \mathbf{\Phi}.$$

Os filtros discriminativos bidimensionais $\mathbf{\Theta}_{\phi_1}, \dots, \mathbf{\Theta}_{\phi_S}$ serão projetados para S componentes principais ϕ_1, \dots, ϕ_S com autovalores associados iguais à $\lambda_1, \dots, \lambda_S$. Os filtros serão obtidos utilizando-se a Equação (3.19), que descreve a matriz \mathbf{F} e a Equação (3.13), que define a matriz \mathbf{A} . Por comodidade, a notação das matrizes \mathbf{F} e \mathbf{A} será modificada para \mathbf{F}_{ϕ_i} e \mathbf{A}_{ϕ_i} com $i = \{1, \dots, S\}$. Dessa forma, a Equação (3.13) poderá ser reescrita como:

$$\mathbf{A}_{\phi_i} = \mathbf{F}_{\phi_i}^{*t} \left[\mathbf{F}_{\phi_i} \mathbf{F}_{\phi_i}^{*t} + \mathbf{C}_{\mathbf{b}i} \right]^{-1}. \quad (4.8)$$

onde o super-escrito *t indica o transposto conjugado. Relembre que iremos utilizar a restrição imposta à matriz \mathbf{A}_{ϕ_i} (apresentada na Equação (3.20)) para determinarmos o filtro $\mathbf{\Theta}_{\phi_i}$ por inspeção.

Por último, vamos analisar o termo $\mathbf{C}_{\mathbf{b}i}$ da Equação (4.8). Vamos iniciar lembrando que em [14, 24, 25, 27] explana-se a dificuldade encontrada para determinarmos a matriz $\mathbf{C}_{\mathbf{b}}$ de tal forma que o termo $[\mathbf{F}\mathbf{F}^{*t} + \mathbf{C}_{\mathbf{b}}]$ seja invertível. Assim, nestes trabalhos, a matriz $\mathbf{C}_{\mathbf{b}}$ ou é feita igual à matriz nula ou é feita proporcional à identidade. Em nosso caso, o uso das componentes principais fornece um interessante modelo para determinarmos a matriz $\mathbf{C}_{\mathbf{b}i}$ associada a cada um dos filtros discriminativos $\mathbf{\Theta}_{\phi_i}$ projetado para a componente ϕ_i . Para determinarmos tal modelo, vamos supor que nós intencionamos detectar a componente principal ϕ_i . Assim, os padrões que não são de interesse irão se localizar no subespaço ortogonal a ϕ_i . Conseqüentemente, o ruído associado à componente ϕ_i será considerado como a projeção de padrões típicos no subespaço ortogonal à ϕ_i . Assim, a matriz de covariância $\mathbf{C}_{\mathbf{b}i}$ associada a direção ϕ_i poderá ser escrita como:

$$\mathbf{C}_{\mathbf{b}i} = \sum_{\substack{j=1 \\ j \neq i}}^N \psi_j \mathbf{F}_{\phi_j} \mathbf{F}_{\phi_j}^{*t}, \quad (4.9)$$

onde ψ_j são constantes que indicam a estatística do ruído, e \mathbf{F}_{ϕ_j} é a matriz circulante por blocos obtida a partir da componente ϕ_j . Nos experimentos realizados, se usou ψ_j proporcionais a λ_j , ou seja, $\psi_j = \alpha \lambda_j$. Esta escolha se deve ao fato de que, no conjunto de treinamento, as demais componentes principais irão ter na média

energias proporcionais a λ_j .

Finalmente, se substituirmos a Equação (4.9) na Equação (4.8) iremos obter a forma definitiva das matrizes \mathbf{A}_{ϕ_i} :

$$\mathbf{A}_{\phi_i} = p_i \mathbf{F}_{\phi_i}^{*t} \left[p_i \mathbf{F}_{\phi_i} \mathbf{F}_{\phi_i}^{*t} + (1 - p_i) \sum_{\substack{j=1 \\ j \neq i}}^N \alpha \lambda_j \mathbf{F}_{\phi_j} \mathbf{F}_{\phi_j}^{*t} \right]^{-1}, \quad (4.10)$$

onde p_i é a probabilidade a priori² de ϕ_i . Vale lembrar novamente que iremos utilizar a restrição imposta à matriz \mathbf{A}_{ϕ_i} (Equação (3.20)) para determinarmos o filtro Θ_{ϕ_i} por inspeção.

4.5 Considerações Finais

Neste capítulo, apresentamos uma das contribuições deste trabalho. Nós demonstramos como projetar filtros discriminativos robustos utilizando as componentes principais. Os filtros são projetados para as direções principais de maior variância. Eles podem ser utilizados para detectar padrões com leves rotações e deformações. Verificamos que o uso das componentes principais fornece uma interessante maneira para determinarmos a matriz de covariância usada no projeto dos filtros discriminativos. Neste contexto, os padrões que não são de interesse irão se localizar no subespaço ortogonal à componente de interesse.

No próximo capítulo, apresentamos os procedimentos experimentais e as simulações utilizando-se o método proposto no contexto do problema da detecção de pontos fiduciais em faces humanas. Além disso, realizam-se comparações entre o método proposto, seu antecessor e um método similar baseado em SVM. As simulações são realizadas para duas bases de dados (*BioID* e *Feret*) com, respectivamente, 503 e 2004 imagens de faces.

²Na prática podemos considerar, sem perda de generalidade que todos os p_i 's são equiprováveis.

Capítulo 5

Filtragem Discriminativa utilizando Análise de Componentes Principais (DF-B-PCA): Procedimentos Experimentais e Simulações

5.1 Introdução

Com o objetivo de validar o método por Filtragem Discriminativa utilizando Análise de Componentes Principais, desenvolvido no Capítulo 4, abordamos o problema da detecção de pontos fiduciais em faces humanas¹ que vem recebendo especial atenção da comunidade científica. A detecção de pontos fiduciais pode ser usada para reconhecimento de faces [40, 41] e expressões [42], estimação de pose [41, 43], modelagem de faces em 3D [44] e rastreamento de faces [45, 46]. Não existe uma definição universal para pontos fiduciais em faces, porém pode-se comentar que eles determinam características salientes como os cantos dos olhos esquerdo e direito, ponta do nariz ou cantos da boca.

¹Pode-se encontrar outras denominações, em *inglês*, tais como *facial landmarks* e *facial feature point*.

5.2 Rótulos Manuais e Automáticos

Os rótulos de classe de um conjunto de padrões podem ser divididos como rótulos manuais ou automáticos. Os rótulos automáticos representam a saída do classificador para um padrão de entrada. Considerando-se que o problema da aprendizagem é supervisionado (veja o Capítulo 2) então temos que todos os padrões, utilizados para treinamento, possuem uma classe pré-especificada. Os rótulos de classe pré-especificados serão denominados por rótulos manuais. Iremos utilizar o super-escrito \sim para explicitar rótulos manuais.

5.3 Bases de Dados

Os experimentos foram realizados em duas bases de dados, denominadas *BioID* [91] e *Feret* [92–94]. Em uma breve descrição geral, na base de dados *BioID* temos um total de 1521 imagens contendo faces humanas frontais em nível de cinza de 23 indivíduos diferentes. A resolução das imagens é de 384×286 pixels. As faces possuem variações de escala (algumas estão perto da câmera e outras não), iluminação e pequenas rotações. Alguns indivíduos usam óculos, outros têm barba e/ou bigode. As imagens desta base de dados possuem rótulos manuais para 20 pontos fiduciais, porém estes rótulos são imprecisos em diversas imagens. Na base de dados *Feret* (versão *color*) temos um total de 11338 imagens coloridas, codificadas em 24 bits, contendo faces humanas em diferentes perspectivas. A resolução das imagens é de 512×768 pixels. As faces variam em escala e iluminação. Assim como na *BioID*, existem indivíduos com óculos, barba e/ou bigode. As imagens possuem rótulos manuais para 4 pontos fiduciais. Por fim, veja alguns exemplos de imagens para *BioID* e *Feret* no Apêndice A.

Consideramos um subconjunto de cada base de dados para realizarmos dois experimentos. No primeiro, utilizamos 503 imagens da base de dados *BioID* e no segundo experimento separamos 2004 imagens da base de dados *Feret*. Os critérios de seleção utilizados para originar os subconjuntos consistem em separar faces em perspectiva frontal, sem óculos, sem barba e sem bigode. Os rótulos manuais nessas imagens, tanto na *BioID* quanto na *Feret*, foram cuidadosamente realizados para 11 pontos fiduciais. As numerações, descrições e os acrônimos destes pontos fiduciais e

sua localização na face é fornecida na Tabela 5.1.

Tabela 5.1: Numerações, descrições e acrônimos dos pontos fiduciais das bases de dados *BioID* e *Feret*.

Pontos Fiduciais		
Número	Descrição	Acrônimo
00	Canto Externo do Olho Direito	CEOD
01	Centro do Olho Direito	COD
02	Canto Interno do Olho Direito	CIOD
03	Canto Externo do Olho Esquerdo	CEOE
04	Centro do Olho Esquerdo	COE
05	Canto Externo do Olho Esquerdo	CIOE
06	Canto Direito do Nariz	CDN
07	Ponta do Nariz	PN
08	Canto Esquerdo do Nariz	CEN
09	Canto Direito da Boca	CDB
10	Canto Esquerdo da Boca	CEB

5.4 Métrica de Desempenho

Nos experimentos, projeta-se um classificador $f_{\mathbf{PF}}$ para cada ponto fiducial \mathbf{PF} . Dessa forma, temos que nosso problema será classificar um candidato \mathbf{x} como o ponto fiducial \mathbf{PF} ou não, ou seja, este é um problema a duas classes. Assim, temos que:

$$f_{\mathbf{PF}}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \text{ é o ponto fiducial } \mathbf{PF} \\ 0, & \text{indica que } \mathbf{x} \text{ não é o ponto fiducial } \mathbf{PF} \end{cases}. \quad (5.1)$$

Neste trabalho, usaremos as distâncias normalizadas por \tilde{d}_o que corresponde à distância entre os rótulos manuais dos centros dos olhos esquerdo e direito². Matematicamente, \tilde{d}_o define-se da seguinte forma:

$$\tilde{d}_o = \sqrt{(\widetilde{\text{COE}}(x_1) - \widetilde{\text{COD}}(x_1))^2 + (\widetilde{\text{COE}}(y_1) - \widetilde{\text{COD}}(y_1))^2}, \quad (5.2)$$

²É comum encontrarmos outras denominações para \tilde{d}_o . As mais comuns são *distância interocular* e *distância entre as pupilas*.

onde: $\widetilde{\mathbf{COE}}(x_1, y_1)$ corresponde às coordenadas do rótulo manual do centro do olho esquerdo e $\widetilde{\mathbf{COD}}(x_1, y_1)$ são as coordenadas do rótulo manual do centro do olho direito. Supondo que as coordenadas do rótulo automático para o candidato \mathbf{x} sejam $\mathbf{x}(x_1, y_1)$ e que o rótulo manual do ponto fiducial \mathbf{PF} tenha coordenadas iguais a $\widetilde{\mathbf{PF}}(x_1, y_1)$ então define-se $d_{\mathbf{PF}}$ como:

$$d_{\mathbf{PF}} = \sqrt{(\mathbf{x}(x_1) - \widetilde{\mathbf{PF}}(x_1))^2 + (\mathbf{x}(y_1) - \widetilde{\mathbf{PF}}(y_1))^2}. \quad (5.3)$$

Finalmente, a decisão será feita da seguinte maneira:

$$f_{\mathbf{PF}}(\mathbf{x}) = \begin{cases} 1, & \text{se } d_{\mathbf{PF}} \leq \alpha \tilde{d}_o \\ 0, & \text{se } d_{\mathbf{PF}} > \alpha \tilde{d}_o \end{cases}, \quad (5.4)$$

onde: α indica qual percentual de \tilde{d}_o será considerado para confirmar se um candidato é ponto fiducial. Dessa forma α está no intervalo $0 \leq \alpha \leq 1$. É comum usarmos $\alpha = 0, 1$, ou seja, 10% de \tilde{d}_o .

5.4.1 Desempenho do Classificador

O desempenho do classificador é medido no contexto de dois cenários que levam em consideração os rótulos automáticos fornecidos na saída do método. No primeiro, levamos em conta todos os rótulos automáticos. Neste caso, o desempenho é medido através das taxas de Verdadeiros Positivos (TP) e Falsos Positivos (FP) conforme as expressões:

$$\text{TP}(\%) = 100 \frac{\text{Total de faces em que o algoritmo detecta o ponto fiducial}}{\text{Total de faces}}, \quad (5.5)$$

$$\text{FP}(\%) = 100 \frac{\text{Total de rótulos automáticos fora da margem}}{\text{Total de negativos}}. \quad (5.6)$$

O numerador da Equação (5.5) é determinado utilizando-se a Equação (5.4). O numerador da Equação (5.6) é determinado usando-se os rótulos automáticos com distância, em relação ao rótulo manual, superior à $\alpha \tilde{d}_o$. Na Figura 5.1, apresentamos uma ilustração da margem. Ela é definida pela circunferência de raio igual a $\alpha \tilde{d}_o$.

Na Figura 5.2, temos um exemplo deste cenário. Neste caso, apresentamos três imagens onde se considera todos os rótulos automáticos como saída. Temos,

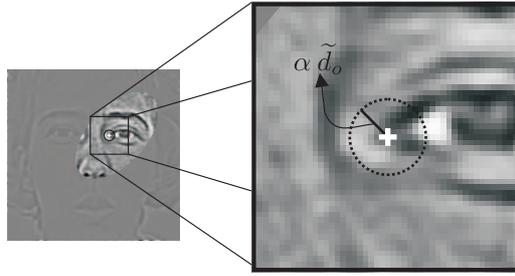


Figura 5.1: Exemplo da margem usada para determinarmos a taxa de FP.

respectivamente, quatro, três e seis rótulos automáticos para as imagens 1, 2 e 3. Neste caso, a taxa de Verdadeiro Positivo é igual a 100% e a taxa de Falso Positivo é igual a $FP = 100 \frac{3+1+2}{3000}\%$.

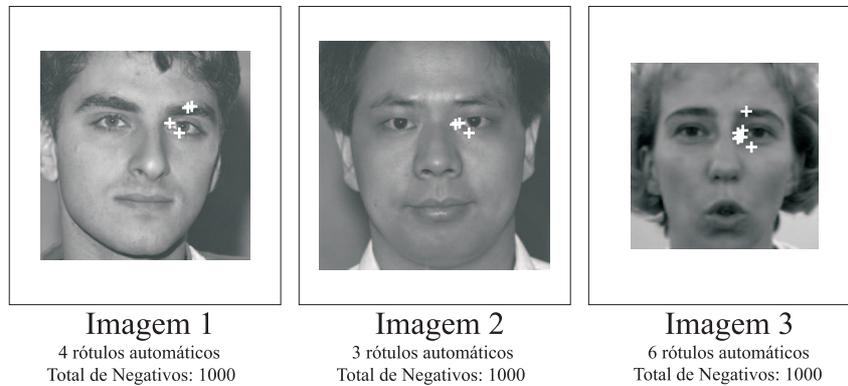


Figura 5.2: Exemplo do cenário que considera todos os rótulos automáticos como saída do método.

No segundo, consideramos um único rótulo automático por imagem. Neste caso, o desempenho é computado através da distribuição de acerto média acumulada *versus* distância inter-ocular. Para obtermos esta distribuição basta calcularmos, para todas as imagens, a que percentual da distância inter-ocular está o rótulo automático do rótulo manual. Na Figura 5.3, temos um exemplo deste cenário. Neste caso, a taxa de acerto é igual a 33% quando $\alpha = 2\%$, 66% quando $\alpha = 3\%$ e 100% quando α é maior que 6%.

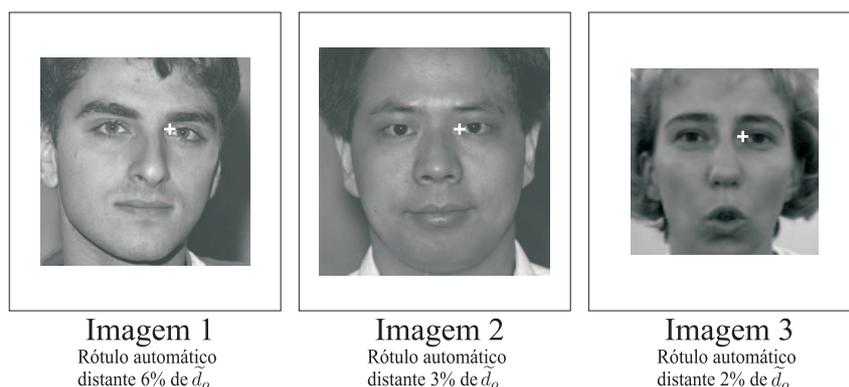


Figura 5.3: Exemplo para distribuição de acerto média acumulada *versus* distância inter-ocular.

5.5 Procedimentos para Pré-processamento, Treinamento e Teste

Utilizamos em todas as simulações³, validação cruzada com 7 partições (em inglês, *folds*) o que é comum na literatura. Dessa forma, uma simulação totaliza 7 experimentos. Em cada experimento temos dois conjuntos disjuntos, denominados por treinamento e teste. No conjunto de treinamento utilizamos 6/7 do total de imagens da base de dados para treinamento e verificação de parâmetros e no conjunto de teste utilizamos 1/7 do total de imagens para validar o modelo treinado.

Nesta seção, verificaremos primeiramente os procedimentos necessários para pré-processamento. A seguir, apresentamos as peculiaridades para treinamento dos métodos e, por fim, os procedimentos para teste dos métodos são expostos.

5.5.1 Pré-processamento das Imagens

Utilizamos, em todas as simulações, um procedimento comum para pré-processamento das imagens. Na Figura 5.4, temos o diagrama em blocos empregado sendo composto por três etapas diferentes. A primeira consiste no enquadramento da face da imagem de entrada utilizando um detector *Viola-Jones* [95,96], a seguir realizamos um escalonamento da imagem para uma resolução pré-definida⁴ e por último, fazemos a correção de iluminação [97,98].

³Cada simulação corresponde a um ponto fiducial. Dessa forma, temos 11 simulações distintas para cada base de dados.

⁴Em todas as simulações a resolução da imagem, após o escalonamento, é igual a 220×220 .

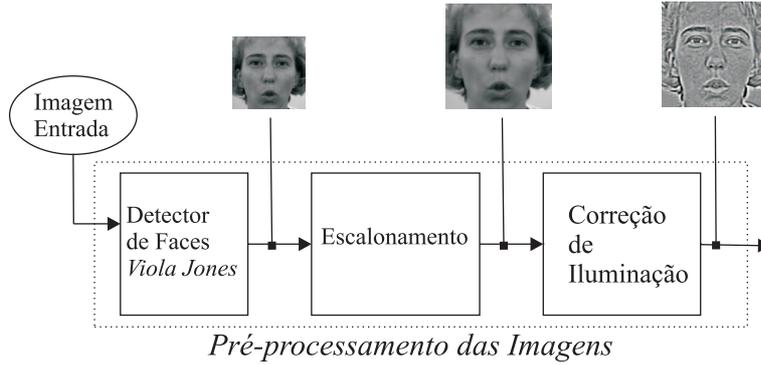


Figura 5.4: Diagrama em blocos para o pré-processamento das imagens. Neste exemplo, empregamos o pré-processamento à uma imagem em particular. As imagens, conectadas ao diagrama, fornecem uma visualização da saída das etapas.

5.5.2 Treinamento (Fase de Aprendizagem)

Nesta fase, efetuamos o aprendizado dos métodos utilizando, para cada experimento, o conjunto de treinamento que corresponde a 6/7 do total de imagens da base de dados. Primeiramente verificaremos os procedimentos iniciais, comuns para todos os métodos. A seguir, complementaremos os procedimentos para cada método utilizado.

5.5.2.1 Procedimentos Iniciais para Treinamento

Utilizamos dois procedimentos iniciais para treinamento, são eles: região de interesse e definição das classes de positivos e negativos.

Região de Interesse

A região de interesse (em inglês, *region of interest*) será obtida através de uma função densidade de probabilidade gaussiana com média $\boldsymbol{\mu}_{\mathcal{X}}$ e matriz de covariância $\boldsymbol{\Sigma}_{\mathcal{X}}$, onde \mathcal{X} é uma variável aleatória com N realizações iguais às coordenadas dos pontos fiduciais $\mathbf{x}_i = (x_{1i}, x_{2i})$ do conjunto de treinamento (onde $i = \{1, 2, \dots, N\}$). A distância euclidiana máxima, acrescida em 5%, entre $\boldsymbol{\mu}_{\mathcal{X}}$ e todos os \mathbf{x}_i definirá a região elíptica de interesse. Escolhe-se o percentual de 5% para uma margem de segurança da região elíptica. Dessa forma, um candidato a ponto fiducial **PF** com coordenadas $\mathbf{y} = (y_1, y_2)$, somente será considerado caso:

$$\max_{\text{rótulo}(\mathbf{x}_i)=\mathbf{PF}} \{1.05|\mathbf{x}_i - \boldsymbol{\mu}_{\mathcal{X}}|^2\} \geq (\mathbf{y} - \boldsymbol{\mu}_{\mathcal{X}})^t \boldsymbol{\Sigma}_{\mathcal{X}}^{-1} (\mathbf{y} - \boldsymbol{\mu}_{\mathcal{X}}). \quad (5.7)$$

Na Figura 5.5, apresentamos um exemplo de uma imagem após a obtenção da região de interesse obtida através da Equação (5.7). Neste exemplo, o ponto fiducial 03 (CEOE) é considerado.

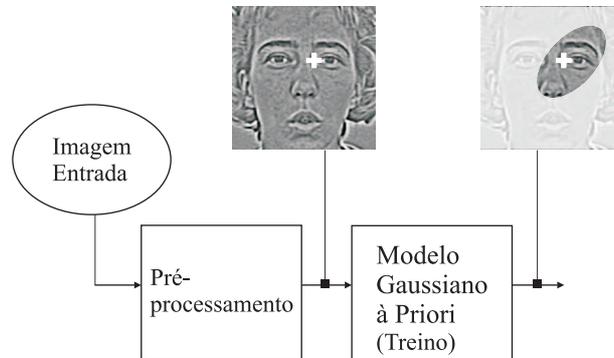


Figura 5.5: Aplicação em uma imagem da região de interesse, obtida através da Equação (5.7). A marcação "+" indica as coordenadas para o rótulo manual do ponto fiducial 03 (CEOE) desta imagem. Observe a região elíptica de interesse em evidência na imagem de saída do diagrama em blocos.

Definição das Classes de Positivos e Negativos

Após a definição da região de interesse realizamos a separação dos blocos em duas classes que serão denominadas pelo termo *positivos*, quando nos referirmos a blocos com centros em pontos fiduciais e por *negativos* quando nos referirmos aos outros blocos. Na Figura 5.6, ampliamos a região de interesse de uma imagem para observarmos de forma didática a localização dos blocos pertencentes à classe de positivos e negativos. Neste exemplo, novamente o ponto fiducial 03 (CEOE) é considerado. Na imagem da esquerda visualizamos o único bloco, centrado no ponto fiducial considerado, pertencente à classe de positivos. Claramente, o número total de blocos da classe de positivos é igual ao número de imagens de treinamento. Na imagem da direita podemos visualizar os outros blocos que pertencem à classe de negativos. Observe que o total de blocos desta classe será limitado pela região elíptica, determinada através da Equação (5.7), de tal forma que o centro dos blocos não ultrapasse esta região. Em todas as simulações, foram utilizados blocos com dimensões 13×13 .

Observe que o total de blocos da classe de negativos é bem superior ao total de blocos da classe de positivos. Porém, ao observarmos a imagem à direita percebemos intuitivamente que os blocos adjacentes da classe de negativos são semelhantes.

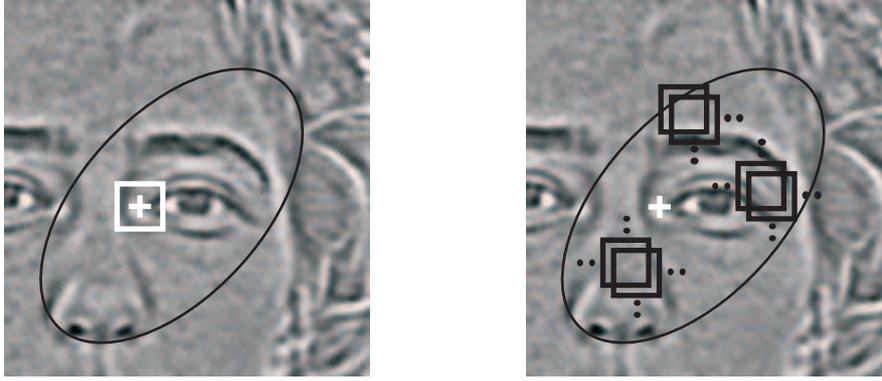


Figura 5.6: Região de interesse ampliada de uma imagem.

Sendo assim, visando a um melhor desempenho dos sistemas, realizaremos, em cada imagem de treinamento, uma seleção aleatória uniformemente distribuída de 50 blocos da classe de negativos. Por último, na Figura 5.7, apresentamos o diagrama em blocos para os procedimentos iniciais para treinamento.

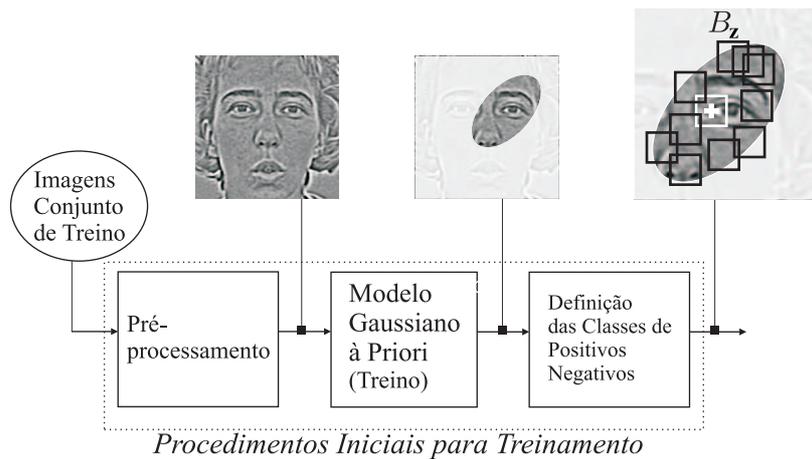


Figura 5.7: Diagrama em blocos para os procedimentos iniciais para treinamento.

5.5.2.2 Procedimentos para Treinamento dos Métodos

Na Figura 5.8, apresentamos o diagrama em blocos completo para treinamento do método DF-B-PCA. Para facilitar o entendimento, inserimos, ao longo do diagrama, uma visualização do processamento de uma imagem. Primeiramente, realizamos os procedimentos iniciais para treinamento (Seção 5.5.2.1). A seguir, os filtros discriminativos Θ_{ϕ_i} , onde $i = \{1, \dots, k\}$, serão projetados para cada uma das componentes principais com maiores variâncias utilizando a Equação (4.10). Cada bloco pertencente a classe de positivos e negativos, subtraído do bloco mé-

dio da classe de positivos (denominado por μ_U), será filtrado por todos os filtros Θ_{ϕ_i} gerando k matrizes $C_{\phi_i}(B_z)$. Aplicando a Equação (3.14) obtemos os valores de $DSNR_2$ das matrizes $C_{\phi_i}(B_z)$. Assim, cada bloco B_z terá um vetor $DSNR_2$ associado denominado \mathbf{d}_{B_z} com dimensões $1 \times k$. Os vetores \mathbf{d}_{B_z} serão utilizados para treinamento do classificador *Gentle AdaBoost* [52] utilizando a *toolbox* GML AdaBoost Matlab [53] de tal forma que possamos obter a superfície de decisão que separa a classe de positivos da classe de negativos. Os filtros Θ_{ϕ_i} e o classificador AdaBoost correspondem à saída do diagrama em blocos (o bloco médio μ_U também será utilizado no teste). Por comodidade, todas as etapas estão descritas na legenda da Figura 5.8.

Com o propósito de comparação, realizamos outras três implementações. A primeira, denominada por DF-B, consiste na Filtragem Discriminativa sem utilizar a análise de componentes principais. A segunda, denominada por DF, é a versão original do método por Filtragem Discriminativa⁵ [14]. Em ambos, o filtro discriminativo Θ é projetado utilizando-se o bloco médio da classe de positivos. A saída para DF consiste no filtro discriminativo Θ e a saída para DF-B compreende o filtro Θ e o classificador AdaBoost. O classificador será obtido como segue: cada bloco pertencente à classe de positivos e negativos será filtrado por Θ gerando a matriz $C(B_z)$. Aplicando a Equação (3.14) obtemos os valores de $DSNR_2$ das matrizes $C(B_z)$. Assim, cada bloco B_z terá um escalar d_{B_z} associado (valor da $DSNR_2$). Os escalares d_{B_z} serão utilizados para treinamento do classificador *Gentle AdaBoost* [52] utilizando a *toolbox* GML AdaBoost Matlab [53].

No terceiro método, nomeado por SVM, empregamos classificadores *Support Vector Machines* (SVM) lineares e não lineares. Neste caso, utilizamos classificadores por *Support Vector Machines* (SVM) lineares e não lineares (original desenvolvido por Vapnik em [47], outras referências em [20, 29, 55]) empregando a *toolbox* SVM-*light* [57]. A configuração em cascata de classificadores SVM será formada por estágios. Cada estágio será projetado utilizando-se os positivos (falsos ou verdadeiros) do estágio imediatamente anterior bem como os rótulos do conjunto de treinamento. Quanto aos parâmetros (veja a Seção 2.1.4), para o classificador SVM linear variamos a constante C (entre 1 e 128) e para o SVM não linear (*kernel* poli-

⁵Neste caso, um único filtro é usado. Ele será projetado utilizando-se a média dos blocos pertencentes à classe de positivos.

nomial) variamos o grau do polinômio (entre 2 e 3) e a constante C (mesma variação do caso linear).

Na Figura 5.9(a), apresentamos o diagrama em blocos para os métodos DF-B e DF e na Figura 5.9(b), temos o diagrama para o método SVM. Por comodidade, a descrição de cada um está localizada nas legendas das figuras.

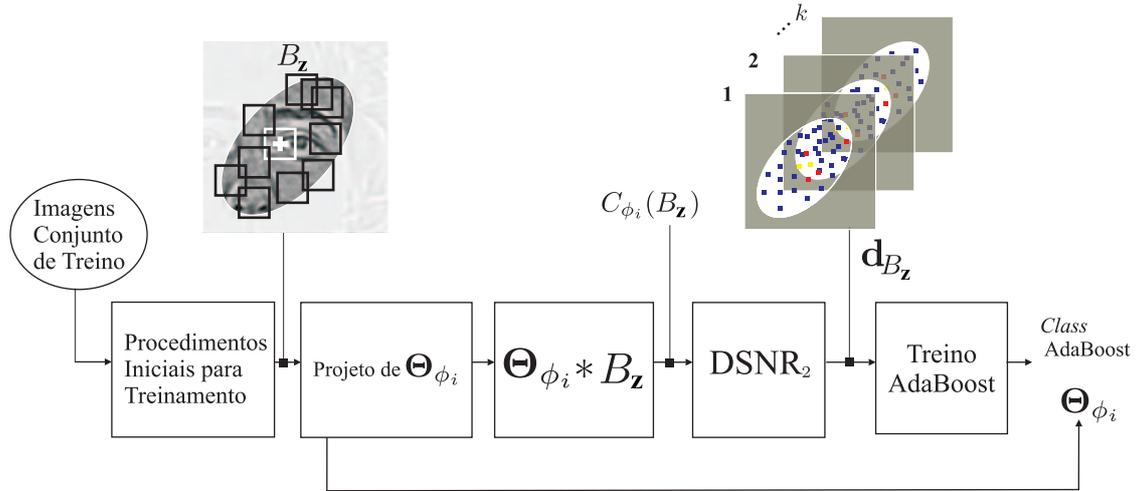
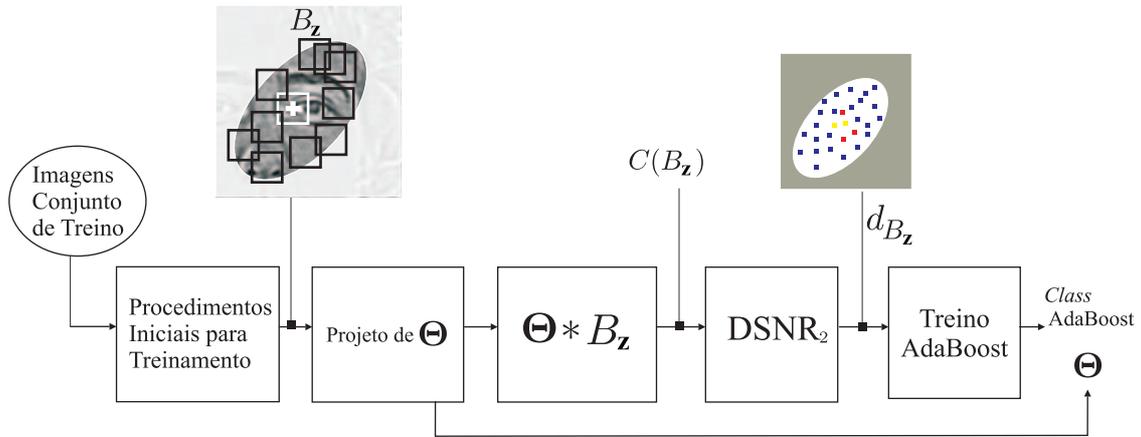
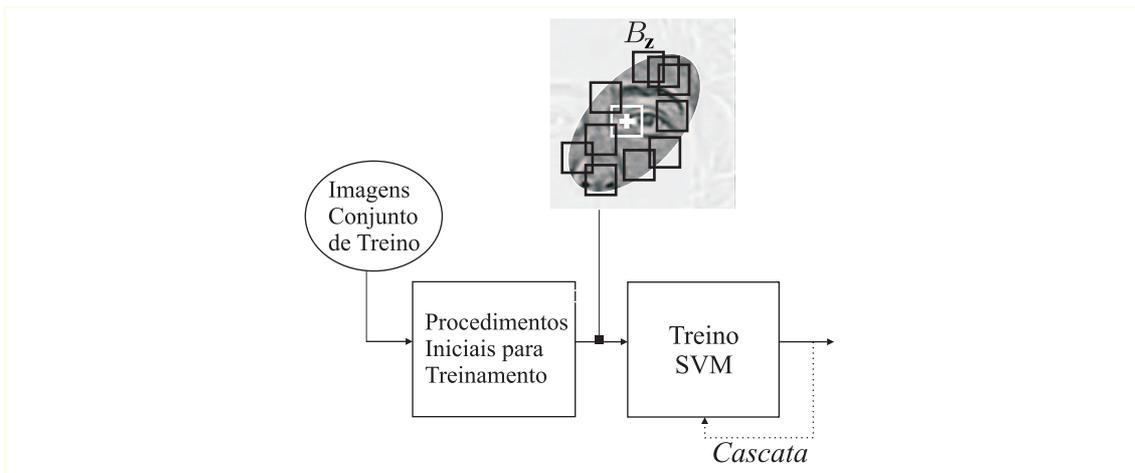


Figura 5.8: Diagrama em blocos para fase de treinamento do método DF-B-PCA. Primeiramente, realizamos os procedimentos iniciais para treinamento (Seção 5.5.2.1). A seguir, os filtros discriminativos Θ_{ϕ_i} , onde $i = \{1, \dots, k\}$, serão projetados para cada uma das componentes principais com maiores variâncias utilizando a Equação (4.10). Cada bloco pertencente a classe de positivos e negativos, subtraído do bloco médio da classe de positivos (denominado por μ_U), será filtrado por todos os filtros Θ_{ϕ_i} gerando k matrizes $C_{\phi_i}(B_z)$. Aplicando a Equação (3.14) obtemos os valores de $DSNR_2$ das matrizes $C_{\phi_i}(B_z)$. Assim, cada bloco B_z terá um vetor $DSNR_2$ associado denominado \mathbf{d}_{B_z} com dimensões $1 \times k$. Os vetores \mathbf{d}_{B_z} serão utilizados para treinamento do classificador *Gentle* AdaBoost [52] utilizando a *toolbox* GML AdaBoost Matlab [53] de tal forma que possamos obter a superfície de decisão que separa a classe de positivos da classe de negativos. Os filtros Θ_{ϕ_i} e o classificador AdaBoost correspondem à saída do diagrama em blocos (o bloco médio μ_U também será utilizado no teste).



(a) Diagrama em blocos para treinamento dos métodos DF-B e DF. Em ambos, o filtro discriminativo Θ é projetado utilizando-se o bloco médio da classe de positivos. A saída para DF consiste no filtro discriminativo Θ e a saída para DF-B compreende o filtro Θ e o classificador AdaBoost. O classificador será obtido como segue: cada bloco pertencente à classe de positivos e negativos será filtrado por Θ gerando a matriz $C(B_z)$. Aplicando a Equação (3.14) obtemos os valores de $DSNR_2$ das matrizes $C(B_z)$. Assim, cada bloco B_z terá um escalar d_{B_z} associado (valor da $DSNR_2$). Os escalares d_{B_z} serão utilizados para treinamento do classificador *Gentle* AdaBoost [52] utilizando a *toolbox* GML AdaBoost Matlab [53].



(b) Diagrama em blocos para treinamento do método SVM (linear e não linear). Neste caso, utilizamos classificadores por *Support Vector Machines* (SVM) lineares e não lineares (original desenvolvido por Vapnik em [47], outras referências em [20,29,55]) empregando a *toolbox* SVM-light [57]. A configuração em cascata de classificadores SVM será formada por estágios. Cada estágio será projetado utilizando-se os positivos (falsos ou verdadeiros) do estágio imediatamente anterior bem como os rótulos do conjunto de treinamento. Quanto aos parâmetros (veja a Seção 2.1.4), para o classificador SVM linear variamos a constante C (entre 1 e 128) e para o SVM não linear (*kernel* polinômio) variamos o grau do polinômio (entre 2 e 3) e a constante C (mesma variação do caso linear).

Figura 5.9: Diagramas em blocos para os métodos DF-B, DF e SVM.

5.5.3 Teste (Fase de Operação)

Nesta seção, realizamos o teste dos modelos, treinados conforme descrição realizada na Seção 5.5.2, dos métodos DF-B-PCA, DF-B, DF e SVM. Em cada experimento temos que o conjunto de teste será formado por 1/7 do total de imagens da base de dados. Inicialmente, apresentaremos os procedimentos comuns e a seguir verificaremos os procedimentos individuais para cada método.

Procedimentos Iniciais e Pós-processamento para Teste

Na Figura 5.10, verificamos os procedimentos iniciais para teste dos métodos. A descrição das etapas é feita a seguir. Inicialmente, realizamos o pré-processamento das imagens conforme descrição realizada na Seção 5.5.1. Por fim, aplicamos o modelo gaussiano a priori, obtido no treinamento (Seção 5.5.2), nas imagens de teste para determinarmos a região de interesse.

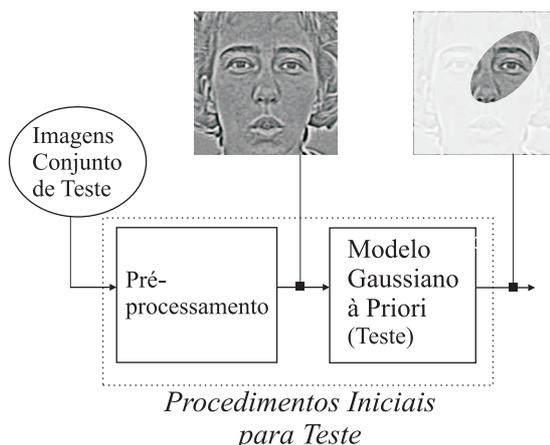


Figura 5.10: Diagrama em blocos para os procedimentos iniciais para teste.

Na Figura 5.12, verificamos os procedimentos para pós-processamento. Primeiramente, aplicamos um simples escalonamento para as dimensões da imagem original. A seguir, podemos optar entre aplicarmos ou não estratégias para obtermos somente um rótulo automático. Tais estratégias podem ser utilizadas devido à proximidade ou entre rótulos automáticos ou entre grupos de rótulos automáticos (observe a Figura 5.11). Iremos utilizar os acrônimos NA, ML, GML e A para indicar qual estratégia foi utilizada. Na Tabela 5.2, podemos verificar os acrônimos com as respectivas descrições.



Figura 5.11: Exemplo de rótulos automáticos agrupados. À esquerda, temos um exemplo de vários rótulos automáticos para uma imagem. À direita, temos o resultado obtido aplicando-se uma técnica de pós-processamento. Neste caso, apenas um rótulo automático foi selecionado.

Tabela 5.2: Estratégias utilizadas no pós-processamento para obtermos somente um rótulo automático. Cada estratégia será referenciada através dos acrônimos apresentados na primeira coluna. A descrição de cada estratégia é apresentada na segunda coluna.

Acrônimo	Descrição
NA	Indica que não será utilizada nenhuma estratégia para agrupamento. Dessa forma, todos os rótulos automáticos serão considerados como saída do método.
ML	Neste caso, iremos obter somente um rótulo automático como saída do método. Iremos utilizar o rótulo automático mais provável, supondo o modelo gaussiano a priori, como critério de escolha.
GML	Novamente, iremos obter somente um rótulo automático como saída do método. Neste caso, agrupamos os rótulos automáticos com distância menor que P pixels e a seguir escolhemos o rótulo automático mais provável (supondo o modelo gaussiano à priori).
A	Neste, teremos somente um rótulo automático como saída do método. Ele será igual à média dos rótulos automáticos.

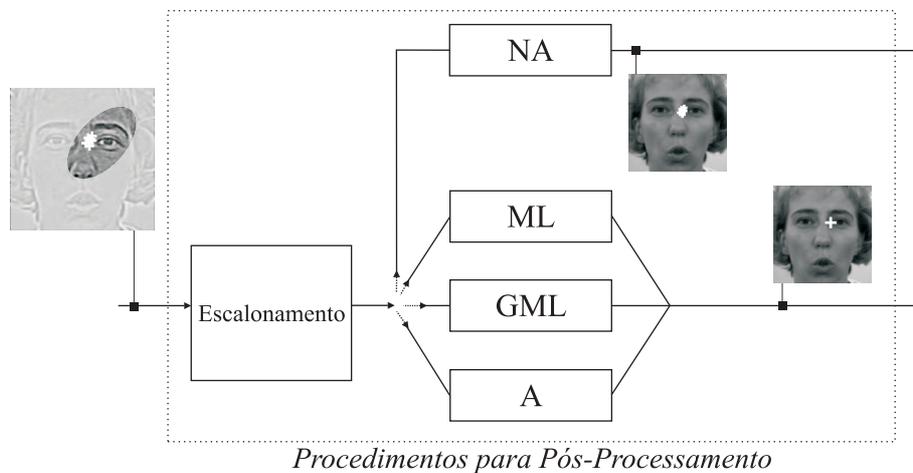


Figura 5.12: Diagrama em blocos para pós-processamento na etapa de teste.

Procedimentos para Teste

Na Figura 5.13, apresentamos o diagrama em blocos para teste do método DB-B-PCA e nas Figuras 5.14(a) e 5.14(b) temos os diagramas em blocos para teste dos métodos DF-B, DF e SVM. O processamento de uma imagem pode ser visualizado ao longo do diagrama em blocos. Para o diagrama apresentado na Figura 5.13, primeiramente, efetuamos os procedimentos iniciais para teste. A seguir, as imagens serão processadas por uma janela deslizante $B_{\mathbf{z}}$, onde \mathbf{z} é a coordenada correspondente ao centro do bloco. As matrizes $C_{\phi_i}(B_{\mathbf{z}})$ serão obtidas através da filtragem entre $B_{\mathbf{z}}$, subtraído por $\mu_{\mathcal{U}}$ (determinado no treinamento), com cada Θ_{ϕ_i} (obtidos no treinamento), onde $i = \{1, \dots, k\}$, k corresponde ao número de componentes principais utilizadas no treinamento. A saída do *box* DSNR_2 fornece a DSNR_2 (Equação (3.14)) para cada $C_{\phi_i}(B_{\mathbf{z}})$. Dessa forma, cada bloco $B_{\mathbf{z}}$ terá um vetor DSNR_2 associado denominado $\mathbf{d}_{B_{\mathbf{z}}}$. O classificador AdaBoost, construído no treinamento (Seção 5.5.2), usará $\mathbf{d}_{B_{\mathbf{z}}}$ para classificar $B_{\mathbf{z}}$. Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).

Para os diagramas dos métodos DF-B e DF, apresentados na Figura 5.14(a), primeiramente, efetuamos os procedimentos iniciais para teste. A seguir, as imagens serão processadas por uma janela deslizante $B_{\mathbf{z}}$, onde \mathbf{z} é a coordenada do centro do bloco. As matrizes $C(B_{\mathbf{z}})$ serão obtidas através da filtragem entre os blocos $B_{\mathbf{z}}$ e o filtro Θ (obtido no treinamento). A saída do *box* DSNR_2 fornece a DSNR_2 (Equação (3.14)) para $C(B_{\mathbf{z}})$. Dessa forma, cada bloco $B_{\mathbf{z}}$ terá um escalar DSNR_2 associado denominado $d_{B_{\mathbf{z}}}$. Neste ponto os métodos diferem. Para o método DF-B seguimos a rota **A** (observe no diagrama), neste caso, o classificador AdaBoost, construído no treinamento (Seção 5.5.2), usará $d_{B_{\mathbf{z}}}$ para classificar $B_{\mathbf{z}}$. Para o método DF seguimos a rota **B**, neste caso, a saída em cada imagem será igual aos valores de DSNR_2 máximos. Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).

Para o diagrama utilizado para os métodos baseados em SVM, apresentado na Figura 5.14(b), Primeiramente realizamos os procedimentos iniciais para teste. A seguir, as imagens serão processadas por uma janela deslizante $B_{\mathbf{z}}$, onde \mathbf{z} é a coordenada do centro do bloco. O classificador SVM, determinado no treinamento, será utilizado para classificar cada bloco $B_{\mathbf{z}}$. Por último, realizamos a etapa de pós-

processamento conforme a estratégia escolhida (veja a Tabela 5.2). Por comodidade apresentamos as descrições dos diagramas nos rodapés das figuras.

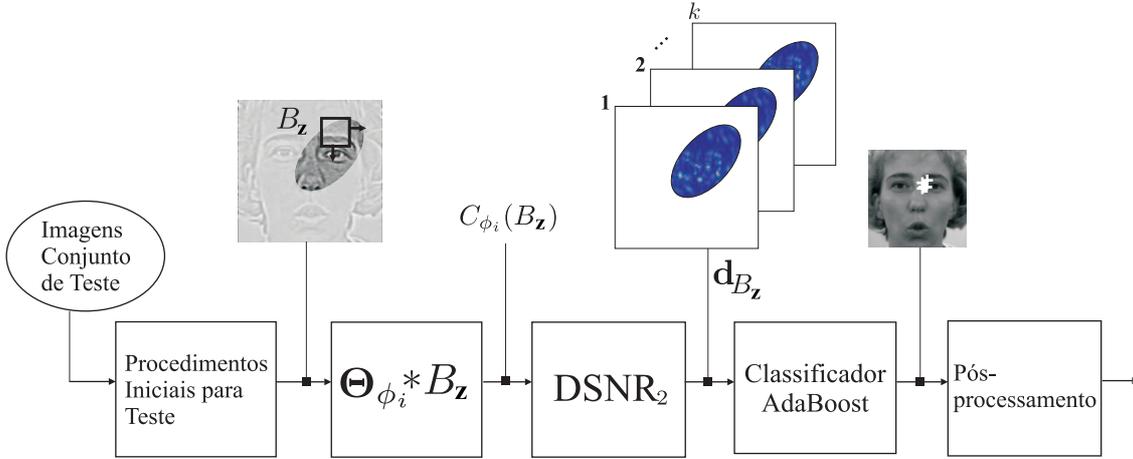
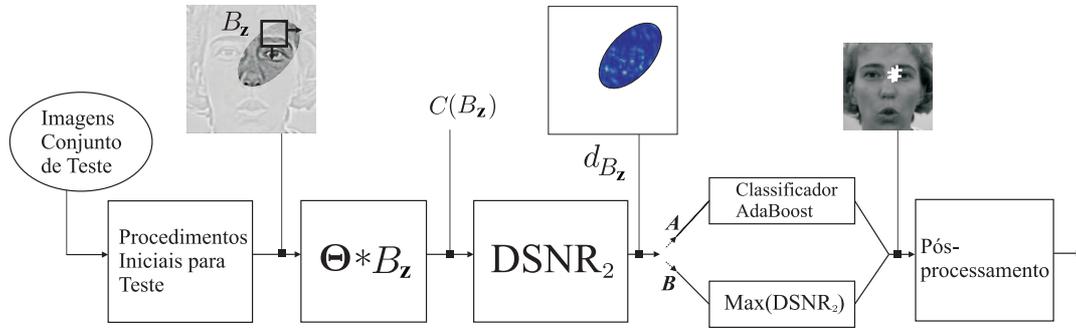
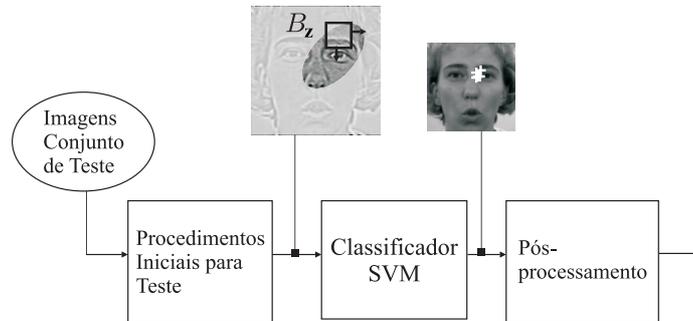


Figura 5.13: Diagrama em blocos para teste do método DF-B-PCA. Primeiramente, efetuamos os procedimentos iniciais para teste. A seguir, as imagens serão processadas por uma janela deslizante B_z , onde \mathbf{z} é a coordenada correspondente ao centro do bloco. As matrizes $C_{\phi_i}(B_z)$ serão obtidas através da filtragem entre B_z , subtraído por μ_U (determinado no treinamento), com cada Θ_{ϕ_i} (obtidos no treinamento), onde $i = \{1, \dots, k\}$, k corresponde ao número de componentes principais utilizadas no treinamento. A saída do *box* $DSNR_2$ fornece a $DSNR_2$ (Equação (3.14)) para cada $C_{\phi_i}(B_z)$. Dessa forma, cada bloco B_z terá um vetor $DSNR_2$ associado denominado \mathbf{d}_{B_z} . O classificador AdaBoost, construído no treinamento (Seção 5.5.2), usará \mathbf{d}_{B_z} para classificar B_z . Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).



(a) Métodos DF-B e DF. Em ambos, primeiramente efetuamos os procedimentos iniciais para teste. A seguir, as imagens serão processadas por uma janela deslizante $B_{\mathbf{z}}$, onde \mathbf{z} é a coordenada do centro do bloco. As matrizes $C(B_{\mathbf{z}})$ serão obtidas através da filtragem entre os blocos $B_{\mathbf{z}}$ e o filtro Θ (obtido no treinamento). A saída do *box* $DSNR_2$ fornece a $DSNR_2$ (Equação (3.14)) para $C(B_{\mathbf{z}})$. Dessa forma, cada bloco $B_{\mathbf{z}}$ terá um escalar $DSNR_2$ associado denominado $d_{B_{\mathbf{z}}}$. Neste ponto os métodos diferem. Para o método DF-B seguimos a rota **A** (observe no diagrama), neste caso, o classificador AdaBoost, construído no treinamento (Seção 5.5.2), usará $d_{B_{\mathbf{z}}}$ para classificar $B_{\mathbf{z}}$. Para o método DF seguimos a rota **B**, neste caso, a saída em cada imagem será igual aos valores de $DSNR_2$ máximos. Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).



(b) Método SVM. Primeiramente realizamos os procedimentos iniciais para teste. A seguir, as imagens serão processadas por uma janela deslizante $B_{\mathbf{z}}$, onde \mathbf{z} é a coordenada do centro do bloco. O classificador SVM, determinado no treinamento, será utilizado para classificar cada bloco $B_{\mathbf{z}}$. Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).

Figura 5.14: Diagrama em blocos para teste dos métodos DF-B, DF e SVM.

5.6 Influência da Quantidade de Componentes Principais

Nesta seção, apresentamos os resultados para a influência da quantidade de componentes principais, empregada no projeto do filtro discriminativo, no desempenho do método DF-B-PCA, para um subconjunto das bases de dados *BioID* e *Feret* com, respectivamente, 503 e 2004 imagens (Seção 5.3). Os resultados foram obtidos utilizando-se validação cruzada com 7 *folds*, onde o conjunto de treinamento corresponde a 6/7 do total de imagens e o conjunto de teste corresponde a 1/7 do total de imagens. Os procedimentos de treinamento foram desenvolvidos conforme descrição realizada na Seção 5.5.2. Os procedimentos para teste estão na Seção 5.5.3; neste caso, utilizamos o pós-processamento NA⁶ (veja a Tabela 5.2).

A disposição dos resultados é feita da seguinte maneira. Primeiramente, temos dois gráficos que indicam, usando barras, as percentagens médias de Verdadeiros Positivos (TP) e Falsos Positivos (FP). Nestes gráficos, a quantidade de componentes principais utilizada está localizada na horizontal. A pequena barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão. A seguir, apresentamos um terceiro grupo de gráficos. Nestes, as taxas de TP e FP, para cada componente, são representadas por coordenadas cuja localização está associada ao desempenho obtido. Dessa forma, podemos verificar qual componente proporciona o melhor desempenho. Observe que quanto mais acima e à esquerda a coordenada estiver melhor será o desempenho obtido.

Utilizamos $\alpha = 0, 1$ (Equação (5.4)), ou seja, consideramos que se acerta um ponto fiducial quando ele está a menos de 10% de \tilde{d}_o (distância inter-ocular da face) do seu rótulo manual. Observe que acrescentamos, em cada página, uma imagem de referência onde consta a localização dos pontos fiduciais.

Ao final desta seção, verificamos a análise dos resultados e uma síntese dos melhores resultados obtidos, formatados em uma tabela.

⁶Lembramos que o acrônimo NA indica que não será utilizada nenhuma estratégia para agrupamento. Dessa forma, todos os rótulos automáticos serão considerados como saída do método.

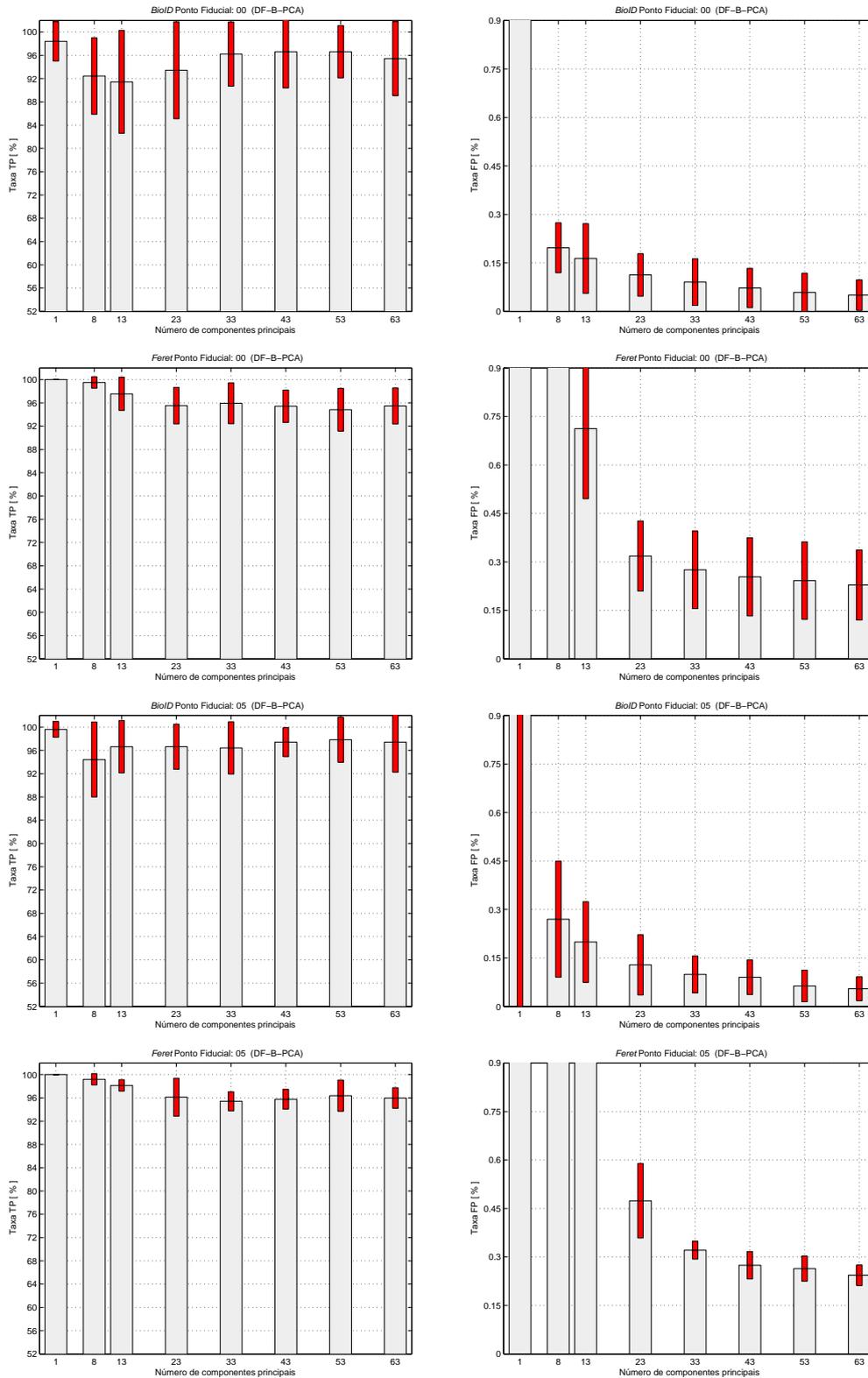


Figura 5.15: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 00 e 05. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

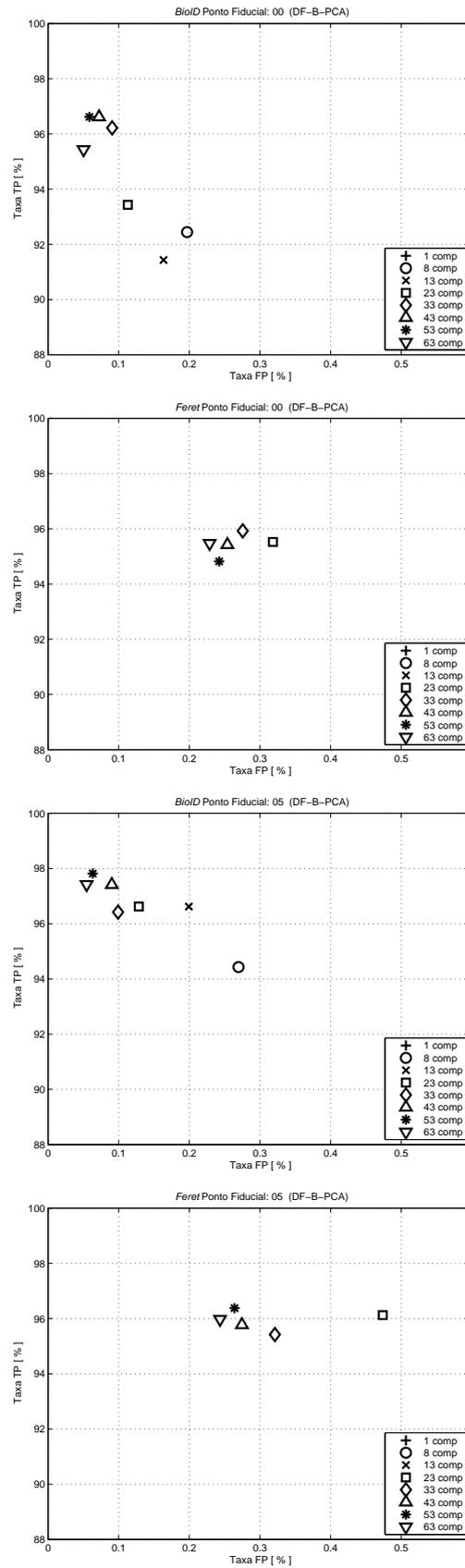


Figura 5.16: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 00 e 05. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

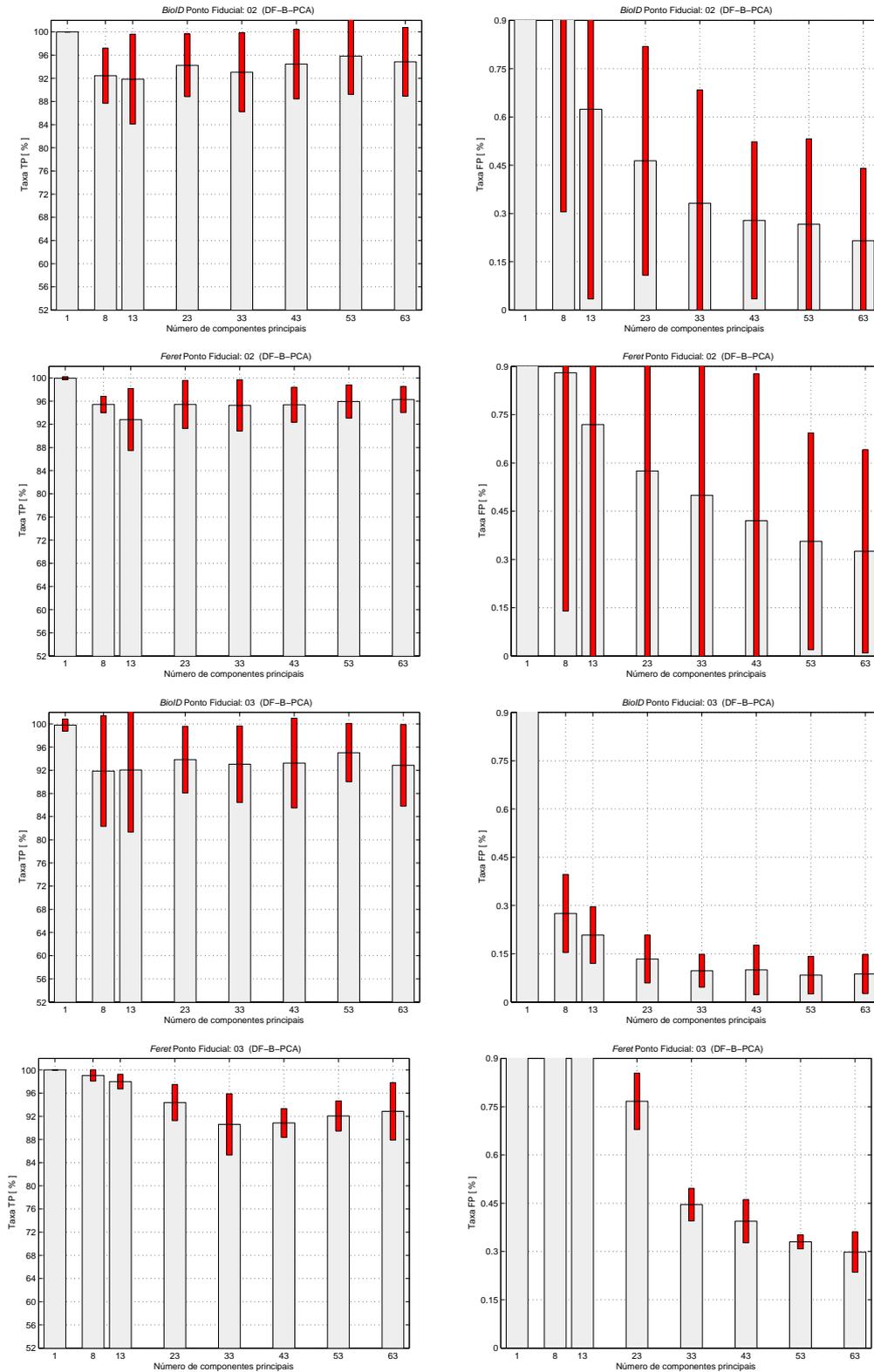
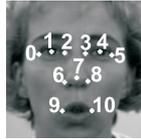


Figura 5.17: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 02 e 03. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

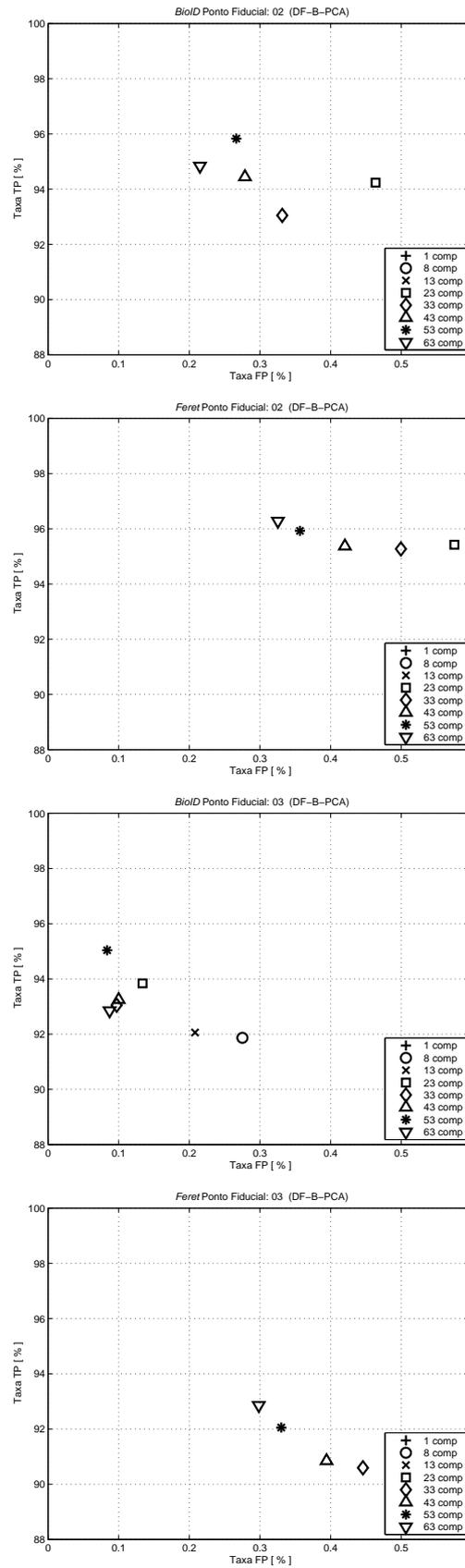


Figura 5.18: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 02 e 03. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

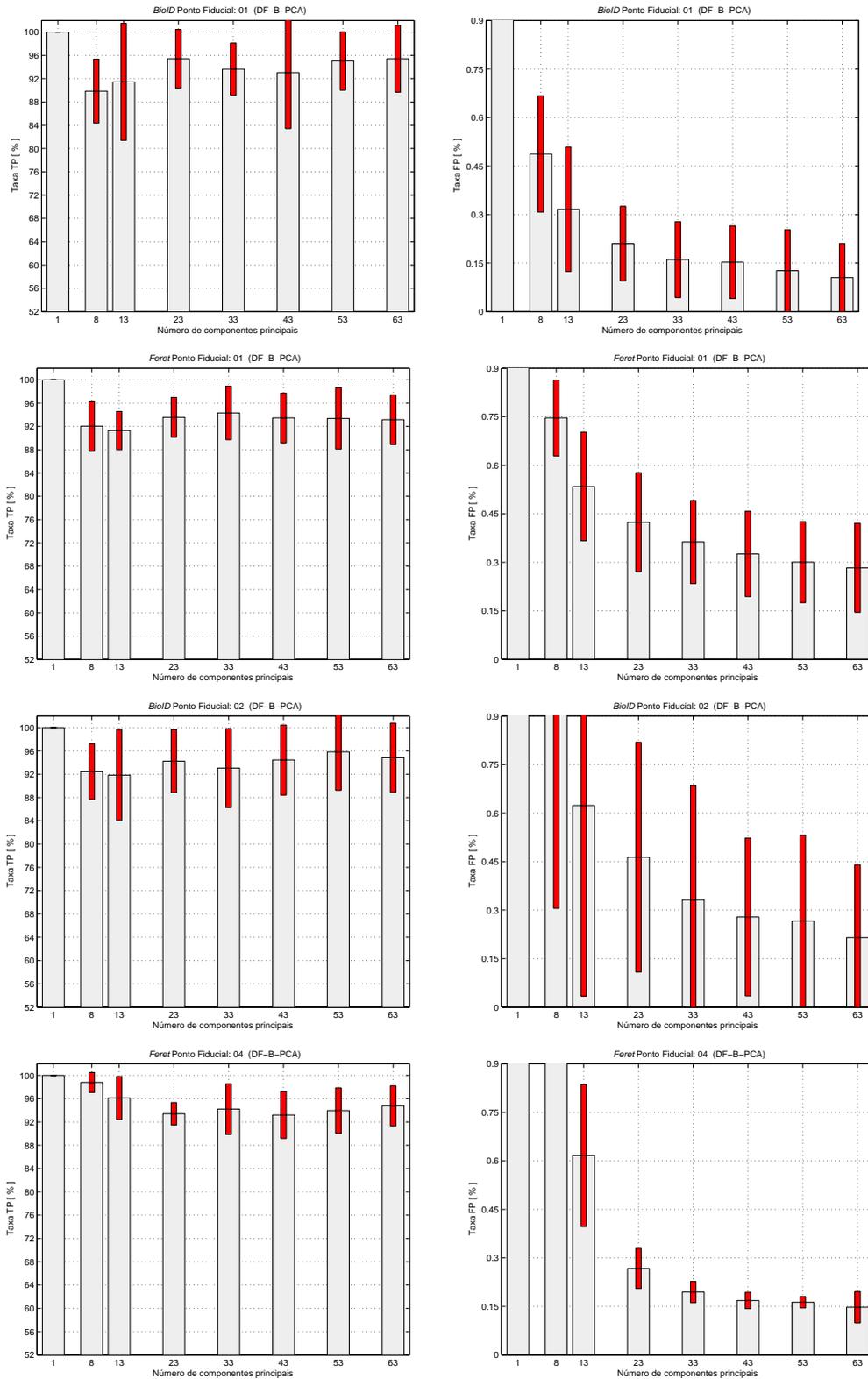
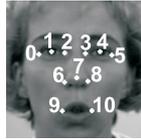


Figura 5.19: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 01 e 04. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

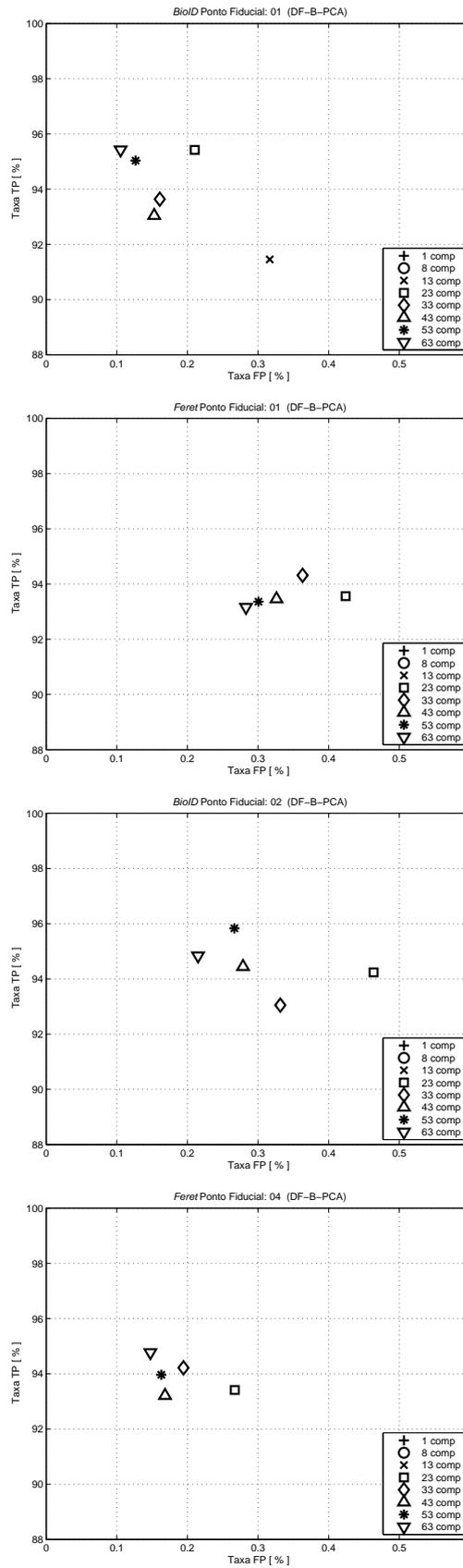


Figura 5.20: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 01 e 04. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

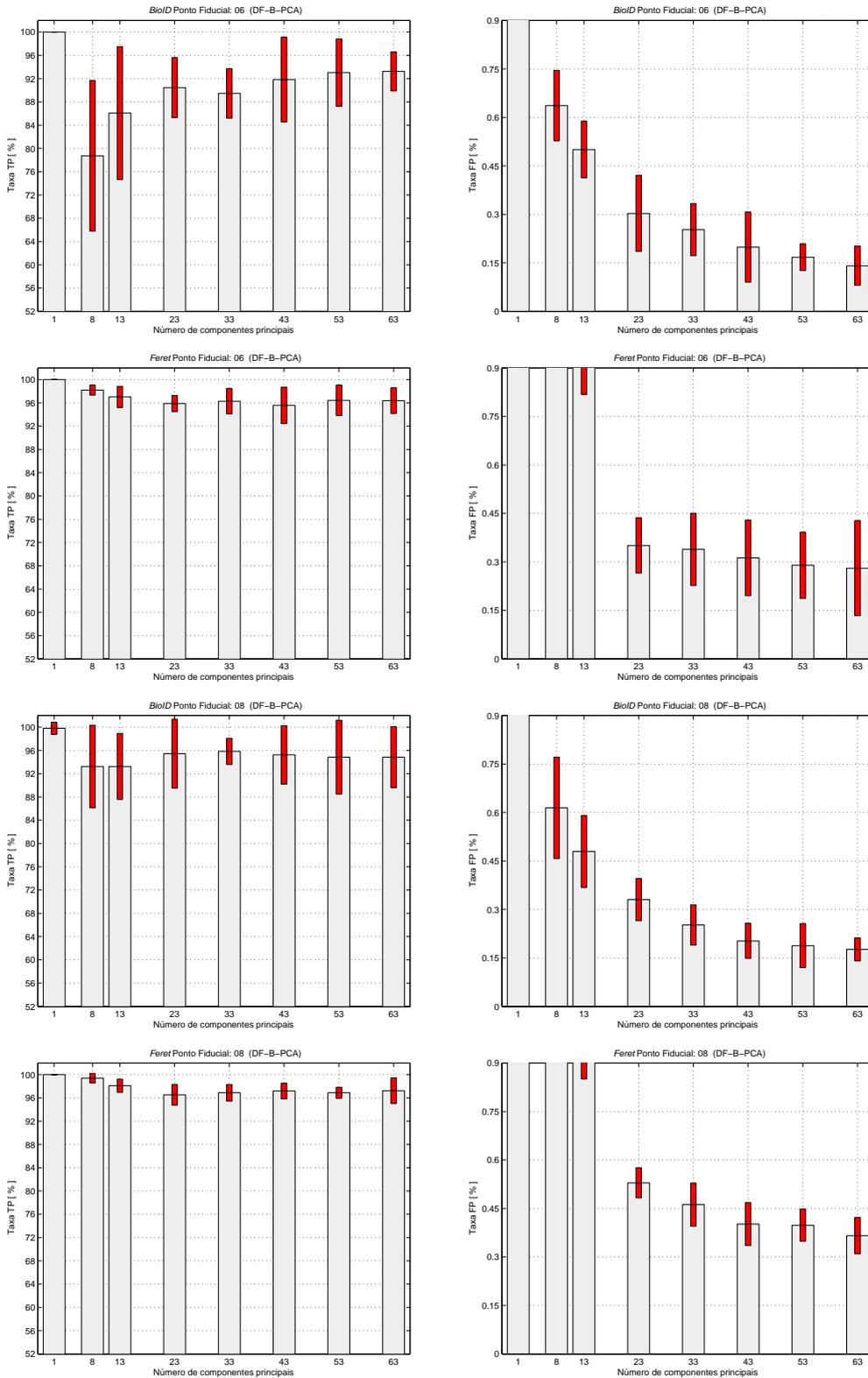


Figura 5.21: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 06 e 08. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

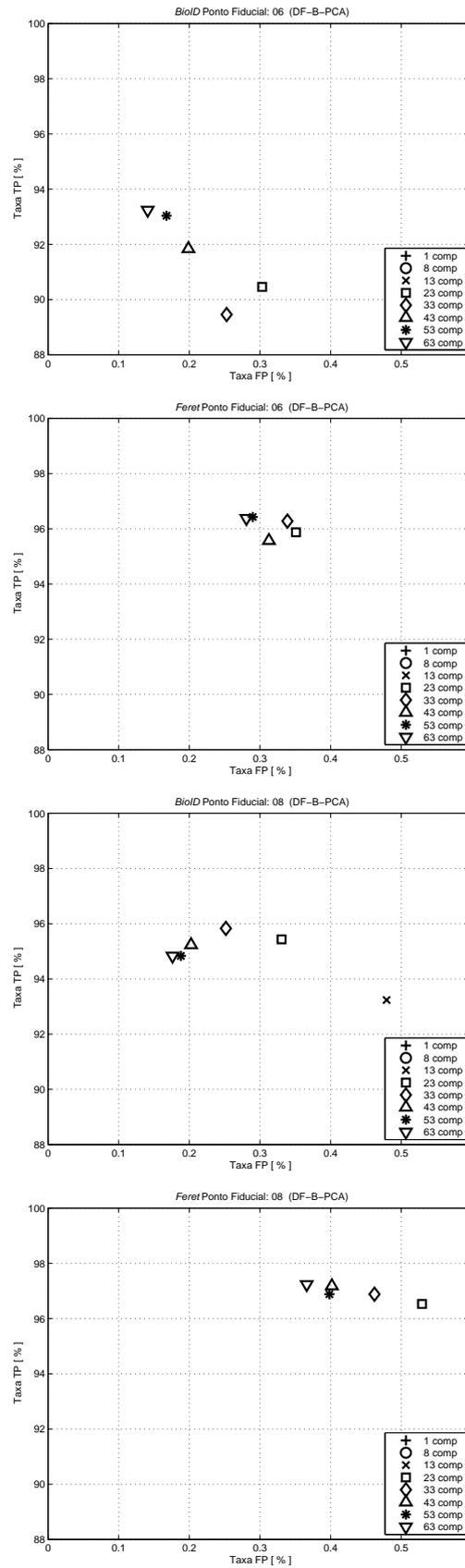


Figura 5.22: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 06 e 08. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

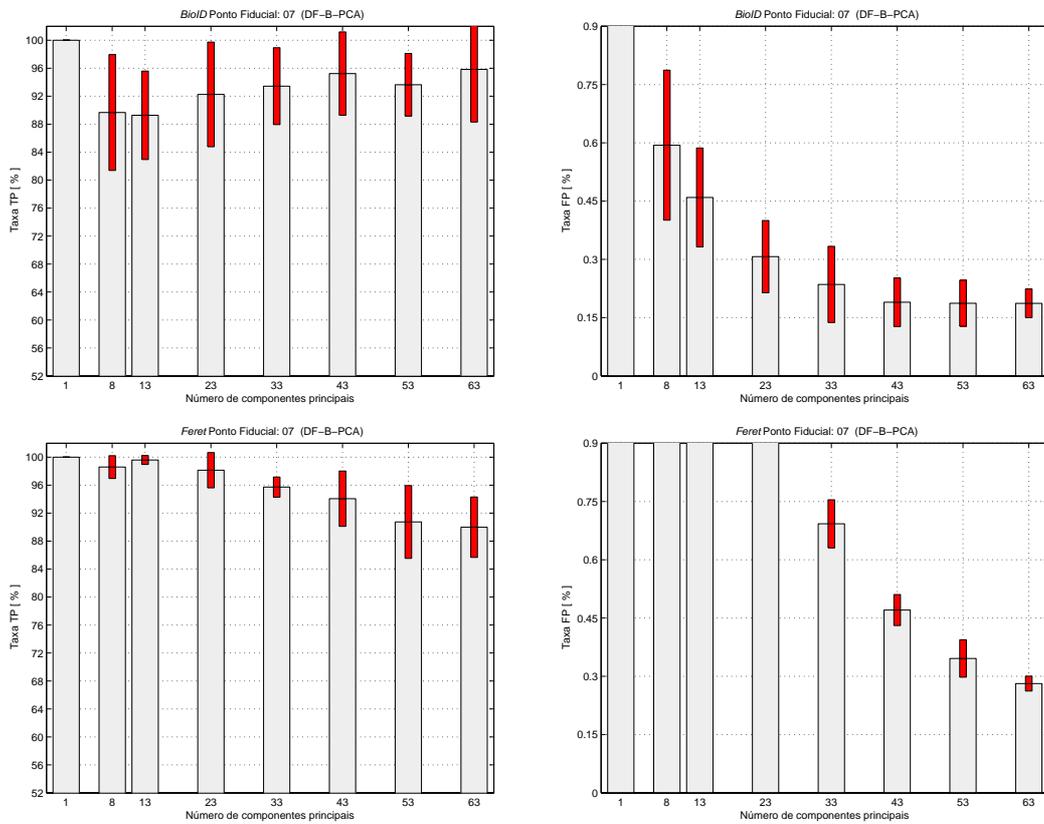


Figura 5.23: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para o ponto fiducial 07. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

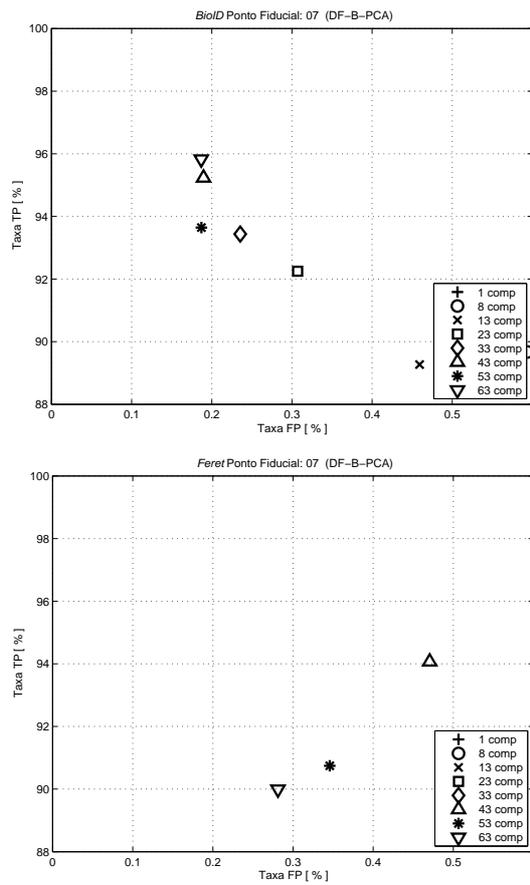


Figura 5.24: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para o ponto fiducial 07. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

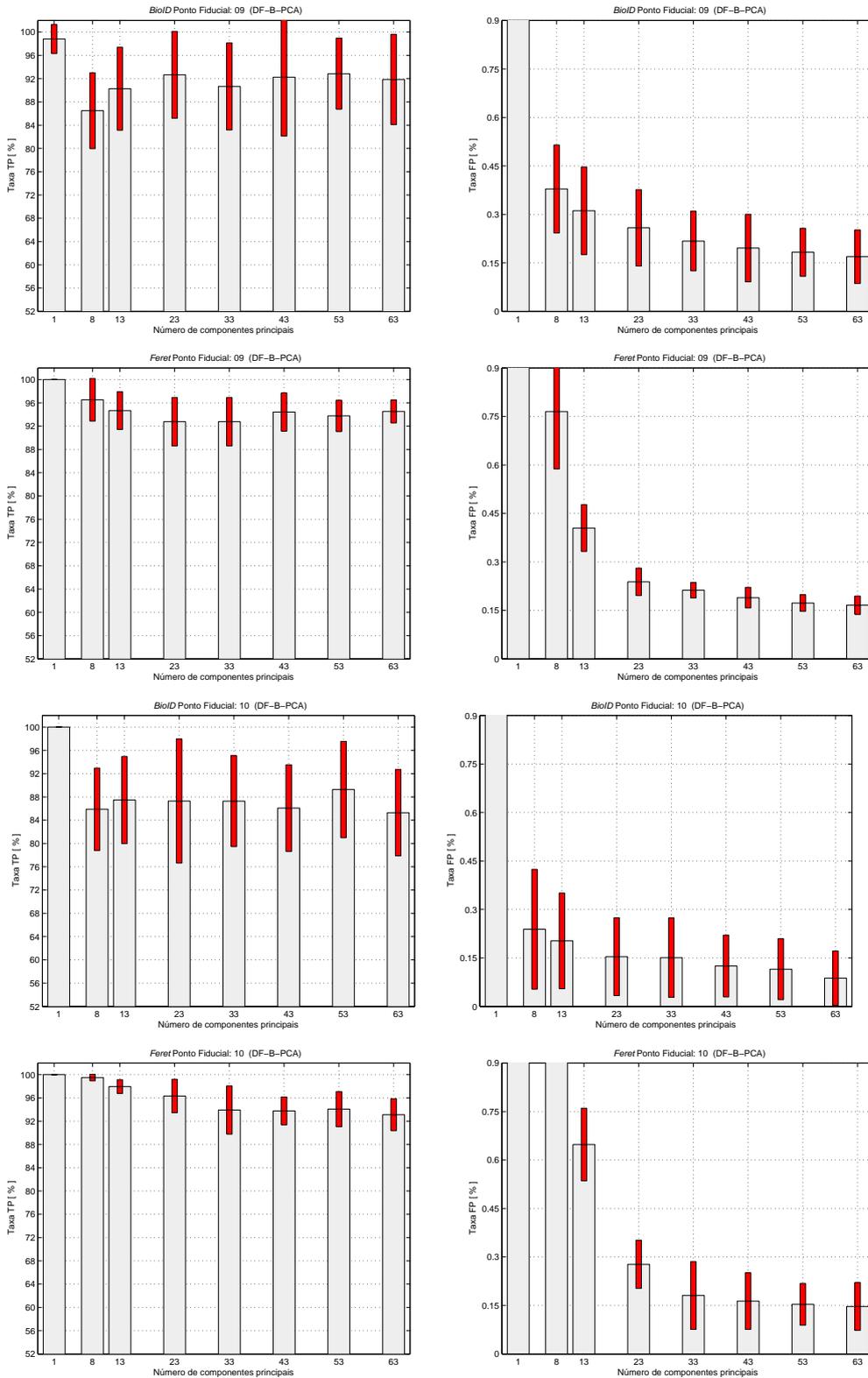


Figura 5.25: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 09 e 10. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

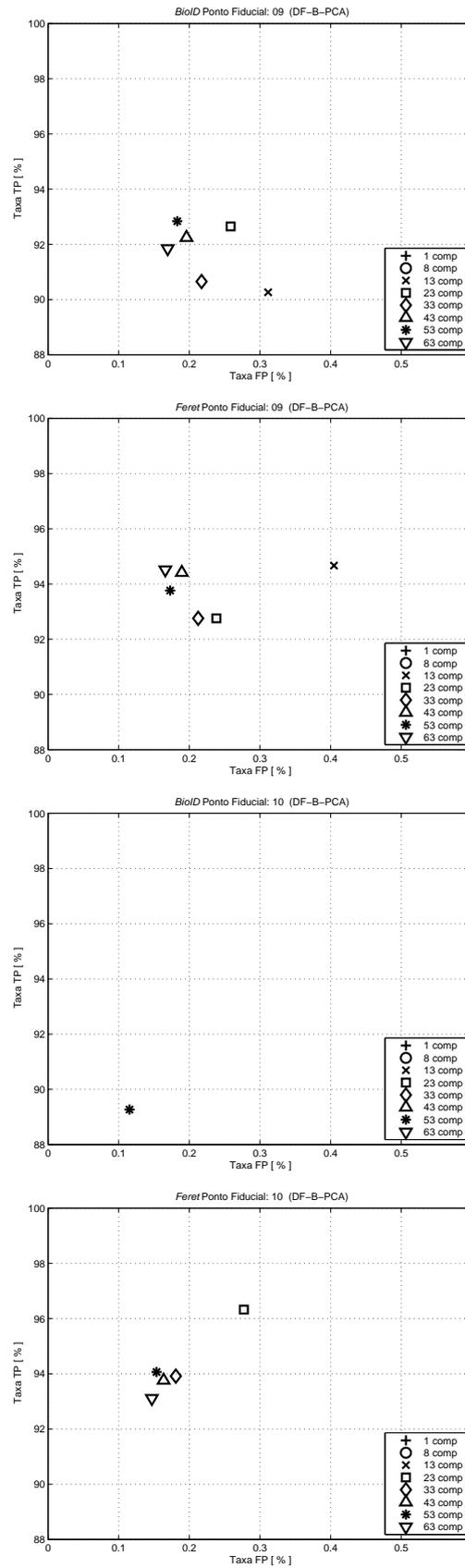


Figura 5.26: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 09 e 10. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

5.6.1 Análise dos Resultados

Nas Figuras 7.5 até 7.15 apresentamos os gráficos de TP (Verdadeiros Positivos) e FP (Falsos Positivos) para todos os pontos fiduciais das bases de dados *BioID* e *Feret*. Após analisarmos as figuras podemos verificar que:

1. Para as bases de dados *BioID* e *Feret* temos, em todos os pontos fiduciais, um considerável equilíbrio da taxa de TP quando o número de componentes k é superior a 23. Exceto para os pontos fiduciais 10 da base *BioID* e 07 da base *Feret*, percebemos que para k superior a 53 temos que a taxa de TP é superior a 92%.
2. Para as bases *BioID* e *Feret* temos, em todos os pontos fiduciais, uma redução brusca da taxa de FP quando o número de componentes k está entre 1 e 23.
3. Para as bases de dados *BioID* e *Feret* para k superior a 23 percebe-se uma tendência decrescente para a taxa de FP. Para k superior a 53 temos que a taxa de FP é inferior a 1,0%;
4. Na base *Feret*, o ponto fiducial 07 apresenta valores de taxa de FP com uma tendência decrescente. Porém, observe que a taxa de TP segue a mesma tendência.

As figuras evidenciam que a utilização das componentes principais proporciona um ganho significativo no desempenho⁷ do método DF-B-PCA. Por exemplo:

1. Na base *BioID* para o ponto fiducial 07 temos que a taxa de FP decresce de $\cong 0,6\%$ ($k = 8$) para $\cong 0,18\%$ ($k = 63$) o que equivale a uma redução de $\cong 70\%$ e a taxa de TP eleva-se de 90% para 96%; para o ponto fiducial 02 a taxa de FP decresce de $\cong 1,0\%$ ($k = 8$) para $\cong 0,26\%$ ($k = 63$), equivalente a uma redução de $\cong 75\%$ enquanto que a taxa de TP permanece em $\cong 92\%$.
2. Na base *Feret* para o ponto fiducial 00 temos que a taxa de FP decresce de $\cong 0,75\%$ ($k = 13$) para $\cong 0,26\%$ ($k = 63$) o que equivale a uma redução de $\cong 70\%$ e a taxa de TP se mantém em $\cong 95\%$.

⁷Perceba que idealmente, buscamos $FP = 0\%$ e $TP = 100\%$. Porém, na realidade é necessário um compromisso entre as taxas de FP e TP. Dessa forma, iremos considerar que o método possui resultados eficazes quando temos um valor baixo para a taxa de FP e um valor elevado para a taxa de TP. Note, porém, que é mais importante ter uma taxa de FP baixa do que uma taxa de TP alta.

É importante comentar que não existem evidências de que a taxa de FP decresça indefinidamente com o aumento do número de componentes. Para todos os pontos fiduciais, observe que com k superior a 43 temos que as taxas de FP praticamente estabilizam.

Os resultados obtidos para as bases de dados *BioID* e *Feret* reforçam as evidências de que:

1. A utilização das componentes principais proporciona um ganho significativo no desempenho do método DF-B-PCA.
2. O número de componentes principais é fator determinante na eficácia do método DF-B-PCA.
3. Não existem evidências de que a taxa de FP decresça indefinidamente.

Por fim, apresentamos uma síntese dos melhores resultados obtidos, em forma de tabela, em termos do número de componentes principais para todos os pontos fiduciais (observe a Tabela 5.3). Nesta, listamos as taxas médias de Verdadeiros Positivos (TP) e Falsos Positivos (FP) e os respectivos desvios padrões (σ) obtidos por validação cruzada com 7 *folds*.

Síntese dos Resultados para DF-B-PCA

Tabela 5.3: Bases de dados *BioID* e *Feret*: síntese dos melhores desempenhos, em termos da quantidade de componentes principais, para o método DF-B-PCA. As taxas médias de Verdadeiros Positivos (TP) e Falsos Positivos (FP) e os respectivos desvios padrões (σ) foram obtidos por validação cruzada com 7 *fold*s. O pós-processamento NA foi utilizado.

					
<i>BioID</i>					
PFs	qPC	DF-B-PCA			
		TP	σ	FP	σ
00	63	96,6	2,2	0,05	0,03
01	58	95,4	2,8	0,10	0,05
02	53	95,0	2,1	0,24	0,14
03	53	95,0	2,5	0,08	0,03
04	53	97,8	2,8	0,11	0,04
05	58	97,4	2,5	0,05	0,02
06	63	93,2	1,6	0,14	0,03
07	63	95,8	3,7	0,18	0,02
08	63	94,8	2,6	0,17	0,02
09	63	92,8	3,0	0,18	0,04
10	63	87,4	2,9	0,18	0,02
<i>Feret</i>					
PFs	qPC	DF-B-PCA			
		TP	σ	FP	σ
00	58	95,4	1,5	0,22	0,05
01	58	93,1	2,1	0,27	0,05
02	63	96,2	1,0	0,32	0,16
03	63	92,8	2,4	0,29	0,03
04	53	94,7	1,7	0,16	0,01
05	58	95,9	0,8	0,25	0,01
06	63	96,3	1,1	0,28	0,07
07	58	89,9	2,1	0,30	0,02
08	63	97,2	1,1	0,36	0,03
09	63	94,5	0,9	0,16	0,01
10	63	93,1	1,3	0,14	0,04

5.7 Comparação entre DF-B-PCA, DF e DF-B

Nesta seção, apresentamos uma comparação dos desempenhos para os métodos DF-B-PCA, DF-B e DF. Os procedimentos de treinamento dos métodos DF-B e DF estão descritos na Seção 5.5.2 e os procedimentos para teste estão na Seção 5.5.3. Os métodos são treinados para cada ponto fiducial de cada base de dados. Utilizamos o pós-processamento NA, dessa forma, todos os rótulos automáticos serão considerados (veja a Tabela 5.2) e a valor de α é igual a 0, 1. (Equação (5.4)). Nesta comparação, listamos as taxas médias de Verdadeiros Positivos (TP) e Falsos Positivos (FP) e os respectivos desvios padrões (σ) obtidos por validação cruzada com 7 *folds*. Para que a comparação seja efetiva precisamos considerar alguns critérios:

1. Valores elevados de taxa de FP indicam que o método não possui um desempenho satisfatório. Obviamente, valores reduzidos de taxa de FP indicam que o método é eficaz. Na prática, valores de taxa de FP inferiores a 1,5% serão considerados aceitáveis.
2. Valores reduzidos de taxa de TP indicam que o método não possui um desempenho satisfatório. Na prática, valores de taxa de TP superiores a 80% podem ser considerados satisfatórios.
3. Supondo valores de taxa de FP reduzidos e similares, observa-se a taxa de TP obtida. Caso a taxa de TP seja superior então podemos inferir que um método supera o outro (obviamente $TP > 80\%$).

Os resultados para o método DF-B-PCA, apresentados na Tabela 5.3, foram acrescentados na Tabela 5.4. Ao analisarmos os resultados apresentados na Tabela 5.4 podemos verificar que:

1. Para as bases de dados *BioID* e *Feret* o método DF-B-PCA possui taxas de FP abaixo da taxa aceitável para todos os pontos fiduciais. As taxas de TP superam 80% em todos os casos.
2. Para as bases *BioID* e *Feret* o método DF-B apresenta taxas de FP elevadas para todos os pontos fiduciais. Elas estão muito acima de 1,5% (valor considerado aceitável). Sendo assim, podemos inferir que este método não é eficaz.

3. Para as bases *BioID* e *Feret* o método DF possui taxas de FP bem abaixo da taxa aceitável para todos os pontos fiduciais. Porém, as taxas de TP não atingem valores satisfatórios tornando o método ineficaz.

A Tabela 5.4 evidencia que o método DF-B-PCA proporciona um ganho significativo em relação aos métodos DF-B e DF para todos os pontos fiduciais. Perceba que o método DF-B-PCA proporciona um bom compromisso entre taxas de TP e FP, ou seja, temos altas taxas de TP e baixas taxas de FP. Note que os métodos DF-B-PCA, DF-B e DF apresentam um comportamento de resultados equivalente para as duas bases de dados (*BioID* e *Feret*).

Tabela 5.4: Bases de dados *BioID* e *Feret*: comparação entre os métodos DF-B-PCA, DF e DF-B para os pontos fiduciais apresentados na face localizada na parte superior da tabela. A denominação NA indica a estratégia de agrupamento utilizada (sem agrupamento).

												
<i>BioID</i>												
PFs	DF-B-PCA-NA (proposto)				DF-NA				DF-B-NA			
	TP	σ	FP	σ	TP	σ	FP	σ	TP	σ	FP	σ
00	96,6	2,2	0,05	0,03	21,1	4,7	0,22	0,03	100,0	0,0	21,64	6,2
01	95,4	2,8	0,10	0,05	59,2	6,8	0,30	0,04	100,0	0,0	7,27	2,3
02	95,0	2,1	0,24	0,14	53,7	6,1	0,39	0,03	100,0	0,0	20,25	1,6
03	95,0	2,5	0,08	0,03	65,6	13,7	0,30	0,03	100,0	0,0	8,41	0,8
04	97,8	2,8	0,11	0,04	65,0	9,6	0,40	0,02	99,8	0,5	5,30	1,3
05	97,4	2,5	0,05	0,02	70,3	12,4	0,23	0,04	99,6	1,1	6,99	4,5
06	93,2	1,6	0,14	0,03	56,7	9,3	0,25	0,04	99,8	0,5	9,59	1,6
07	95,8	3,7	0,18	0,02	54,1	8,1	0,20	0,01	99,8	0,5	5,66	1,7
08	94,8	2,6	0,17	0,02	54,9	5,0	0,23	0,01	99,8	0,5	5,49	0,5
09	92,8	3,0	0,18	0,04	43,5	5,5	0,22	0,01	99,8	0,5	14,44	4,2
10	87,4	2,9	0,18	0,02	22,1	6,2	0,11	0,04	100,0	0,0	15,93	2,5
<i>Feret</i>												
PFs	DF-B-PCA-NA (proposto)				DF-NA				DF-B-NA			
	TP	σ	FP	σ	TP	σ	FP	σ	TP	σ	FP	σ
00	95,4	1,5	0,22	0,05	35,1	6,5	0,18	0,04	100,0	0,0	20,09	8,08
01	93,1	2,1	0,27	0,05	15,8	1,8	0,31	0,06	100,0	0,0	39,07	1,39
02	96,2	1,0	0,32	0,16	32,2	11,3	0,22	0,09	99,9	0,1	19,21	4,22
03	92,8	2,4	0,29	0,03	64,9	3,7	0,52	0,01	100,0	0,0	19,21	2,51
04	94,7	1,7	0,16	0,01	25,7	3,0	0,40	0,01	100,0	0,0	27,14	1,84
05	95,9	0,8	0,25	0,01	33,0	2,8	0,26	0,00	100,0	0,0	23,83	0,83
06	96,3	1,1	0,28	0,07	74,1	2,5	0,14	0,04	99,9	0,1	12,37	0,75
07	89,9	2,1	0,30	0,02	42,6	3,9	0,22	0,01	100,0	0,0	23,71	1,69
08	97,2	1,1	0,36	0,03	77,8	2,2	0,15	0,01	100,0	0,0	18,76	0,72
09	94,5	0,9	0,16	0,01	18,4	1,6	0,12	0,00	100,0	0,0	28,46	3,60
10	93,1	1,3	0,14	0,04	9,8	4,0	0,07	0,02	100,0	0,0	15,78	3,23

5.8 Comparação entre DF-B-PCA e SVM

Nesta seção, temos uma comparação dos resultados para o método DF-B-PCA e para os métodos por SVM (linear e *kernel* polinomial). O SVM é treinado para cada ponto fiducial das duas bases de dados. Utilizamos o pós-processamento NA (veja a Tabela 5.2) e consideramos $\alpha = 0,1$ (Equação (5.4)). Os procedimentos para treino e teste do método SVM estão descritos, respectivamente, nas Seções 5.5.2 e 5.5.3. Nesta comparação, apresentamos as taxas médias de Verdadeiros Positivos (TP) e Falsos Positivos (FP) e os respectivos desvios padrões obtidos por validação cruzada com 7 *folds*.

Considerando os critérios estabelecidos na Seção 5.7 e analisando os resultados apresentados para Tabela 5.5 podemos verificar que:

1. Para a base *BioID* o método SVM linear apresenta taxas de FP próximas de 1,5% para seis pontos fiduciais, ou seja, 55% dos pontos fiduciais estão com taxas de FP acima do valor aceitável. Nestes casos, estes valores não ultrapassam 3 %. As taxas de TP para os casos em que $FP < 1,5\%$ ficam acima de 92% exceto para o ponto fiducial 07. Dessa forma, podemos inferir que este método é eficaz em um total de três pontos fiduciais (00, 01 e 08).
2. A diversidade da base *Feret* piorou o desempenho do método SVM linear. Em alguns casos, a taxa de FP foi bem superior a 1,5%. Observe que os valores de desvio padrão das taxas de TP estão acima de 12% (exceto para o ponto fiducial 03 onde $\sigma = 9,7\%$).
3. Para as bases de dados *BioID* e *Feret* o método SVM com *kernel* polinomial apresenta taxas de FP abaixo de 1,5% para quase todos os pontos fiduciais. As taxas de TP estão acima de 90%, exceto para o ponto fiducial 07 da base *Feret*. Note que o ponto fiducial 07 da base *Feret* apresentou uma baixa taxa de TP e o ponto fiducial 06 apresentou um alto valor de desvio padrão para a taxa de TP.

Por fim, após os comentários acima referentes aos resultados apresentados na Tabela 5.5 podemos concluir que o método SVM com *kernel* polinomial possui excelentes resultados, ao contrário do desempenho para o método SVM linear que não é satisfatório. Ao confrontarmos os resultados obtidos para o método DB-B-PCA com os do SVM-Pol percebemos que ambos possuem um bom compromisso entre taxas de TP e FP. Perceba que as taxas de TP no SVM-Pol são superiores ao passo que as taxas de FP no DF-B-PCA são inferiores.

Tabela 5.5: Bases de dados *BioID* e *Feret*: comparação entre os métodos DF-B-PCA, SVM linear e SVM com *kernel* polinomial.

												
<i>BioID</i>												
PFs	DF-B-PCA-NA (proposto)				SVM-L-NA				SVM-Pol-NA			
	TP	σ	FP	σ	TP	σ	FP	σ	TP	σ	FP	σ
00	96,6	2,2	0,05	0,03	92,4	9,8	1,09	0,74	99,6	0,7	0,39	0,07
01	95,4	2,8	0,10	0,05	99,6	1,0	0,32	0,09	98,8	1,2	0,04	0,02
02	95,0	2,1	0,24	0,14	93,0	14,3	1,58	0,59	94,2	2,7	0,15	0,03
03	95,0	2,5	0,08	0,03	96,0	1,7	0,46	0,21	99,8	0,5	0,33	0,09
04	97,8	2,8	0,11	0,04	97,2	4,2	2,49	4,76	98,8	1,5	0,08	0,05
05	97,4	2,5	0,05	0,02	98,8	3,1	2,27	0,62	99,6	0,7	0,42	0,09
06	93,2	1,6	0,14	0,03	99,6	1,0	1,40	1,44	99,6	0,7	0,31	0,05
07	95,8	3,7	0,18	0,02	47,1	9,6	0,06	0,03	99,6	0,7	0,60	0,05
08	94,8	2,6	0,17	0,02	95,4	3,9	0,40	0,09	99,0	1,1	0,24	0,03
09	92,8	3,0	0,18	0,04	96,8	5,5	2,91	5,12	98,8	1,7	0,39	0,06
10	87,4	2,9	0,18	0,02	98,6	1,9	1,47	0,49	99,6	0,7	0,35	0,04
<i>Feret</i>												
PFs	DF-B-PCA-NA (proposto)				SVM-L-NA				SVM-Pol-NA			
	TP	σ	FP	σ	TP	σ	FP	σ	TP	σ	FP	σ
00	95,4	1,5	0,22	0,05	91,1	12,4	3,00	2,04	91,9	6,7	0,26	0,16
01	93,1	2,1	0,27	0,05	80,0	15,7	1,85	3,30	99,1	0,8	0,17	0,03
02	96,2	1,0	0,32	0,16	89,8	25,3	19,51	15,60	97,1	7,0	0,38	0,11
03	92,8	2,4	0,29	0,03	93,6	9,7	4,82	5,47	98,8	1,2	0,22	0,09
04	94,7	1,7	0,16	0,01	63,7	39,6	0,43	0,36	99,4	0,5	0,18	0,02
05	95,9	0,8	0,25	0,01	73,3	27,6	0,84	0,61	94,8	6,1	0,30	0,18
06	96,3	1,1	0,28	0,07	73,9	33,7	2,06	3,32	91,8	16,8	0,35	0,39
07	89,9	2,1	0,30	0,02	80,7	13,0	1,16	0,62	68,4	33,4	13,64	23,08
08	97,2	1,1	0,36	0,03	86,2	16,3	2,13	2,42	93,8	7,6	0,94	1,16
09	94,5	0,9	0,16	0,01	84,7	17,4	1,57	1,43	90,7	6,4	0,12	0,11
10	93,1	1,3	0,14	0,04	54,4	38,0	0,76	1,33	93,1	9,3	0,19	0,15

5.9 Estratégias para Pós-processamento para o Método DF-B-PCA

Conforme mencionado, podemos aplicar estratégias para pós-processamento dos métodos (observe a Figura 5.12 e a descrição de cada estratégia na Tabela 5.2). Por comodidade, lembramos que as estratégias ML, GML e A têm por objetivo selecionar somente um rótulo automático por imagem (consulte o rodapé ⁽⁸⁾ para detalhes).

Nas Figuras 5.27 até 5.32, apresentamos os resultados obtidos para o método DF-B-PCA utilizando as bases de dados *BioID* e *Feret*, utilizando estratégias de pós-processamento ML, GML e A. A disposição dos resultados será feita utilizando-se um gráfico por ponto fiducial. Cada gráfico será composto por três curvas relacionadas à estratégia utilizada, sendo que cada uma corresponde à distribuição de acerto média acumulada *versus* distância inter-ocular (note que, nestes casos, não faz sentido computarmos falsos positivos). A linha vertical pontilhada, exposta em cada gráfico, auxilia a visualização da distância inter-ocular igual a 10% ($\alpha = 0,1$).

Os gráficos foram alinhados, para facilitar a comparação dos resultados, levando-se em consideração a simetria da face em cada base de dados. Por exemplo, observe a disposição dos gráficos dos pontos fiduciais 01 e seu simétrico 04, note que os gráficos obtidos para a base *BioID* estão acima dos obtidos para a *Feret*. Após analisarmos os resultados podemos verificar que:

1. Naturalmente, as estratégias ML, GML e A proporcionam resultados inferiores aos da NA. É importante lembrar que na estratégia NA não existe a preocupação de selecionarmos apenas um rótulo automático por imagem.
2. Para as bases de dados *BioID* e *Feret*, em todos os pontos fiduciais, observamos que as estratégias GML e ML proporcionam resultados equivalentes.
3. O uso da estratégia A não produz resultados satisfatórios.

⁸Nas estratégias ML, GML e A iremos obter somente um rótulo automático como saída do método. Na ML, iremos utilizar o rótulo automático mais provável, supondo o modelo gaussiano a priori. Na GML, agrupamos os rótulos automáticos com distância menor que 8 pixels e a seguir escolhemos o rótulo automático mais provável (supondo o modelo gaussiano a priori). Na estratégia A, o rótulo automático será igual à média de todos os rótulos automáticos.

4. Perceba que, de maneira geral, as curvas para pontos fiduciais simétricos possuem um comportamento equivalente.
5. Para a base *BioID*, nos pontos fiduciais localizados na região dos olhos, podemos observar que as taxas de acerto são superiores a 92% e para os pontos fiduciais localizados nas regiões da boca e do nariz as taxas são superiores a 82%, exceto para o ponto fiducial 10 (observe a interseção entre as curvas e a linha vertical pontilhada que representa 10% da distância inter-ocular).
6. Para a base *Feret*, nos pontos fiduciais localizados nos centros e nos cantos internos dos olhos, podemos observar que as taxas ficam acima de 90% e nos cantos externos as taxas estão acima de 85%. Para os pontos nos cantos do nariz as taxas ficam acima de 90%. Para os pontos localizados no centro do nariz e nos cantos da boca as taxas de acerto ficam por volta de 80%.

Podemos concluir que os resultados apresentados nas Figuras 5.27 e 5.32 demonstram que tanto a estratégia ML quanto a GML podem ser utilizadas no método DF-B-PCA para seleção de apenas um rótulo automático com resultados satisfatórios. Por fim, assim como para o método DF-B-PCA, utilizamos as estratégias ML, GML e A nos métodos SVM linear e SVM com *kernel* polinomial. Para o SVM linear obtivemos resultados superiores com a estratégia ML e para o SVM-Pol obtivemos resultados similares com as estratégias GML e ML. Em ambos, o uso da estratégia A não produz resultados satisfatórios.

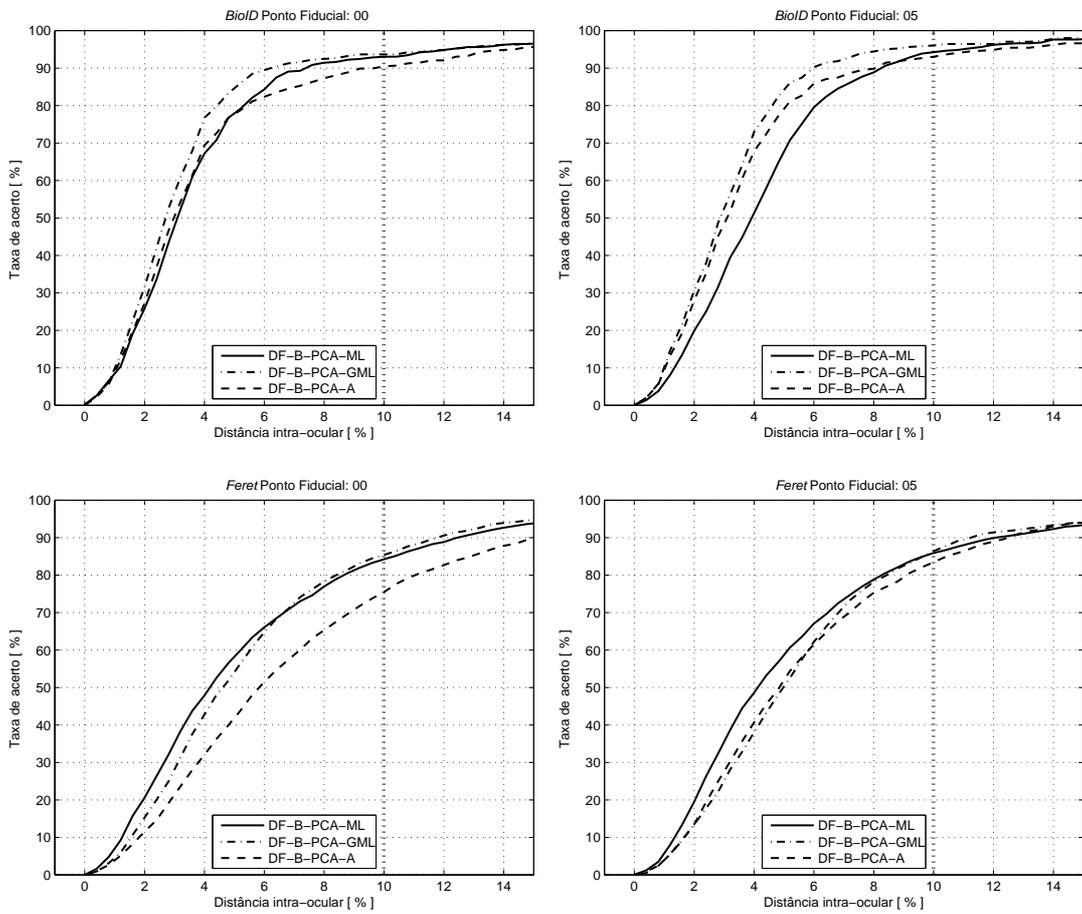


Figura 5.27: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 00 e 05.

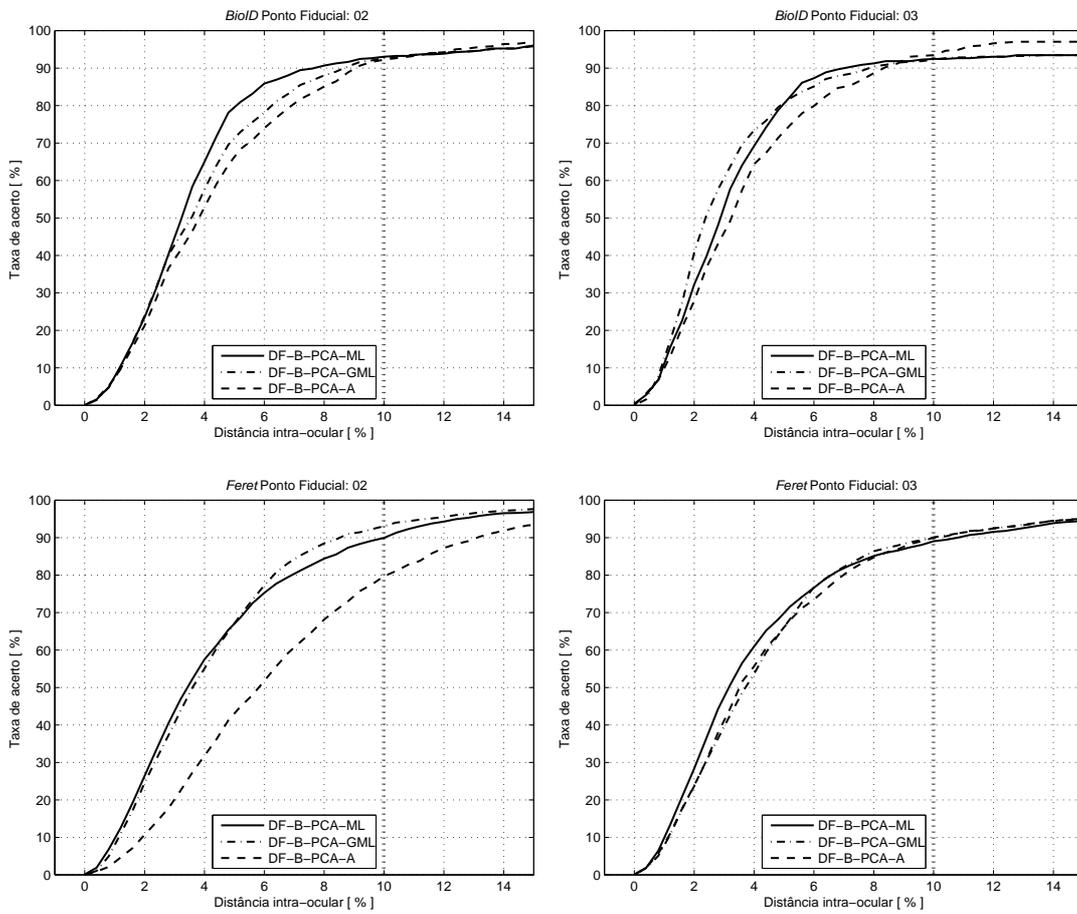


Figura 5.28: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 02 e 03.

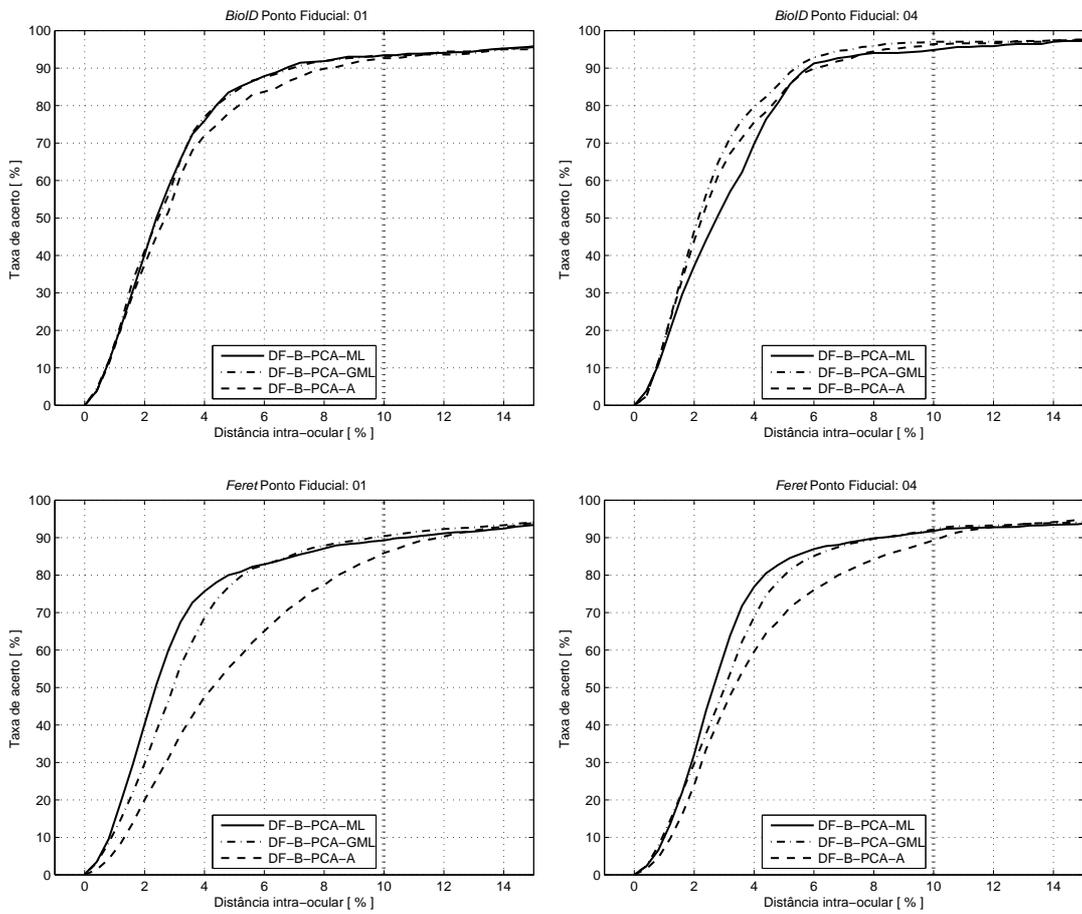
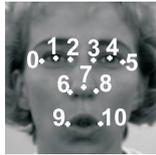


Figura 5.29: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 01 e 04.

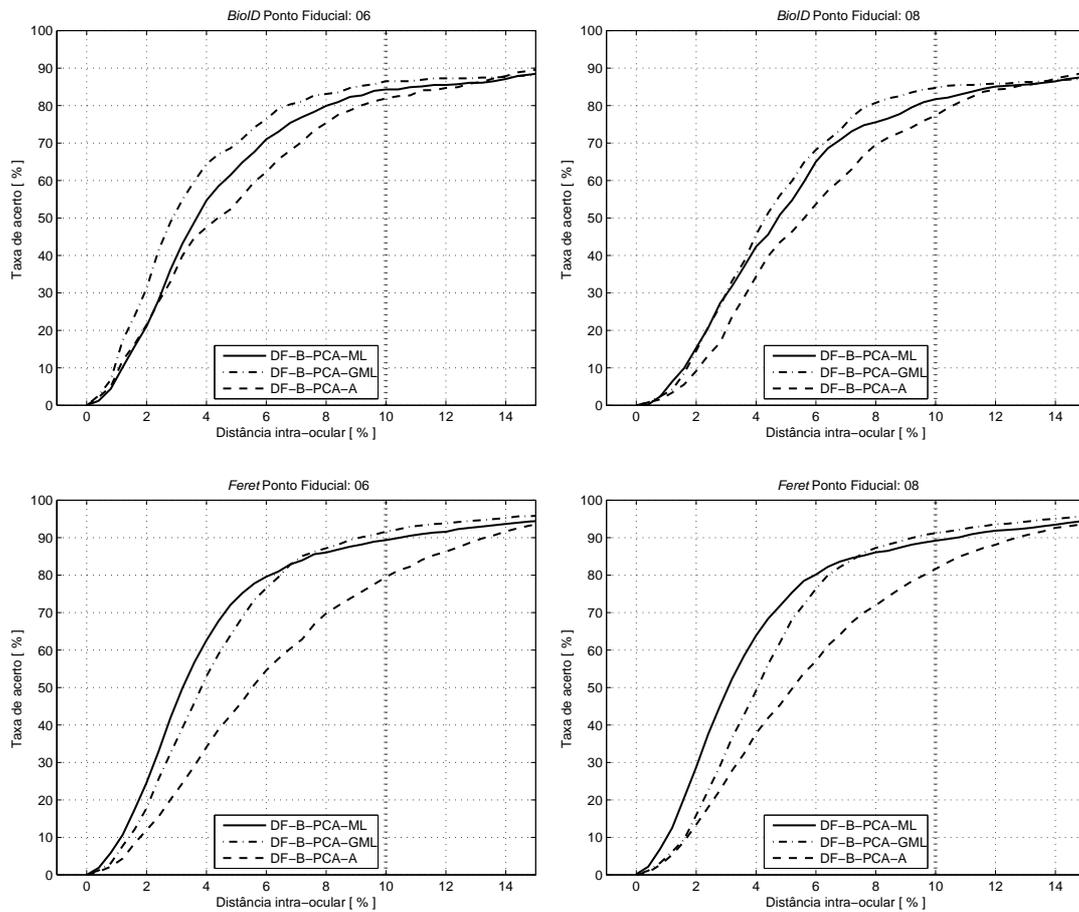


Figura 5.30: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 06 e 08.

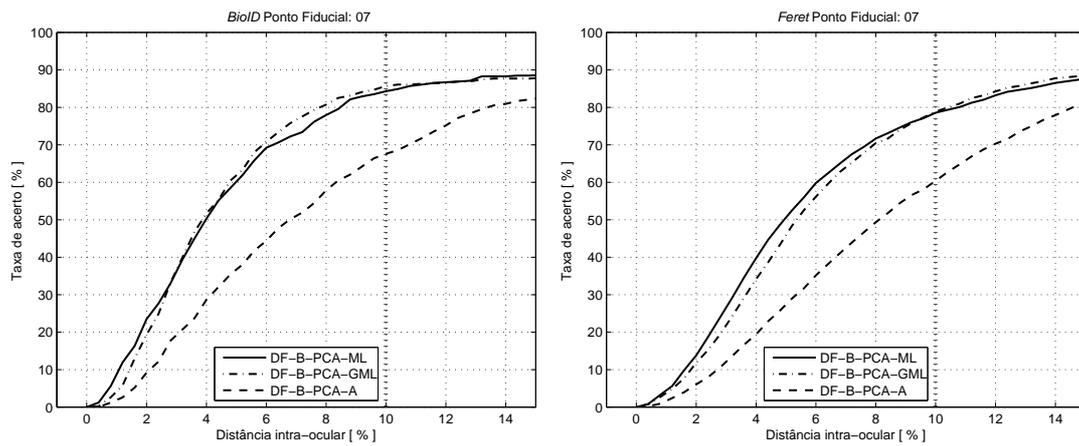


Figura 5.31: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para o ponto fiducial 07.

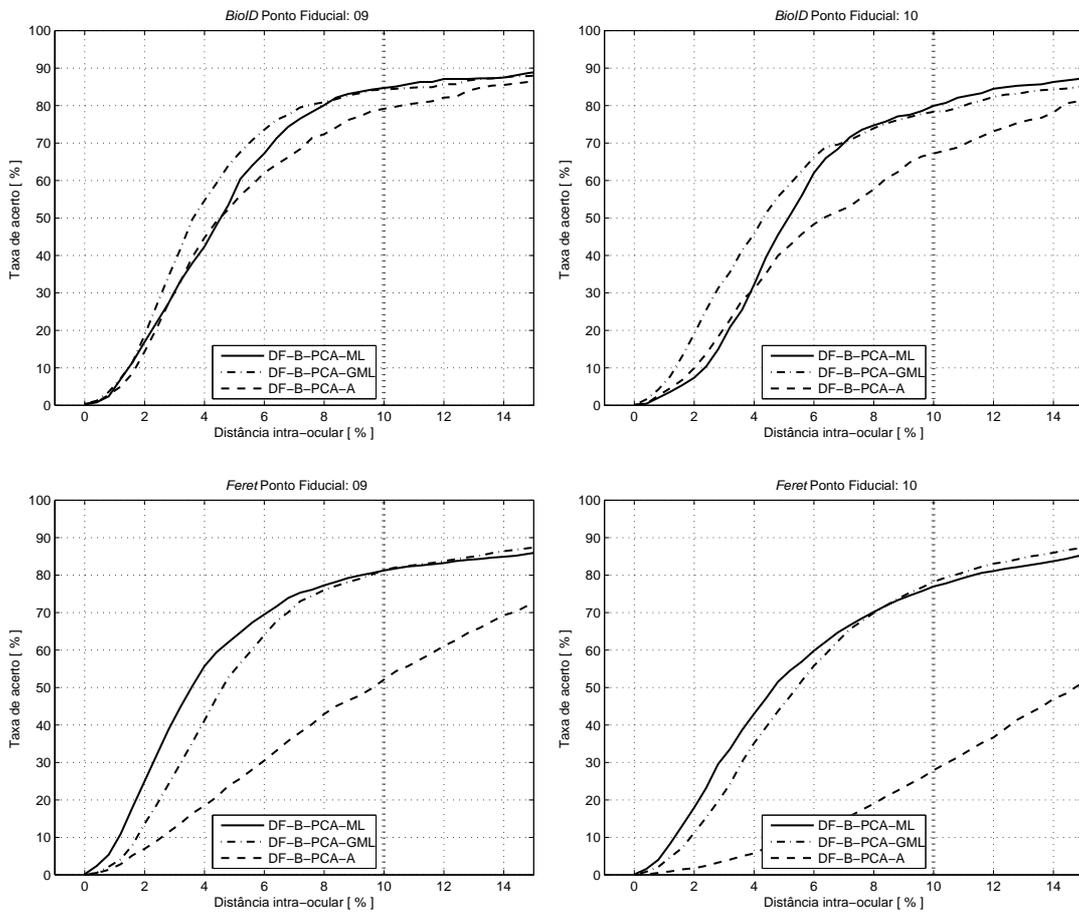


Figura 5.32: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 09 e 10.

5.10 Estratégias para Pós-processamento: Comparação entre DF-B-PCA e SVM

Nesta seção, utilizaremos as estratégias para seleção de somente um rótulo automático para compararmos o desempenho dos métodos DF-B-PCA, SVM linear e SVM com *kernel* polinomial. A estratégia GML foi utilizada para os métodos DF-B-PCA e SVM-Pol e para o método SVM linear utilizamos a estratégia ML.

Os gráficos, apresentados nas Figuras 5.33 até 5.38, estão dispostos no mesmo formato apresentado na Seção 5.9. A única diferença está na correspondência entre as curvas de cada gráfico. Observe que cada uma especificará um método (veja as legendas dos gráficos). Após análise dos gráficos podemos verificar que:

1. Para as bases de dados *BioID* e *Feret*, em quase todos os pontos fiduciais temos que as taxas de acerto para o método SVM com *kernel* polinomial são superiores às do SVM linear. Apenas no ponto fiducial 08 da base *BioID* o SVM linear é ligeiramente superior ao SVM-Pol.
2. Em termos de taxas de acerto, para a base *BioID* o método SVM-Pol proporciona, para os pontos fiduciais localizados na região dos olhos, taxas de acerto superiores a 90%. Observe que as taxas para os pontos correspondentes aos centros dos olhos estão acima de 95%. Para os pontos fiduciais localizados nas regiões da boca e do nariz as taxas são superiores a 82%. Para a base *Feret* o método SVM-Pol proporciona, para os pontos fiduciais localizados na região dos olhos, taxas de acerto superiores a 85%. Note que as taxas para os pontos localizados nos centros e nos cantos internos dos olhos estão acima de 95%. Para os pontos fiduciais nos cantos do nariz as taxas ficam pouco abaixo de 90% e nos cantos da boca as taxas ficam pouco acima de 80%. Para o ponto fiducial 07 o SVM-Pol não apresenta um resultado satisfatório.

Os gráficos demonstram que o método proposto (DF-B-PCA) possui resultados competitivos quando comparados com os do método SVM com *kernel* polinomial. Para a base de dados *BioID* observe que, em três pontos fiduciais (00, 05 e 07), os resultados para o método DF-B-PCA são superiores aos do SVM-Pol. Para a base *Feret* note que as taxas obtidas para o método DF-B-PCA nos pontos fiduciais 00 e 05 mantêm os bons resultados e nos pontos fiduciais 06, 07 e 08 os resultados são superiores aos do SVM-Pol.

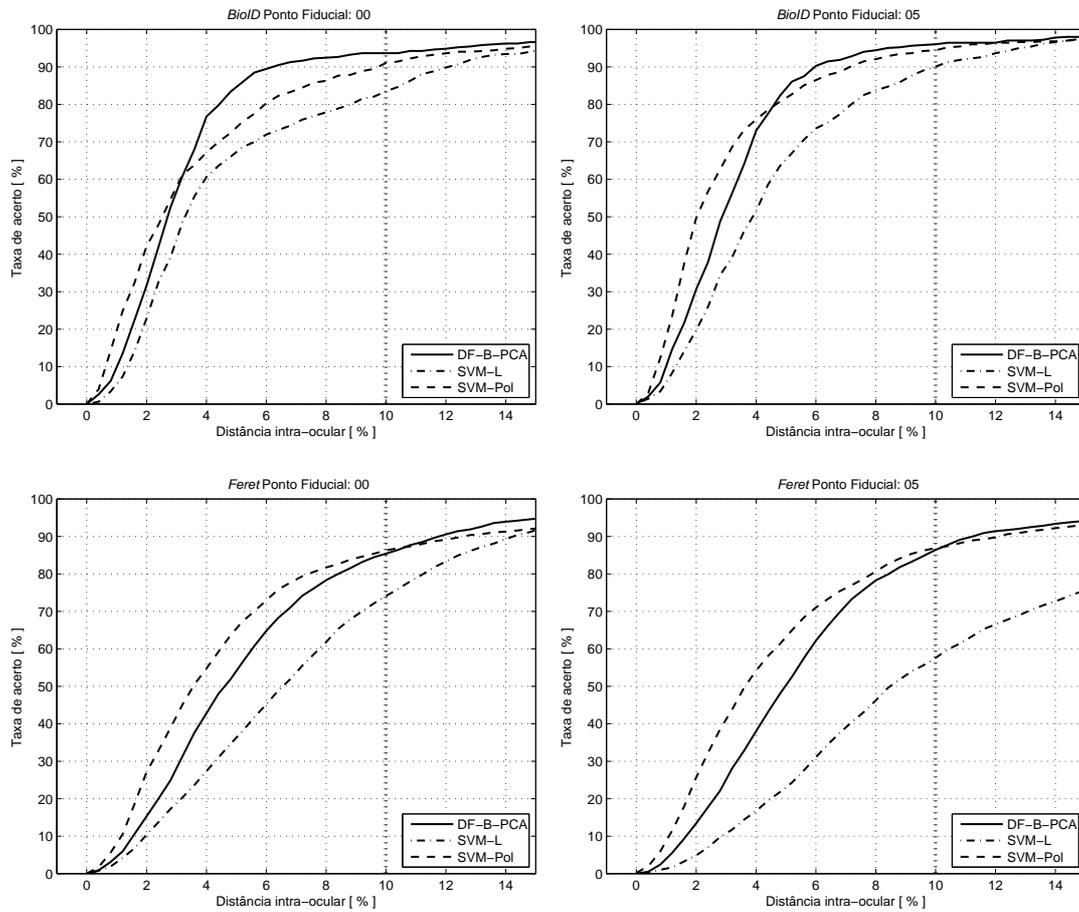
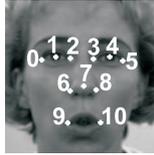


Figura 5.33: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA, SVM com *kernel* polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 00 e 05.

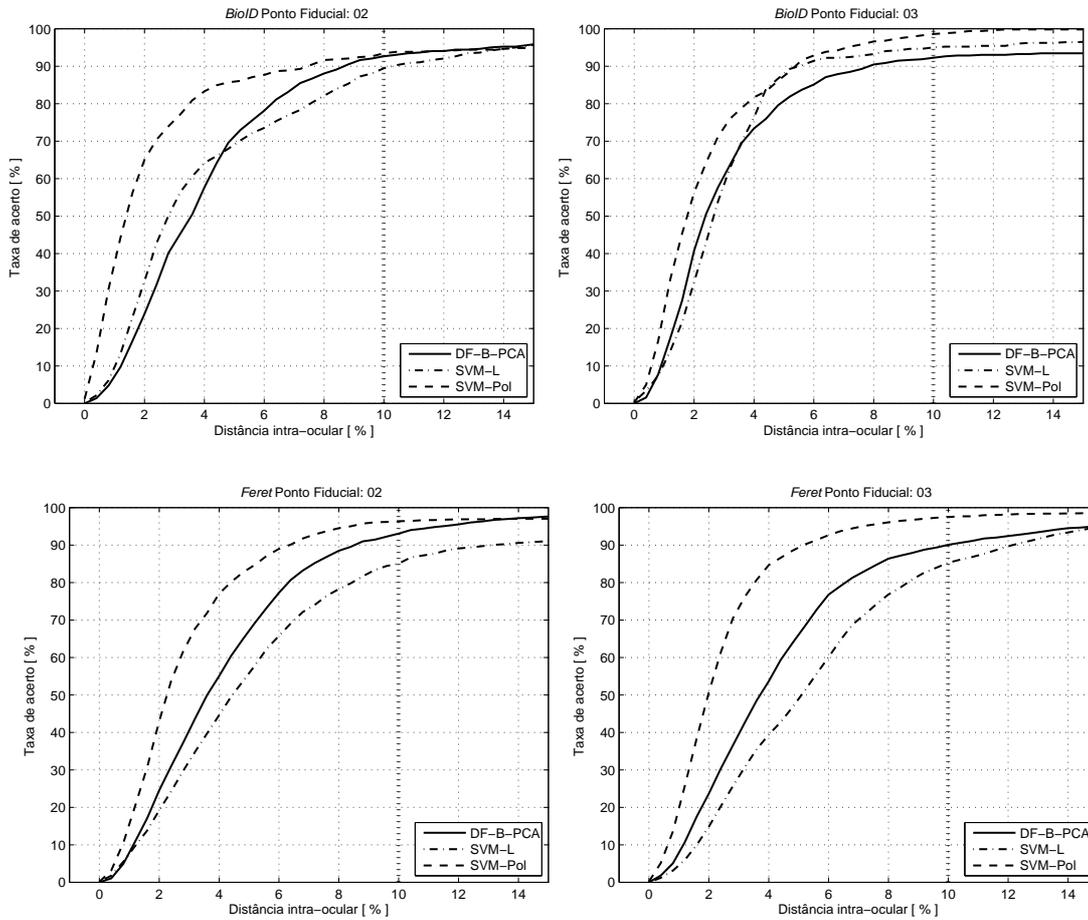


Figura 5.34: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA, SVM com *kernel* polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 02 e 03.

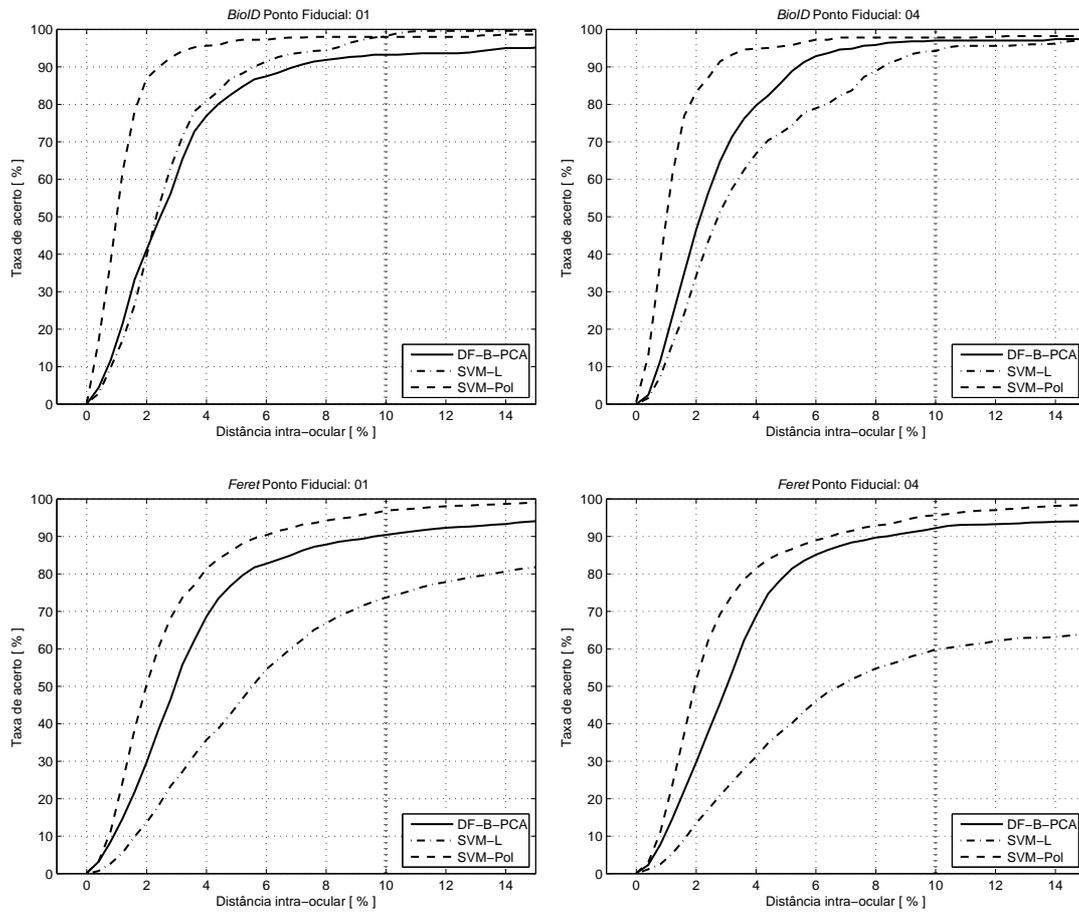
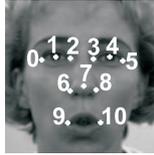


Figura 5.35: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA, SVM com *kernel* polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 01 e 04.

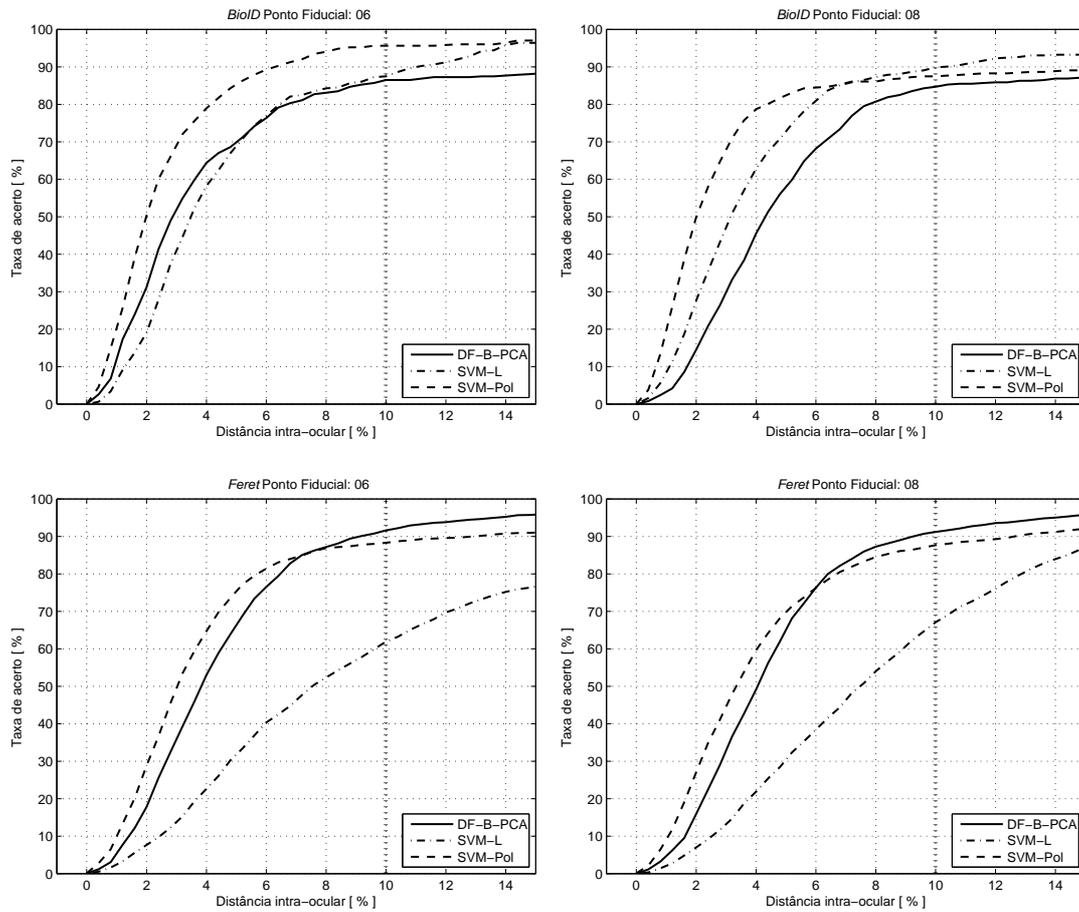
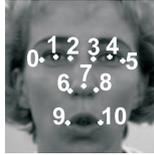


Figura 5.36: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA, SVM com *kernel* polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 06 e 08.

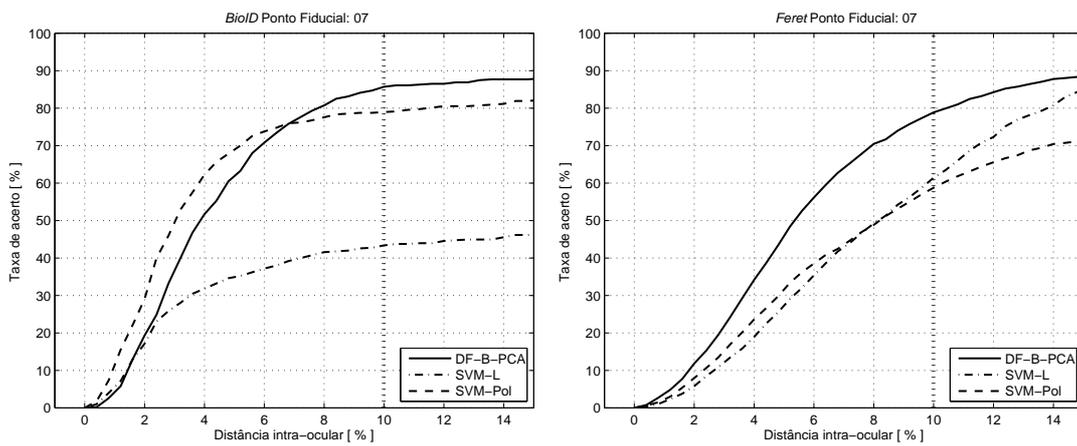


Figura 5.37: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA, SVM com *kernel* polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para o ponto fiducial 07.

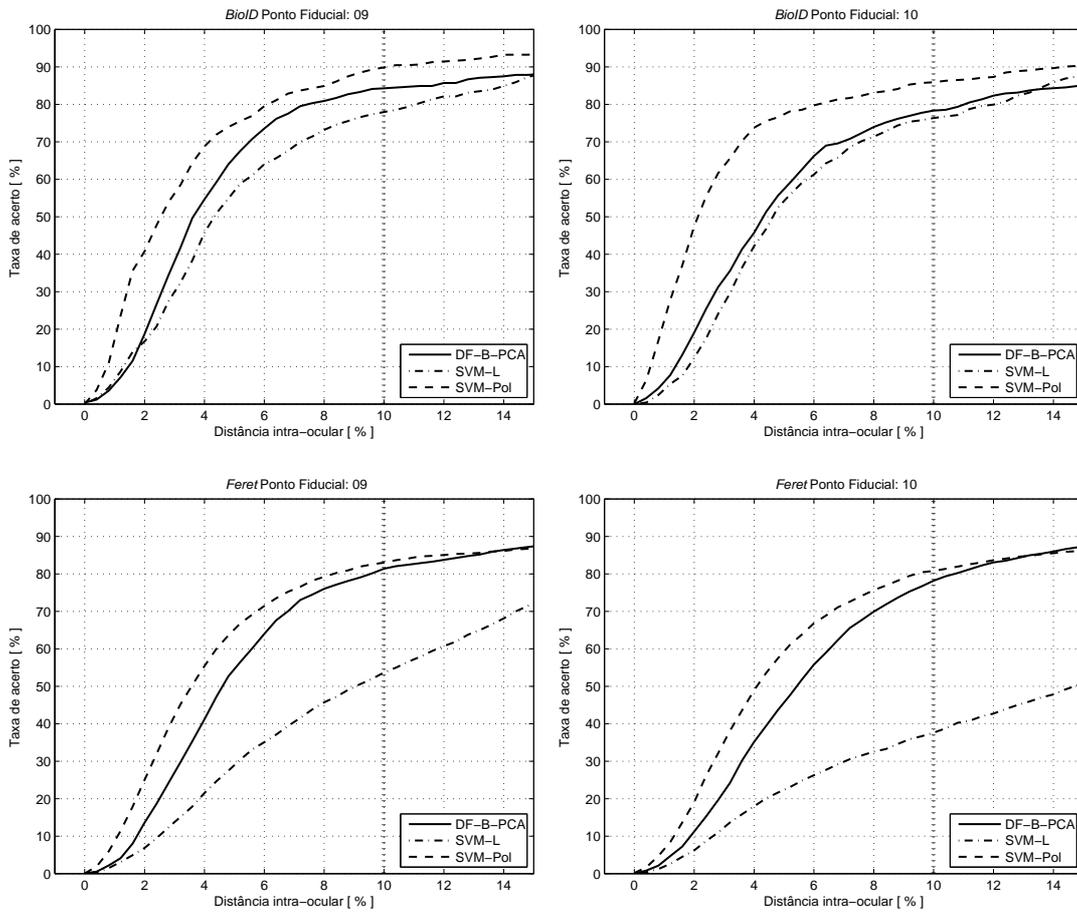
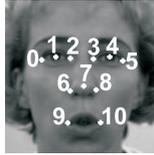


Figura 5.38: Bases de dados *BioID* e *Feret*: resultados para o método DF-B-PCA, SVM com *kernel* polinomial e SVM linear utilizando as estratégias, respectivamente, GML, GML e ML para os pontos fiduciais 09 e 10.

5.11 Considerações Finais

Neste capítulo, atacamos o problema da detecção de pontos fiduciais em faces humanas com o objetivo de validarmos o método proposto. Com o propósito de comparação, realizamos outras implementações denominadas por DF-B, DF, SVM-L e SVM-Pol. As simulações para todos os métodos, foram realizadas utilizando-se dois subconjuntos das bases de dados *BioID* e *Feret*, respectivamente com 503 e 2004 imagens no total, com 11 pontos fiduciais cuidadosamente rotulados. Os resultados foram obtidos utilizando-se validação cruzada com 7 *folds*.

Os resultados obtidos comprovam que o método proposto proporciona resultados bem superiores aos seus antecessores (DF-B e DF) para todos os pontos fiduciais demonstrando que as modificações propostas foram eficazes. Os resultados adquiridos para o método proposto são competitivos quando comparados aos do método SVM-Pol, que representa o estado da arte, e bem superiores aos do método SVM-L.

No próximo capítulo, apresentamos uma classe de detectores denominado por Detectores por Produto Interno (IPD), sendo a próxima contribuição desta Tese, que utiliza um produto interno para determinar se um candidato é ou não o padrão de interesse. Demonstramos que a Filtragem Discriminativa e os Filtros de Correlação são casos particulares do IPD. Por fim, apresentamos uma formulação para obtermos IPD robustos.

Capítulo 6

Detectores por Produto Interno com Minimização do Erro Quadrático Médio (IPD)

Recentemente, pode-se encontrar na literatura diversas técnicas originadas do tradicional método denominado Filtros de Correlação ou CF (em inglês, *Correlation Filters* [33,99]). Os algoritmos desenvolvidos podem ser aplicados no reconhecimento de: faces humanas [31–35], íris [36], características biométricas [37–39], dentre outras [100]. Neste capítulo, desenvolvemos uma classe de detectores denominados Detectores por Produto Interno (IPD). O classificador obtido para o IPD, na forma de um vetor \mathbf{h} , será ótimo no sentido de minimizar o erro médio quadrático e sua saída será obtida através do produto interno da entrada com o vetor \mathbf{h} . Demonstramos a relação existente entre os métodos baseados nos Filtros de Correlação e a Teoria de Filtragem Discriminativa por Restauração do Impulso, apresentada no Capítulo 3. A seguir, mostramos como utilizar a Análise de Componentes Principais (veja o Capítulo 4) para estender a teoria desenvolvida para o IPD.

6.1 Detectores por Produto Interno com Minimização do Erro Quadrático Médio para uma Única Classe de Padrões

Nesta seção, partindo de uma solução LMMSE para o problema de classificação definiremos o detector por produto interno onde poderão ser dados diferentes pesos na aceitação/rejeição de determinados conjuntos de padrões.

Suponha uma variável aleatória $\mathcal{X}_{d \times 1}$ de tal forma que suas realizações \mathbf{x}_i possam ser classificadas ou nas classes $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$ ou na classe \mathcal{B} que concentra os demais padrões (observe a Figura 6.1 para $n = 4$). A probabilidade de \mathcal{X} pertencer à classe \mathcal{B} será definida por $p(\mathcal{B})$. A probabilidade de \mathcal{X} pertencer à \mathcal{A}_i será definida como $p(\mathcal{A}_i) = p_i$, com $i = \{1, 2, \dots, n\}$. Dessa forma, podemos resumir:

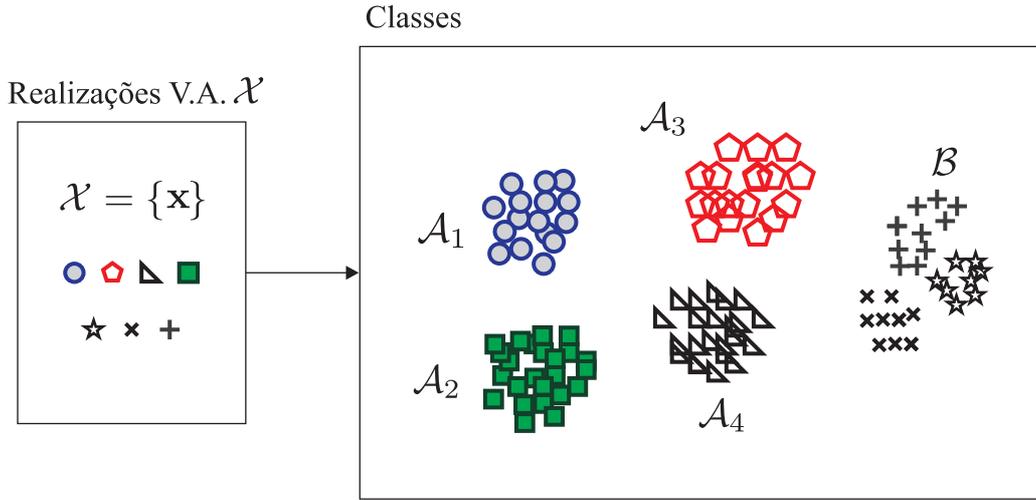


Figura 6.1: Representação das classes para $n = 4$. À esquerda temos as realizações possíveis da variável aleatória \mathcal{X} e à direita as classes $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$ e \mathcal{B} .

$$\begin{aligned}
 p(\mathcal{A}_i) &= p(\mathcal{X} \in \mathcal{A}_i) = p_i, \\
 p(\mathcal{B}) &= p(\mathcal{X} \in \mathcal{B}).
 \end{aligned}
 \tag{6.1}$$

Nesta seção, nosso objetivo é obter um classificador cuja saída é dada pelo produto interno dos vetores $\mathbf{h}_{\mathcal{A}_i}$ (com dimensão $d \times 1$) e \mathbf{x} tal que, se a entrada \mathbf{x} é

uma realização da variável aleatória \mathcal{X} pertencente a \mathcal{A}_i então:

$$\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}_i} \rangle = \mathbf{h}_{\mathcal{A}_i}^t \mathbf{x} = 1$$

e, caso contrário,

$$\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}_i} \rangle = \mathbf{h}_{\mathcal{A}_i}^t \mathbf{x} = 0.$$

Genericamente, pode-se escrever:

$$\langle \mathcal{X}, \mathbf{h}_{\mathcal{A}_i} \rangle = \mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} = \mathcal{C}, \quad (6.2)$$

onde $\mathcal{C} = 1$ para $\mathcal{X} \in \mathcal{A}_i$ e $\mathcal{C} = 0$ para $\mathcal{X} \notin \mathcal{A}_i$.

Na verdade procuramos um vetor $\mathbf{h}_{\mathcal{A}_i}$, na Equação (6.2), que produza o melhor classificador possível no sentido dos mínimos quadrados, isto é, definindo o erro como:

$$\mathcal{E} = \mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} - \mathcal{C}.$$

Temos que o erro quadrático será dado por:

$$\|\mathcal{E}\|^2 = (\mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} - \mathcal{C}) (\mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} - \mathcal{C})^t. \quad (6.3)$$

Considerando-se $\mathbf{h}_{\mathcal{A}_i}$, \mathcal{X} e \mathcal{C} reais podemos desenvolver a Equação (6.3) da seguinte maneira:

$$\begin{aligned} \|\mathcal{E}\|^2 &= (\mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} - \mathcal{C}) (\mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} - \mathcal{C})^t \\ &= \mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} (\mathbf{h}_{\mathcal{A}_i}^t \mathcal{X})^t - \mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} \mathcal{C}^t - \mathcal{C} (\mathbf{h}_{\mathcal{A}_i}^t \mathcal{X})^t + \mathcal{C} \mathcal{C}^t \\ &= \mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} \mathcal{X}^t \mathbf{h}_{\mathcal{A}_i} - \mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} \mathcal{C} - \mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} \mathcal{C} + \mathcal{C}^2 \\ &= \mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} \mathcal{X}^t \mathbf{h}_{\mathcal{A}_i} - 2\mathbf{h}_{\mathcal{A}_i}^t \mathcal{X} \mathcal{C} + \mathcal{C}^2. \end{aligned}$$

O valor esperado (ou média) do erro quadrático será igual a:

$$E [\|\mathcal{E}\|^2] = \mathbf{h}_{\mathcal{A}_i}^t E [\mathcal{X} \mathcal{X}^t] \mathbf{h}_{\mathcal{A}_i} - 2\mathbf{h}_{\mathcal{A}_i}^t E [\mathcal{X} \mathcal{C}] + E [\mathcal{C}^2]. \quad (6.4)$$

O valor de $\mathbf{h}_{\mathcal{A}_i}$ que minimiza o erro quadrático pode ser calculado igualando

a zero (0) o gradiente da Equação (6.4) em relação a $\mathbf{h}_{\mathcal{A}_i}$. Dessa forma temos:

$$\begin{aligned}\frac{\partial E [\|\mathcal{E}\|^2]}{\partial \mathbf{h}_{\mathcal{A}_i}} &= \frac{\partial}{\partial \mathbf{h}_{\mathcal{A}_i}} \{ \mathbf{h}_{\mathcal{A}_i}^t E [\mathcal{X} \mathcal{X}^t] \mathbf{h}_{\mathcal{A}_i} - 2 \mathbf{h}_{\mathcal{A}_i}^t E [\mathcal{X} \mathcal{C}] + E [\mathcal{C}^2] \} \\ &= \left\{ E [\mathcal{X} \mathcal{X}^t] + E [\mathcal{X} \mathcal{X}^t]^t \right\} \mathbf{h}_{\mathcal{A}_i} - 2 E [\mathcal{X} \mathcal{C}].\end{aligned}\quad (6.5)$$

Igualando a zero (0) a Equação (6.5) encontramos a forma básica do nosso problema. Assim, iremos obter:

$$\begin{aligned}&\left\{ E [\mathcal{X} \mathcal{X}^t] + E [\mathcal{X} \mathcal{X}^t]^t \right\} \mathbf{h}_{\mathcal{A}_i} - 2 E [\mathcal{X} \mathcal{C}] = \\ &= 2 E [\mathcal{X} \mathcal{X}^t] \mathbf{h}_{\mathcal{A}_i} - 2 E [\mathcal{X} \mathcal{C}] = 0,\end{aligned}$$

então,

$$\begin{aligned}E [\mathcal{X} \mathcal{X}^t] \mathbf{h}_{\mathcal{A}_i} &= E [\mathcal{X} \mathcal{C}] \\ \mathbf{h}_{\mathcal{A}_i} &= \{ E [\mathcal{X} \mathcal{X}^t] \}^{-1} E [\mathcal{X} \mathcal{C}].\end{aligned}\quad (6.6)$$

Dessa forma, precisamos encontrar o valor de $E [\mathcal{X} \mathcal{X}^t]$ e $E [\mathcal{X} \mathcal{C}]$. Podemos desenvolver o termo $E [\mathcal{X} \mathcal{X}^t]$ da seguinte maneira:

$$E [\mathcal{X} \mathcal{X}^t] = E [\mathcal{X} \mathcal{X}^t | \mathcal{B}] p(\mathcal{B}) + \sum_{j=1}^n E [\mathcal{X} \mathcal{X}^t | \mathcal{A}_j] p(\mathcal{A}_j).\quad (6.7)$$

Se considerarmos os elementos típicos da classe \mathcal{B} iguais a $\{\mathbf{b}_1, \dots, \mathbf{b}_r\}$ e os elementos típicos das classes \mathcal{A}_i iguais a $\{\mathbf{a}_{i1}, \mathbf{a}_{i2}, \dots, \mathbf{a}_{iL_i}\}$ (com $i = \{1, 2, \dots, n\}$) então podemos calcular as médias da Equação (6.7) conforme a seguir:

$$\begin{aligned}E [\mathcal{X} \mathcal{X}^t] &= E [\mathcal{X} \mathcal{X}^t | \mathcal{B}] p(\mathcal{B}) + \sum_{j=1}^n E [\mathcal{X} \mathcal{X}^t | \mathcal{A}_j] p(\mathcal{A}_j) \\ &= p(\mathcal{B}) \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^t + \sum_{j=1}^n p(\mathcal{A}_j) \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} \mathbf{a}_{jk}^t \\ &= p(\mathcal{B}) \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^t + \sum_{j=1}^n p_j \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} \mathbf{a}_{jk}^t.\end{aligned}\quad (6.8)$$

Para desenvolvermos o termo $E [\mathcal{X} \mathcal{C}]$ devemos recordar que a variável aleatória \mathcal{C} é igual a um (1) para o caso em que a variável aleatória \mathcal{X} for classificada na classe \mathcal{A}_i e \mathcal{C} será igual a zero (0) para o caso em que \mathcal{X} for classificada nas demais

classes representadas através do complemento de \mathcal{A}_i dado por \mathcal{A}_i^C .

$$\begin{aligned}
E[\mathcal{X}\mathcal{C}] &= E[\mathcal{X}\mathcal{C} | \mathcal{A}_i]p(\mathcal{A}_i) + E[\mathcal{X}\mathcal{C} | \mathcal{A}_i^C](1 - p(\mathcal{A}_i)) \\
&= E[\mathcal{X} | \mathcal{A}_i]p(\mathcal{A}_i) + 0(1 - p(\mathcal{A}_i)) \\
&= p_i \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{a}_{ik}.
\end{aligned} \tag{6.9}$$

Dessa forma, podemos concluir que a Equação (6.6) pode ser escrita utilizando-se os resultados apresentados nas Equações (6.8) e (6.9), em função dos elementos típicos das classes, como:

$$\mathbf{h}_{\mathcal{A}_i} = \left\{ p(\mathcal{B}) \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^t + \sum_{j=1}^n p_j \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} \mathbf{a}_{jk}^t \right\}^{-1} \left\{ p_i \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{a}_{ik} \right\} \tag{6.10}$$

Podemos expressar os resultados em função dos momentos das variáveis aleatórias considerando-se a matriz de autocorrelação de \mathcal{B} igual a $\mathbf{R}_{\mathcal{B}}$, a matriz de autocorrelação de \mathcal{A}_i igual a $\mathbf{R}_{\mathcal{A}_i}$ e a média de \mathcal{A}_i igual a $\boldsymbol{\mu}_{\mathcal{A}_i}$, onde:

$$\begin{aligned}
\mathbf{R}_{\mathcal{B}} &= \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j (\mathbf{b}_j)^{*t}, \\
\mathbf{R}_{\mathcal{A}_j} &= \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} (\mathbf{a}_{jk})^{*t}, \\
\boldsymbol{\mu}_{\mathcal{A}_i} &= \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{a}_{ik}.
\end{aligned} \tag{6.11}$$

Considerando \mathbf{b}_j e \mathbf{a}_{jk} reais temos:

$$\begin{aligned}
\mathbf{R}_{\mathcal{B}} &= \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j (\mathbf{b}_j)^{*t} = \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^t, \\
\mathbf{R}_{\mathcal{A}_j} &= \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} (\mathbf{a}_{jk})^{*t} = \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} \mathbf{a}_{jk}^t, \\
\boldsymbol{\mu}_{\mathcal{A}_i} &= \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{a}_{ik}.
\end{aligned} \tag{6.12}$$

Dessa forma a Equação (6.10) ficará, em função dos momentos, igual a:

$$\mathbf{h}_{\mathcal{A}_i} = \left\{ p(\mathcal{B}) \mathbf{R}_{\mathcal{B}} + \sum_{j=1}^n p_j \mathbf{R}_{\mathcal{A}_j} \right\}^{-1} \{p_i \boldsymbol{\mu}_{\mathcal{A}_i}\}. \quad (6.13)$$

Para concluir podemos usar a Equação (6.2) para classificar um vetor de entrada \mathbf{x} (sendo realização da variável aleatória \mathcal{X}) através do valor de \mathcal{C} . Neste caso \mathcal{C} será próximo de um (1) se \mathbf{x} pertencer à classe \mathcal{A}_i e próximo de (0) se \mathbf{x} pertence a qualquer outra classe (observe a Figura 6.2, neste caso o classificador detecta os padrões pertencentes à \mathcal{A}_2 , conforme definição das classes na Figura 6.1). Devemos observar que a inversa da matriz $\left\{ p(\mathcal{B}) \mathbf{R}_{\mathcal{B}} + \sum_{j=1}^n p_j \mathbf{R}_{\mathcal{A}_j} \right\}$ precisa existir e a dimensão dos vetores de entrada deve ser menor que a soma dos números de elementos de todas as classes \mathcal{A}_j e \mathcal{B} , ou seja, $d \leq r + \sum_{j=1}^n L_j$. O caso em que $d > r + \sum_{j=1}^n L_j$ será abordado na Seção 6.5.

Realizações V.A. \mathcal{X}

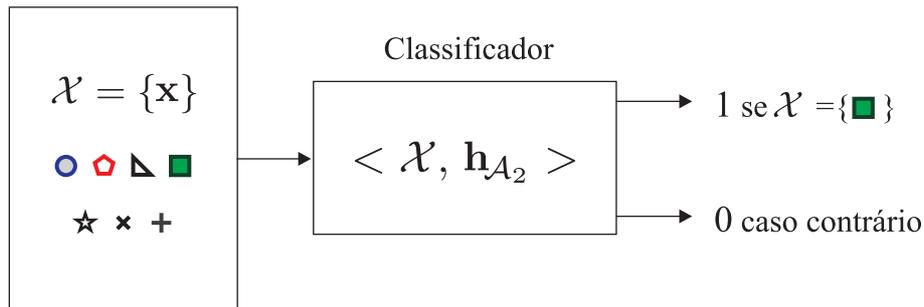


Figura 6.2: Representação do classificador ideal para detectar os elementos pertencentes à classe \mathcal{A}_2 (conforme definição das classes na Figura 6.1). A saída será igual a um (1) caso a entrada (realização de \mathcal{X}) pertença à \mathcal{A}_2 .

6.2 Detectores por Produto Interno com Minimização do Erro Quadrático Médio para Múltiplas Classes de Padrões

O vetor $\mathbf{h}_{\mathcal{A}_i}$ é projetado para produzir um classificador que detecta uma única classe. Nesta seção, iremos propor um classificador capaz de detectar diversas classes que representam padrões os quais desejamos detectar. Dessa forma, se considerarmos

as classes sendo $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$ então nosso objetivo é obter um classificador para detectar todos os padrões que pertencem a classe $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m\}$ onde $m \leq n$.

Assim, de forma semelhante a Seção 6.1, nosso objetivo é obter um classificador com saída igual ao produto interno entre os vetores \mathbf{x} e $\mathbf{h}_{\mathcal{A}}$ (com dimensão $d \times 1$) tal que, se a entrada \mathbf{x} é uma realização da variável aleatória $\mathcal{X} \in \mathcal{A}$ então:

$$\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}} \rangle = \mathbf{h}_{\mathcal{A}}^t \mathbf{x} = 1$$

e, caso contrário,

$$\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}} \rangle = \mathbf{h}_{\mathcal{A}}^t \mathbf{x} = 0$$

a forma genérica pode ser expressa como:

$$\langle \mathcal{X}, \mathbf{h}_{\mathcal{A}} \rangle = \mathbf{h}_{\mathcal{A}}^t \mathcal{X} = \mathcal{C}, \quad (6.14)$$

onde $\mathcal{C} = 1$ para $\mathcal{X} \in \mathcal{A}$ e $\mathcal{C} = 0$ para $\mathcal{X} \notin \mathcal{A}$.

Como na Seção 6.1 devemos encontrar um vetor $\mathbf{h}_{\mathcal{A}}$, na Equação (6.14), que produza o melhor classificador possível no sentido dos mínimos quadrados. Desenvolvendo a Equação (6.14) de forma semelhante à da Seção 6.1 chegamos ao seguinte resultado:

$$\mathbf{h}_{\mathcal{A}} = \{E[\mathcal{X}\mathcal{X}^t]\}^{-1} E[\mathcal{X}\mathcal{C}]. \quad (6.15)$$

Para determinarmos o valor de $\mathbf{h}_{\mathcal{A}}$, na Equação (6.15), precisamos encontrar $E[\mathcal{X}\mathcal{X}^t]$ e $E[\mathcal{X}\mathcal{C}]$. O valor do termo $E[\mathcal{X}\mathcal{X}^t]$ não será modificado sendo exatamente igual a Equação (6.8). Porém, o termo $E[\mathcal{X}\mathcal{C}]$ sofrerá algumas modificações. Neste caso, \mathcal{C} será igual a um (1) se \mathcal{X} for classificada nas classes desejadas $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m\}$ e \mathcal{C} será igual a zero (0) se \mathcal{X} for classificada nas demais

classes representadas através do complemento de \mathcal{A} dado por \mathcal{A}^C .

$$\begin{aligned}
E[\mathcal{X}\mathcal{C}] &= E[\mathcal{X}\mathcal{C} | \mathcal{A}]p(\mathcal{A}) + E[\mathcal{X}\mathcal{C} | \mathcal{A}^C](1 - p(\mathcal{A})) \\
&= E[\mathcal{X} | \mathcal{A}]p(\mathcal{A}) + 0(1 - p(\mathcal{A})) \\
&= \{E[\mathcal{X} | \mathcal{A}_1]p(\mathcal{A}_1 | \mathcal{A}) + \dots + E[\mathcal{X} | \mathcal{A}_m]p(\mathcal{A}_m | \mathcal{A})\}p(\mathcal{A}) \\
&= E[\mathcal{X} | \mathcal{A}_1]p(\mathcal{A}_1) + \dots + E[\mathcal{X} | \mathcal{A}_m]p(\mathcal{A}_m) \\
&= E[\mathcal{X} | \mathcal{A}_1]p_1 + \dots + E[\mathcal{X} | \mathcal{A}_m]p_m \\
&= \sum_{i=1}^m p_i \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{a}_{ik}.
\end{aligned} \tag{6.16}$$

O vetor $\mathbf{h}_{\mathcal{A}}$ poderá ser escrito como:

$$\mathbf{h}_{\mathcal{A}} = \left\{ p(\mathcal{B}) \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^t + \sum_{j=1}^n p_j \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} \mathbf{a}_{jk}^t \right\}^{-1} \left\{ \sum_{i=1}^m p_i \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{a}_{ik} \right\}. \tag{6.17}$$

Assim como na Seção 6.1 podemos expressar a Equação (6.17) em função dos momentos das variáveis aleatórias. Neste caso, podemos utilizar as Equações em (6.12) para obter:

$$\mathbf{h}_{\mathcal{A}} = \left\{ p(\mathcal{B}) \mathbf{R}_{\mathcal{B}} + \sum_{j=1}^n p_j \mathbf{R}_{\mathcal{A}_j} \right\}^{-1} \left\{ \sum_{i=1}^m p_i \boldsymbol{\mu}_{\mathcal{A}_i} \right\}. \tag{6.18}$$

Observe que podemos utilizar a Equação (6.14) para realizar a classificação de um vetor de entrada \mathbf{x} através de \mathcal{C} . Assim, \mathcal{C} será próximo de um (1) nos casos em que \mathbf{x} pertencer às classes \mathcal{A}_1 ou \mathcal{A}_2, \dots , ou \mathcal{A}_m e será próximo de zero (0) caso \mathbf{x} pertença às demais classes (observe a Figura 6.3 para $m = 2$, neste caso, o classificador detecta os padrões pertencentes à $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$, conforme definição das classes na Figura 6.1). As condições apresentadas no final da Seção 6.1 são as mesmas para $\mathbf{h}_{\mathcal{A}}$, ou seja, a matriz $\left\{ p(\mathcal{B}) \mathbf{R}_{\mathcal{B}} + \sum_{j=1}^n p_j \mathbf{R}_{\mathcal{A}_j} \right\}$ precisa ter inversa e a dimensão dos vetores de entrada \mathbf{x} deve ser menor que a soma dos números de elementos de todas as classes \mathcal{A}_j e \mathcal{B} , ou seja, $d \leq r + \sum_{j=1}^n L_j$. Lembrando que o caso $d > r + \sum_{j=1}^n L_j$ será abordado na Seção 6.5.

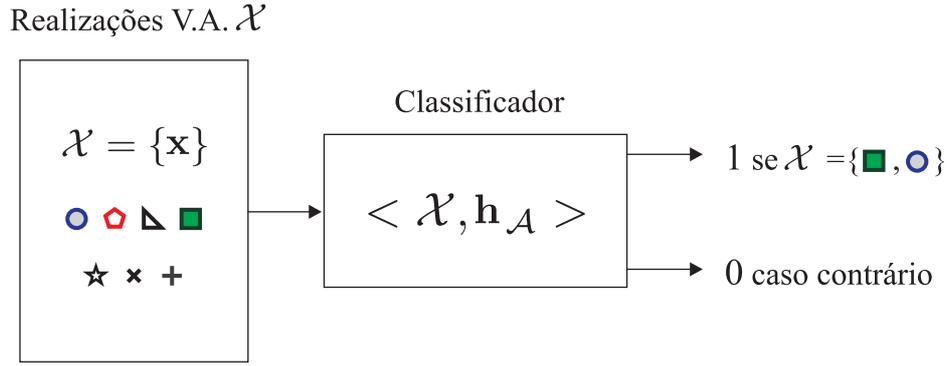


Figura 6.3: Representação do classificador para detectar os elementos pertencentes à $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$ (conforme definição das classes na Figura 6.1). Caso a entrada pertença à $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$ então a saída será igual a um (1).

6.3 Detectores por Produto Interno Complexo com Minimização do Erro Quadrático Médio

Até então fizemos a suposição de que as realizações \mathbf{x} da variável aleatória \mathcal{X} e $\mathbf{h}_{\mathcal{A}}$ são reais. Nesta seção, nosso objetivo será determinar um classificador para o caso de $\mathbf{h}_{\mathcal{A}}$ e a variável aleatória \mathcal{X} complexos. Esta abordagem, além de expressar uma forma geral para o classificador nos permitirá analisar qual será seu comportamento ao usarmos transformadas lineares complexas em \mathcal{X} .

Vamos iniciar considerando \mathbf{x} uma realização complexa da variável aleatória \mathcal{X} . As classes são iguais a $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$ e o classificador irá detectar todos os padrões que pertencem a classe $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m\}$ onde $m \leq n$. O vetor $\mathbf{h}_{\mathcal{A}}$ será projetado tal que, se \mathbf{x} é uma realização da variável aleatória $\mathcal{X} \in \mathcal{A}$ então

$$\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}} \rangle = \mathbf{h}_{\mathcal{A}}^{*t} \mathbf{x} = 1$$

e, caso contrário,

$$\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}} \rangle = \mathbf{h}_{\mathcal{A}}^{*t} \mathbf{x} = 0.$$

A forma genérica de $\mathbf{h}_{\mathcal{A}}$ será:

$$\langle \mathcal{X}, \mathbf{h}_{\mathcal{A}} \rangle = \mathbf{h}_{\mathcal{A}}^{*t} \mathcal{X} = \mathcal{C}, \quad (6.19)$$

onde $\mathcal{C} = 1$ para $\mathcal{X} \in \mathcal{A}$ e $\mathcal{C} = 0$ para $\mathcal{X} \notin \mathcal{A}$.

Assim como na Seção 6.2 devemos procurar $\mathbf{h}_{\mathcal{A}}$, na Equação (6.19), como o

melhor classificador possível no sentido dos mínimos quadrados. O erro quadrático, dado pela Equação (6.3), será igual a:

$$\begin{aligned}
\|\mathcal{E}\|^2 &= (\mathbf{h}_A^{*t} \mathcal{X} - \mathcal{C}) (\mathbf{h}_A^{*t} \mathcal{X} - \mathcal{C})^{*t} \\
&= \mathbf{h}_A^{*t} \mathcal{X} (\mathbf{h}_A^{*t} \mathcal{X})^{*t} - \mathbf{h}_A^{*t} \mathcal{X} \mathcal{C}^{*t} - \mathcal{C} (\mathbf{h}_A^{*t} \mathcal{X})^{*t} + \mathcal{C} \mathcal{C}^{*t} \\
&= \mathbf{h}_A^{*t} \mathcal{X} \mathcal{X}^{*t} \mathbf{h}_A - \mathbf{h}_A^{*t} \mathcal{X} \mathcal{C}^{*t} - \mathcal{C} \mathcal{X}^{*t} \mathbf{h}_A + \|\mathcal{C}\|^2.
\end{aligned} \tag{6.20}$$

Como \mathbf{h}_A é complexo então iremos considerar $\mathbf{h}_A = \mathbf{h}_R + j\mathbf{h}_I$, onde \mathbf{h}_R e \mathbf{h}_I representam, respectivamente, as partes real e imaginária de \mathbf{h}_A . Assim temos:

$$\begin{aligned}
\|\mathcal{E}\|^2 &= (\mathbf{h}_R + j\mathbf{h}_I)^{*t} \mathcal{X} \mathcal{X}^{*t} (\mathbf{h}_R + j\mathbf{h}_I) - (\mathbf{h}_R + j\mathbf{h}_I)^{*t} \mathcal{X} \mathcal{C}^{*t} \\
&\quad - \mathcal{C} \mathcal{X}^{*t} (\mathbf{h}_R + j\mathbf{h}_I) + \|\mathcal{C}\|^2 \\
&= (\mathbf{h}_R^t - j\mathbf{h}_I^t) \mathcal{X} \mathcal{X}^{*t} (\mathbf{h}_R + j\mathbf{h}_I) - (\mathbf{h}_R^t - j\mathbf{h}_I^t) \mathcal{X} \mathcal{C}^{*t} \\
&\quad - \mathcal{C} \mathcal{X}^{*t} (\mathbf{h}_R + j\mathbf{h}_I) + \|\mathcal{C}\|^2 \\
&= \mathbf{h}_R^t \mathcal{X} \mathcal{X}^{*t} \mathbf{h}_R + j \mathbf{h}_R^t \mathcal{X} \mathcal{X}^{*t} \mathbf{h}_I - j \mathbf{h}_I^t \mathcal{X} \mathcal{X}^{*t} \mathbf{h}_R + \mathbf{h}_I^t \mathcal{X} \mathcal{X}^{*t} \mathbf{h}_I \\
&\quad - \mathbf{h}_R^t \mathcal{X} \mathcal{C}^{*t} + j \mathbf{h}_I^t \mathcal{X} \mathcal{C}^{*t} - \mathcal{C} \mathcal{X}^{*t} \mathbf{h}_R - j \mathcal{C} \mathcal{X}^{*t} \mathbf{h}_I + \|\mathcal{C}\|^2.
\end{aligned}$$

O valor esperado de $\|\mathcal{E}\|^2$ será dado por:

$$\begin{aligned}
E [\|\mathcal{E}\|^2] &= \mathbf{h}_R^t E [\mathcal{X} \mathcal{X}^{*t}] \mathbf{h}_R + j \mathbf{h}_R^t E [\mathcal{X} \mathcal{X}^{*t}] \mathbf{h}_I - j \mathbf{h}_I^t E [\mathcal{X} \mathcal{X}^{*t}] \mathbf{h}_R \\
&\quad + \mathbf{h}_I^t E [\mathcal{X} \mathcal{X}^{*t}] \mathbf{h}_I - \mathbf{h}_R^t E [\mathcal{X} \mathcal{C}^{*t}] + j \mathbf{h}_I^t E [\mathcal{X} \mathcal{C}^{*t}] \\
&\quad - E [\mathcal{C} \mathcal{X}^{*t}] \mathbf{h}_R - j E [\mathcal{C} \mathcal{X}^{*t}] \mathbf{h}_I + E [\|\mathcal{C}\|^2].
\end{aligned} \tag{6.21}$$

Neste caso, o valor do gradiente deve ser calculado conforme a seguir, lembrando que o valor de \mathbf{h}_A que minimiza o erro médio quadrático é obtido igualando a zero (0) o gradiente de $E [\|\mathcal{E}\|^2]$ apresentado na Equação (6.21).

$$\begin{aligned}
\frac{\partial E [\|\mathcal{E}\|^2]}{\partial \mathbf{h}_R} &= 0, \\
\frac{\partial E [\|\mathcal{E}\|^2]}{\partial \mathbf{h}_I} &= 0.
\end{aligned}$$

Desenvolvendo os termos $\partial E [||\mathcal{E}^2||] / \partial \mathbf{h}_R$ e $\partial E [||\mathcal{E}^2||] / \partial \mathbf{h}_I$ encontramos:

$$\begin{aligned} \frac{\partial E [||\mathcal{E}||^2]}{\partial \mathbf{h}_R} &= [E [\mathcal{X}\mathcal{X}^{*t}] + E [\mathcal{X}^* \mathcal{X}^t]] \mathbf{h}_R + j E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_I \\ &\quad - j E [\mathcal{X}^* \mathcal{X}^t] \mathbf{h}_I - E [\mathcal{X}\mathcal{C}^{*t}] - E [\mathcal{X}^* \mathcal{C}^t]. \end{aligned} \quad (6.22)$$

$$\begin{aligned} \frac{\partial E [||\mathcal{E}||^2]}{\partial \mathbf{h}_I} &= j E [\mathcal{X}^* \mathcal{X}^t] \mathbf{h}_R - j E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_R \\ &\quad + [E [\mathcal{X}\mathcal{X}^{*t}] + E [\mathcal{X}^* \mathcal{X}^t]] \mathbf{h}_I + j E [\mathcal{X}\mathcal{C}^{*t}] \\ &\quad - j E [\mathcal{X}^* \mathcal{C}^t]. \end{aligned} \quad (6.23)$$

Igualando os gradientes $\partial [\mathcal{E}\mathcal{E}^{*t}] / \partial \mathbf{h}_R$ e $\partial [\mathcal{E}\mathcal{E}^{*t}] / \partial \mathbf{h}_I$ (Equações (6.22) e (6.23)) a zero (0), multiplicando a Equação (6.23) por j e organizando os termos iremos obter:

$$\begin{aligned} E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_R + E [\mathcal{X}^* \mathcal{X}^t] \mathbf{h}_R + j E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_I - j E [\mathcal{X}^* \mathcal{X}^t] \mathbf{h}_I \\ - E [\mathcal{X}\mathcal{C}^{*t}] - E [\mathcal{X}^* \mathcal{C}^t] = 0. \end{aligned} \quad (6.24)$$

$$\begin{aligned} E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_R - E [\mathcal{X}^* \mathcal{X}^t] \mathbf{h}_R + j E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_I + j E [\mathcal{X}^* \mathcal{X}^t] \mathbf{h}_I \\ - E [\mathcal{X}\mathcal{C}^{*t}] + E [\mathcal{X}^* \mathcal{C}^t] = 0. \end{aligned} \quad (6.25)$$

Efetuando a soma das Equações (6.24) e (6.25) temos:

$$\begin{aligned} 2E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_R + j 2 \mathbf{h}_I E [\mathcal{X}\mathcal{X}^{*t}] - 2 E [\mathcal{X}\mathcal{C}^{*t}] &= 0 \\ E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_R + j \mathbf{h}_I E [\mathcal{X}\mathcal{X}^{*t}] - E [\mathcal{X}\mathcal{C}^{*t}] &= 0 \\ E [\mathcal{X}\mathcal{X}^{*t}] (\mathbf{h}_R + j \mathbf{h}_I) &= E [\mathcal{X}\mathcal{C}^{*t}] \\ E [\mathcal{X}\mathcal{X}^{*t}] \mathbf{h}_A &= E [\mathcal{X}\mathcal{C}^{*t}] \\ \mathbf{h}_A &= E [\mathcal{X}\mathcal{X}^{*t}]^{-1} E [\mathcal{X}\mathcal{C}^{*t}]. \end{aligned} \quad (6.26)$$

A Equação (6.26) representa a forma final do vetor complexo \mathbf{h}_A . Perceba a semelhança entre as Equações (6.15) (caso real para múltiplas classes) e (6.26). Semelhante à Seção 6.2, precisamos encontrar, neste caso, $E [\mathcal{X}\mathcal{X}^{*t}]$ e $E [\mathcal{X}\mathcal{C}^{*t}]$. O termo $E [\mathcal{X}\mathcal{C}^{*t}]$ não será modificado e o termo $E [\mathcal{X}\mathcal{X}^{*t}]$ poderá ser encontrado de

forma análoga ao desenvolvimento da Seção 6.1. Dessa forma, $E[\mathcal{X}\mathcal{X}^{*t}]$ e $E[\mathcal{X}\mathcal{C}^{*t}]$ são iguais a:

$$\begin{aligned} E[\mathcal{X}\mathcal{X}^{*t}] &= p(\mathcal{B}) \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j (\mathbf{b}_j)^{*t} + \sum_{j=1}^n p_j \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} (\mathbf{a}_{jk})^{*t} \\ E[\mathcal{X}\mathcal{C}^{*t}] &= \sum_{i=1}^m p_i \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{a}_{ik}. \end{aligned} \quad (6.27)$$

Assim, $\mathbf{h}_{\mathcal{A}}$ será expresso por:

$$\mathbf{h}_{\mathcal{A}} = \left\{ p(\mathcal{B}) \frac{1}{r} \sum_{j=1}^r \mathbf{b}_j (\mathbf{b}_j)^{*t} + \sum_{j=1}^n p_j \frac{1}{L_j} \sum_{k=1}^{L_j} \mathbf{a}_{jk} (\mathbf{a}_{jk})^{*t} \right\}^{-1} \left\{ \sum_{i=1}^m p_i \frac{1}{L_i} \sum_{k=1}^{L_i} \mathbf{a}_{ik} \right\}. \quad (6.28)$$

Se utilizarmos as Equações em (6.11) poderemos expressar $\mathbf{h}_{\mathcal{A}}$ em função dos momentos $\mathbf{R}_{\mathcal{B}}$, $\mathbf{R}_{\mathcal{A}_j}$ e $\boldsymbol{\mu}_{\mathcal{A}_i}$. A expressão ficará:

$$\mathbf{h}_{\mathcal{A}} = \left\{ p(\mathcal{B}) \mathbf{R}_{\mathcal{B}} + \sum_{j=1}^n p_j \mathbf{R}_{\mathcal{A}_j} \right\}^{-1} \left\{ \sum_{i=1}^m p_i \boldsymbol{\mu}_{\mathcal{A}_i} \right\}. \quad (6.29)$$

Observe que as condições para a existência do classificador e detecção de um vetor de entrada \mathbf{x} , apresentadas no final das Seções 6.1 e 6.2 são semelhantes à esta seção. Neste caso, $\mathbf{h}_{\mathcal{A}}$ existirá se a inversa da matriz $E[\mathcal{X}\mathcal{X}^{*t}]^{-1}$ existir e se $d \leq r + \sum_{j=1}^n L_j$. O vetor de entrada \mathbf{x} poderá ser classificado através da Equação (6.19) com \mathcal{C} próximo de um (1) caso \mathbf{x} seja o padrão o qual desejamos detectar e \mathcal{C} próximo de zero (0) caso contrário. A seguir, iremos analisar qual será o comportamento do classificador quando realizamos uma transformação linear em \mathcal{X} .

6.3.1 Detectores por Produto Interno Complexo com Transformação Linear

Muitas vezes é interessante que antes de fazermos a classificação dos vetores \mathcal{X} possamos realizar uma transformação linear em \mathcal{X} de forma que após esta transformação as possíveis realizações de \mathcal{X} estejam mais separadas no espaço n dimensional e dessa forma o erro de classificação seja diminuído. Para tal, vamos

considerar uma transformação linear complexa \mathbf{Q} de tal forma que

$$\mathcal{X}' = \mathbf{Q} \mathcal{X},$$

onde \mathcal{X} é a variável aleatória complexa definida no início desta seção e \mathbf{Q} é uma matriz quadrada com determinante diferente de zero. Neste caso, podemos utilizar a Equação (6.26) para encontrarmos o vetor transformado $\mathbf{h}'_{\mathcal{A}}$. Assim, podemos escrever:

$$\begin{aligned} \mathbf{h}'_{\mathcal{A}} &= E [\mathcal{X}' \mathcal{X}'^{*t}]^{-1} E [\mathcal{X}' \mathcal{C}^{*t}] \\ &= E [\mathbf{Q} \mathcal{X} (\mathbf{Q} \mathcal{X})^{*t}]^{-1} E [\mathbf{Q} \mathcal{X} \mathcal{C}^{*t}] \\ &= E [\mathbf{Q} \mathcal{X} \mathcal{X}^{*t} \mathbf{Q}^{*t}]^{-1} E [\mathbf{Q} \mathcal{X} \mathcal{C}^{*t}] \\ &= (\mathbf{Q} E [\mathcal{X} \mathcal{X}^{*t}] \mathbf{Q}^{*t})^{-1} \mathbf{Q} E [\mathcal{X} \mathcal{C}^{*t}] \\ &= (\mathbf{Q}^{*t})^{-1} E [\mathcal{X} \mathcal{X}^{*t}]^{-1} \mathbf{Q}^{-1} \mathbf{Q} E [\mathcal{X} \mathcal{C}^{*t}] \\ &= (\mathbf{Q}^{*t})^{-1} E [\mathcal{X} \mathcal{X}^{*t}]^{-1} E [\mathcal{X} \mathcal{C}^{*t}] \\ &= (\mathbf{Q}^{*t})^{-1} \mathbf{h}_{\mathcal{A}}. \end{aligned} \tag{6.30}$$

A Equação (6.30) define o vetor transformado sendo obtido quando \mathcal{X} sofre uma transformação linear igual a $\mathcal{X}' = \mathbf{Q} \mathcal{X}$. Vamos, neste momento, fazer o produto interno no domínio transformado, ou seja, $\langle \mathcal{X}', \mathbf{h}'_{\mathcal{A}} \rangle = (\mathbf{h}'_{\mathcal{A}})^{*t} \mathcal{X}'$:

$$\begin{aligned} (\mathbf{h}'_{\mathcal{A}})^{*t} \mathcal{X}' &= [(\mathbf{Q}^{*t})^{-1} \mathbf{h}_{\mathcal{A}}]^{*t} \mathbf{Q} \mathcal{X} \\ &= \mathbf{h}_{\mathcal{A}}^{*t} [(\mathbf{Q}^{*t})^{-1}]^{*t} \mathbf{Q} \mathcal{X} \\ &= \mathbf{h}_{\mathcal{A}}^{*t} \mathbf{Q}^{-1} \mathbf{Q} \mathcal{X} \\ &= \mathbf{h}_{\mathcal{A}}^{*t} \mathcal{X}. \end{aligned} \tag{6.31}$$

A Equação (6.31) expressa que o vetor ótimo (para o domínio transformado) aplicado no domínio transformado produz exatamente o mesmo resultado quando o vetor ótimo (para o domínio original) é aplicado no domínio original. Este fato tem uma importância fundamental pois, considerando o uso de vetores ótimos, temos que \mathbf{Q} não irá produzir nenhum efeito no que diz respeito à classificação dos padrões envolvidos. Entretanto, utilizando um \mathbf{Q} derivado da Transformada *Karhunen-*

Loève (KLT), também conhecida como Análise de Componentes Principais (PCA) [29, 30, 85], então podemos ter uma interessante interpretação para o classificador conforme veremos a seguir.

6.3.2 Detectores por Produto Interno Complexo com Transformação Linear - Interpretação no Domínio Transformado

O vetor \mathbf{h}_A (dimensão $d \times 1$), mostrado na Equação (6.26), pode ser representado em função da matriz de autocorrelação \mathbf{R}_X de \mathcal{X} da seguinte forma:

$$\begin{aligned}\mathbf{h}_A &= E[\mathcal{X}\mathcal{X}^{*t}]^{-1} E[\mathcal{X}\mathcal{C}^{*t}] \\ &= (\mathbf{R}_X)^{-1} E[\mathcal{X}\mathcal{C}^{*t}].\end{aligned}\tag{6.32}$$

Considerando a KLT de \mathbf{v} (dimensão $d \times 1$) igual a \mathbf{V} e $\Phi = [\phi_1, \dots, \phi_d]$ igual a matriz de autovetores ortonormais de \mathbf{R}_X . Temos que:

$$\mathbf{V} = \Phi^{*t}\mathbf{v}, \quad \mathbf{v} = \Phi\mathbf{V}.\tag{6.33}$$

Podemos encontrar na literatura [20, 85] a relação entre Λ , Φ e \mathbf{R}_X de tal forma que Λ será diagonal com seus elementos da diagonal principal sendo os autovalores $\lambda_1 \geq \dots \geq \lambda_d$ de \mathbf{R}_X correspondentes a cada autovetor de Φ (os valores λ_i correspondem a variância de \mathcal{X} [20, 85]). Assim:

$$\Lambda = \Phi^{*t}\mathbf{R}_X\Phi.\tag{6.34}$$

Podemos manipular a Equação (6.34) como segue:

$$\begin{aligned}\Phi^{*t}\mathbf{R}_X\Phi &= \Lambda \\ \Phi(\Phi^{*t}\mathbf{R}_X\Phi)\Phi^{*t} &= \Phi\Lambda\Phi^{*t} \\ \mathbf{R}_X &= \Phi\Lambda\Phi^{*t}.\end{aligned}\tag{6.35}$$

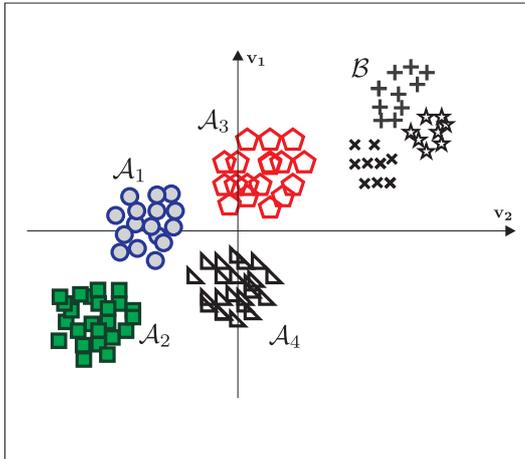
Substituindo a Equação (6.35) na Equação (6.32) iremos obter:

$$\begin{aligned}
\mathbf{h}_{\mathcal{A}} &= (\Phi \Lambda \Phi^{*t})^{-1} E [\mathcal{X} \mathcal{C}^{*t}] \\
\mathbf{h}_{\mathcal{A}} &= \Phi \Lambda^{-1} \Phi^{*t} E [\mathcal{X} \mathcal{C}^{*t}] \\
\Phi^{*t} \mathbf{h}_{\mathcal{A}} &= \Phi^{*t} \Phi \Lambda^{-1/2} \Lambda^{-1/2} \Phi^{*t} E [\mathcal{X} \mathcal{C}^{*t}] \\
\Phi^{*t} \mathbf{h}_{\mathcal{A}} &= \Lambda^{-1/2} E [\Lambda^{-1/2} \Phi^{*t} \mathcal{X} \mathcal{C}^{*t}] \\
\Lambda^{1/2} \Phi^{*t} \mathbf{h}_{\mathcal{A}} &= E [\Lambda^{-1/2} \Phi^{*t} \mathcal{X} \mathcal{C}^{*t}] \\
[(\Lambda^{-1/2} \Phi^{*t})^{*t}]^{-1} \mathbf{h}_{\mathcal{A}} &= E [\Lambda^{-1/2} \Phi^{*t} \mathcal{X} \mathcal{C}^{*t}]. \tag{6.36}
\end{aligned}$$

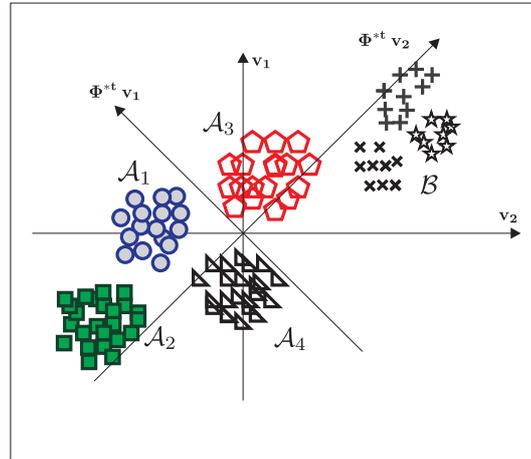
Comparando o resultado apresentado na Equação (6.30) (exposta abaixo por conveniência) com a Equação (6.36) encontramos a expressão abaixo onde a transformação \mathbf{Q} será dada por $\mathbf{Q} = \Lambda^{-1/2} \Phi^{*t}$:

$$\mathbf{h}'_{\mathcal{A}} = (\mathbf{Q}^{*t})^{-1} \mathbf{h}_{\mathcal{A}} = E [\mathbf{Q} \mathcal{X} \mathcal{C}^{*t}]. \tag{6.37}$$

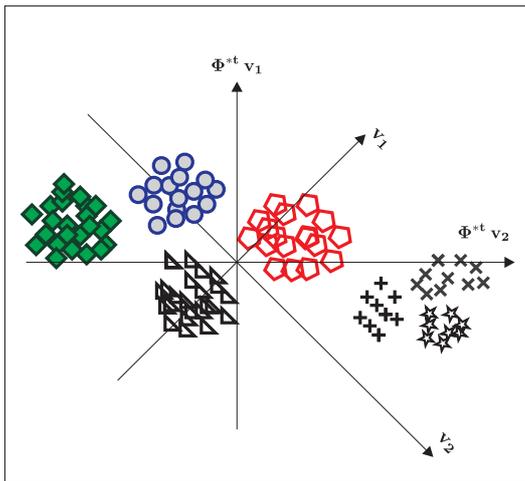
A Equação (6.37) nos permite vislumbrar uma interessante interpretação para o vetor transformado $\mathbf{h}'_{\mathcal{A}}$. Perceba que, na verdade, o termo $\mathbf{Q} \mathcal{X}$ é igual a KLT de \mathcal{X} dada por $\Phi^{*t} \mathcal{X}$ (veja a Equação (6.33)) multiplicada por $\Lambda^{-1/2}$. Assim, estamos aplicando uma rotação nas realizações de \mathcal{X} a fim de torná-las descorrelacionadas. O produto por $\Lambda^{-1/2}$ pode ser interpretado como o branqueamento (em inglês, *whitening* [20,101–103]) pois $\Phi^{*t} \mathcal{X}$ será descorrelacionado. O branqueamento normaliza as realizações de \mathcal{X} em relação ao desvio padrão fazendo com que elas fiquem com separação angular máxima. Este efeito pode ser observado na Figura 6.4.



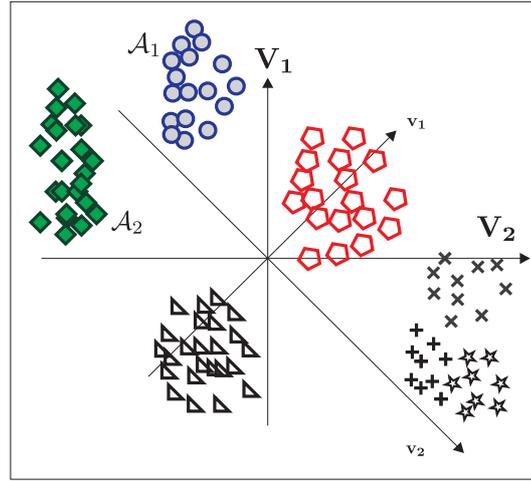
(a) Dados originais agrupados nas classes \mathcal{A}_1 , \mathcal{A}_2 , \mathcal{A}_3 , \mathcal{A}_4 e \mathcal{B} .



(b) Descorrelacionando os dados, ou seja, aplicando a KLT. Eixos originais são \mathbf{v}_1 e \mathbf{v}_2 .



(c) Rotacionando os dados para centralizar os novos eixos $\Phi^{*t} \mathbf{v}_1$ e $\Phi^{*t} \mathbf{v}_2$.



(d) Fazendo o branqueamento dos dados decorrelacionados da Figura 6.4(c). Os novos eixos serão $\mathbf{V}_1 = \Lambda^{-1/2} \Phi^{*t} \mathbf{v}_1$ e $\mathbf{V}_2 = \Lambda^{-1/2} \Phi^{*t} \mathbf{v}_2$. Perceba o aumento da separação entre as classes \mathcal{A}_1 e \mathcal{A}_2 .

Figura 6.4: Exemplo para o branqueamento de um conjunto de dados agrupados em classes conforme a Figura 6.4(a). Na Figura 6.4(b) temos os dados originais. Os dados decorrelacionados e a rotação para centralizar os eixos são apresentados nas Figuras 6.4(b) e 6.4(c). Na Figura 6.4(d) temos o branqueamento dos dados decorrelacionados. Podemos verificar o aumento da separação angular entre as classes \mathcal{A}_1 e \mathcal{A}_2 ao compararmos a Figura 6.4(a) e a Figura 6.4(d).

Para o caso do produto interno $\langle \mathcal{X}', \mathbf{h}'_{\mathcal{A}} \rangle$ (saída do classificador no domínio transformado) lembrando que as dimensões dos vetores $\mathbf{h}'_{\mathcal{A}}$ e \mathcal{X}' valem $d \times 1$ e considerando $\mathbf{h}'_{\mathcal{A}}[n] = [h'(1) h'(2) \dots h'(d)]^t$ e $\mathcal{X}'[n] = [x'(1)x'(2) \dots x'(d)]^t$ podemos escrever:

$$\langle \mathcal{X}', \mathbf{h}'_{\mathcal{A}} \rangle = \sum_{k=1}^d x'[k]h'^*[k]. \quad (6.38)$$

A convolução discreta [30, 73], por definição, é igual a

$$s_1[n] * s_2[n] = \sum_{k=-\infty}^{\infty} s_1[k]s_2[n-k],$$

podendo ser apresentada como:

$$\begin{aligned} s_1[n] * s_2^*[-n] &= \sum_{k=-\infty}^{\infty} s_1[k]s_2^*[k-n] \\ s_1[n] * s_2^*[-n]|_{n=0} &= \sum_{k=-\infty}^{\infty} s_1[k]s_2^*[k], \end{aligned} \quad (6.39)$$

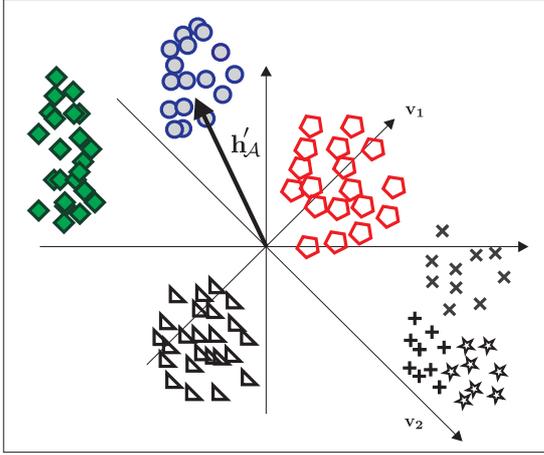
finalmente, utilizando a Equação (6.39) na Equação (6.38) encontramos:

$$\langle \mathcal{X}', \mathbf{h}'_{\mathcal{A}} \rangle = \mathcal{X}'[n] * \mathbf{h}'_{\mathcal{A}}^*[-n]|_{n=0}. \quad (6.40)$$

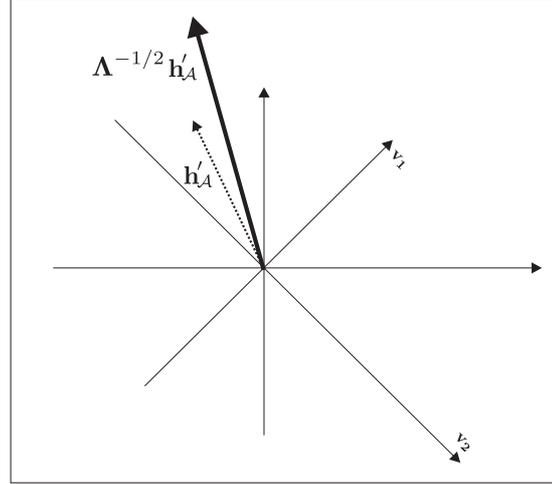
Note que a Equação (6.40), representa a forma clássica do filtro casado (em inglês, *matched filter* [73]). Assim o produto escalar $\langle \mathcal{X}', \mathbf{h}'_{\mathcal{A}} \rangle$ (saída do classificador no domínio transformado) será igual à filtragem (no domínio transformado) entre $\mathcal{X}'[n]$ e $\mathbf{h}'_{\mathcal{A}}^*[-n]$ para $n = 0$. Por último, podemos manipular $\mathbf{h}'_{\mathcal{A}}$, utilizando as Equações (6.27) e (6.11) da seguinte forma:

$$\begin{aligned} \mathbf{h}'_{\mathcal{A}} &= E[\mathbf{Q}\mathcal{X}\mathcal{C}^{*t}] \\ &= \mathbf{Q}E[\mathcal{X}\mathcal{C}^{*t}] \\ &= \mathbf{Q} \sum_{i=1}^m p_i \boldsymbol{\mu}_{\mathcal{A}_i} \\ &= \sum_{i=1}^m p_i \boldsymbol{\mu}'_{\mathcal{A}_i}. \end{aligned} \quad (6.41)$$

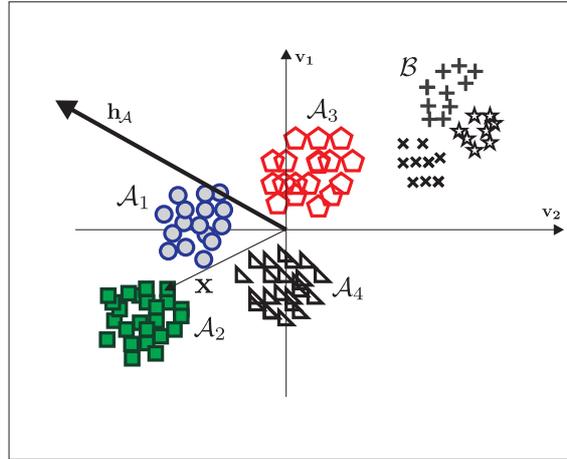
A Equação (6.41) nos apresenta que $\mathbf{h}'_{\mathcal{A}}$, na verdade, é a soma ponderada das médias $\boldsymbol{\mu}'_{\mathcal{A}_i}$ por p_i , dos m vetores transformados \mathcal{X}' os quais desejamos detectar, ou seja é o filtro casado com a média dos padrões que se quer detectar com $\mathbf{h}_{\mathcal{A}}$ (observe um exemplo na Figura 6.5).



(a) Vetor $\mathbf{h}'_{\mathcal{A}}$ para detecção da classe \mathcal{A}_1 . O valor de $\mathbf{h}'_{\mathcal{A}}$ é igual a $\boldsymbol{\mu}'_{\mathcal{A}_1}$ (média da classe \mathcal{A}_1 transformada).



(b) Fazendo o produto do vetor $\mathbf{h}'_{\mathcal{A}}$ por $\boldsymbol{\Lambda}^{-1/2}$. Passo intermediário para encontrarmos $\mathbf{h}_{\mathcal{A}}$.



(c) Vetor $\mathbf{h}_{\mathcal{A}} = \boldsymbol{\Phi} \boldsymbol{\Lambda}^{-1/2} \mathbf{h}'_{\mathcal{A}}$ no domínio original. Perceba que $\mathbf{h}_{\mathcal{A}}$ é aproximadamente ortogonal aos vetores das classes $\mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$ e \mathcal{B} e aproximadamente colinear aos vetores da classe \mathcal{A}_1 . Perceba que $\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}} \rangle \cong 1$ para \mathbf{x} próximo da classe \mathcal{A}_1 .

Figura 6.5: A Figura 6.5(a) apresenta o vetor $\mathbf{h}'_{\mathcal{A}}$ ($\mathbf{h}_{\mathcal{A}}$ no domínio transformado) utilizado para detectar a classe \mathcal{A}_1 , seu valor é igual média da classe \mathcal{A}_1 transformada (veja a Equação (6.41)). O vetor $\mathbf{h}_{\mathcal{A}}$ será igual a $\mathbf{h}_{\mathcal{A}} = (\mathbf{Q}^{*t}) \mathbf{h}'_{\mathcal{A}}$ (observe a Equação (6.37)) onde $\mathbf{Q} = \boldsymbol{\Lambda}^{-1/2} \boldsymbol{\Phi}^{*t}$, logo, $\mathbf{h}_{\mathcal{A}} = \boldsymbol{\Phi} \boldsymbol{\Lambda}^{-1/2} \mathbf{h}'_{\mathcal{A}}$. A Figura 6.5(b) mostra $\boldsymbol{\Lambda}^{-1/2} \mathbf{h}'_{\mathcal{A}}$ sendo um passo intermediário para obtermos $\mathbf{h}_{\mathcal{A}}$. Por fim, na Figura 6.5(c), temos o vetor $\mathbf{h}_{\mathcal{A}}$ obtido através da rotação de $\boldsymbol{\Lambda}^{-1/2} \mathbf{h}'_{\mathcal{A}}$. Se um vetor qualquer \mathbf{x} estiver próximo a classe \mathcal{A}_1 então $\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}} \rangle \cong 1$, caso contrário, temos $\langle \mathbf{x}, \mathbf{h}_{\mathcal{A}} \rangle \cong 0$.

6.4 Detectores por Produto Interno e Filtragem Discriminativa

Nesta seção verificaremos como o detector para um único padrão desenvolvido através da Filtragem Discriminativa por Restauração do Impulso e o detector para múltiplos padrões obtido através do Filtro OU apresentados no Capítulo 3 podem ser desenvolvidos como um caso particular do método de Detecção por Produto Interno (Seção 6.3).

6.4.1 Detectores por Produto Interno e Filtragem Discriminativa: Caso para Único Padrão

Nesta seção iremos verificar a relação existente entre a Filtragem Discriminativa por Restauração do Impulso (Capítulo 3) e o Detector por Produto Interno (Seção 6.3). Nosso objetivo, ao final desta seção será verificar como podemos obter o estimador linear \mathbf{A} o qual restaura um impulso¹ (Equação (3.13) repetida abaixo por comodidade²) a partir de vetores \mathbf{h}_A .

$$\hat{\delta} = \mathbf{A}\mathbf{g} = \mathbf{F}^t [\mathbf{F}\mathbf{F}^t + \mathbf{C}_b]^{-1} \mathbf{g}.$$

Para tal vamos considerar as classes; $A_0 = \{\mathbf{a} + \mathcal{N}_0\}$ formada por um único padrão com dimensão $d \times 1$ corrompido por ruído aditivo representado por uma variável aleatória \mathcal{N}_0 de média zero e covariância igual a \mathbf{R}_N ; A_k (com $k = 1, 2, \dots, d-1$) formadas por apenas um elemento, contendo o deslocamento circular do padrão \mathbf{a} corrompido por \mathcal{N}_k com média zero e covariância \mathbf{R}_N ; \mathcal{B} que concentra todos os demais padrões os quais desejamos rejeitar. Para condensar as expressões envolvendo deslocamento circular iremos utilizar a notação $D^{-k}\mathbf{a}$ para representar o deslocamento circular do vetor \mathbf{a} de k unidades³. Dessa forma:

- A classe A_0 será formada por um padrão \mathbf{a} com dimensão $d \times 1$ corrompido por \mathcal{N}_0 .

¹O produto $\mathbf{A}\mathbf{g}$ é uma estimativa do impulso.

²Lembrando que: \mathbf{g} resulta da concatenação das linhas transpostas de um padrão bidimensional g ; \mathbf{F} é uma matriz formada por \mathbf{g} e todos os seus deslocamentos circulares.

³Como exemplo se $\mathbf{a} = [a(1) a(2) a(3) \dots a(d)]^t$ então $D^{-1}\mathbf{a} = [a(d) a(1) a(2) \dots a(d-1)]^t$

Logo: $A_0 = \{D^0 \mathbf{a} + \mathcal{N}_0\} = \{\mathbf{a} + \mathcal{N}_0\}$.

- Genericamente, a classe A_i irá possuir somente um elemento igual $D^{-i} \mathbf{a}$ (deslocamento circular de \mathbf{a} de i unidades) corrompido por ruído aditivo \mathcal{N}_i (média zero e covariância $\mathbf{R}_{\mathcal{N}}$). Ao todo teremos d classes neste formato. Assim:
 $A_i = \{D^{-i} \mathbf{a} + \mathcal{N}_i\}$, com $i = \{0, 1, \dots, d-1\}$.
- A classe \mathcal{B} irá conter todos os demais padrões o qual desejamos rejeitar. Assim:
 $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_r\}$.

Vamos considerar que desejamos obter um classificador cuja saída será o produto interno da entrada \mathbf{x} (dimensão $d \times 1$) e do vetor \mathbf{h}_{A_i} , da seguinte forma:

$$\langle \mathbf{x}, \mathbf{h}_{A_i} \rangle = \mathbf{h}_{A_i}^{*t} \mathbf{x} = c.$$

O objetivo do classificador será detectar o padrão $D^{-i} \mathbf{a}$ (escalar $c = 1$) e rejeitar todas as suas versões deslocadas circularmente (exceto $D^{-i} \mathbf{a}$) bem como o conjunto \mathcal{B} que concentra os demais padrões (escalar $c = 0$). Para tal, utilizando o Detector por Produto Interno (Seção 6.3) apresentado na Equação (6.26) (repetida abaixo por comodidade) podemos encontrar o vetor \mathbf{h}_{A_i} .

$$\mathbf{h}_{A_i} = \{E[\mathbf{x}\mathbf{x}^{*t}]\}^{-1} E[\mathbf{x}c^{*t}]. \quad (6.42)$$

Os termos $E[\mathbf{x}c^{*t}]$ e $E[\mathbf{x}\mathbf{x}^{*t}]$ podem ser determinados de forma semelhante ao desenvolvimento realizado na Seção 6.1. Para $E[\mathbf{x}\mathbf{x}^{*t}]$ (observe as Equações (6.7) e (6.8)) teremos:

$$\begin{aligned} E[\mathbf{x}\mathbf{x}^{*t}] &= E[\mathbf{x}\mathbf{x}^{*t}|\mathcal{B}]p(\mathcal{B}) + E[\mathbf{x}\mathbf{x}^{*t}|A_0]p_0 + E[\mathbf{x}\mathbf{x}^{*t}|A_1]p_1 \\ &\quad + E[\mathbf{x}\mathbf{x}^{*t}|A_2]p_2 + \dots + E[\mathbf{x}\mathbf{x}^{*t}|A_{d-1}]p_{d-1} \\ &= p(\mathcal{B})\frac{1}{r} \sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^{*t} + p_0 E[(\mathbf{a} + \mathcal{N}_0)(\mathbf{a} + \mathcal{N}_0)^{*t}] \\ &\quad + p_1 E[(D^{-1}\mathbf{a} + \mathcal{N}_1)(D^{-1}\mathbf{a} + \mathcal{N}_1)^{*t}] + \dots \\ &\quad + p_{d-1} E[(D^{-(d-1)}\mathbf{a} + \mathcal{N}_{d-1})(D^{-(d-1)}\mathbf{a} + \mathcal{N}_{d-1})^{*t}]. \end{aligned} \quad (6.43)$$

Considerando $(D^{-k}\mathbf{a})$ e \mathcal{N}_k temos:

$$\begin{cases} E[(D^{-k}\mathbf{a})\mathcal{N}_k^{*t}] = (D^{-k}\mathbf{a})E[\mathcal{N}_k]^{*t} = 0 \\ E[\mathcal{N}_k(D^{-k}\mathbf{a})^{*t}] = E[\mathcal{N}_k](D^{-k}\mathbf{a})^{*t} = 0 \end{cases}, \quad (6.44)$$

Substituindo as Equações em (6.44) na Equação (6.43) obtemos:

$$\begin{aligned} E[\mathbf{xx}^{*t}] &= \\ &= p(\mathcal{B})\frac{1}{r}\sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^{*t} + p_0 \mathbf{a} \mathbf{a}^{*t} + p_0 E[\mathcal{N}_0 \mathcal{N}_0^{*t}] \\ &\quad + p_1 (D^{-1}\mathbf{a})(D^{-1}\mathbf{a})^{*t} + p_1 E[\mathcal{N}_1 \mathcal{N}_1^{*t}] + \dots \\ &\quad + p_{d-1} (D^{-(d-1)}\mathbf{a})(D^{-(d-1)}\mathbf{a})^{*t} + p_{d-1} E[\mathcal{N}_{d-1} \mathcal{N}_{d-1}^{*t}] \\ &= p(\mathcal{B})\frac{1}{r}\sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^{*t} + \sum_{j=1}^{d-1} p_j (D^{-j}\mathbf{a})(D^{-j}\mathbf{a})^{*t} + \sum_{j=1}^{d-1} p_j E[\mathcal{N}_j \mathcal{N}_j^{*t}]. \end{aligned} \quad (6.45)$$

Se os vetores $\mathbf{b}_1, \dots, \mathbf{b}_r$ são as realizações possíveis para a classe \mathcal{B} então o termo $\frac{1}{r}\sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^{*t}$ (primeira parcela da Equação (6.45)) é igual a matriz de autocorrelação $\mathbf{R}_{\mathcal{B}}$, ou seja:

$$\mathbf{R}_{\mathcal{B}} = \frac{1}{r}\sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^{*t}. \quad (6.46)$$

Podemos considerar o padrão \mathbf{a} (dimensão $d \times 1$) e seus deslocamentos circulares equiprováveis⁴. Dessa forma:

$$p_0 = p_1 = \dots = p_{d-1} = \frac{1 - p(\mathcal{B})}{d} = p. \quad (6.47)$$

Aplicando as Equações (6.47) e (6.46) na Equação (6.45) e considerando que $\mathbf{R}_{\mathcal{N}} = E[\mathcal{N}_j \mathcal{N}_j^{*t}]$, temos:

$$E[\mathbf{xx}^{*t}] = p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + p \sum_{j=1}^{d-1} (D^{-j}\mathbf{a})(D^{-j}\mathbf{a})^{*t} + p d \mathbf{R}_{\mathcal{N}}. \quad (6.48)$$

A Equação (6.48) pode ser representada como:

$$E[\mathbf{xx}^{*t}] = p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + p \mathbf{F}\mathbf{F}^{*t} + p d \mathbf{R}_{\mathcal{N}}, \quad (6.49)$$

⁴O que é razoável na prática pois o produto interno será aplicado deslocando-se o vetor \mathbf{h}_{A_0} sobre uma janela de tamanho d do sinal de entrada \mathbf{x} .

onde:

$$\mathbf{F} = \begin{bmatrix} \mathbf{a} & D^{-1}\mathbf{a} & \dots & D^{-(d-1)}\mathbf{a} \end{bmatrix}. \quad (6.50)$$

Perceba que a matriz \mathbf{F} , definida na Equação (6.50), é composta por um padrão \mathbf{a} e todas as suas versões deslocadas circularmente, ou seja, temos a estrutura da matriz apresentada na Equação (3.5) (página 20)⁵.

Para determinarmos $E[\mathbf{x}c^{*t}]$ do detector do padrão $D^{-i}\mathbf{a}$ (observe a Equação (6.9)) devemos lembrar que $c = 1$ se \mathbf{x} for igual à $A_i = \{D^{-i}\mathbf{a} + \mathcal{N}_i\}$ e $c = 0$ caso contrário. As probabilidades são iguais a p (veja a Equação (6.47)). Dessa forma:

$$\begin{aligned} E[\mathbf{x}c^{*t}] &= \sum_{j=0}^{d-1} E[Xc^{*t}|A_j] p \\ &= E[\mathbf{x}c^{*t}|A_i] p + 0 p + \dots + 0 p \\ &= p E[D^{-i}\mathbf{a} + \mathcal{N}_i] \\ &= p (D^{-i}\mathbf{a} + E[\mathcal{N}_i]) \\ &= p (D^{-i}\mathbf{a}). \end{aligned} \quad (6.51)$$

Substituindo os resultados das Equações (6.51) e (6.49) na Equação (6.42) temos a expressão final para o vetor \mathbf{h}_{A_i} sendo igual a:

$$\mathbf{h}_{A_i} = \{p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + p\mathbf{F}\mathbf{F}^{*t} + p d \mathbf{R}_{\mathcal{N}}\}^{-1} (D^{-i}\mathbf{a}) p. \quad (6.52)$$

Dessa forma o classificador com saída igual a $\langle \mathbf{x}, \mathbf{h}_{A_i} \rangle$ (com \mathbf{h}_{A_i} definido na Equação (6.52)) irá detectar o padrão $D^{-i}\mathbf{a}$ e rejeitar todos os outros (inclusive os deslocamentos circulares exceto $D^{-i}\mathbf{a}$). Podemos compor uma matriz \mathbf{H} formada por i vetores $(\mathbf{h}_{A_i})^{*t}$, conforme a Equação (6.52), concatenados linha a linha da

⁵A matriz definida na Equação (3.5) possui a mesma designação.

seguinte forma:

$$\begin{aligned}
\mathbf{H} &= \begin{bmatrix} (\mathbf{h}_{A_0})^{*t} \\ (\mathbf{h}_{A_1})^{*t} \\ \vdots \\ (\mathbf{h}_{A_{d-1}})^{*t} \end{bmatrix} \\
&= \begin{bmatrix} (D^0 \mathbf{a})^{*t} p \{p(\mathcal{B}) \mathbf{R}_B + p \mathbf{F}\mathbf{F}^{*t} + p d \mathbf{R}_N\}^{-1} \\ (D^{-1} \mathbf{a})^{*t} p \{p(\mathcal{B}) \mathbf{R}_B + p \mathbf{F}\mathbf{F}^{*t} + p d \mathbf{R}_N\}^{-1} \\ \vdots \\ (D^{-(d-1)} \mathbf{a})^{*t} p \{p(\mathcal{B}) \mathbf{R}_B + p \mathbf{F}\mathbf{F}^{*t} + p d \mathbf{R}_N\}^{-1} \end{bmatrix} \\
&= p \begin{bmatrix} \mathbf{a}^{*t} \\ (D^{-1} \mathbf{a})^{*t} \\ \vdots \\ (D^{-(d-1)} \mathbf{a})^{*t} \end{bmatrix} \{p(\mathcal{B}) \mathbf{R}_B + p \mathbf{F}\mathbf{F}^{*t} + p d \mathbf{R}_N\}^{-1}. \quad (6.53)
\end{aligned}$$

Perceba que a Equação (6.53) pode ser reescrita em função da matriz \mathbf{F} definida na Equação (6.50). Dessa forma temos:

$$\mathbf{H} = p \mathbf{F}^{*t} \{p(\mathcal{B}) \mathbf{R}_B + p \mathbf{F}\mathbf{F}^{*t} + p d \mathbf{R}_N\}^{-1} \quad (6.54)$$

Manipulando a Equação (6.54) e substituindo o valor da probabilidade p (Equação (6.47)) temos a forma final da matriz \mathbf{H} :

$$\begin{aligned}
\mathbf{H} &= p \mathbf{F}^{*t} \{p(\mathcal{B}) \mathbf{R}_B + p d \mathbf{R}_N + p \mathbf{F}\mathbf{F}^{*t}\}^{-1} \\
&= pp^{-1} \mathbf{F}^{*t} \left\{ \frac{p(\mathcal{B})}{p} \mathbf{R}_B + d \mathbf{R}_N + \mathbf{F}\mathbf{F}^{*t} \right\}^{-1} \\
&= \mathbf{F}^{*t} \left\{ d \frac{p(\mathcal{B})}{1 - p(\mathcal{B})} \mathbf{R}_B + d \mathbf{R}_N + \mathbf{F}\mathbf{F}^{*t} \right\}^{-1} \\
&= \mathbf{F}^{*t} \left\{ d \left(\frac{p(\mathcal{B})}{1 - p(\mathcal{B})} \mathbf{R}_B + \mathbf{R}_N \right) + \mathbf{F}\mathbf{F}^{*t} \right\}^{-1}. \quad (6.55)
\end{aligned}$$

Vamos repetir, por comodidade, a equação do estimador linear, apresentada originalmente no Capítulo 3.

$$\mathbf{A} = \mathbf{F}^t [\mathbf{F}\mathbf{F}^t + \mathbf{C}_b]^{-1}.$$

Na formulação por Filtragem Discriminativa a matriz \mathbf{H} equivale ao estimador linear \mathbf{A} quando:

$$\mathbf{C}_b = d \left(\frac{p(\mathcal{B})}{1 - p(\mathcal{B})} \mathbf{R}_B + \mathbf{R}_N \right).$$

Para finalizar, podemos verificar que \mathbf{H} irá restaurar aproximadamente um impulso (no sentido do erro médio quadrático) se definirmos um classificador com entrada \mathbf{x} e saída igual ao produto matricial entre \mathbf{H} e \mathbf{x} , ou seja, $\mathbf{H}\mathbf{x}$. Fazendo a entrada \mathbf{x} igual ao padrão \mathbf{a} (mesmo que $D^0\mathbf{a}$), lembrando que $\langle \mathbf{a}, \mathbf{h}_{A_0} \rangle \cong 1$ (no sentido do erro médio quadrático) teremos:

$$\mathbf{H}\mathbf{x} = \mathbf{H}\mathbf{a} = \begin{bmatrix} (\mathbf{h}_{A_0})^{*t} \\ (\mathbf{h}_{A_1})^{*t} \\ \vdots \\ (\mathbf{h}_{A_{d-1}})^{*t} \end{bmatrix} \mathbf{a} \cong \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Assim quando $\mathbf{x} = \mathbf{a}$ temos que a saída do classificador, dada por $\mathbf{H}\mathbf{x}$, irá restaurar aproximadamente um impulso. A seguir verificaremos a relação existente entre os Detectores por Produto Interno e o Filtro OU.

6.4.2 Detectores por Produto Interno e Filtro OU: caso para Múltiplos Padrões

Nesta seção iremos demonstrar a relação existente entre os Detectores por Produto Interno (Seção 6.3) e o detector para múltiplos padrões denominado Filtro OU (Seção 3.3). Nosso objetivo é mostrar que os vetores \mathbf{h}_A podem formar o estimador linear (matriz \mathbf{A}) representado através da Equação (3.24) (repetida abaixo por comodidade⁶).

$$\hat{\delta} = \mathbf{A}\mathbf{g} = \sum_{i=1}^m \{p_i \mathbf{F}_i^t\} \left\{ \mathbf{C}_b + \sum_{i=1}^m p_i [\mathbf{F}_i \mathbf{F}_i^t] \right\}^{-1} \mathbf{g}.$$

Para tal vamos considerar as classes; $A_{ij} = \{D^{-j}\mathbf{a}_i + \mathcal{N}_{ij}\}$ formadas por um único padrão de dimensão $d \times 1$ corrompido por ruído aditivo representado

⁶Lembrando que: \mathbf{g} resulta da concatenação das linhas transpostas de um padrão bidimensional g ; o número de padrões que desejamos detectar é igual a m ; \mathbf{F}_i indica a matriz \mathbf{F} associada a cada padrão i ; a probabilidade de cada padrão i é igual a p_i .

por uma variável aleatória \mathcal{N}_{ij} de média zero e covariância igual a $\mathbf{R}_{\mathcal{N}}$ onde $D^{-j}\mathbf{a}_i$ representa o deslocamento circular de \mathbf{a}_i de j unidades⁷ e; \mathcal{B} que concentra todos os demais padrões os quais desejamos rejeitar. Ao todo temos m padrões, ou seja, $i = \{1, 2, \dots, m\}$. Considere que desejamos obter um classificador cuja saída será o produto interno da entrada \mathbf{x} (dimensão $d \times 1$) e do vetor \mathbf{h}_{A_j} , da seguinte forma:

$$\langle \mathbf{x}, \mathbf{h}_{A_j} \rangle = \mathbf{h}_{A_j}^{*t} \mathbf{x} = c.$$

O objetivo do classificador será detectar todos os $i = \{1, \dots, m\}$ padrões para um deslocamento circular fixo igual a j , ou seja, iremos detectar a classe A_j igual a $A_j = \{A_{1j}, A_{2j}, \dots, A_{mj}\}$. Podemos usar o Detector por Produto Interno apresentado na Seção 6.3 (repetido abaixo por comodidade) para detectar A_j .

$$\mathbf{h}_{A_j} = \{E[\mathbf{xx}^{*t}]\}^{-1} E[Xc^{*t}]. \quad (6.56)$$

Assim como na seção anterior precisamos determinar os termos $E[\mathbf{xc}^{*t}]$ e $E[\mathbf{xx}^{*t}]$. Eles podem ser desenvolvidos de forma semelhante aos da Seção 6.4.1. Para $E[\mathbf{xx}^{*t}]$ temos as classes A_{ij} (com $i = \{1, \dots, m\}$ e $j = \{0, \dots, d-1\}$) e a classe \mathcal{B} :

$$\begin{aligned} E[\mathbf{xx}^{*t}] &= \\ &= E[\mathbf{xx}^{*t}|\mathcal{B}]p(\mathcal{B}) + \sum_{i=1}^m \sum_{j=0}^{d-1} E[\mathbf{xx}^{*t}|A_{ij}]p_{ij} \\ &= p(\mathcal{B})\frac{1}{r} \sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^{*t} + \sum_{i=1}^m \sum_{j=0}^{d-1} p_{ij} E[(D^{-j}\mathbf{a}_i + \mathcal{N}_{ij})(D^{-j}\mathbf{a}_i + \mathcal{N}_{ij})^{*t}] \\ &= p(\mathcal{B})\frac{1}{r} \sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^{*t} + \sum_{i=1}^m \sum_{j=0}^{d-1} p_{ij} E[(D^{-j}\mathbf{a}_i + \mathcal{N}_{ij})(D^{-j}\mathbf{a}_i + \mathcal{N}_{ij})^{*t}]. \end{aligned} \quad (6.57)$$

Considerando $(D^{-k}\mathbf{a}_l)$ e \mathcal{N}_{kl} temos que:

$$\begin{cases} E[(D^{-k}\mathbf{a}_l) \mathcal{N}_{kl}^{*t}] = (D^{-k}\mathbf{a}_l) E[\mathcal{N}_{kl}^{*t}] = 0 \\ E[\mathcal{N}_{kl} (D^{-k}\mathbf{a}_l)^{*t}] = E[\mathcal{N}_{kl}] (D^{-k}\mathbf{a}_l)^{*t} = 0 \end{cases}. \quad (6.58)$$

⁷Veja um exemplo na página 116.

Podemos simplificar a Equação (6.57) utilizando as Equações em 6.58 (lembrando que $\mathbf{R}_{\mathcal{N}} = E[\mathcal{N}_{ij}\mathcal{N}_{ij}^{*t}]$). Dessa forma:

$$\begin{aligned}
E[\mathbf{xx}^{*t}] &= \\
&= p(\mathcal{B})\frac{1}{r}\sum_{j=1}^r \mathbf{b}_j \mathbf{b}_j^{*t} + \sum_{i=1}^m \sum_{j=0}^{d-1} p_{ij}(D^{-j}\mathbf{a}_i)(D^{-j}\mathbf{a}_i)^{*t} + \sum_{i=1}^m \sum_{j=0}^{d-1} p_{ij}E[\mathcal{N}_{ij}\mathcal{N}_{ij}^{*t}] \\
&= p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + \sum_{i=1}^m \sum_{j=0}^{d-1} p_{ij}(D^{-j}\mathbf{a}_i)(D^{-j}\mathbf{a}_i)^{*t} + \sum_{i=1}^m \sum_{j=0}^{d-1} p_{ij}\mathbf{R}_{\mathcal{N}}. \tag{6.59}
\end{aligned}$$

Podemos considerar um dado padrão \mathbf{a}_i e todos os seus j deslocamentos circulares equiprováveis. Assim, podemos associar uma única probabilidade p'_i para um padrão \mathbf{a}_i e todos os seus deslocamentos circulares. Dessa forma:

$$\begin{cases} p'_i = \sum_{j=0}^{d-1} p_{ij}, \text{ se } p_{ik} = p_{ir} \forall k, r \\ p_{ij} = \frac{p'_i}{d} \\ \sum_{i=1}^m p'_i = 1 - p(\mathcal{B}). \end{cases} \tag{6.60}$$

Aplicando as Equações em (6.60) na Equação (6.59) obtemos:

$$\begin{aligned}
E[\mathbf{xx}^{*t}] &= \\
&= p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + \frac{1}{d}\sum_{i=1}^m p'_i \sum_{j=0}^{d-1} (D^{-j}\mathbf{a}_i)(D^{-j}\mathbf{a}_i)^{*t} + \frac{1}{d}\sum_{i=1}^m p'_i \sum_{j=0}^{d-1} \mathbf{R}_{\mathcal{N}} \\
&= p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + \frac{1}{d}\sum_{i=1}^m p'_i \sum_{j=0}^{d-1} (D^{-j}\mathbf{a}_i)(D^{-j}\mathbf{a}_i)^{*t} + \frac{1}{d}\sum_{i=1}^m p'_i d \mathbf{R}_{\mathcal{N}} \\
&= p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + \frac{1}{d}\sum_{i=1}^m p'_i \sum_{j=0}^{d-1} (D^{-j}\mathbf{a}_i)(D^{-j}\mathbf{a}_i)^{*t} + (1 - p(\mathcal{B}))\mathbf{R}_{\mathcal{N}}. \tag{6.61}
\end{aligned}$$

A Equação (6.61) na forma matricial é igual a:

$$E[\mathbf{xx}^{*t}] = p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + \frac{1}{d}\sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} + (1 - p(\mathcal{B}))\mathbf{R}_{\mathcal{N}}, \tag{6.62}$$

onde:

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{a}_i & D^{-1}\mathbf{a}_i & \dots & D^{-(d-1)}\mathbf{a}_i \end{bmatrix}. \tag{6.63}$$

Genericamente podemos determinar o termo $E[\mathbf{x}c^{*t}]$ para o caso do detector dos padrões $D^{-j}\mathbf{a}_i$ (onde $i = \{1, \dots, m\}$) com deslocamento circular j fixo. Neste caso, devemos observar que $c = 1$ para $\mathbf{x} \in A_j = \{A_{1j}, A_{2j}, \dots, A_{mj}\}$ e $c = 0$ caso contrário. Dessa forma:

$$\begin{aligned}
E[\mathbf{x}c^{*t}] &= \\
&= E[\mathbf{x}c^{*t}|A_j] p(A_j) \\
&= p(A_j) \{ E[\mathbf{x}c^{*t}|A_{1j}] p(A_{1j}|A_j) + E[\mathbf{x}c^{*t}|A_{2j}] p(A_{2j}|A_j) \\
&\quad + \dots + E[\mathbf{x}c^{*t}|A_{mj}] p(A_{mj}|A_j) \} \\
&= p_{1j} E[D^{-j}\mathbf{a}_1 + \mathcal{N}_{1j}] + p_{2j} E[D^{-j}\mathbf{a}_2 + \mathcal{N}_{2j}] + \dots + p_{mj} E[D^{-j}\mathbf{a}_m + \mathcal{N}_{mj}] \\
&= \sum_{i=1}^m p_{ij} \{ (D^{-j}\mathbf{a}_i) + E[\mathcal{N}_{ij}] \} \\
&= \frac{1}{d} \sum_{i=1}^m p'_i (D^{-j}\mathbf{a}_i). \tag{6.64}
\end{aligned}$$

Substituindo os resultados das Equações (6.64) e (6.62) na Equação (6.56) temos a expressão final para o vetor \mathbf{h}_{A_j} sendo igual a:

$$\mathbf{h}_{A_j} = \left\{ p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + (1 - p(\mathcal{B}))\mathbf{R}_{\mathcal{N}} + \frac{1}{d} \sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1} \frac{1}{d} \sum_{i=1}^m p'_i (D^{-j}\mathbf{a}_i). \tag{6.65}$$

Dessa forma o classificador com saída igual a $\langle \mathbf{x}, \mathbf{h}_{A_j} \rangle$ (com \mathbf{h}_{A_j} definido na Equação (6.65)) irá detectar os padrões $D^{-j}\mathbf{a}_1, D^{-j}\mathbf{a}_2, \dots, D^{-j}\mathbf{a}_m$ e rejeitar todos os outros. Podemos compor uma matriz \mathbf{H} formada por j vetores $(\mathbf{h}_{A_j})^{*t}$ (com

$j = \{0, 1, \dots, d-1\}$ concatenados linha a linha da seguinte forma:

$$\begin{aligned}
\mathbf{H} &= \\
&= \begin{bmatrix} (\mathbf{h}_{A_0})^{*t} \\ (\mathbf{h}_{A_1})^{*t} \\ \vdots \\ (\mathbf{h}_{A_{d-1}})^{*t} \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{d} \sum_{i=1}^m p'_i (D^0 \mathbf{a}_i) \left\{ p(\mathcal{B}) \mathbf{R}_B + (1-p(\mathcal{B})) \mathbf{R}_N + \frac{1}{d} \sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1} \\ \frac{1}{d} \sum_{i=1}^m p'_i (D^{-1} \mathbf{a}_i) \left\{ p(\mathcal{B}) \mathbf{R}_B + (1-p(\mathcal{B})) \mathbf{R}_N + \frac{1}{d} \sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1} \\ \vdots \\ \frac{1}{d} \sum_{i=1}^m p'_i (D^{-(d-1)} \mathbf{a}_i) \left\{ p(\mathcal{B}) \mathbf{R}_B + (1-p(\mathcal{B})) \mathbf{R}_N + \frac{1}{d} \sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1} \end{bmatrix} \\
&= \frac{1}{d} \sum_{i=1}^m \left\{ p'_i \begin{bmatrix} \mathbf{a}_i^{*t} \\ (D^{-1} \mathbf{a}_i)^{*t} \\ \vdots \\ (D^{-(d-1)} \mathbf{a}_i)^{*t} \end{bmatrix} \right\} \left\{ p(\mathcal{B}) \mathbf{R}_B + (1-p(\mathcal{B})) \mathbf{R}_N + \frac{1}{d} \sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1}.
\end{aligned} \tag{6.66}$$

A Equação (6.66) pode ser apresentada em função da matriz \mathbf{F}_i definida na Equação (6.63). Dessa forma temos:

$$\mathbf{H} = \frac{1}{d} \sum_{i=1}^m \left\{ p'_i \mathbf{F}_i^{*t} \right\} \left\{ p(\mathcal{B}) \mathbf{R}_B + (1-p(\mathcal{B})) \mathbf{R}_N + \frac{1}{d} \sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1} \tag{6.67}$$

Manipulando a Equação (6.66) temos a forma final da matriz \mathbf{H} :

$$\begin{aligned}
\mathbf{H} &= \frac{1}{d} d \sum_{i=1}^m \left\{ p'_i \mathbf{F}_i^{*t} \right\} \left\{ d p(\mathcal{B}) \mathbf{R}_B + d (1-p(\mathcal{B})) \mathbf{R}_N + \sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1} \\
&= \sum_{i=1}^m \left\{ p'_i \mathbf{F}_i^{*t} \right\} \left\{ d [p(\mathcal{B}) \mathbf{R}_B + (1-p(\mathcal{B})) \mathbf{R}_N] + \sum_{i=1}^m p'_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1}.
\end{aligned} \tag{6.68}$$

O estimador linear \mathbf{A} para o Filtro OU (abaixo por comodidade) mostrado originalmente no Capítulo 3 é igual a:

$$\mathbf{A} = \sum_{i=1}^m \left\{ p_i \mathbf{F}_i^t \right\} \left\{ \mathbf{C}_b + \sum_{i=1}^m p_i \mathbf{F}_i \mathbf{F}_i^t \right\}^{-1}. \tag{6.69}$$

Podemos relacionar as probabilidades p'_i e p_i respectivamente nas Equações (6.71) e (6.69) da seguinte maneira:

- Seja $A = \bigcup_{i=1}^m A_i$. Neste caso $p(A) = 1 - p(\mathcal{B})$. A probabilidade⁸ p'_i será igual a $p'_i = p(A_i)$.
- A probabilidade p_i será igual a $p(A_i|A)$. Como $p(A_i) = p(A_i|A)p(A)$ então:

$$\begin{aligned} p(A_i) &= p(A_i|A)p(A) = p_i(1 - p(\mathcal{B})) \\ p'_i &= p_i(1 - p(\mathcal{B})). \end{aligned} \quad (6.70)$$

Substituindo a Equação (6.70) na Equação (6.71) temos:

$$\begin{aligned} \mathbf{H} &= \sum_{i=1}^m \left\{ p_i(1 - p(\mathcal{B})) \mathbf{F}_i^{*t} \right\} \left\{ d[p(\mathcal{B})\mathbf{R}_{\mathcal{B}} + (1 - p(\mathcal{B}))\mathbf{R}_{\mathcal{N}}] \right. \\ &\quad \left. + \sum_{i=1}^m p_i(1 - p(\mathcal{B})) \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1} \\ &= \sum_{i=1}^m \left\{ p_i \mathbf{F}_i^{*t} \right\} \left\{ d \left(\frac{p(\mathcal{B})}{1 - p(\mathcal{B})} \mathbf{R}_{\mathcal{B}} + \mathbf{R}_{\mathcal{N}} \right) + \sum_{i=1}^m p_i \mathbf{F}_i \mathbf{F}_i^{*t} \right\}^{-1}. \end{aligned} \quad (6.71)$$

O estimador linear \mathbf{A} para o Filtro OU equivale à matriz \mathbf{H} (Equações (6.69) e (6.71)) para a mesma matriz $\mathbf{C}_{\mathbf{b}}$ (abaixo por comodidade) apresentada na Seção 6.4.1 o que prova nossa verificação.

$$\mathbf{C}_{\mathbf{b}} = d \left(\frac{p(\mathcal{B})}{1 - p(\mathcal{B})} \mathbf{R}_{\mathcal{B}} + \mathbf{R}_{\mathcal{N}} \right).$$

Para finalizar, semelhante ao desenvolvimento realizado no final da Seção 6.4.1 podemos verificar que \mathbf{H} restaura aproximadamente um impulso (no sentido do erro médio quadrático) se definirmos um classificador com entrada \mathbf{x} e saída igual a $\mathbf{H}\mathbf{x}$. Fazendo \mathbf{x} igual a um padrão \mathbf{a}_i (mesmo que $D^0\mathbf{a}_i$) com $i = \{1, \dots, m\}$,

⁸Associada ao padrão a_i e todos os seus deslocamentos circulares.

lembrando que $\langle \mathbf{a}_i, \mathbf{h}_{A_0} \rangle \cong 1$ teremos:

$$\mathbf{H} \mathbf{x} = \mathbf{H} \mathbf{a} = \begin{bmatrix} (\mathbf{h}_{A_0})^{*t} \\ (\mathbf{h}_{A_1})^{*t} \\ \vdots \\ (\mathbf{h}_{A_{d-1}})^{*t} \end{bmatrix} \quad \mathbf{a}_i \cong \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Assim quando \mathbf{x} for igual a qualquer padrão \mathbf{a}_i com $i = \{1, \dots, m\}$ temos que a saída do classificador, dada por $\mathbf{H}\mathbf{x}$, irá restaurar aproximadamente um impulso (no sentido do erro médio quadrático).

6.5 Detectores por Produto Interno para um Pequeno Número de Imagens (IPD-hd)

Na Seção 6.3 apresentamos a solução para calcularmos o vetor \mathbf{h}_A de tal forma a obtermos a saída do classificador através do produto interno de \mathbf{h}_A com um vetor de entrada \mathbf{x} (mesmo que $\langle \mathbf{x}, \mathbf{h}_A \rangle$) onde \mathbf{x} é uma realização da variável aleatória \mathcal{X} . A solução para \mathbf{h}_A , apresentada na Equação (6.27) será válida quando o número total de padrões (igual a $r + \sum_{i=1}^n L_i$) for maior ou igual à dimensão do sinal de entrada (igual a d), conforme comentamos ao final das Seções 6.1, 6.2 e 6.3. Caso contrário, precisamos modificar a abordagem do nosso problema para encontrarmos uma solução. Para tal, vamos considerar as classes A_i com $i = \{1, 2, \dots, n\}$. Cada classe A_i possui L_i elementos (cada um com dimensão $d \times 1$) onde:

$$A_i = \{\mathbf{a}_{i1}, \mathbf{a}_{i2}, \dots, \mathbf{a}_{iL_i}\}.$$

Podemos compor uma matriz \mathbb{A} com dimensão $d \times N$ onde $d > N$ através das matrizes \mathbf{A}_i da seguinte maneira:

$$\mathbb{A} = [\mathbf{A}_1 \mathbf{A}_2 \dots \mathbf{A}_n]_{d \times N}, \quad (6.72)$$

onde a matriz \mathbf{A}_i será formada através da concatenação dos L_i elementos da classe

A_i conforme a definição abaixo:

$$\mathbf{A}_i = [\mathbf{a}_{i1} \ \mathbf{a}_{i2} \ \dots \ \mathbf{a}_{iL_i}],$$

perceba que dessa forma \mathbb{A} terá $N = \sum_{i=1}^n L_i$ colunas.

Nosso objetivo é obter um classificador na forma do vetor \mathbf{h}_{A_i} de tal forma que o produto matricial de $\mathbf{h}_{A_i}^{*t}$ com \mathbb{A} seja igual ao vetor $\mathbf{c} = [c_w]$ (dimensão $1 \times N$) com $w = \sum_{l=1}^{i-1} L_l + j$ onde c_w será igual a um (1) para todo j com $1 \leq j \leq L_i$ e zero (0) para os demais valores. Dessa forma:

$$\begin{aligned} \mathbf{h}_{A_i}^{*t} \mathbb{A} &= \mathbf{c}, \\ \text{onde } \mathbf{c} = [c_w] &= \begin{cases} 1, & \text{se } w = \sum_{l=1}^{i-1} L_l + j, \forall j \text{ com } 1 \leq j \leq L_i \\ 0, & \text{caso contrário} \end{cases} \end{aligned} \quad (6.73)$$

Assim, semelhante à Seção 6.3 devemos encontrar \mathbf{h}_{A_i} , na Equação (6.73), que produza o melhor classificador possível. Modificando a Equação (6.73) temos:

$$\begin{aligned} \mathbf{h}_{A_i}^{*t} \mathbb{A} &= \mathbf{c}, \\ \mathbb{A}^{*t} \mathbf{h}_{A_i} &= \mathbf{c}^{*t}. \end{aligned} \quad (6.74)$$

A Equação (6.74) possui solução exata para $d > N$, uma delas sendo igual à pseudo-inversa à direita⁹ [104, 105]. Assim, o vetor \mathbf{h}_{A_i} será igual a:

$$\mathbf{h}_{A_i} = \mathbb{A} (\mathbb{A}^{*t} \mathbb{A})^{-1} \mathbf{c}^{*t}. \quad (6.75)$$

Podemos verificar se uma entrada \mathbf{x} pertence à classe A_i . Neste caso, conforme a Equação (6.73) fazendo $\mathbb{A} = \mathbf{x}$ temos que o produto matricial $\mathbf{h}_{A_i} \mathbf{x}$ (equivalente ao produto interno de \mathbf{h}_{A_i} e \mathbf{x}) será igual a um (1) caso \mathbf{x} pertença a classe A_i e zero (0) caso \mathbf{x} não pertença a A_i .

A abordagem por IPD-hd, poderá ser usada quando não tivermos estatística suficiente dos padrões os quais desejamos detectar. A dimensão do padrão possui relação direta com a abordagem a ser utilizada.

Na prática, o caso em que $d > N$ será utilizado para um conjunto de treina-

⁹Neste caso a matriz $\mathbf{X}^{*t} \mathbf{X}$ com dimensão $N \times N$ é não singular.

mento apenas; ou seja, a solução é exata para o conjunto de treinamento. Porém, usamos o vetor \mathbf{h}_{A_i} para uma entrada não pertencente ao conjunto de treinamento; logo, a solução deixa de ser exata. Neste caso, a solução ótima será obtida através do IPD apresentado na Seção 6.3 que leva em conta as estatísticas. A desvantagem deste método é ter que inverter uma matriz $d \times d$ o que pode ser computacionalmente complexo para dimensão elevada. Assim, o método IPD-hd deve ser utilizado se a dimensão do padrão for maior que a quantidade total de elementos, caso contrário, utiliza-se o IPD apresentado na Seção 6.3. Uma forma de contornar isso é encontrar a matriz \mathbf{Q} tal que, dada a estatística dos padrões fora do conjunto de treinamento a serem classificados, a capacidade discriminatória é máxima.

Por fim, iremos verificar na próxima seção que os métodos *Minimum Average Correlation Energy Filter* (ou *MACE filter*) [32] e *Optimal Tradeoff Filter* (ou *OTF filter*) [31] são casos particulares do IPD-hd quando se usa a versão transformada da matriz \mathbf{A} igual a $\mathbf{A}' = \mathbf{Q}\mathbf{A}$.

6.5.1 Detectores por Produto Interno para um Pequeno Número de Imagens (IPD-hd) com Transformação Linear

Semelhante à Seção 6.3.1 podemos aplicar uma transformação na matriz \mathbf{A} com o objetivo de realizar um pré-processamento a fim de melhorar a classificação dos dados. Para tal, vamos considerar uma transformação linear complexa \mathbf{Q} de tal forma que:

$$\mathbf{A}' = \mathbf{Q} \mathbf{A}.$$

Podemos utilizar a Equação (6.75) para encontrarmos o vetor transformado \mathbf{h}'_{A_i} . Assim, podemos escrever:

$$\begin{aligned} \mathbf{h}'_{A_i} &= \mathbf{A}' (\mathbf{A}'^{*t} \mathbf{A}')^{-1} \mathbf{c}^{*t} \\ &= \mathbf{Q} \mathbf{A} ((\mathbf{Q}\mathbf{A})^{*t} \mathbf{Q}\mathbf{A})^{-1} \mathbf{c}^{*t} \\ &= \mathbf{Q} \mathbf{A} (\mathbf{A}^{*t} \mathbf{Q}^{*t} \mathbf{Q}\mathbf{A})^{-1} \mathbf{c}^{*t}. \end{aligned} \tag{6.76}$$

A Equação (6.76) define o vetor transformado \mathbf{h}'_{A_i} obtido quando \mathbf{A} sofre uma transformação linear $\mathbf{A}' = \mathbf{Q}\mathbf{A}$. A saída do classificador, no domínio transformado,

será igual a $(\mathbf{h}'_{A_i})^{*t} \mathbf{A}'$. Dessa forma:

$$\begin{aligned} (\mathbf{h}'_{A_i})^{*t} \mathbf{A}' &= [(\mathbf{h}'_{A_i})^{*t} \mathbf{Q}] \mathbf{A} \\ &= [\mathbf{Q}^{*t} \mathbf{h}'_{A_i}]^{*t} \mathbf{A}. \end{aligned} \quad (6.77)$$

Comparando a Equação (6.77) com a Equação (6.73)¹⁰ temos a seguinte alternativa à solução da Equação (6.75):

$$\mathbf{h}_{A_i} = \mathbf{Q}^{*t} \mathbf{h}'_{A_i}. \quad (6.78)$$

Substituindo a Equação (6.76) na Equação (6.78) obtemos:

$$\mathbf{h}_{A_i} = \mathbf{Q}^{*t} \mathbf{Q} \mathbf{A} (\mathbf{A}^{*t} \mathbf{Q}^{*t} \mathbf{Q} \mathbf{A})^{-1} \mathbf{c}^{*t}. \quad (6.79)$$

A Equação (6.79), define o vetor \mathbf{h}_{A_i} para o domínio original com \mathbf{A} modificado através da transformação linear \mathbf{Q} . Neste caso, a escolha da matriz \mathbf{Q} implica em uma solução para o vetor \mathbf{h}_{A_i} (caso exista a inversa de $\mathbf{A}^{*t} \mathbf{Q}^{*t} \mathbf{Q} \mathbf{A}$); dessa forma, temos infinitas soluções para \mathbf{h}_{A_i} . A transformação ótima seria a que realizasse o branqueamento [101,103] dos dados a fim de melhorar o poder discriminatório. Para tal, podemos utilizar a KLT, semelhante ao desenvolvimento feito na Seção 6.3.2.

Considere a matriz de autocorrelação de \mathcal{X} igual a $\mathbf{R}_{\mathcal{X}}$. Neste caso temos que $\mathbf{R}_{\mathcal{X}}$ não pode ser determinada através de $\mathbf{A} \mathbf{A}^{*t}$ pois $d > N$, ou seja, $\mathbf{R}_{\mathcal{X}}$ não poderá ser determinada a partir do conjunto de treinamento¹¹. Semelhante ao desenvolvimento da Equação (6.35) podemos considerar a KLT de \mathbf{v} (dimensão $d \times 1$) igual a \mathbf{V} e $\Phi = [\phi_1, \dots, \phi_d]$ igual à matriz de autovetores ortonormais de $\mathbf{R}_{\mathcal{X}}$. Assim:

$$\mathbf{V} = \Phi^{*t} \mathbf{v}, \quad \mathbf{v} = \Phi \mathbf{V}.$$

A relação entre \mathbf{A} , Φ e $\mathbf{R}_{\mathcal{X}}$ (observe as Equações (6.34) e (6.35)) será¹²:

$$\mathbf{R}_{\mathcal{X}} = \Phi \mathbf{A} \Phi^{*t}. \quad (6.80)$$

¹⁰Produto $\mathbf{h}'_{A_i} \mathbf{A} = \mathbf{c}$

¹¹Deve-se estimar $\mathbf{R}_{\mathcal{X}}$ de outra forma.

¹²A matriz \mathbf{A} é diagonal com seus elementos da diagonal principal sendo os autovalores $\lambda_1 \geq \dots \geq \lambda_d$ de $\mathbf{R}_{\mathcal{X}}$ correspondentes a cada autovetor de Φ (os valores λ_i correspondem a variância de \mathcal{X} [20, 85]).

A transformação ótima, ou seja, a que realiza o branqueamento será igual a:

$$\mathbf{Q} = \mathbf{\Lambda}^{-1/2} \mathbf{\Phi}^{*t}. \quad (6.81)$$

Substituindo a Equação (6.81) na Equação (6.79) e utilizando a Equação (6.80) temos:

$$\begin{aligned} \mathbf{h}_{A_i} &= (\mathbf{\Lambda}^{-1/2} \mathbf{\Phi}^{*t})^{*t} \mathbf{\Lambda}^{-1/2} \mathbf{\Phi}^{*t} \mathbb{A} [\mathbb{A}^{*t} (\mathbf{\Lambda}^{-1/2} \mathbf{\Phi}^{*t})^{*t} \mathbf{\Lambda}^{-1/2} \mathbf{\Phi}^{*t} \mathbb{A}]^{-1} \mathbf{c}^{*t} \\ &= \mathbf{\Phi} \mathbf{\Lambda}^{-1} \mathbf{\Phi}^{*t} \mathbb{A} (\mathbb{A}^{*t} \mathbf{\Phi} \mathbf{\Lambda}^{-1} \mathbf{\Phi}^{*t} \mathbb{A})^{-1} \mathbf{c}^{*t} \end{aligned} \quad (6.82)$$

$$\begin{aligned} &= (\mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Phi}^{*t})^{-1} \mathbb{A} [\mathbb{A}^{*t} (\mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Phi}^{*t})^{-1} \mathbb{A}]^{-1} \mathbf{c}^{*t} \\ &= (\mathbf{R}_{\mathcal{X}})^{-1} \mathbb{A} [\mathbb{A}^{*t} (\mathbf{R}_{\mathcal{X}})^{-1} \mathbb{A}]^{-1} \mathbf{c}^{*t}. \end{aligned} \quad (6.83)$$

Para encontrarmos \mathbf{h}_{A_i} temos duas opções, fornecidas através das Equações (6.82) e (6.83), da seguinte maneira:

- Conforme a Equação (6.83), precisamos da matriz inversa de $\mathbf{R}_{\mathcal{X}}$. Neste caso, como $d > N$ temos que a inversa de $\mathbf{R}_{\mathcal{X}}$ não existe se considerarmos apenas o conjunto de treinamento, dessa forma, a solução para \mathbf{h}_{A_i} será possível para uma estimativa de $\mathbf{R}_{\mathcal{X}}$.
- Manipulando a Equação (6.82) encontramos:

$$\begin{aligned} \mathbf{h}_{A_i} &= \mathbf{\Phi} \mathbf{\Lambda}^{-1} \mathbf{\Phi}^{*t} \mathbb{A} (\mathbb{A}^{*t} \mathbf{\Phi} \mathbf{\Lambda}^{-1} \mathbf{\Phi}^{*t} \mathbb{A})^{-1} \mathbf{c}^{*t} \\ &= \mathbf{\Phi} \mathbf{\Lambda}^{-1} \mathbf{\Phi}^{*t} \mathbb{A} [(\mathbf{\Phi}^{*t} \mathbb{A})^{*t} \mathbf{\Lambda}^{-1} \mathbf{\Phi}^{*t} \mathbb{A}]^{-1} \mathbf{c}^{*t}. \end{aligned} \quad (6.84)$$

Neste caso, a Equação (6.84) mostra que é necessário encontrar a KLT de \mathbb{A} igual a $\mathbf{\Phi}^{*t} \mathbb{A}$, ou seja, precisamos da estatística dos padrões a serem reconhecidos ou rejeitados.

Assim, a transformação ótima (branqueamento), apresentada na Equação (6.81) não pode ser aplicada apenas a partir do conjunto de treinamento. Uma possibilidade para contornar este problema seria usarmos a DFT para aproximar a KLT e tentar incorporar a estatística do conjunto de treinamento na aproximação

da matriz $\mathbf{\Lambda}$. Assim, a transformação \mathbf{Q} se tornaria:

$$\mathbf{Q} = \mathbf{P}^{-1/2}\mathbf{F}, \quad (6.85)$$

onde \mathbf{P} é diagonal e \mathbf{F} é a matriz DFT. Aplicando a transformação \mathbf{Q} apresentada na Equação (6.85) na Equação (6.79) temos:

$$\begin{aligned} \mathbf{h}_{A_i} &= (\mathbf{P}^{-1/2}\mathbf{F})^{*t} \mathbf{P}^{-1/2}\mathbf{F} \mathbf{\Lambda} [\mathbf{\Lambda}^{*t}(\mathbf{P}^{-1/2}\mathbf{F})^{*t}\mathbf{P}^{-1/2}\mathbf{F}\mathbf{\Lambda}]^{-1} \mathbf{c}^{*t} \\ \mathbf{h}_{A_i} &= \mathbf{F}^{*t}\mathbf{P}^{-1}\mathbf{F} \mathbf{\Lambda} [\mathbf{\Lambda}^{*t}\mathbf{F}^{*t}\mathbf{P}^{-1}\mathbf{F}\mathbf{\Lambda}]^{-1} \mathbf{c}^{*t} \\ \mathbf{F} \mathbf{h}_{A_i} &= \mathbf{P}^{-1}\mathbf{F} \mathbf{\Lambda} [(\mathbf{F}\mathbf{\Lambda})^{*t}\mathbf{P}^{-1}\mathbf{F}\mathbf{\Lambda}]^{-1} \mathbf{c}^{*t}. \end{aligned} \quad (6.86)$$

Perceba que os termos $\mathbf{F}\mathbf{h}_{A_i}$ e $\mathbf{F}\mathbf{\Lambda}$ na Equação (6.86) representam, respectivamente, a DFT de \mathbf{h}_{A_i} e $\mathbf{\Lambda}$.

Podemos encontrar na literatura métodos originados da Equação (6.86) onde a matriz \mathbf{P} incorpora as estatísticas dos padrões de variadas formas. Quando escolhermos \mathbf{P} como o espectro de potência médio da DFT do conjunto de treinamento (matriz $\mathbf{F}\mathbf{\Lambda}$) recaímos no método *Minimum Average Correlation Energy Filter* (ou *MACE filter*) [32], neste caso, a matriz \mathbf{P} nomeia-se \mathbf{D} . Considerando a matriz \mathbf{P} igual a $\mathbf{T} = \alpha\mathbf{D} + \sqrt{1 - \alpha^2} \mathbf{C}$ onde a matriz diagonal \mathbf{C} representa a densidade espectral de potência do ruído. O termo α regula a importância do ruído na estatística dos padrões, e provê uma certa tolerância ao ruído. Neste caso, recaímos no método *Optimal Tradeoff Filter* (ou OTF) [31] (outras referências para OTF e *MACE filter* em [106–111]).

6.6 Detector por Produto Interno com Dimensão Extra

O classificador obtido para o IPD, na forma de um vetor \mathbf{h} , proporciona uma saída próxima de um (1) caso um candidato \mathbf{x} seja o padrão o qual desejamos encontrar ou próxima de zero (0) caso contrário. Dessa forma, se \mathbf{x} é o padrão desejado,

$$\langle \mathbf{x}, \mathbf{h} \rangle = \mathbf{h}^t \mathbf{x} \cong 1$$

e, caso contrário,

$$\langle \mathbf{x}, \mathbf{h} \rangle = \mathbf{h}^t \mathbf{x} \cong 0.$$

Porém, note que não temos garantias de que as equações acima serão obedecidas caso outros candidatos estejam aproximadamente colineares em relação aos padrões de interesse. Este problema pode ser solucionado acrescentando-se uma dimensão extra, determinada utilizando-se os padrões de treinamento, de tal forma que eles possuam a mesma norma denominada por E_m (naturalmente, E_m será utilizada em todos os padrões de teste).

Podemos determinar E_m da seguinte maneira: vamos supor que temos N padrões \mathbf{x}_i , com $i = \{1, \dots, N\}$, para treinamento com dimensões $d \times 1$. Primeiramente, determinamos qual \mathbf{x}_i possui a maior norma ao quadrado, acrescida em 20%, da seguinte maneira:

$$E_m = 1,2 \max \{ \|\mathbf{x}_i\|^2 \}, \quad i = \{1, \dots, N\}. \quad (6.87)$$

A Equação (6.87) será utilizada para determinarmos a dimensão extra x_E que será acrescentada aos padrões de treinamento $\mathbf{x}_i = [x(1) \ x(2) \ \dots \ x(d)]$. O percentual de 20% consiste apenas em uma margem de segurança. Assim, formaremos novos vetores $\tilde{\mathbf{x}}_i$ da seguinte maneira:

$$\tilde{\mathbf{x}}_i = [x(1) \ x(2) \ \dots \ x(d) \ x_E(d+1)]^t. \quad (6.88)$$

Naturalmente, o valor de x_E na Equação (6.88) que proporciona que todos os $\tilde{\mathbf{x}}_i$ tenham norma ao quadrado E_m será:

$$x_E = \sqrt{(E_m - \|\mathbf{x}_i\|^2)}. \quad (6.89)$$

As Equações (6.87), (6.88) e (6.89) podem ser aplicadas nos padrões \mathbf{x}_i , pertencentes ao conjunto de treinamento, de tal forma que possamos obter vetores $\tilde{\mathbf{x}}_i$ com norma igual a E_m . Esta alteração permite que os padrões de treinamento $\tilde{\mathbf{x}}_i$ estejam distribuídos na superfície de uma hipersfera de dimensão $d+1$ e raio igual a E_m .

6.7 Detector por Produto Interno utilizando Análise de Componentes Principais (IPD-B-PCA)

Nesta seção, iremos apresentar como podemos utilizar os Detectores por Produto Interno em conjunto com a Análise de Componentes Principais (Seção 4.1). As motivações são semelhantes às expostas na Seção 4.4. Assim, nosso objetivo será aumentar a robustez dos detectores por produto interno através do projeto de detectores para as componentes principais (observe a Figura 6.6).

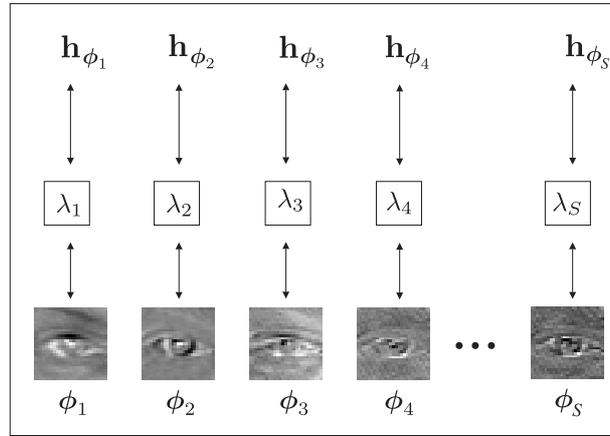


Figura 6.6: Projeto para S detectores por produto interno considerando-se S componentes principais. A base Φ possui um total de N componentes. O índice i das componentes principais ϕ_i indica a ordem decrescente de seus valores de variância (autovalores λ_i).

A formulação matemática do método IPD-B-PCA pode ser obtida conforme é descrito a seguir. Reproduzindo-se abaixo as Equações (4.4) até (4.7), por conveniência, temos que podemos encontrar Φ e Λ usando:

$$\boldsymbol{\mu}_{\mathcal{U}} = \frac{1}{M} \sum_{i=1}^M \mathbf{u}_i,$$

$$\mathbf{U} = [\mathbf{u}_1 - \boldsymbol{\mu}_{\mathcal{U}}, \dots, \mathbf{u}_M - \boldsymbol{\mu}_{\mathcal{U}}],$$

$$\boldsymbol{\Sigma}_{\mathcal{U}} = \frac{1}{M-1} \mathbf{U} \mathbf{U}^{*t},$$

$$\boldsymbol{\Lambda} = \Phi^{*t} \boldsymbol{\Sigma}_{\mathcal{U}} \Phi.$$

Nas equações acima, \mathcal{U} é variável aleatória com realizações iguais aos vetores $\mathbf{u}_1, \dots, \mathbf{u}_M$ (cada vetor \mathbf{u}_i será obtido concatenando-se as colunas do bloco i cujo

centro é o ponto fiducial de interesse), $\Phi = [\phi_1, \dots, \phi_N]$ são os *eigenpoints*, e a diagonal principal da matriz Λ , igual a $\lambda_1, \dots, \lambda_N$, corresponde aos autovetores.

Em nosso caso, desejamos reconhecer a componente ϕ_i e rejeitar as demais componentes. Para o projeto do vetor \mathbf{h} torna-se necessário definir as classes envolvidas. Vamos considerar três classes especificadas da seguinte maneira:

- A classe \mathcal{A}_1 será formada pela componente a qual desejamos detectar, ou seja, $\mathcal{A}_1 = \{\phi_i\}$.
- A classe \mathcal{A}_2 será formada por deslocamentos lineares de ϕ_i . Iremos utilizar a notação $\phi_i^{(n,m)}$ para representar o deslocamento linear da componente i de n pixels na direção horizontal e m na vertical. Assim, $\mathcal{A}_2 = \{\phi_i^{(n,m)}, \text{onde } -L \leq m, n \leq L, \text{ com } n, m \neq 0\}$. Onde L corresponde ao máximo deslocamento.
- A classe \mathcal{B} irá conter as outras componentes. Assim, $\mathcal{B} = \{\phi_j\}$, com $j \neq i$.

Os deslocamentos lineares $\phi_i^{(n,m)}$ da componente ϕ_i serão determinados utilizando-se a componente principal $\tilde{\phi}_i$ que será calculada utilizando-se um bloco com dimensões $(4L + 1) \times (4L + 1)$ quando o máximo deslocamento é de L (observe a Figura 6.7).

Note que, propositalmente, supomos que a classe \mathcal{B} é formada por componentes ortogonais à componente que desejamos detectar. Dessa forma, semelhante às observações realizadas no final da Seção 4.4, temos que os padrões que não são de interesse irão se situar no subespaço ortogonal a ϕ_i . Assim, o ruído associado a ϕ_i será considerado como a projeção de padrões típicos no subespaço ortogonal a ϕ_i .

A Equação (6.29) pode ser utilizada para determinarmos o detector por produto interno para as componentes principais. Assim, para nosso caso:

$$\mathbf{h}_{\phi_i} = \left\{ p_1 \mathbf{R}_{\mathcal{A}_1} + p_2 \mathbf{R}_{\mathcal{A}_2} + p(\mathcal{B}) \mathbf{R}_{\mathcal{B}} \right\}^{-1} p_1 \boldsymbol{\mu}_{\mathcal{A}_1}. \quad (6.90)$$

A Equação (6.90) pode ser escrita, em função dos elementos típicos das classes

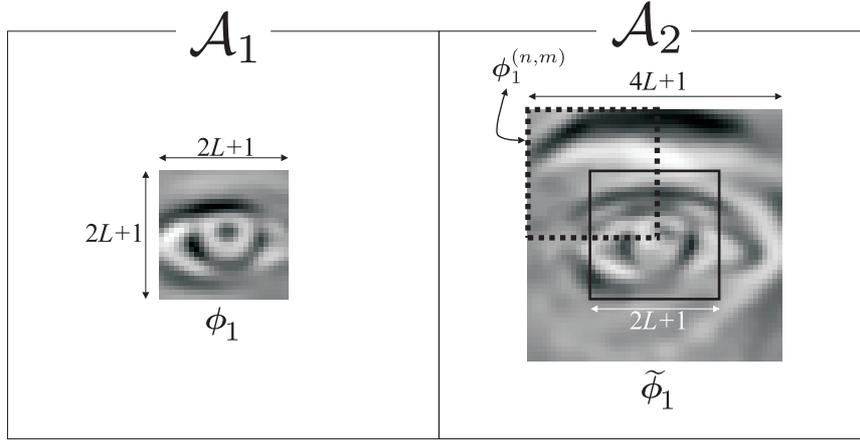


Figura 6.7: Exemplo para determinarmos os deslocamentos lineares $\phi_i^{(n,m)}$ utilizando-se a componente principal $\tilde{\phi}_i$. Conforme mencionamos, $\mathcal{A}_1 = \{\phi_i\}$ e $\mathcal{A}_2 = \{\phi_i^{(n,m)}\}$. Neste exemplo, consideramos a primeira componente principal. A componente ϕ_1 foi obtida empregando-se blocos com dimensões $2L + 1$. Supondo que desejamos formar \mathcal{A}_2 com deslocamentos lineares de até meio bloco então devemos calcular $\tilde{\phi}_1$ empregando-se blocos com dimensões $4L + 1$. Dessa forma, os deslocamentos lineares $\phi_1^{(n,m)}$ podem ser obtidos facilmente através de uma janela deslizando aplicada em ϕ_1 .

\mathcal{A}_1 , \mathcal{A}_2 e \mathcal{B} da seguinte maneira:

$$\mathbf{h}_{\phi_i} = \left\{ p_1 \phi_i (\phi_i)^{*t} + p_2 \frac{1}{(2L+1)^2 - 1} \sum_{\substack{n=-L \\ n \neq 0}}^L \sum_{\substack{m=-L \\ m \neq 0}}^L \phi_i^{(n,m)} (\phi_i^{(n,m)})^{*t} + \right. \\ \left. + p(\mathcal{B}) \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^N \phi_j (\phi_j)^{*t} \right\}^{-1} p_1 \phi_i. \quad (6.91)$$

A Equação (6.91) expressa o detector por produto interno utilizando a análise de componentes principais. Por fim, listamos abaixo algumas particularidades relativas ao método IPD-B-PCA.

- A flexibilidade dos detectores por produto interno pode ser utilizada para construirmos a classe \mathcal{A}_1 dos padrões de interesse. Por exemplo, podemos inserir componentes levemente rotacionadas ou deslocadas. Assim, é possível obtermos detectores ainda mais robustos. Naturalmente, deve-se controlar a inserção de padrões de interesse de tal forma a não descaracterizar a classe \mathcal{A}_1 .
- A ortonormalidade das componentes principais permite uma simplificação no

projeto do detector por produto interno. Note que os padrões de interesse (componentes principais) estão, devido à ortonormalidade das componentes, distribuídos na superfície de uma hiperesfera com raio unitário. Dessa forma, torna-se desnecessário o uso da dimensão extra (Seção 6.6) para o projeto dos detectores do método IPD-B-PCA.

- Vamos observar as classes \mathcal{B} e \mathcal{A}_2 . Note que rigorosamente, bastaria considerarmos a classe dos padrões que não são de interesse, ou seja, somente a classe \mathcal{B} seria necessária para o projeto dos detectores. Porém, a localização espacial dos padrões de interesse é de extrema importância. Sendo assim, é essencial que se possa rejeitar com eficiência pequenos deslocamentos lineares do padrão. Logo, este fato evidencia a importância da classe \mathcal{A}_2 .
- Observe que as classes envolvidas no projeto do detector obtido através do método IPD-B-PCA possuem valores constantes proporcionais às probabilidades p_1 , p_2 e $p(\mathcal{B})$. Os valores destas probabilidades influenciam significativamente no desempenho do detector, pois eles ponderam a importância de cada classe. Note que o treinamento dos detectores deve ser feito utilizando-se um compromisso entre detecção dos padrões de interesse e rejeição dos outros padrões.

6.8 Considerações Finais

Neste capítulo, apresentamos o método denominado por Detectores por Produto Interno (IPD) que compreende uma das contribuições da Tese. Os classificadores desenvolvidos utilizam um simples produto interno para determinar se um candidato é ou não o padrão de interesse. Além disso, demonstramos que a Filtragem Discriminativa e os Filtros de Correlação são casos particulares do IPD. Dessa forma, é possível obtermos filtros discriminativos e de correlação a partir de detectores por produto interno. Por fim, incrementamos a robustez do IPD através do projetando de detectores para as componentes principais.

No próximo capítulo, apresentamos os procedimentos experimentais e as simulações para o método IPD no contexto do problema de detecção de pontos fiduciais de faces humanas. Além disso, comparamos os resultados obtidos para o IPD com os dos métodos utilizados nos capítulos anteriores.

Capítulo 7

Detectores por Produto Interno (IPD): Procedimentos Experimentais e Simulações

7.1 Introdução

Com o objetivo de validar os métodos IPD-B e IPD-B-PCA (Capítulo 6), abordamos o problema da detecção de pontos fiduciais em um esquema semelhante ao desenvolvido no Capítulo 5. Assim, as simulações foram feitas utilizando-se dois subconjuntos das bases de dados *BioID* e *Feret*, com respectivamente 503 e 2004 imagens (veja a Seção 5.3). A métrica de desempenho para os experimentos é a mesma especificada na Seção 5.4. Para comparação, iremos utilizar todos os resultados obtidos no Capítulo 5.

7.2 Procedimentos para Pré-processamento, Treinamento e Teste

Assim como no Capítulo 5, iremos utilizar em todas as simulações, validação cruzada com 7 *folds* (observe o rodapé ⁽¹⁾). Nesta seção, verificaremos os procedimentos necessários para pré-processamento, treinamento e teste.

7.2.1 Pré-processamento das Imagens

Os procedimentos para pré-processamento de todas as imagens das bases de dados são idênticos aos desenvolvidos na Seção 5.5.1.

7.2.2 Treinamento (Fase de Aprendizagem)

Nesta etapa, efetuamos o aprendizado de todos os modelos dos métodos IPD e IPD-PCA utilizando, para cada experimento, o conjunto de treinamento que corresponde a 6/7 do total de imagens da base de dados (idêntico ao aprendizado feito na Seção 5.5.2). Os procedimentos iniciais para treinamento são idênticos aos apresentados na Seção 5.5.2.1. Na Figura 7.1 temos o diagrama em blocos para o método IPD-B adaptado de [112]. Primeiramente, realizamos os procedimentos iniciais para treinamento com uma pequena diferença, proposta em [112], em relação ao diagrama apresentado na Seção 5.5.2.1. Nesta, utiliza-se a flexibilidade do IPD para definição das classes. Assim, consideram-se três classes: \mathcal{A}_1 , como classe de positivos formada por blocos com centros em pontos fiduciais e; \mathcal{A}_2 e \mathcal{B} , como classes de negativos. Iremos separar as classes de negativos da seguinte maneira: a classe \mathcal{A}_2 será formado por blocos vizinhos aos blocos que pertencem à \mathcal{A}_1 (em [112], considera-se uma vizinhança de 1 até 10 pixels) e; a classe \mathcal{B} será formada pelos outros blocos. A seguir, acrescenta-se uma dimensão extra (veja a Seção 6.6) aos vetores de treinamento, de tal forma que eles possuam a mesma norma. Esta dimensão extra utiliza a máxima norma E_m dentre os blocos das classes \mathcal{A}_1 , \mathcal{A}_2 e \mathcal{B} . Antes da etapa de projeto dos

¹Lembrando que cada simulação corresponde a um ponto fiducial. Assim, temos 11 simulações distintas para cada base de dados. Uma simulação possui 7 experimentos e em cada um temos dois conjuntos disjuntos, denominados por treinamento e teste. No conjunto de treinamento utilizamos 6/7 do total de imagens da base de dados para treinamento e verificação de parâmetros e no conjunto de teste utilizamos 1/7 do total de imagens para validar o modelo treinado.

detectores \mathbf{h} , retira-se a média global ($\boldsymbol{\mu}$) dos vetores com dimensão extra. A configuração em cascata de classificadores IPD será formada por estágios. Cada estágio será projetado utilizando-se os positivos do estágio anterior bem como os rótulos das classes de treinamento (\mathcal{A}_1 , \mathcal{A}_2 e \mathcal{B}) e cada IPD será projetado utilizando-se as Equações (6.10) e (6.18). O produto interno $\mathbf{h}^{*t} B_{\mathbf{z}} = d_{B_{\mathbf{z}}}$ será utilizado, em cada estágio, para treinamento do classificador *Gentle AdaBoost* utilizando a *toolbox* GML AdaBoost Matlab [53]. A cascata formada por estágios AdaBoost e vetores \mathbf{h} correspondem à saída do diagrama em blocos. A média global $\boldsymbol{\mu}$ e a máxima norma E_m serão utilizadas no teste.

Na Figura 7.2, apresentamos o diagrama em blocos para o método IPD-B-PCA. Primeiramente, realizamos os procedimentos iniciais para treinamento (Seção 5.5.2.1). A seguir, os detectores \mathbf{h}_{ϕ_i} , onde $i = \{1, \dots, k\}$, serão projetados para cada uma das componentes principais com maiores variâncias utilizando a Equação (6.91). Cada bloco pertencente à classe de positivos e negativos, subtraído do bloco médio da classe de positivos (denominado por $\boldsymbol{\mu}_{\mathcal{U}}$), será processado utilizando-se a equação $\mathbf{h}_{\phi_i}^{*t} B_{\mathbf{z}}$. Assim, cada bloco $B_{\mathbf{z}}$ terá um vetor associado denominado $\mathbf{d}_{B_{\mathbf{z}}}$ com dimensões $1 \times k$. Os vetores $\mathbf{d}_{B_{\mathbf{z}}}$ serão utilizados para treinamento do classificador *Gentle AdaBoost* [52] utilizando a *toolbox* GML AdaBoost Matlab [53] de tal forma que possamos obter a superfície de decisão que separa a classe de positivos da classe de negativos. Os vetores \mathbf{h}_{ϕ_i} e o classificador AdaBoost correspondem à saída do diagrama em blocos. O bloco médio da classe de positivos ($\boldsymbol{\mu}_{\mathcal{U}}$) também será usado no teste. Note que não é utilizada a dimensão extra.

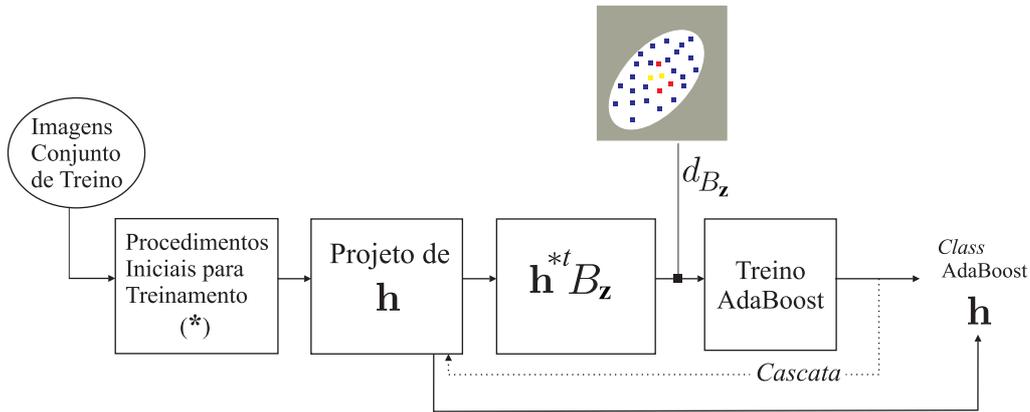


Figura 7.1: Diagrama em blocos para fase de treinamento do método IPD-B adaptado de [112]. Primeiramente, realizamos os procedimentos iniciais para treinamento com uma pequena diferença, proposta em [112], em relação ao diagrama apresentado na Seção 5.5.2.1. Nesta, utiliza-se a flexibilidade do IPD para definição das classes. Assim, consideram-se três classes: \mathcal{A}_1 , como classe de positivos formada por blocos com centros em pontos fiduciais e; \mathcal{A}_2 e \mathcal{B} , como classes de negativos. Iremos separar as classes de negativos da seguinte maneira: a classe \mathcal{A}_2 será formado por blocos vizinhos aos blocos que pertencem à \mathcal{A}_1 (em [112], considera-se uma vizinhança de 1 até 10 pixels) e; a classe \mathcal{B} será formada pelos outros blocos. A seguir, acrescenta-se uma dimensão extra (veja a Seção 6.6) aos vetores de treinamento, de tal forma que eles possuam a mesma norma. Esta dimensão extra utiliza a máxima norma E_m dentre os blocos das classes \mathcal{A}_1 , \mathcal{A}_2 e \mathcal{B} . Antes da etapa de projeto dos detectores \mathbf{h} , retira-se a média global (μ) dos vetores com dimensão extra. A configuração em cascata de classificadores IPD será formada por estágios. Cada estágio será projetado utilizando-se os positivos do estágio anterior bem como os rótulos das classes de treinamento (\mathcal{A}_1 , \mathcal{A}_2 e \mathcal{B}) e cada IPD será projetado utilizando-se as Equações (6.10) e (6.18). O produto interno $\mathbf{h}^{*t} B_z = d_{B_z}$ será utilizado, em cada estágio, para treinamento do classificador *Gentle* AdaBoost utilizando a *toolbox* GML AdaBoost Matlab [53]. A cascata formada por estágios AdaBoost e vetores \mathbf{h} correspondem à saída do diagrama em blocos. A média global μ e a máxima norma E_m serão utilizadas no teste.

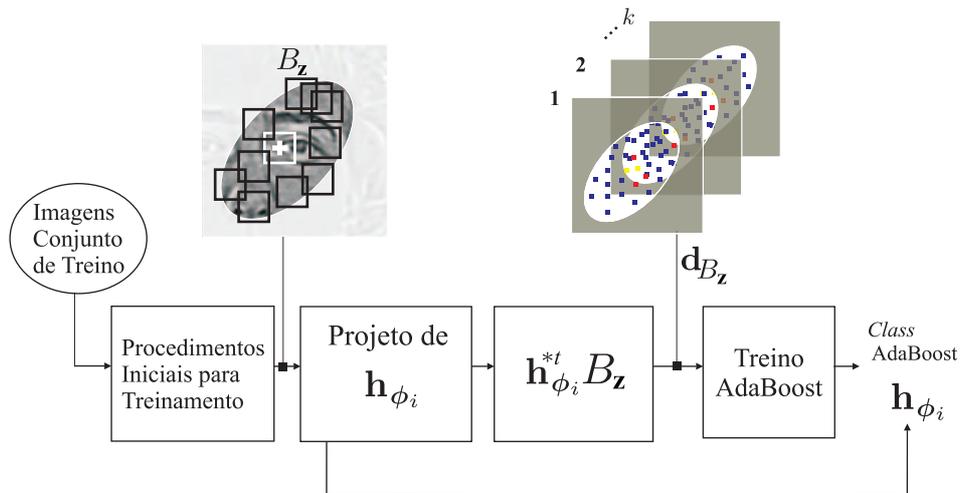


Figura 7.2: Diagrama em blocos para fase de treinamento do método IPD-B-PCA. Primeiramente, realizamos os procedimentos iniciais para treinamento (Seção 5.5.2.1). A seguir, os detectores \mathbf{h}_{ϕ_i} , onde $i = \{1, \dots, k\}$, serão projetados para cada uma das componentes principais com maiores variâncias utilizando a Equação (6.91). Cada bloco pertencente à classe de positivos e negativos, subtraído do bloco médio da classe de positivos (denominado por μ_U), será processado utilizando-se a equação $\mathbf{h}_{\phi_i}^{*t} B_z$. Assim, cada bloco B_z terá um vetor associado denominado \mathbf{d}_{B_z} com dimensões $1 \times k$. Os vetores \mathbf{d}_{B_z} serão utilizados para treinamento do classificador *Gentle AdaBoost* [52] utilizando a *toolbox* GML AdaBoost Matlab [53] de tal forma que possamos obter a superfície de decisão que separa a classe de positivos da classe de negativos. Os vetores \mathbf{h}_{ϕ_i} e o classificador AdaBoost correspondem à saída do diagrama em blocos. O bloco médio da classe de positivos (μ_U) também será usado no teste. Note que não é utilizada a dimensão extra.

7.2.3 Teste (Fase de Operação)

Nesta seção, semelhante a Seção 5.5.3, apresentamos os procedimentos para teste dos modelos treinados conforme a Seção 7.2.2. Em cada experimento, temos a mesma especificação para o conjunto de teste, ou seja, ele será formado por 1/7 do total de imagens da base de dados. Os procedimentos iniciais e pós-processamento para teste são idênticos aos da Seção 5.5.3. Na Figura 7.3 apresentamos o diagrama em blocos para o IPD-B. Primeiramente, efetuamos os procedimentos iniciais para teste. As imagens serão processadas por uma janela deslizante $B_{\mathbf{z}}$, onde \mathbf{z} é a coordenada correspondente ao centro do bloco. A seguir, acrescentamos a dimensão extra em $B_{\mathbf{z}}$ utilizando E_m e subtraímos por $\boldsymbol{\mu}$. O vetor resultante será denominado $\mathbf{b}_{\mathbf{z}}$ (lembrando que E_m e $\boldsymbol{\mu}$ são determinados no treinamento). Assim, cada bloco $B_{\mathbf{z}}$ terá um correspondente $\mathbf{b}_{\mathbf{z}}$. A seguir, cada $\mathbf{b}_{\mathbf{z}}$ será classificado pela cascata de classificadores IPD determinados no treinamento. Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).

Na Figura 7.4 apresentamos o diagrama em blocos para teste do método IPD-B. Primeiramente, efetuamos os procedimentos iniciais para teste. As imagens serão processadas por uma janela deslizante $B_{\mathbf{z}}$ (\mathbf{z} é a coordenada correspondente ao centro do bloco). A seguir, utilizamos os detectores \mathbf{h}_{ϕ_i} (obtidos no treinamento) para efetuar o produto interno com $B_{\mathbf{z}}$ subtraído por $\boldsymbol{\mu}_{\mathcal{U}}$ (determinado no treinamento). Dessa forma, cada bloco $B_{\mathbf{z}}$ terá um vetor associado denominado $\mathbf{d}_{B_{\mathbf{z}}}$. O classificador AdaBoost, construído no treinamento, usará $\mathbf{d}_{B_{\mathbf{z}}}$ para classificar $B_{\mathbf{z}}$. Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).

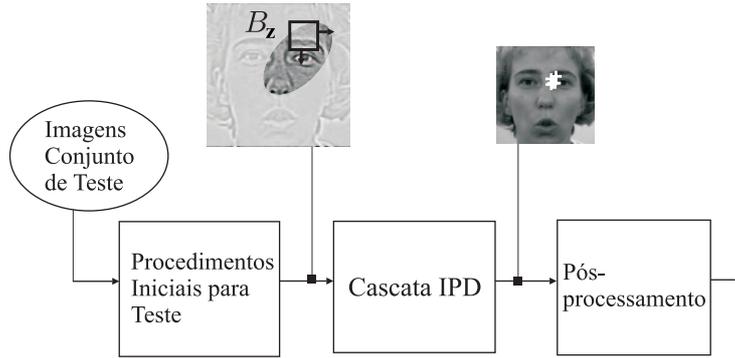


Figura 7.3: Diagrama em blocos para teste do método IPD-B. Primeiramente, efetuamos os procedimentos iniciais para teste. As imagens serão processadas por uma janela deslizante B_z , onde z é a coordenada correspondente ao centro do bloco. A seguir, acrescentamos a dimensão extra em B_z utilizando E_m e subtraímos por μ . O vetor resultante será denominado \mathbf{b}_z (lembrando que E_m e μ são determinados no treinamento). Assim, cada bloco B_z terá um correspondente \mathbf{b}_z . A seguir, cada \mathbf{b}_z será classificado pela cascata de classificadores IPD determinados no treinamento. Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).

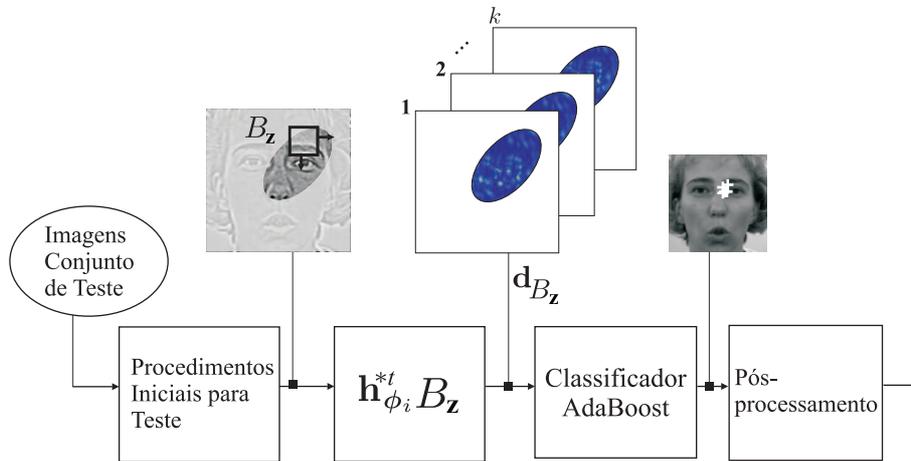


Figura 7.4: Diagrama em blocos para teste do método IPD-B-PCA. Primeiramente, efetuamos os procedimentos iniciais para teste. As imagens serão processadas por uma janela deslizante B_z (z é a coordenada correspondente ao centro do bloco). A seguir, utilizamos os detectores \mathbf{h}_{ϕ_i} (obtidos no treinamento) para efetuar o produto interno com B_z subtraído por $\mu_{\mathcal{U}}$ (determinado no treinamento). Dessa forma, cada bloco B_z terá um vetor associado denominado \mathbf{d}_{B_z} . O classificador AdaBoost, construído no treinamento, usará \mathbf{d}_{B_z} para classificar B_z . Por último, realizamos a etapa de pós-processamento conforme a estratégia escolhida (veja a Tabela 5.2).

7.3 Influência da Quantidade de Componentes Principais

Nesta seção, apresentamos os resultados obtidos para o método IPD-B-PCA utilizando-se as bases de dados *BioID* e *Feret* com, respectivamente, 503 e 2004 imagens. Os procedimentos de treinamento, teste e pré-processamento foram desenvolvidos conforme descrição realizada na Seção 7.2. O pós-processamento² NA foi utilizado (veja a Tabela 5.2). O valor de α é igual a 0,1. Os resultados, assim como para os métodos anteriores, foram obtidos empregando-se validação cruzada com 7 *folds*. Os conjuntos de treinamento e teste são idênticos aos especificados no Capítulo 5. A organização dos resultados é feita de forma idêntica à Seção 5.6. Dessa forma, os resultados serão apresentados através de dois gráficos que indicam, através de barras, as percentagens médias TP e FP. Nos gráficos, a quantidade de componentes principais utilizada está localizada na horizontal. A pequena barra na cor escura, acima de cada barra de TP e FP indica duas vezes o desvio padrão. O terceiro grupo de gráficos é formado por coordenadas que representam as taxas de TP e FP para cada componente. A localização da coordenada está associada ao desempenho obtido. Quanto mais acima e à esquerda a coordenada estiver melhor será o desempenho obtido.

²Lembramos que o acrônimo NA indica que não será utilizada nenhuma estratégia para agrupamento.

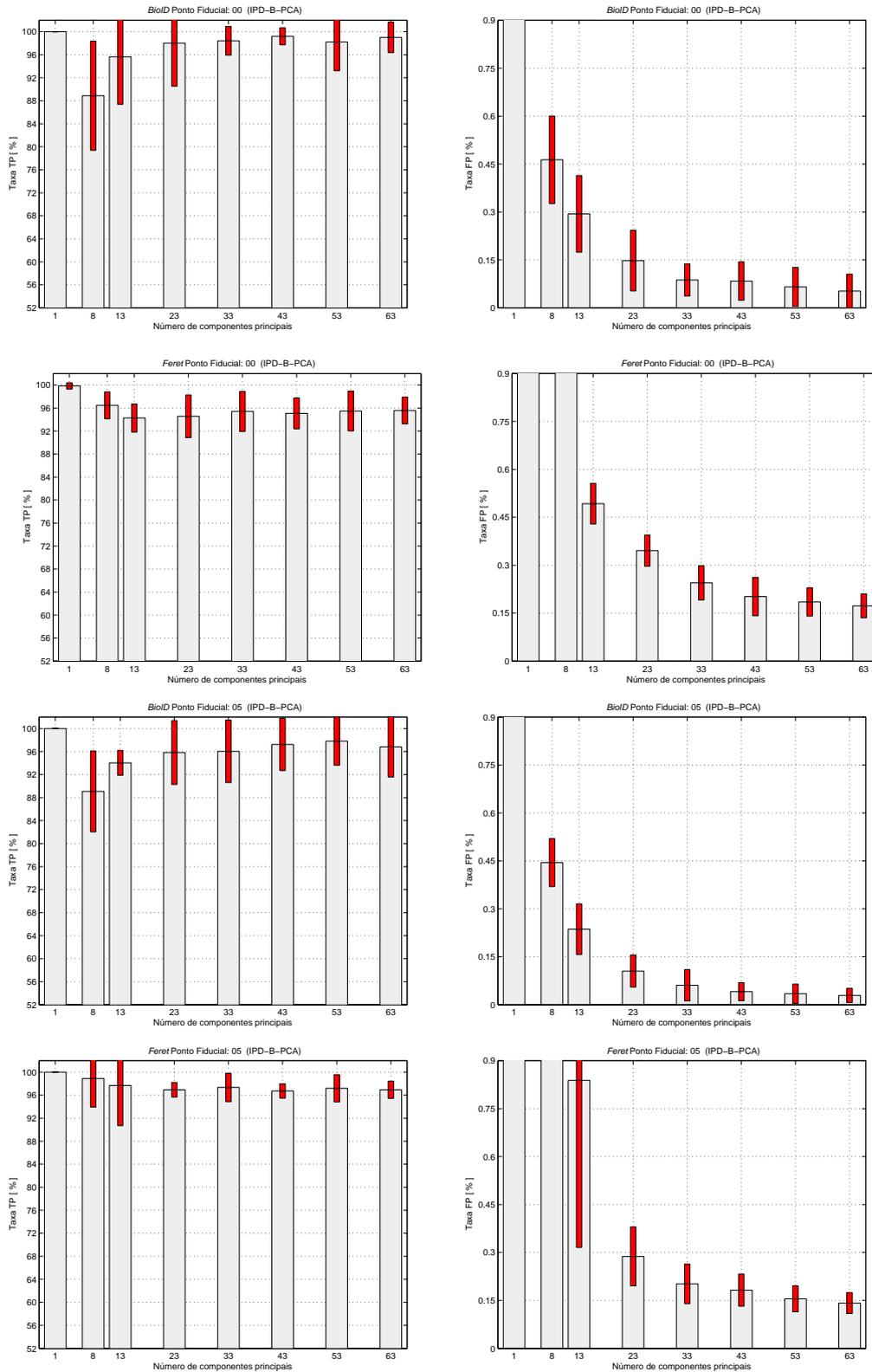


Figura 7.5: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 00 e 05. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

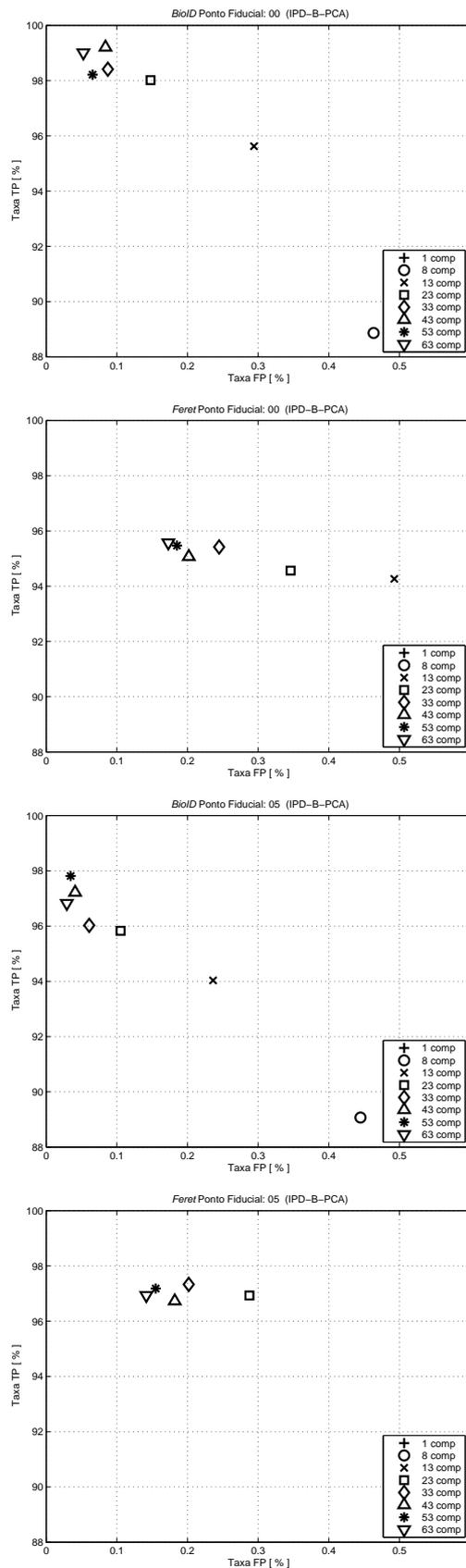


Figura 7.6: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 00 e 05. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

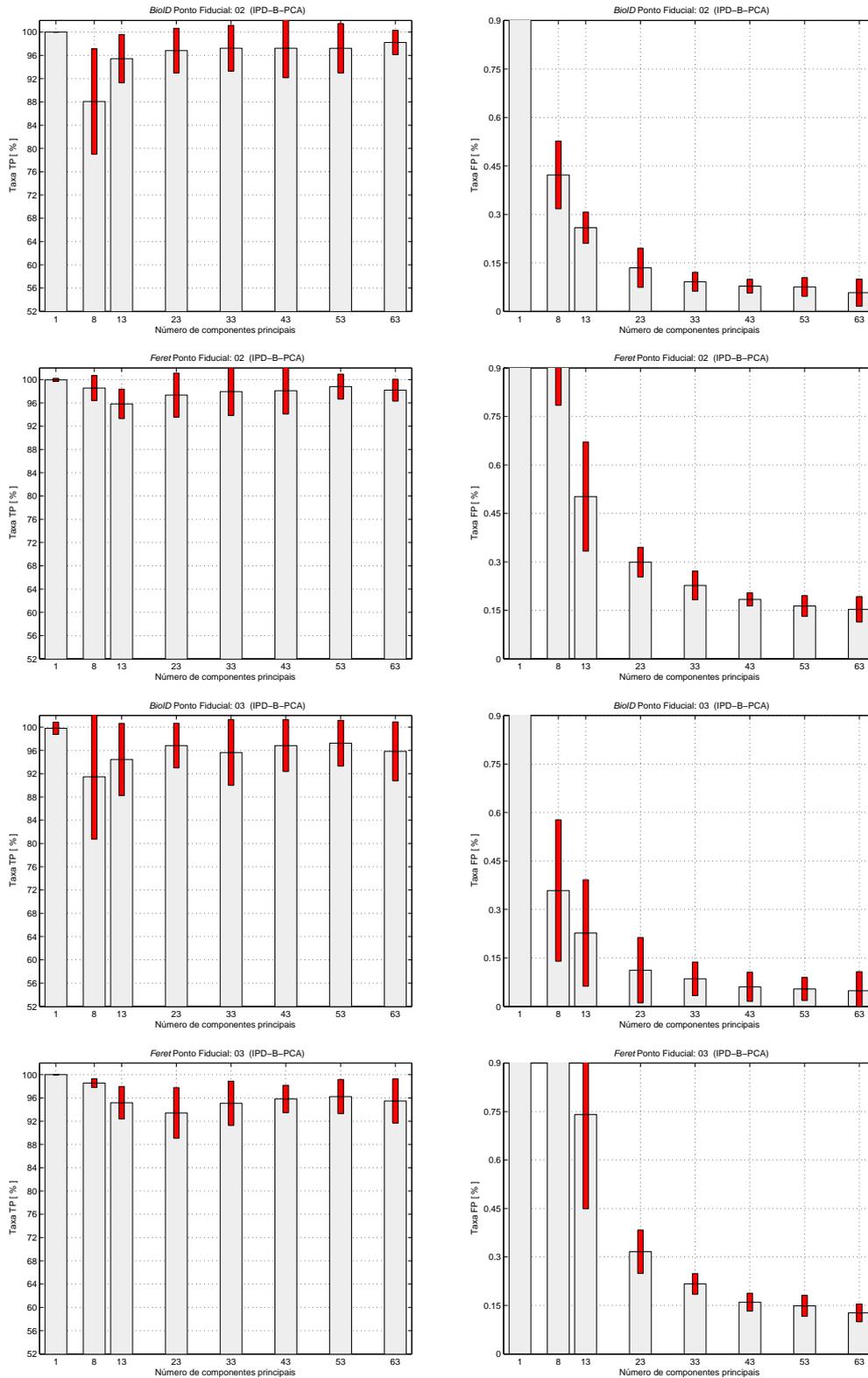


Figura 7.7: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 02 e 03. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

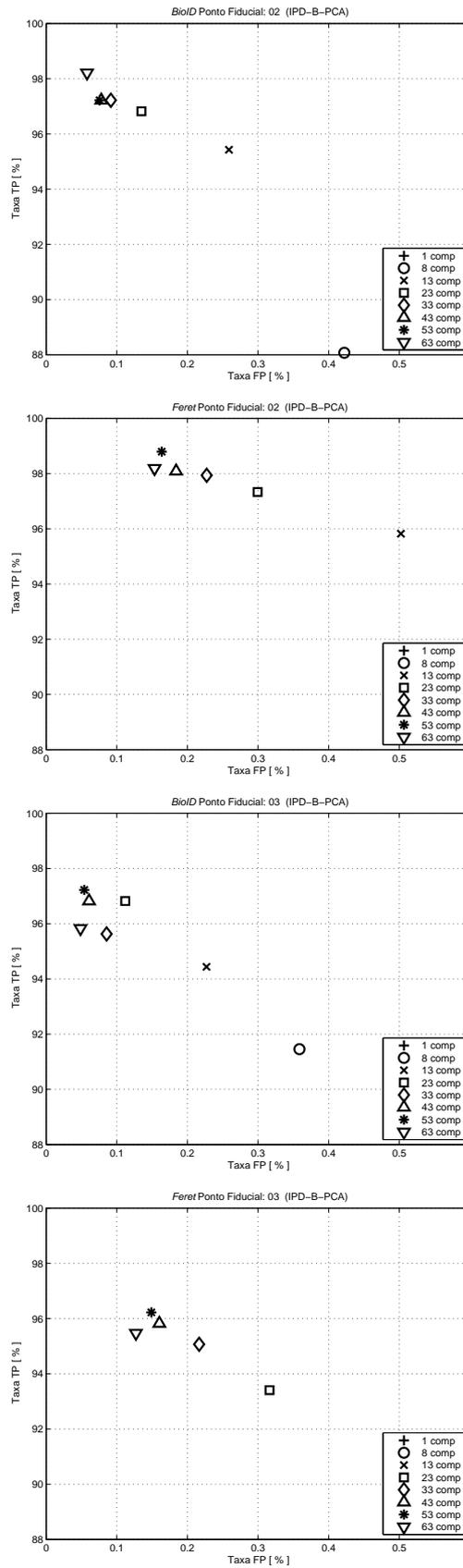


Figura 7.8: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 02 e 03. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

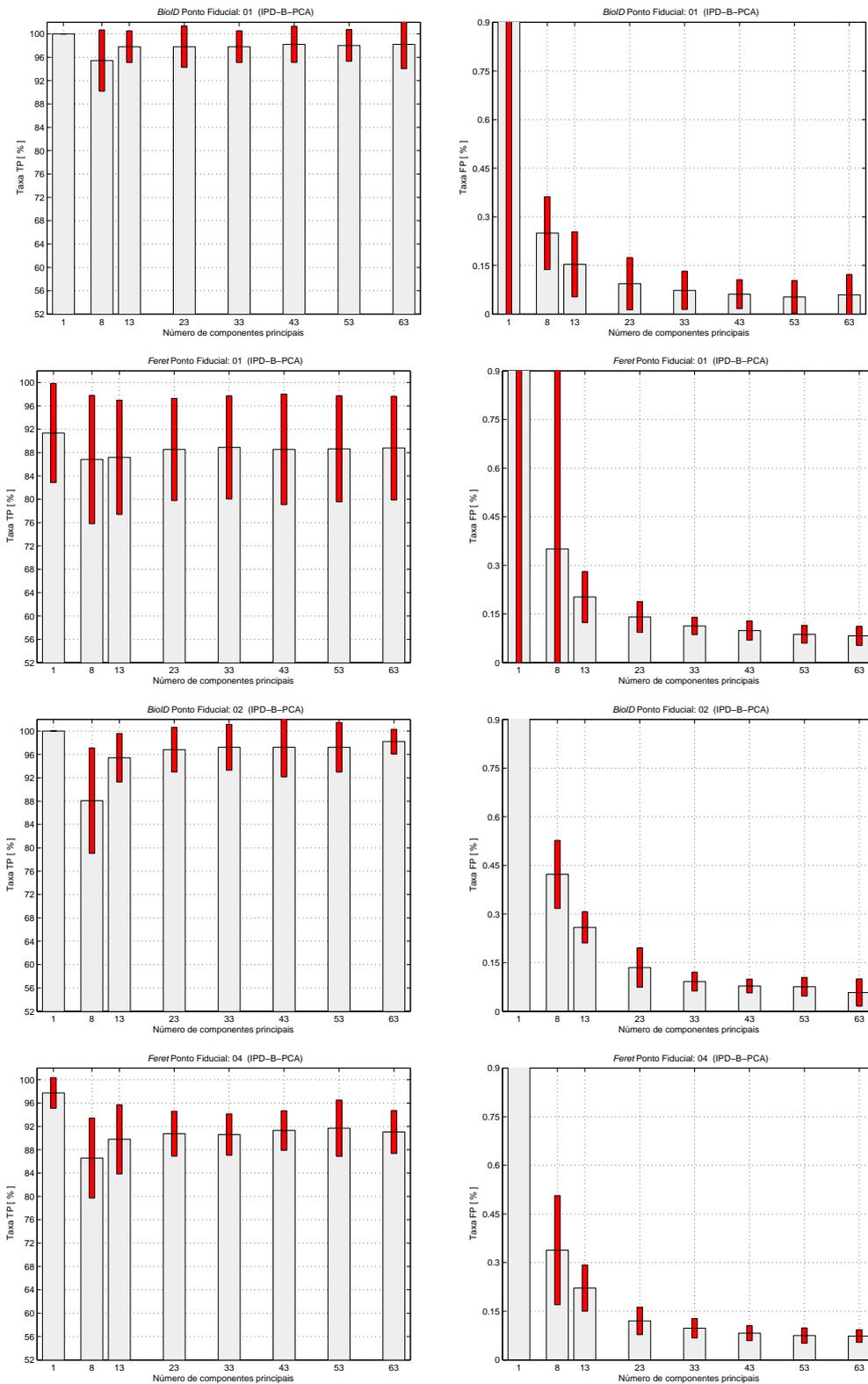


Figura 7.9: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 01 e 04. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

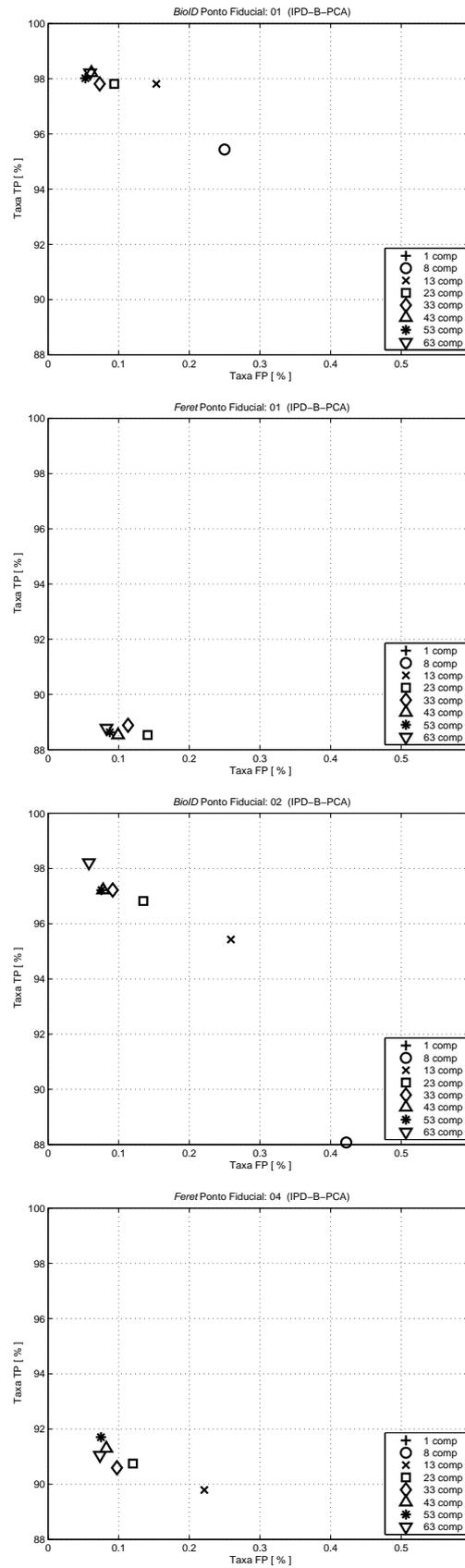


Figura 7.10: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 01 e 04. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

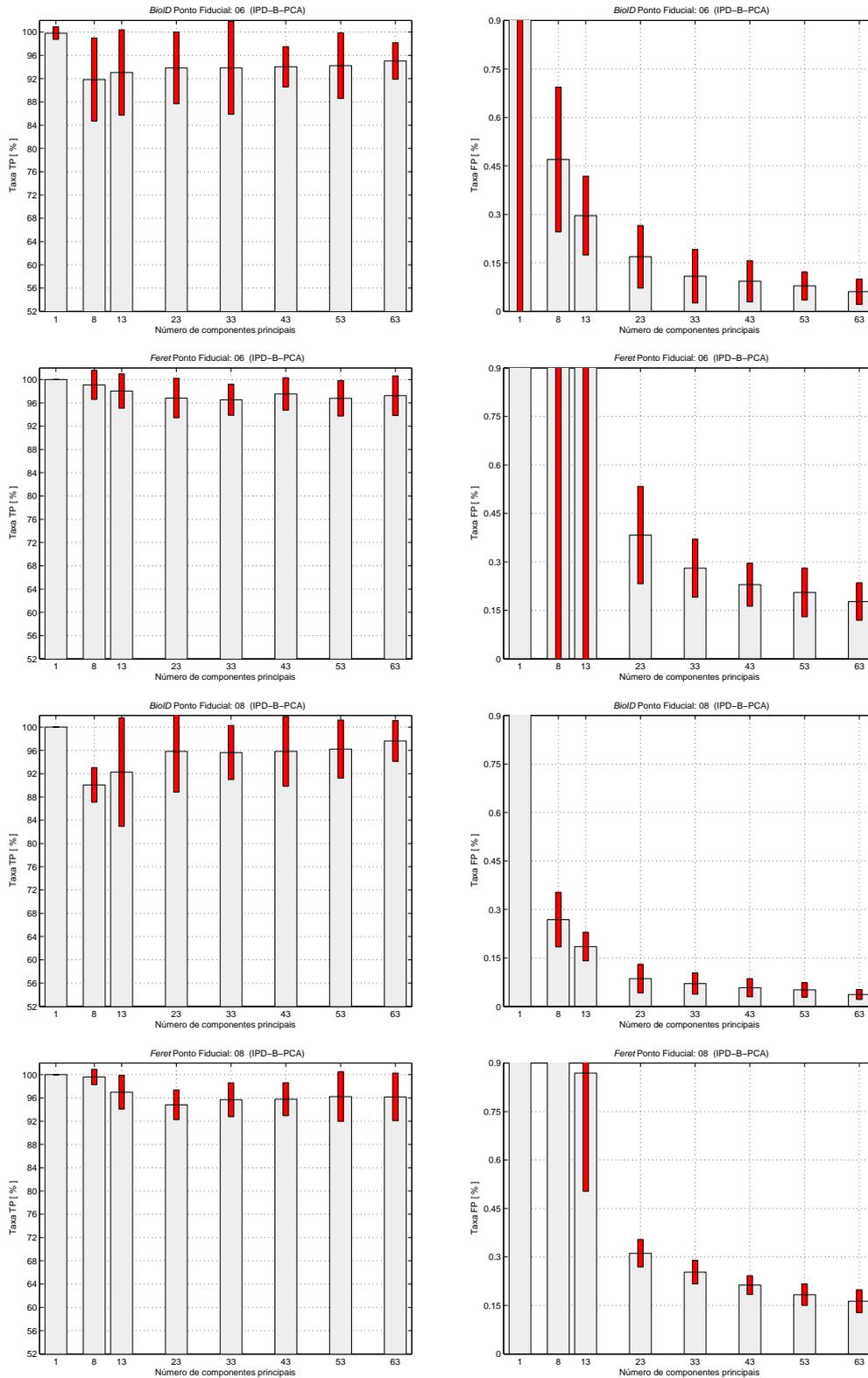


Figura 7.11: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 06 e 08. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

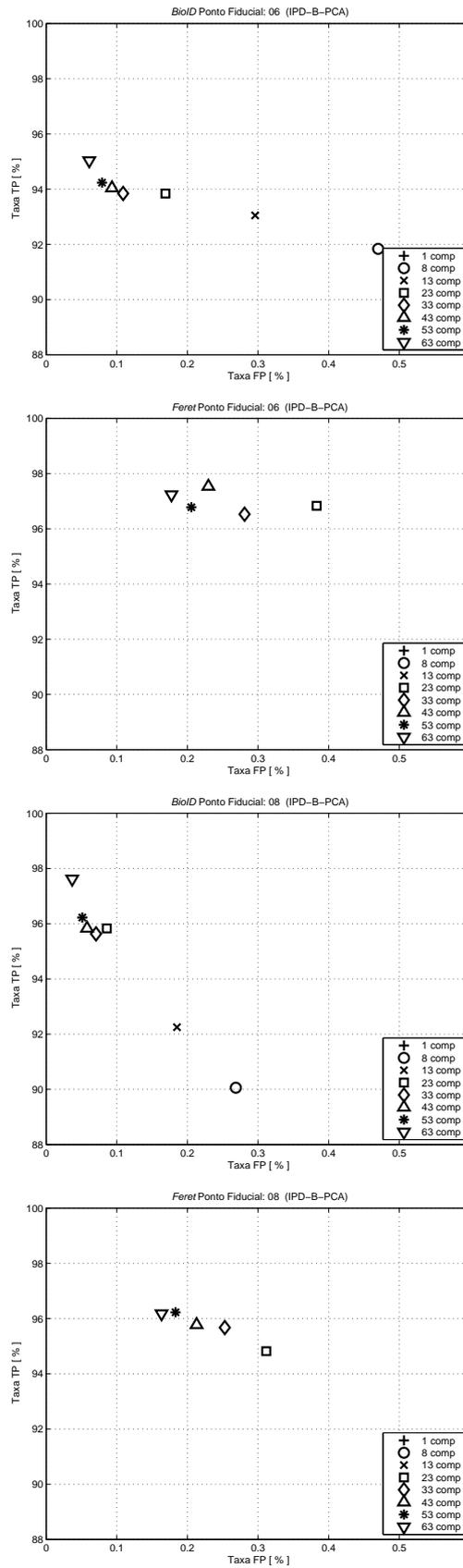


Figura 7.12: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 06 e 08. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

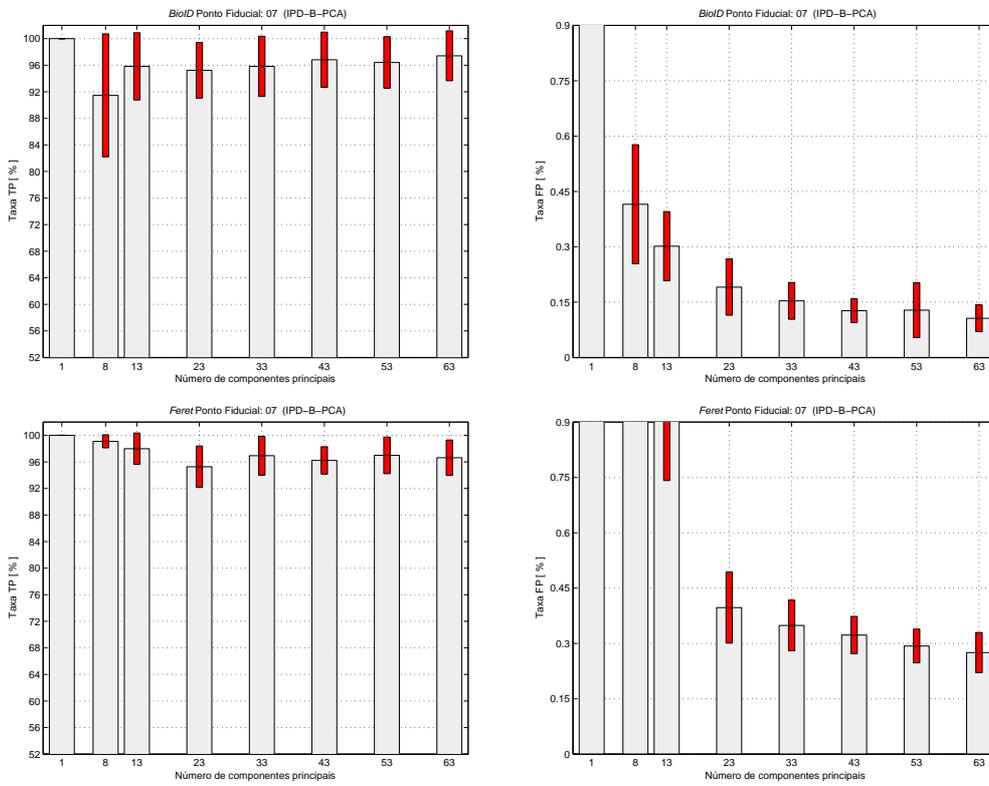


Figura 7.13: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para o ponto fiducial 07. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

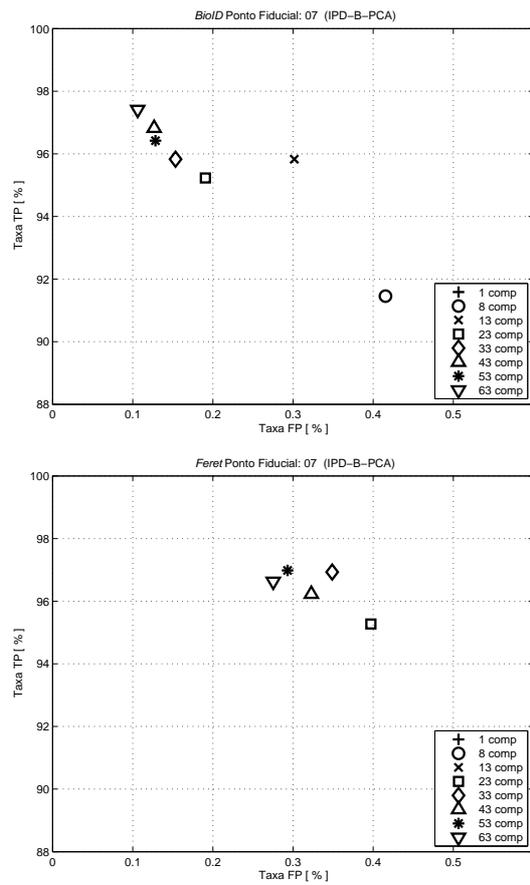


Figura 7.14: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para o ponto fiducial 07. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

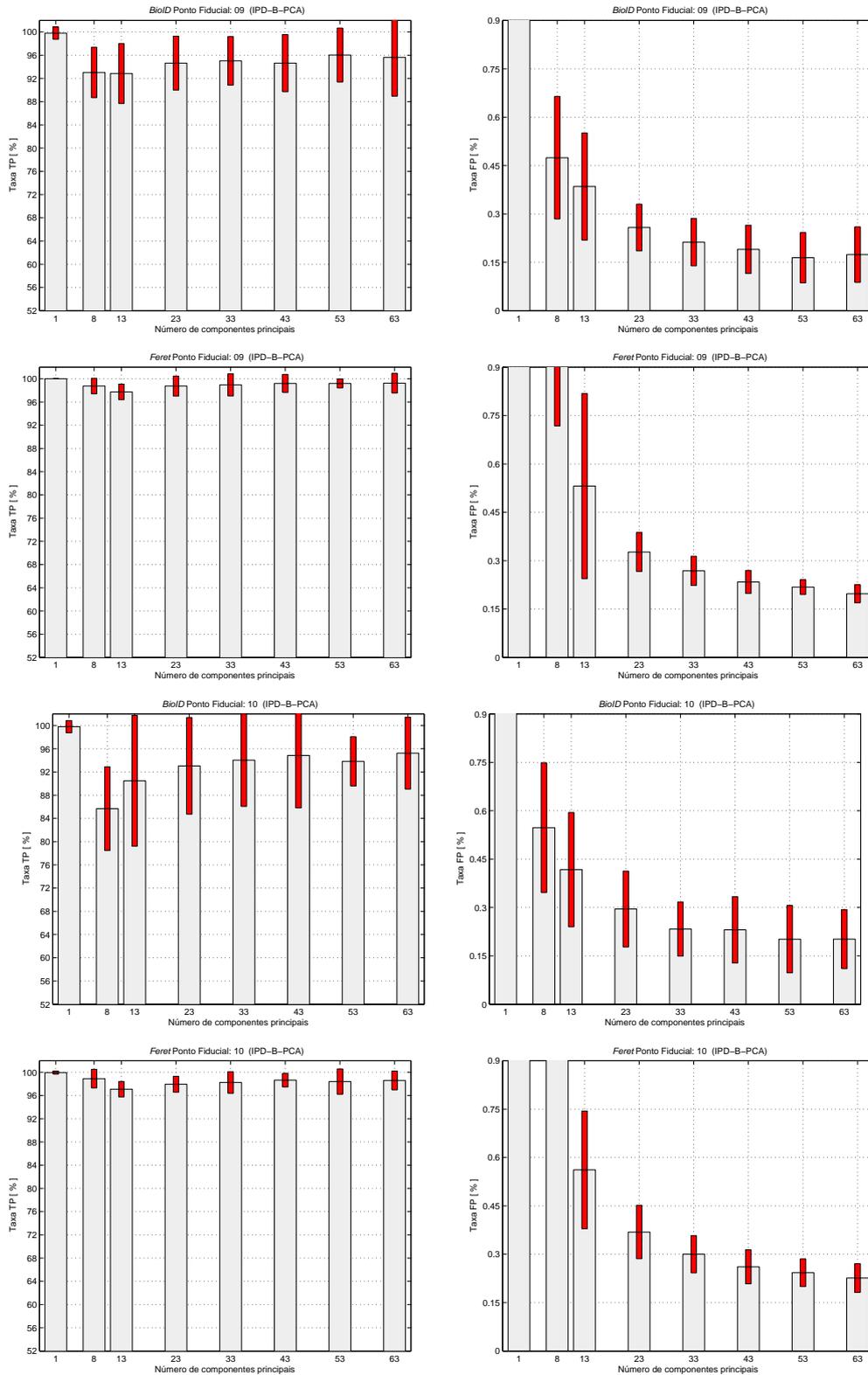
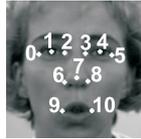


Figura 7.15: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 09 e 10. A barra na cor escura, situada no topo de cada barra de TP e FP indica duas vezes o desvio padrão.

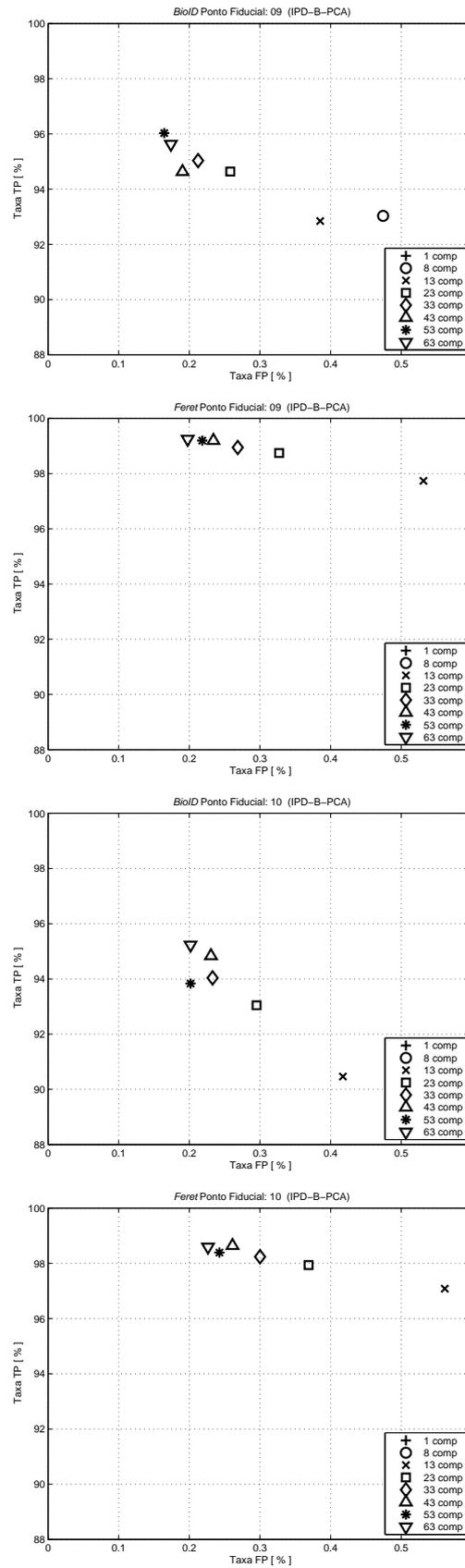


Figura 7.16: Base de dados *BioID* e *Feret*: resultados ao variarmos o número de componentes principais para os pontos fiduciais 09 e 10. Cada coordenada indica as taxas de TP e FP obtidas por cada componente.

7.3.1 Análise dos Resultados

Nas Figuras 7.5 até 7.15 apresentamos, respectivamente, os resultados obtidos utilizando bases de dados *BioID* e *Feret*. Após análise das figuras podemos verificar que:

1. As taxas de TP foram levemente superiores às obtidas pelo método DF-B-PCA. Os valores de desvio padrão obtidos para o método IPD-B-PCA foram levemente menores que os do método DF-B-PCA.
2. As taxas de FP foram compatíveis às do método DF-B-PCA. Note que seus valores estão por volta de 0,45% para um número de componentes (k) superior a 8. Para o método DF-B-PCA as taxas alcançavam este valor com k superior a 23.
3. A utilização das componentes principais proporciona um ganho significativo no desempenho do método IPD-B-PCA.
4. Exceto para a base *Feret* nos pontos fiduciais 01 e 04 temos que as taxas de TP se estabilizam em aproximadamente 95%.

Os gráficos das figuras evidenciam que a quantidade de componentes principais usadas tem alto impacto no desempenho do método IPD-B-PCA. Os resultados obtidos são levemente superiores aos do método DF-B-PCA. Ao final desta seção, apresentamos uma síntese dos melhores resultados obtidos, em forma de tabela, em termos do número de componentes principais para todos os pontos fiduciais.

Síntese dos Resultados para IPD-B-PCA

Tabela 7.1: Base de dados *BioID* e *Feret*: síntese dos melhores desempenhos, em termos da quantidade de componentes principais, para o método IPD-PCA. As taxas médias de Verdadeiros Positivos (TP) e Falsos Positivos (FP) e os respectivos desvios padrões (σ) foram obtidos por validação cruzada com 7 *fold*s.

					
<i>BioID</i>					
PFs	qPC	IPD-B-PCA			
		TP	σ	FP	σ
00	63	99,0	1,3	0,05	0,03
01	53	98,0	1,3	0,05	0,03
02	63	98,2	1,0	0,05	0,02
03	53	97,2	1,9	0,05	0,02
04	63	98,8	0,9	0,03	0,02
05	53	97,8	2,1	0,03	0,02
06	63	95,0	1,5	0,06	0,02
07	63	97,4	1,8	0,10	0,02
08	63	97,6	1,7	0,03	0,01
09	53	96,0	2,3	0,16	0,04
10	63	95,2	3,0	0,20	0,05
<i>Feret</i>					
PFs	qPC	IPD-B-PCA			
		TP	σ	FP	σ
00	53	95,6	1,2	0,18	0,02
01	53	88,6	4,5	0,08	0,01
02	63	98,1	0,9	0,15	0,02
03	63	95,4	1,8	0,12	0,01
04	63	91,0	1,8	0,07	0,01
05	53	97,1	1,1	0,15	0,02
06	53	96,7	1,5	0,20	0,04
07	63	96,6	1,3	0,27	0,03
08	53	96,2	2,1	0,18	0,02
09	63	99,2	0,8	0,19	0,01
10	63	98,5	0,8	0,22	0,02

7.4 Comparação entre IPD-B-PCA and IPD-B

Nesta seção, realizamos uma comparação entre os resultados obtidos para os métodos IPD-B-PCA e IPD-B. Os resultados para o método IPD-B, utilizando a base de dados *BioID* foram extraídos de [112]. Para a base de dados *Feret*, os resultados foram obtidos em conjunto com os autores do trabalho [112]. Analisando os resultados apresentados na Tabela 7.2 podemos verificar que:

1. O método IPD-B apresenta um bom compromisso entre taxas de TP e FP. Para as bases de dados *BioID* e *Feret* em todos os pontos fiduciais, temos que as taxas de FP estão bem abaixo de 1,5% e as taxas de TP estão superiores a 86% (exceto para o ponto fiducial 07 na *Feret*).
2. O método IPD-B-PCA apresenta um bom compromisso entre taxas de TP e FP. Para todos os pontos fiduciais temos que as taxas de FP estão bem abaixo de 1,5% e as taxas de TP estão superiores a 95% (exceto para os pontos fiduciais 01 e 04 localizados no centro dos olhos).

Os resultados apresentados na tabela demonstram que os métodos IPD-B-PCA e IPD-B possuem desempenho satisfatório. Observe que de forma geral, o compromisso entre taxas de TP e FP obtido pelo método IPD-B-PCA é superior ao obtido pelo método IPD-B sendo que em quatro pontos fiduciais (07, 08, 09 e 10) temos uma diferença expressiva.

Tabela 7.2: Base de dados *BioID*: comparação entre os métodos IPD-B-PCA e IPD para todos os pontos fiduciais. A denominação NA indica a estratégia de agrupamento utilizada.

								
<i>BioID</i>								
PFs	IPD-B-PCA-NA (proposto)				IPD-B-NA [112]			
	TP	σ	FP	σ	TP	σ	FP	σ
00	99,0	1,3	0,05	0,03	95,2	2,2	0,07	0,02
01	98,0	1,3	0,05	0,03	97,4	2,9	0,03	0,01
02	98,2	1,0	0,05	0,02	95,6	2,0	0,23	0,03
03	97,2	1,9	0,05	0,02	97,8	2,7	0,40	0,08
04	98,8	0,9	0,03	0,02	98,4	1,8	0,04	0,02
05	97,8	2,1	0,03	0,02	93,6	3,1	0,05	0,02
06	95,0	1,5	0,06	0,02	93,2	3,1	0,08	0,03
07	97,4	1,8	0,10	0,02	88,6	3,2	0,10	0,02
08	97,6	1,7	0,03	0,01	90,6	3,5	0,14	0,05
09	96,0	2,3	0,16	0,04	87,2	3,1	0,13	0,04
10	95,2	3,0	0,20	0,05	86,0	5,8	0,20	0,06
<i>Feret</i>								
PFs	IPD-B-PCA-NA (proposto)				IPD-B-NA Em conjunto com [112]			
	TP	σ	FP	σ	TP	σ	FP	σ
00	95,6	1,2	0,18	0,02	90,5	2,4	0,07	0,01
01	88,6	4,5	0,08	0,01	93,0	2,6	0,06	0,01
02	98,1	0,9	0,15	0,02	87,7	2,7	0,13	0,06
03	95,4	1,8	0,12	0,01	80,6	3,6	0,11	0,02
04	91,0	1,8	0,07	0,01	89,5	2,1	0,11	0,02
05	97,1	1,1	0,15	0,02	89,5	2,4	0,11	0,01
06	96,7	1,5	0,20	0,04	87,6	0,7	0,10	0,01
07	96,6	1,3	0,27	0,03	90,3	2,5	0,15	0,01
08	96,2	2,1	0,18	0,02	88,2	2,6	0,11	0,02
09	99,2	0,8	0,19	0,01	84,6	2,8	0,05	0,00
10	98,5	0,8	0,22	0,02	86,5	2,6	0,14	0,01

7.5 Comparação entre IPD-B-PCA, DF-B-PCA e SVM

Nesta seção, confrontamos os resultados obtidos para os métodos IPD-B-PCA, DF-B-PCA, SVM-L e SVM-Pol. Os resultados obtidos para o método IPD-B foram suprimidos devido a seu desempenho ser inferior ao dos métodos IPD-B-PCA e SVM-Pol. Ao confrontarmos os resultados podemos verificar que:

1. O método IPD-B-PCA proporciona resultados superiores aos do DF-B-PCA para quase todos os pontos fiduciais. Na base de dados *BioID*, o IPD-B-PCA obtém resultados superiores aos do DF-B-PCA em todos os pontos fiduciais e para a base *Feret* o IPD-B-PCA proporciona resultados superiores aos do DF-B-PCA em sete pontos fiduciais (00, 02, 03, 05, 06, 07, 09 e 10).
2. O IPD-B-PCA apresenta resultados competitivos quando comparados aos do método SVM-Pol. Observe que, para a base de dados *BioID*, em três pontos fiduciais (00, 02 e 04) o desempenho do IPD-B-PCA foi superior ao do SVM-Pol. Para a base *Feret* o IPD-B-PCA apresenta resultados superiores aos do método SVM-Pol oito pontos fiduciais (00, 02, 05, 06, 07, 08, 09 e 10)

Por fim, podemos concluir que o IPD-B-PCA permite um ganho de desempenho superior ao obtido pelo DF-B-PCA. A utilização da base de dados *Feret* evidencia o bom desempenho obtido pelo método IPD-B-PCA. Na maioria dos pontos fiduciais o IPD-B-PCA apresenta um desempenho superior frente aos resultados dos métodos DF-B-PCA e SVM-Pol. Note que os valores de desvio padrão das taxas de FP para o método IPD-B-PCA são menores que os obtidos para os métodos DF-B-PCA e SVM-Pol (exceto para o ponto fiducial 07). Isto indica que o IPD-B-PCA proporciona uma variabilidade de falsos positivos controlada frente à diversidade da base de dados *Feret*.

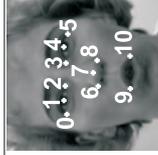
Tabela 7.3: Base de dados *BioID*: comparação entre os métodos IPD-B-PCA, DF-B-PCA, SVM-L e SVM-Pol.



PFs	IPD-B-PCA-NA (proposto)			DF-B-PCA-NA (proposto)			SVM-L-NA			SVM-Pol-NA			
	TP	σ	FP	TP	σ	FP	TP	σ	FP	TP	σ	FP	σ
00	99,0	1,3	0,05	96,6	2,2	0,05	92,4	9,8	1,09	99,6	0,7	0,39	0,07
01	98,0	1,3	0,05	95,4	2,8	0,10	99,6	1,0	0,32	98,8	1,2	0,04	0,02
02	98,2	1,0	0,05	95,0	2,1	0,24	93,0	14,3	1,58	94,2	2,7	0,15	0,03
03	97,2	1,9	0,05	95,0	2,5	0,08	96,0	1,7	0,46	99,8	0,5	0,33	0,09
04	98,8	0,9	0,03	97,8	2,8	0,11	97,2	4,2	2,49	98,8	1,5	0,08	0,05
05	97,8	2,1	0,03	97,4	2,5	0,05	98,8	3,1	2,27	99,6	0,7	0,42	0,09
06	95,0	1,5	0,06	93,2	1,6	0,14	99,6	1,0	1,40	99,6	0,7	0,31	0,05
07	97,4	1,8	0,10	95,8	3,7	0,18	47,1	9,6	0,06	99,6	0,7	0,60	0,05
08	97,6	1,7	0,03	94,8	2,6	0,17	95,4	3,9	0,40	99,0	1,1	0,24	0,03
09	96,0	2,3	0,16	92,8	3,0	0,18	96,8	5,5	2,91	98,8	1,7	0,39	0,06
10	95,2	3,0	0,20	87,4	2,9	0,18	98,6	1,9	1,47	99,6	0,7	0,35	0,04

Tabela 7.4: Base de dados *Feret*: comparação entre os métodos IPD-B-PCA, DF-B-PCA, SVM-L e SVM-Pol.

PFs	IPD-B-PCA-NA (proposto)			DF-B-PCA-NA (proposto)			SVM-L-NA			SVM-Pol-NA		
	TP	σ		TP	σ		TP	σ		TP	σ	
00	95,6	1,2	0,18	95,4	1,5	0,22	91,1	12,4	3,00	91,9	6,7	0,26
01	88,6	4,5	0,08	93,1	2,1	0,27	80,0	15,7	1,85	99,1	0,8	0,17
02	98,1	0,9	0,15	96,2	1,0	0,32	89,8	25,3	19,51	97,1	7,0	0,38
03	95,4	1,8	0,12	92,8	2,4	0,29	93,6	9,7	4,82	98,8	1,2	0,22
04	91,0	1,8	0,07	94,7	1,7	0,16	63,7	39,6	0,43	99,4	0,5	0,18
05	97,1	1,1	0,15	95,9	0,8	0,25	73,3	27,6	0,84	94,8	6,1	0,30
06	96,7	1,5	0,20	96,3	1,1	0,28	73,9	33,7	2,06	91,8	16,8	0,35
07	96,6	1,3	0,27	89,9	2,1	0,30	80,7	13,0	1,16	68,4	33,4	13,64
08	96,2	2,1	0,18	97,2	1,1	0,36	86,2	16,3	2,13	93,8	7,6	0,94
09	99,2	0,8	0,19	94,5	0,9	0,16	84,7	17,4	1,57	90,7	6,4	0,12
10	98,5	0,8	0,22	93,1	1,3	0,14	54,4	38,0	0,76	93,1	9,3	0,19



7.6 Estratégias para Pós-processamento para o Método IPD-B-PCA

Nesta seção, semelhante ao desenvolvimento realizado na Seção 5.8 aplicamos estratégias para pós-processamento do método IPD-B-PCA. Nas Figuras 7.17 até 7.22, apresentamos os resultados obtidos para o método IPD-B-PCA utilizando estratégias de pós-processamento ML, GML e A, para as bases de dados *BioID* e *Feret*. Por comodidade, lembramos que as estratégias ML, GML e A selecionam um rótulo automático por imagem (veja o rodapé ⁽³⁾ para detalhes).

Os gráficos contidos nas figuras têm descrição equivalente aos da Seção 5.8. Assim, a organização dos resultados será feita com um gráfico por ponto fiducial. Cada gráfico será composto por três curvas (distribuição de acerto média acumulada *versus* distância inter-ocular) relacionadas à estratégia utilizada. A linha vertical pontilhada, exposta em cada gráfico, auxilia a visualização da distância inter-ocular igual a 10% ($\alpha = 0,1$). Após análise dos gráficos podemos verificar que:

1. O uso das estratégias ML, GML e A apresentam comportamento semelhante ao obtido pelo método DF-B-PCA, ou seja, as técnicas ML, GML obtêm os melhores resultados ao passo que a técnica A não apresenta resultados satisfatórios.
2. Em termos de taxa de acerto, na base de dados *BioID*, para os pontos fiduciais na região dos olhos as taxas de acerto são superiores a 95%. Para os pontos fiduciais na região do nariz as taxas são superiores a 90% e para os pontos na região da boca as taxas ficam por volta de 84%.

³Nas estratégias ML, GML e A temos, como saída, somente um rótulo automático. Na ML, selecionamos o rótulo automático mais provável, supondo o modelo gaussiano a priori. Na GML, agrupamos os rótulos automáticos com distância menor que 8 pixels e a seguir escolhemos o rótulo automático mais provável (supondo o modelo gaussiano a priori). Na estratégia A, o rótulo automático será igual à média de todos os rótulos automáticos.

3. Na base *Feret*, para os pontos fiduciais localizados nos centros dos olhos as taxas ficam acima de 85%, nos cantos externos as taxas estão em torno de 90% e nos cantos internos as taxas estão acima de 95%. Para os pontos nos cantos do nariz as taxas ficam acima de 92%. Para os pontos localizados no centro do nariz e nos cantos da boca as taxas de acerto ficam acima de 80%.

Por fim, podemos concluir que para o método IPD-B-PCA tanto a estratégia ML quanto a GML podem ser utilizadas para pós-processamento com excelentes resultados.

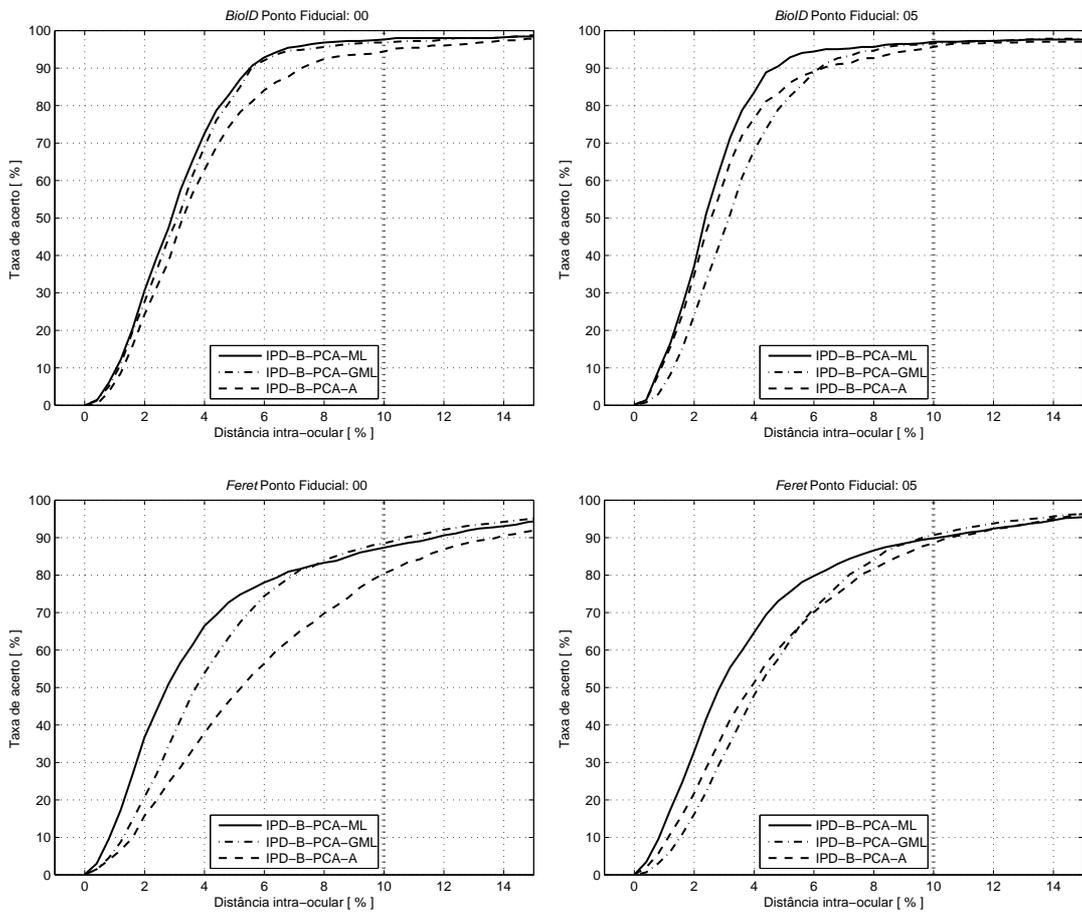


Figura 7.17: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 00 e 05.

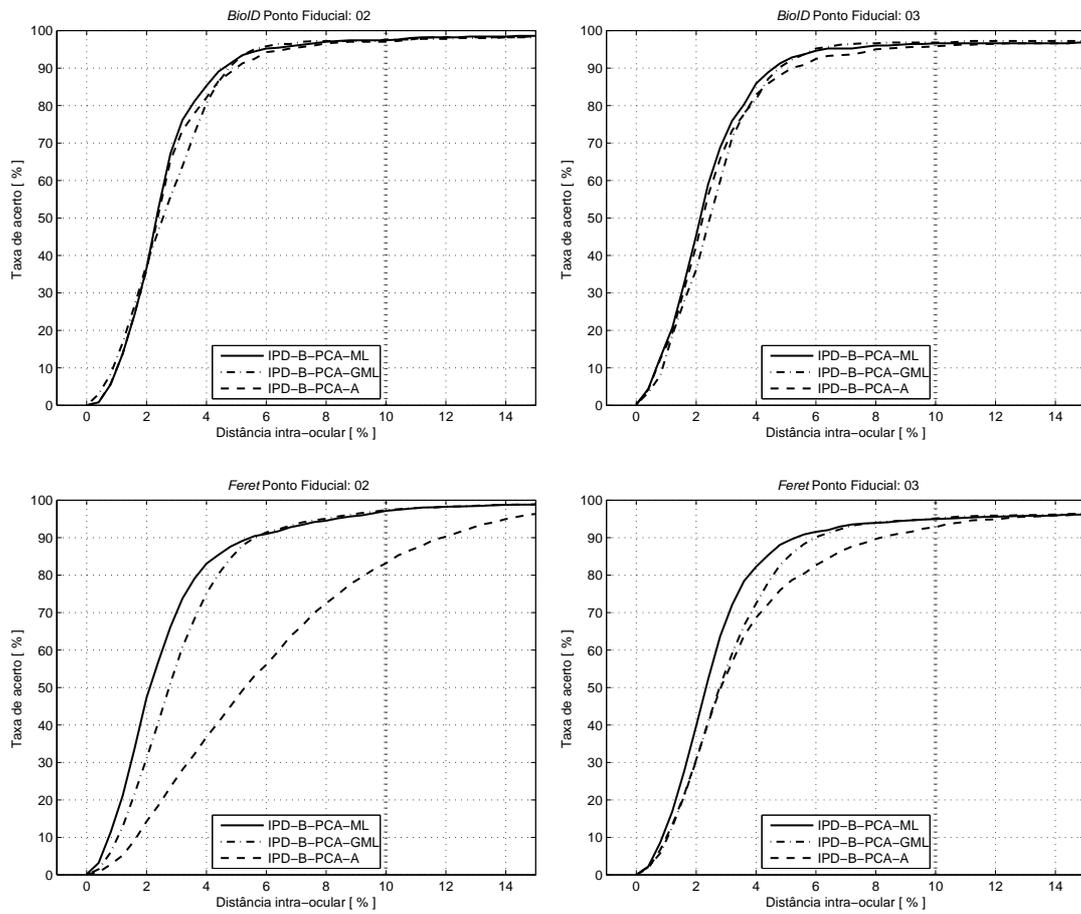


Figura 7.18: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 02 e 03.

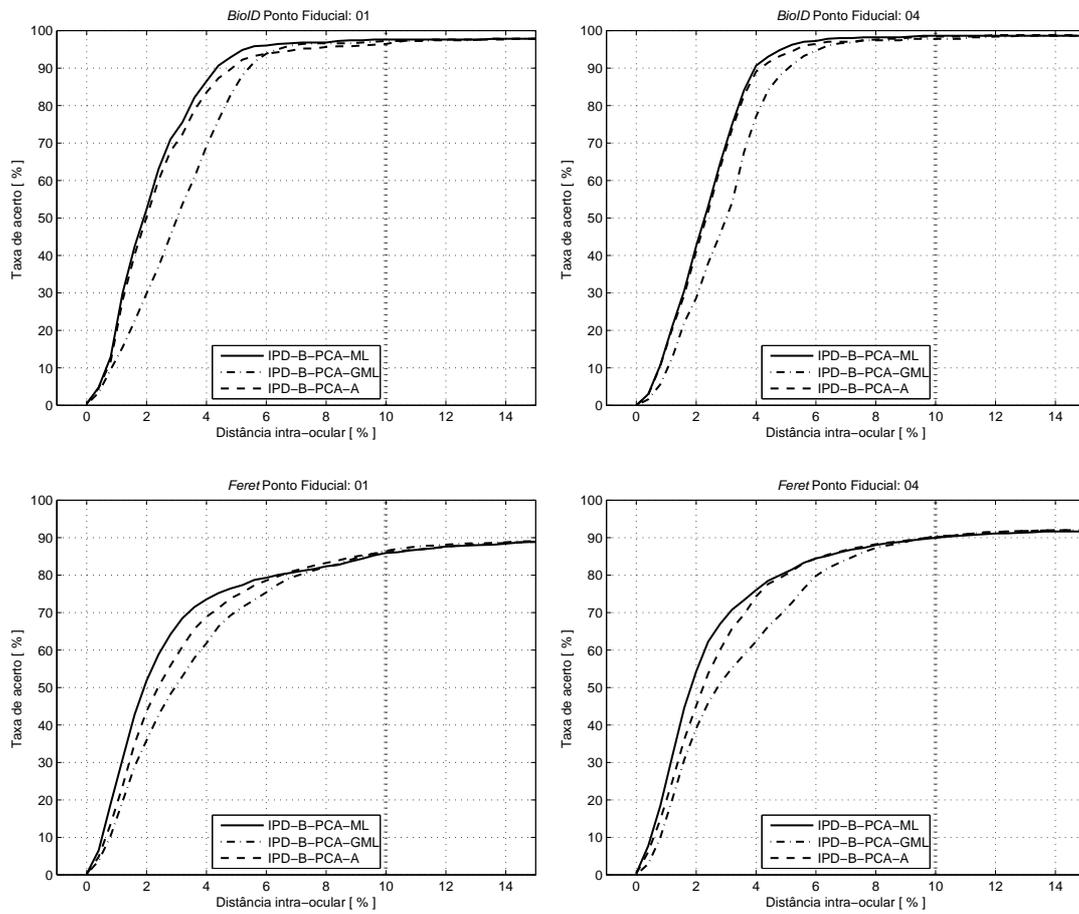


Figura 7.19: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 01 e 04.

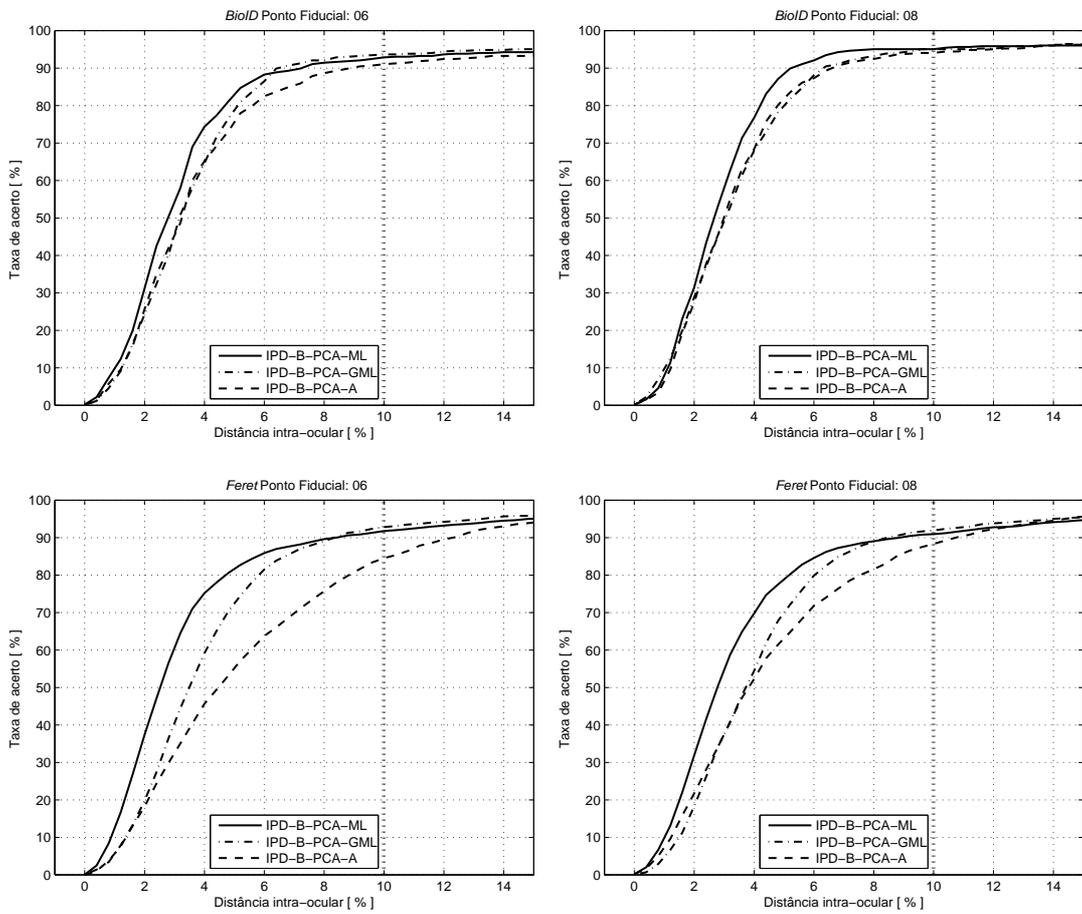


Figura 7.20: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 06 e 08.

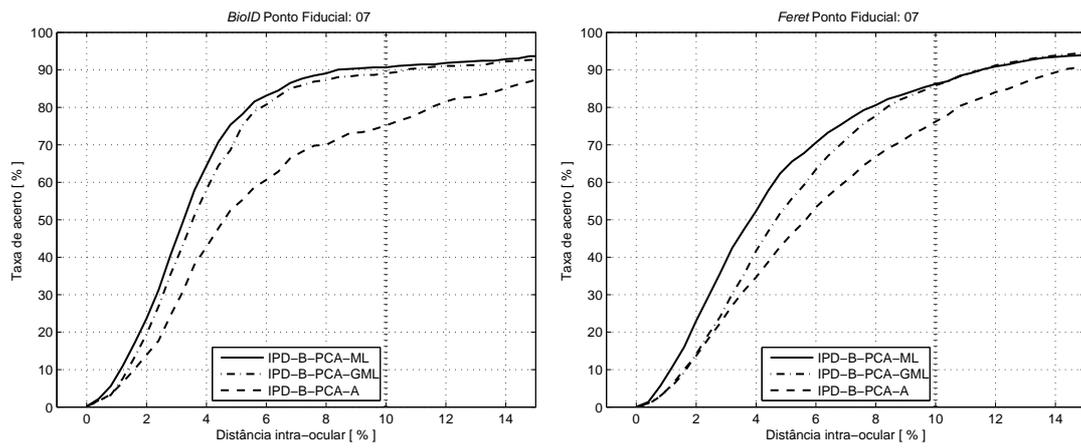


Figura 7.21: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para o ponto fiducial 07.

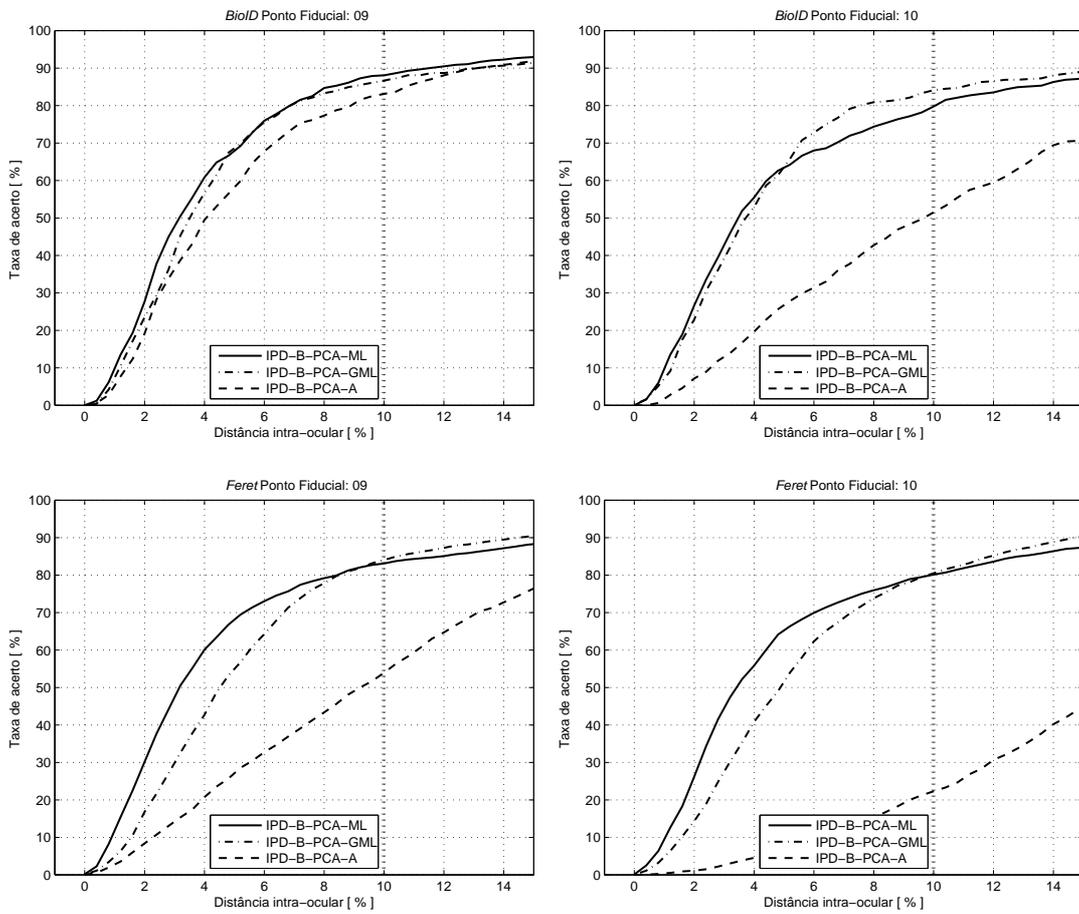


Figura 7.22: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA utilizando estratégias ML, GML e A para os pontos fiduciais 09 e 10.

7.7 Estratégias para Pós-processamento: Comparação entre IPD-B-PCA, DF-B-PCA e SVM

Nesta seção, semelhante à Seção 5.10, iremos comparar o desempenho do método IPD-B-PCA frente aos métodos DF-B-PCA, SVM-L, SVM-Pol utilizando-se estratégias de pós-processamento. Para o método IPD-B-PCA utilizamos a técnica GML. Os gráficos, apresentados nas Figuras 7.23 até 7.28, estão dispostos no mesmo formato apresentado na Seção 7.6. Observe que acrescentamos uma curva referente ao método IPD-B-PCA. Após analisarmos as figuras podemos verificar que:

1. O IPD-B-PCA supera os resultados obtidos pelo método DF-B-PCA em quase todos os pontos fiduciais (as exceções são os pontos fiduciais 01 e 04 da base *Feret*).
2. O IPD-B-PCA fornece resultados expressivos frente aos obtidos pelo SVM-Pol. Para a base *BioID*, em cinco pontos fiduciais (00, 05, 02, 08 e 07) as taxas de acerto do IPD-B-PCA superam as do método SVM-Pol e em dois pontos fiduciais (01 e 04) as taxas possuem diferença mínima. Para a base *Feret*, o IPD-B-PCA supera todos os métodos em um total de sete pontos fiduciais (01, 05, 02, 06, 07, 08 e 09). Os resultados são competitivos em dois pontos fiduciais (03 e 10), nestes casos, o IPD-B-PCA supera os resultados obtidos para o método DF-B-PCA e SVM-L.
3. Para a base *Feret*, o IPD-B-PCA apresenta resultados inferiores aos dos métodos DF-B-PCA e SVM-Pol para os pontos fiduciais localizados no centro dos olhos.

Os gráficos demonstram que o método proposto (IPD-B-PCA) supera os resultados do método DF-B-PCA. Frente aos resultados do método SVM-Pol temos que o IPD-B-PCA fornece resultados superiores, para a base *BioID*, em cinco pontos fiduciais ao passo que nos outros pontos os resultados são competitivos. Novamente, podemos verificar que o uso da base de dados *Feret* confirma os bons resultados obtidos para o método IPD-B-PCA. Na maioria dos pontos fiduciais o IPD-B-PCA apresenta os melhores resultados. Note que os resultados obtidos para o IPD-B-PCA nos pontos fiduciais localizados nos cantos externos dos olhos e no centro do nariz estão com expressivas diferenças para os demais métodos. Observe que os resultados para os pontos fiduciais 03 e 10 ficam um pouco abaixo do melhor resultado, obtido pelo método SVM-Pol.

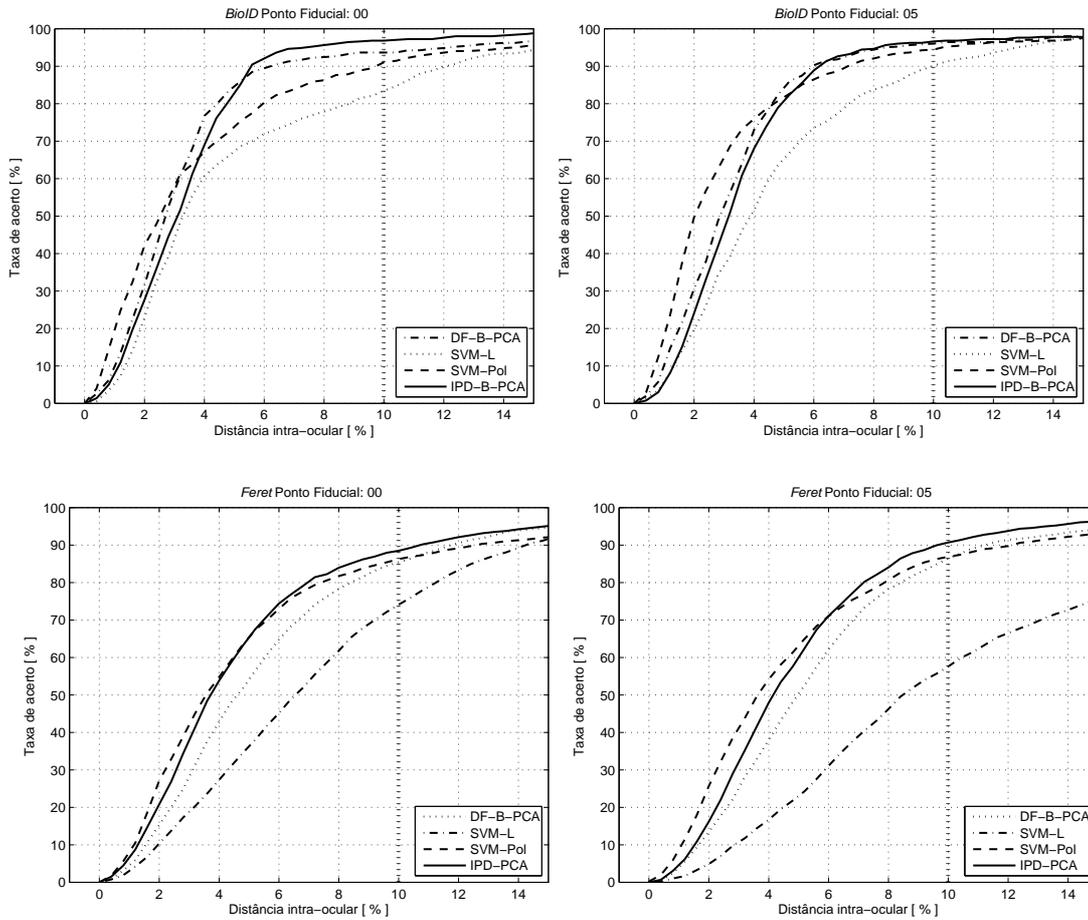


Figura 7.23: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 00 e 05.

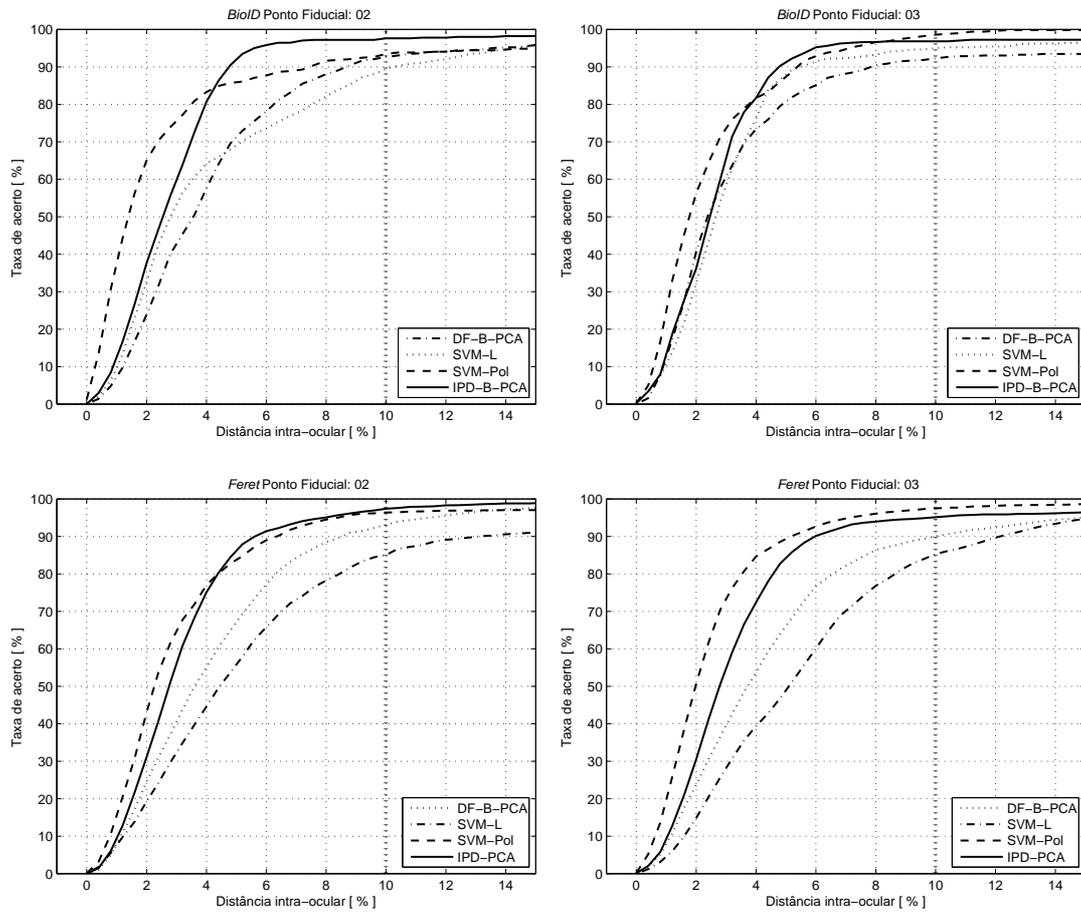
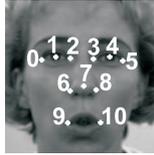


Figura 7.24: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04.

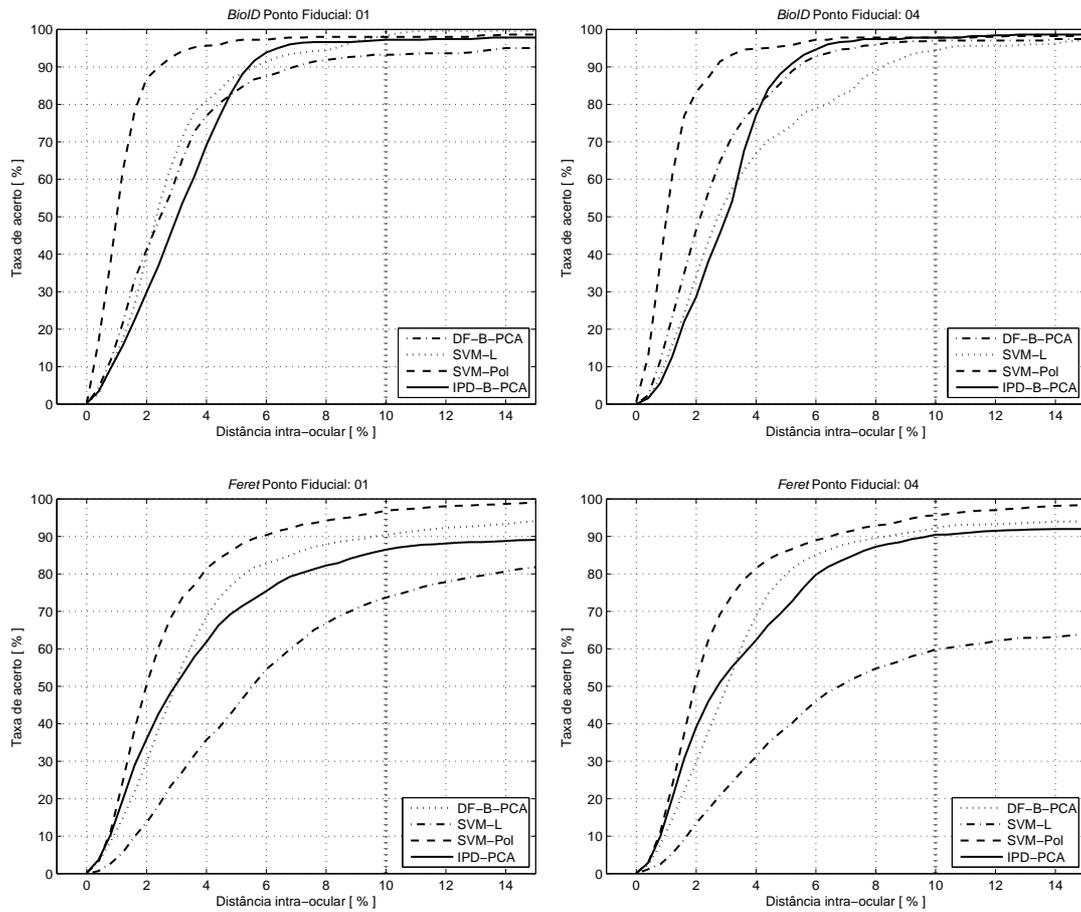
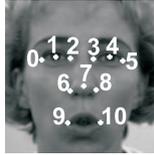


Figura 7.25: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04.

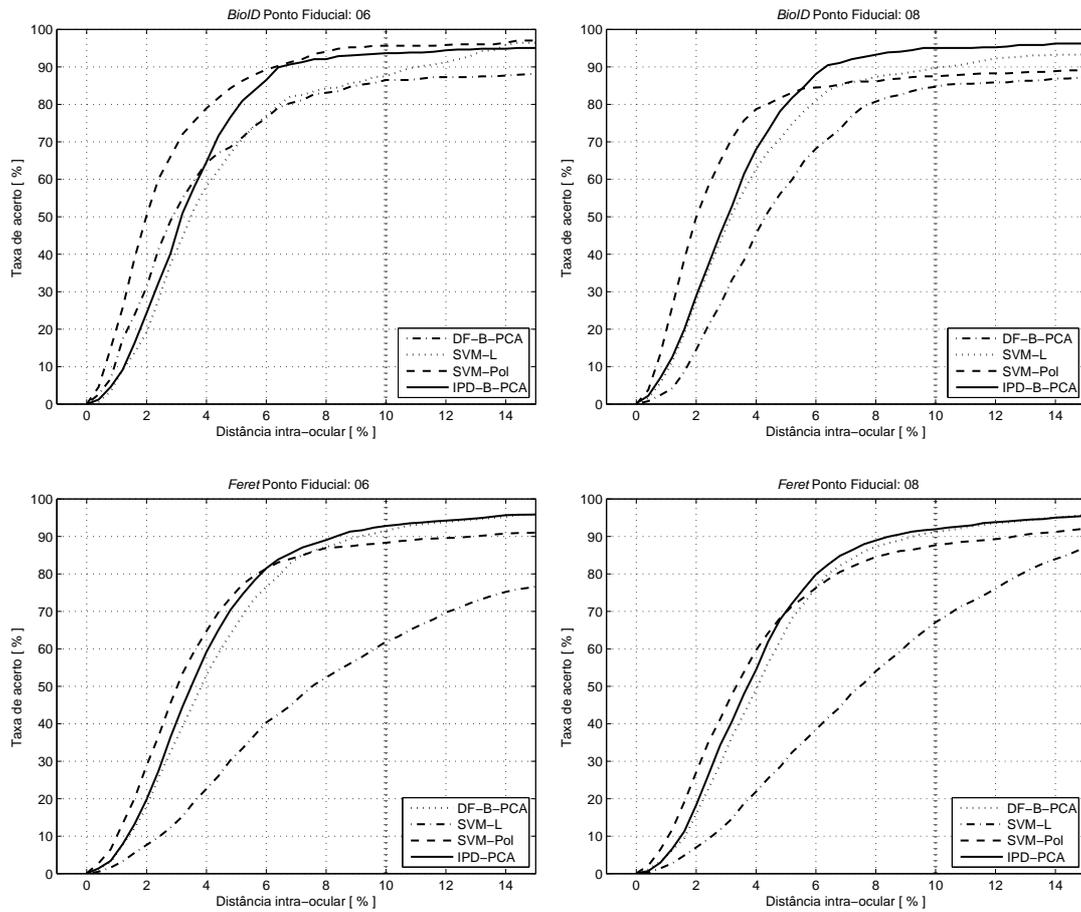
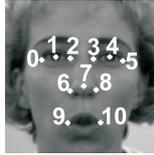


Figura 7.26: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04.

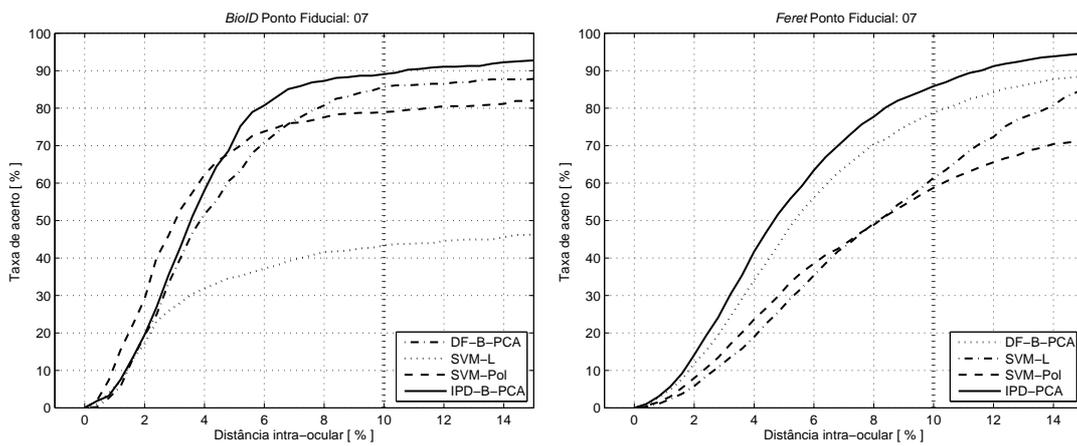


Figura 7.27: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04.

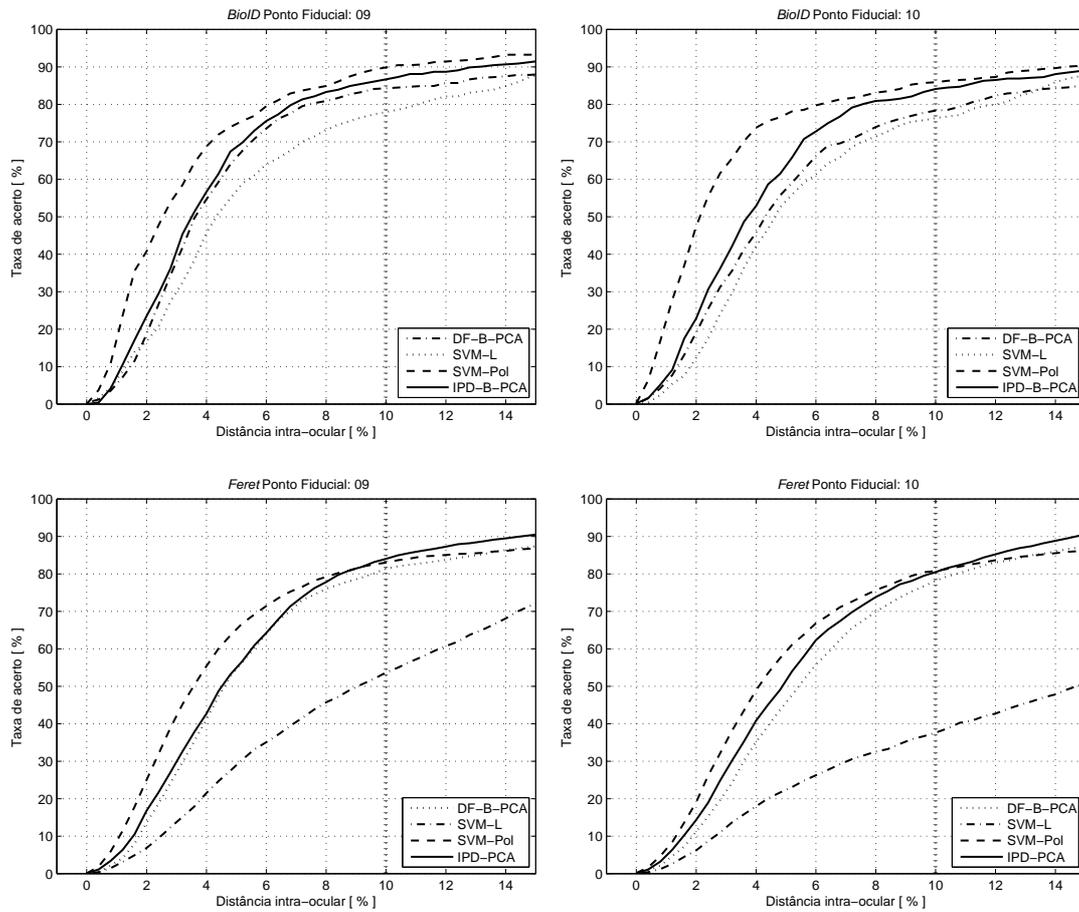
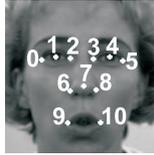


Figura 7.28: Base de dados *BioID* e *Feret*: resultados para o método IPD-B-PCA, DF-B-PCA, SVM-Pol, SVM-L, utilizando as estratégias, respectivamente, ML, GML, GML e ML para os pontos fiduciais 01 e 04.

7.8 Considerações Finais

Neste capítulo, atacamos o problema da detecção de pontos fiduciais em faces humanas com o objetivo de validarmos o método IPD. Nós comparamos os resultados obtidos para o IPD com os dos métodos por Filtragem Discriminativa e por SVM. Os experimentos para todos os métodos, foram realizados utilizando-se dois subconjuntos das bases de dados *BioID* e *Feret*, com respectivamente, 503 e 2004 imagens. Nós consideramos 11 pontos fiduciais e os resultados foram obtidos utilizando-se validação cruzada com 7 *folds*.

Os resultados obtidos evidenciam que o método IPD-B-PCA proporciona um ganho superior ao obtido pelo método por Filtragem Discriminativa com PCA. Quando comparado ao SVM com kernel polinomial o IPD-B-PCA proporciona resultados competitivos e em vários casos superiores.

No próximo capítulo, apresentamos as conclusões e as propostas de trabalhos futuros desta Tese.

Capítulo 8

Conclusões

8.1 Considerações Finais da Tese

Nesta Tese, nós investigamos o desenvolvimento de novos algoritmos para a detecção de padrões. Os métodos propostos se mostraram eficazes no contexto do problema de detecção de pontos fiduciais, que vem sendo utilizado de forma crescente em sistemas de Visão Computacional e constitui um problema sem solução fechada. Em síntese, podemos mencionar que foram realizadas duas contribuições principais.

Na primeira contribuição, apresentada no Capítulo 4, incorporamos robustez à Filtragem Discriminativa de tal forma que os filtros discriminativos sejam capazes de detectar padrões com leves rotações e deformações. Neste novo método, diferentes filtros discriminativos foram projetados para as componentes principais com maiores energias. Vislumbramos que esta abordagem fornece uma interessante interpretação para obtermos a matriz de covariância utilizada no projeto dos filtros discriminativos. Neste contexto, os padrões que não são de interesse situam-se em um subespaço ortogonal ao da componente principal de interesse.

Na segunda, apresentada no Capítulo 6, foi desenvolvido um método para detecção de padrões denominado Detector por Produto Interno (IPD) que utiliza um simples produto interno para determinar se um candidato é ou não o padrão de interesse. Foi demonstrado que o método por Filtragem Discriminativa e os filtros obtidos através da DFT, denominados por filtros MACE e OTF (frequentemente empregados na literatura), são casos particulares dos detectores IPD. Por fim, foi demonstrado como projetar detectores IPD robustos. Para tal, foram projetados

detectores IPD para as componentes principais. Neste contexto, a matriz de covariância pode ser determinada de forma semelhante à realizada para o método por Filtragem Discriminativa.

Os desempenhos dos métodos propostos, apresentados nos Capítulos 5 e 7, foram avaliados considerando-se um problema de detecção de pontos fiduciais em faces humanas. As simulações para onze pontos fiduciais foram feitas utilizando-se validação cruzada para duas bases de dados (*BioID* e *Feret*) com, respectivamente, 503 e 2004 imagens. Para comparação, métodos similares foram projetados utilizando classificadores SVM linear e não-linear.

As modificações propostas para a Filtragem Discriminativa, que deram origem ao método DF-B-PCA (Capítulo 4), proporcionaram resultados superiores frente ao desempenho do método original. Assim, as modificações propostas evidenciam que o método DF-B-PCA é eficaz. Para os detectores por produto interno, o algoritmo final desenvolvido, denominado IPD-B-PCA (Capítulo 6), apresentou desempenho superior ao obtido pelo método DF-B-PCA. O IPD-B-PCA apresentou resultados competitivos ou superiores, em comparação ao desempenho obtido por classificadores similares que utilizam SVM não linear que estão entre os representantes estado da arte em Reconhecimento de Padrões. A abordagem que utiliza PCA se mostrou bastante vantajosa para incrementar a robustez dos métodos.

Por fim, podemos comentar que os métodos propostos, frente aos desempenhos obtidos para bases de dados extensas e diversificadas, foram aplicados com sucesso em um problema de detecção de padrões relevante para a comunidade científica, validando a teoria desenvolvida.

8.2 Propostas para Trabalhos Futuros

Neste seção, apresentamos algumas propostas para desenvolvimentos futuros relacionados aos métodos expostos nesta Tese. Estas propostas podem ser divididas em duas categorias. Na primeira, consideramos modificações que podem ser feitas nos algoritmos e, na segunda, indicamos potenciais aplicações dos algoritmos a outros problemas relevantes. Assim, as propostas futuras são:

Quanto a modificações nos algoritmos:

- O alinhamento dos padrões de interesse no conjunto de treinamento é de fundamental importância para o sucesso dos métodos propostos. Neste sentido, podem-se investigar técnicas de alinhamento automáticas que considerem características peculiares do padrão de interesse. Por exemplo, considerando-se o ponto fiducial localizado no centro dos olhos, pode-se utilizar um detector de bordas para delinear a circunferência da íris, proporcionando melhor alinhamento dos centros dos olhos.
- Algumas técnicas de pós-processamento podem ser incorporadas aos métodos DF-B-PCA e IPD-B-PCA visando um incremento no desempenho. Duas técnicas podem ser implementadas. A primeira é levar em consideração o valor de $DSNR_2$ (ou o valor do produto interno) quando se realiza o pós-processamento seja através do agrupamentos dos rótulos e seleção do mais provável (pós-processamento GML) seja através da média dos rótulos automáticos (pós-processamento A). A segunda é utilizar informações do ponto fiducial simétrico, quando existir, fazendo uma decisão conjunta.
- Podemos modificar o filtro discriminativo de tal forma que ele seja capaz de detectar um padrão mesmo com grandes rotações. Para tal, podemos projetar filtros discriminativos levando-se em consideração as Funções Harmônicas Circulares (CHF) [113, 114] (em inglês, *Circular Harmonic Function*).

Quanto a potenciais aplicações:

- Resultados preliminares obtidos para a complexidade computacional dos métodos evidenciam que eles podem ser utilizados para aplicações em tempo real. Por exemplo, pode-se aplicar os detectores desenvolvidos para rastreamento de pontos fiduciais em faces humanas.
- Tanto os filtros discriminativos quanto os detectores por produto interno podem ser aplicados em uma abordagem conjunta com o *Elastic Bunch Graph Matching* (EBGM) [115, 116] para identificação de faces humanas. Neste contexto, pode-se substituir os filtros de *Gabor* por filtros discriminativos ou detec-

tores por produto interno. Assim, intenciona-se associar cada face aos filtros discriminativos (ou detectores IPD).

- Resultados preliminares demonstram que podemos introduzir informações sobre localizações relativas aos pontos fiduciais utilizando os métodos propostos em conjunto com técnicas de grafo para detecção de faces humanas.
- Os métodos DF e IPD podem ser aplicados em problemas de biometria (tais como identificação por íris e impressões digitais) devido a características semelhantes aos padrões de interesse utilizados nesta Tese.

Referências Bibliográficas

- [1] KIRBY, M., SIROVICH, L., “Application of the Karhunen-Loève Procedure for the Characterization of Human Faces”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 12, 1990.
- [2] TURK, M., PENTLAND, A., “Face Recognition using Eigenfaces”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Maui, June 1991.
- [3] BELHUMEUR, P., HESPANHA, J., KRIEGMAN, D., “Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, January 1997.
- [4] MOGHADDAM, B., WAHID, W., PENTLAND, A., “Beyond Eigenfaces: Probabilistic Matching for Face Recognition”. In: *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, March 1998.
- [5] PENTLAND, A., MOGHADDAM, B., STARNER, T., “View-Based and Modular Eigenspaces for Face Recognition”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 84–91, Seattle, WA, USA, April 1994.
- [6] DA SILVA, P. Q., DE CASTRO SANTA ROSA, A. N., “Reconhecimento Facial Aplicado a Perícia Criminal”. In: *Proceedings da Conferência Internacional de Perícias em Crimes Cibernéticos*, p. 12, Brasília, DF, Brasil, 2004.

- [7] CHEUNG, W., HAMARNEH, G., “N-SIFT: N-Dimensional Scale Invariant Feature Transform for Matching Medical Images”. In: *Proceedings of the International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 720–723, Washington, DC, USA, April 2007.
- [8] LOWE, D. G., “Object Recognition from Local Scale-Invariant Features”. In: *Proceedings of the International Conference on Computer Vision*, pp. 1150–1157, Ft. Collins, CO, USA, June 1999.
- [9] LOWE, D. G., “Distinctive Image Features from Scale Invariant Keypoints”. In: *Proceedings of the International Conference on Computer Vision*, Washington, DC, USA, June 2004.
- [10] NANDY, D., BEN-ARIE, J., “EXM Eigen Templates for Detecting and Classifying Arbitrary Junctions”. In: *Proceedings of the International Conference on Image Processing*, pp. 211–215, Kobe, Japan, October 1998.
- [11] BEN-ARIE, J.; RAO, K. R., “A Novel Approach for Template Matching by Nonorthogonal Image Expansion”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 3, n. 1, pp. 71–84, February 1993.
- [12] ZHIQIAN WANG; RAO, K. R.; BEN-ARIE, J., “Optimal Ramp Edge Detection using Expansion Matching”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 18, n. 11, pp. 1092–1097, November 1996.
- [13] RAO, K. R.; BEN-ARIE, J., “Optimal Edge Detection using Expansion Matching and Restoration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, n. 12, pp. 1169–1182, December 1994.
- [14] MENDONÇA, A. P., *Algoritmos para Discriminação de Padrões em Imagens*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Março de 2003.
- [15] NISHINO, K., SATO, Y., IKEUCHI, K., “Eigen-Texture Method: Appearance Compression and Synthesis Based on a 3D Model”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, pp. 1257–1265, November 2001.

- [16] LIU, J., CHEN, S., ZHOU, H., et al., “Single Image Subspace for Face Recognition”. In: *Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 205–219, Rio de Janeiro, Brazil, October 2007.
- [17] MATEOS, G. G., GARCÍA, A. R., TERUEL, P. E. L. D., “Human Face Processing with 1.5D Models”. In: *Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 220–234, Rio de Janeiro, Brazil, October 2007.
- [18] MALCOLM, J., RATHI, Y., TANNENBAUM, A., “Multi-Object Tracking Though Clutter using Graph Cust”. In: *Proceedings of the International Workshop on Non-rigid Registration and Tracking through Learning*, 2007.
- [19] FORSYTH, D. A., PONCE, J., *Computer Vision: A Modern Approach*. 1st ed. Prentice Hall, 2002.
- [20] BISHOP, C. M., *Pattern Recognition and Machine Learning*. 1st ed. Springer: New York, NY, USA, 2006.
- [21] MENDONÇA, A. P., SILVA, E. A. B., “Filtragem Discriminativa para Detecção de Múltiplos Padrões de Imagem”. In: *Anais do XX Simpósio Brasileiro de Telecomunicações*, Rio de Janeiro, RJ, Brasil, Outubro de 2003.
- [22] MENDONÇA, A. P., SILVA, E. A. B., “Filtros Discriminativos para a Detecção de Gabaritos de Imagens”. In: *Anais do XIX Símposio Brasileiro de Telecomunicações*, Fortaleza, CE, Brasil, Setembro de 2001.
- [23] MENDONÇA, A. P., SILVA, E. A. B., “Reduced Cross-Discrimination for Discriminative Filters”. In: *Proceedings of the International Conference on Image Processing*, pp. 2115–2118, Singapore, October 2004.
- [24] MENDONÇA, A. P., SILVA, E. A. B., “Discriminative Filtering as an Impulse Restoration Problem”. In: *Proceedings of the IEEE International Telecommunication Symposium*, Natal, RN, Brasil, September 2002.

- [25] MENDONÇA, A. P., SILVA, E. A. B., “Two-Dimensional Discriminative Filters for Image Template Detection”. In: *Proceedings of the International Conference on Image Processing*, Thessaloniki, Greece, September 2001.
- [26] MENDONÇA, A. P., SILVA, E. A. B., “Multiple Template Detection using Impulse Restoration and Discriminative Filters”, *IEE Electronics Letters*, v. 39, n. 16, pp. 1172–1174, August 2003.
- [27] MENDONÇA, A. P., SILVA, E. A. B., “Closed-Form Solutions for Discriminative Filtering using Impulse Restoration Techniques”, *IEE Electronics Letters*, v. 38, n. 22, pp. 1332–1333, October 2002.
- [28] SIQUEIRA, C. M. O., *Algoritmos Robustos de Filtragem Discriminativa*, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Março de 2006.
- [29] DUDA, R., HART, P., STORK, D., *Pattern Classification*. 2nd ed. Wiley-Interscience: Secaucus, NJ, USA, 2001.
- [30] JAIN, A. K., *Fundamentals of digital image processing*. 2nd ed. Prentice-Hall: Upper Saddle River, NJ, USA, 1989.
- [31] XIE, C., SAVVIDES, M., VIJAYA KUMAR, B. V. K., “Redundant Class-Dependence Feature Analysis Based on Correlation Filters Using FRGC2.0 Data”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, p. 153, San Diego, CA, USA, June 2005.
- [32] MAHALANOBIS, A., VIJAYA KUMAR, B. V. K., CASASENT, D., “Minimum Average Correlation Energy Filters”, *Applied Optics*, v. 26, pp. 3633–3640, 1987.
- [33] XIE, C., SAVVIDES, M., VIJAYA KUMAR, B. V. K., “Correlation Pattern Recognition for Face Recognition”, *Proceedings of the IEEE*, v. 94, n. 11, pp. 1963–1976, 2006.
- [34] ROBERGE, D., SOUTAR, C., VIJAYA KUMAR, B. V. K., “Reliable Face Recognition using Adaptive and Robust Correlation Filters”, *Elsevier Computer Vision and Image Understanding*, v. 111, pp. 329–350, 2008.

- [35] SAMAD, S. A., RAMLI, D. A., HUSSAIN, A., “Region Selection for Robust Face Verification Using UMACE Filters”. In: *Proceedings of the International Conference on Electrical Engineering and Informatic*, p. 67, Indonesia, June 2007.
- [36] THORNTON, J., SAVVIDES, M., VIJAYA KUMAR, B. V. K., “A Bayesian Approach to Deformed Pattern Matching of Iris Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, pp. 596–606, 2007.
- [37] LAI, H., RAMANATHAN, V., WECHSLER, H., “Optimal Trade-off Filter for the Correlation of Fingerprints”, *Applied Optics*, v. 38, pp. 108–113, 1999.
- [38] LIU, T., ZHU, G., ZHANG, C., et al., “Fingerprint indexing based on singular point correlation”. In: *Proceedings of the IEEE International Conference on Image Processing*, pp. 293–6, Genoa, Italy, September 2005.
- [39] YEOMANS, H., VIJAYA KUMAR, B. V. K., “Palmprint Classification using Pattern-specific Segmentation and Optimal Correlation Filters”, *IEEE Transactions on Information Forensics and Security*, v. 2, pp. 613–622, 2007.
- [40] SHI, J., SAMAL, A., MARX, D., “How effective are landmarks and their geometry for face recognition?” *Computer Vision and Image Understanding*, v. 102, n. 2, pp. 117 – 133, 2006.
- [41] SIVIC, J., EVERINGHAM, M., ZISSERMAN, A., “Who Are You? - Learning Person Specific Classifiers from Video”. In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 1145–1152, Florida, US, jun 2009.
- [42] LI TIAN, Y., KANADE, T., COHN, J. F., “Recognizing Action Units for Facial Expression Analysis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, pp. 97–115, 2001.

- [43] ZHAO, S., GAO, Y., “Automated Face Pose Estimation Using Elastic Energy Models”. In: *Proceedings of the International Conference on Pattern Recognition*, pp. 618–621, Washington, DC, USA, 2006.
- [44] NAIR, P., CAVALLARO, A., “3-D Face Detection, Landmark Localization, and Registration using a Point Distribution Model”, *IEEE Transactions on Multimedia*, v. 11, n. 4, pp. 611–623, 2009.
- [45] COHN, J. F., ZLOCHOWER, A. J., LIEN, J. J., et al., “Feature-Point Tracking by Optical Flow Discriminates Subtle Differences in Facial Expression”. In: *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 396–401, Los Alamitos, CA, USA, 1998.
- [46] NARUNIEC, J., SKARBEEK, W., RAMA, A., “Face Detection and Tracking in Dynamic Background of Street”. In: *Proceedings of the Second International Conference on Signal Processing and Multimedia Applications*, pp. 414–420, Barcelona, Spain, 2007.
- [47] CORTES, C., VAPNIK, V., “Support-Vector Networks”, *Machine Learning*, v. 20, n. 3, pp. 273–297, September 1995.
- [48] JAIN, A. K., DUIN, R. P. W., MAO, J., “Statistical Pattern Recognition: A Review”, v. 22, pp. 4–37, 2000.
- [49] WATANABE, S., *Pattern Recognition: Human and Mechanical*. Wiley, 1985.
- [50] PAPA, J. P., *Classificação Supervisionada de Padrões utilizando Floresta de Caminhos Ótimos*, Tese de D.Sc., UNICAMP/IC, Campinas , SP, Brasil, Novembro de 2008.
- [51] FREUND, Y., SCHAPIRE, R. E., “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, *Journal of Computer System*, v. 55, pp. 119–139, December 1997.
- [52] JEROME, F., TREVOR, H., TIBSHIRANI, R., “Additive Logistic Regression: a Statistical View of Boosting”, *Annals of Statistics*, v. 28, pp. 2000, 1998.

- [53] “The GML AdaBoost Matlab toolbox”, Disponível em: <http://graphics.cs.msu.ru/en/science/research/machinelearning/adaboosttoolbox>. Acesso em: Setembro de 2009.
- [54] DU, C., WU, Q., YANG, J., et al., “SVM based ASM for facial landmarks location”. In: *Proceedings of the International Conference on Computer and Information Technology*, pp. 321–326, Washington, DC, USA, 2008.
- [55] OSUNA, E. E., FREUND, R., GIROSI, F., *Support Vector Machines: Training and Applications*, Tech. rep., 1997.
- [56] BURGESS, C. J. C., “A tutorial on support vector machines for pattern recognition”, *Data Mining and Knowledge Discovery*, v. 2, n. 2, pp. 1–47, September 1998.
- [57] JOACHIMS, T., BURGESS, C., SCHOLKOPF, B., et al., (eds), “Advances in kernel methods: support vector learning”, chap. Making large-scale support vector machine learning practical, pp. 169–184, MIT press: Cambridge, MA, 1999.
- [58] MILBORROW, S., NICOLLS, F., “Locating Facial Features with an Extended Active Shape Model”. In: *Proceedings of the European Conference on Computer Vision*, pp. 504–513, Marseille, France, October 2008.
- [59] CRISTINACCE, D., COOTES, T., “Facial feature detection and tracking with automatic template selection”. In: *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 429–434, July 2006.
- [60] CRISTINACCE, D., COOTES, T., “Automatic Feature Localization with Constrained Local Models”, *Journal of Pattern Recognition*, v. 41, n. 10, pp. 3054–3067, 2008.
- [61] COOTES, T. F., TAYLOR, C. J., COOPER, D. H., et al., “Active shape models—their training and application”, *Computer Vision and Image Understanding*, v. 61, n. 1, pp. 38–59, 1995.

- [62] COOTES, T. F., EDWARDS, G. J., TAYLOR, C. J., “Active Appearance Models”. In: *Proceedings of the European Conference on Computer Vision*, pp. 484–498, Freiburg, DE, jun 1998.
- [63] SIVIC, J., EVERINGHAM, M., ZISSERMAN, A., “Who are you?- Learning person specific classifiers from video”. In: *CVPR*, pp. 1145–1152, 2009.
- [64] ARCA, S., CAMPADELLI, P., LANZAROTTI, R., “An Efficient Method to Detect Facial Fiducial Points for Face Recognition”. In: *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 532–535, Cambridge, UK, August 2004.
- [65] VUKADINOVIC, D., PANTIC, M., “Fully Automatic Facial Feature Point Detection using Gabor Feature Based Boosted Classifiers”. In: *Proceedings of the International Conference on Systems, Man and Cybernetics*, pp. 1692–1698, Waikoloa, Hawaii, December 2005.
- [66] ZHU, Z., JI, Q., “Robust Pose Invariant Facial Feature Detection and Tracking in Real-Time”. In: *Proceedings of the International Conference on Pattern Recognition*, pp. 1092–1095, Hong Kong, China, August 2006.
- [67] Disponível em: <http://www.luxand.com>. Acesso em: Novembro de 2009.
- [68] Disponível em: <http://www.pittpatt.com>. Acesso em: Dezembro de 2009.
- [69] BRANKOV, J. G., GALATSANOS, N. P., YANG, Y., et al., “Similarity Based Clustering using the Expectation Maximization Algorithm”. In: *Proceedings of the International Conference on Image Processing*, pp. 97–100, Rochester, New York, USA, September 2002.
- [70] ROHLFING, T., RUSSAKOFF, D. B., CALVIN, R. M., “An Expectation Maximization-Like Algorithm for Multi-Atlas Multi-Label Segmentation”. In: *Proceedings des Workshops Bildverarbeitung fur die Medizin*, pp. 348–352, Berlin, Germany, März 2003.
- [71] ABU NASER, A., GALATSANOS, N., WERNICK, M., “Impulse Restoration-Based Template-Matching using the Expectation-Maximization Algo-

- rithm”. In: *Proceedings of the International Conference on Image Processing*, p. 158, Washington, DC, October 1997.
- [72] ABU NASER, A., *Impulse Restoration-Based Template-Matching*, Ph.D. Dissertation, University of Illinois, Chicago, USA, 2000.
- [73] HAYKIN, S., *Communication Systems*. 4th ed. John Wiley e Sons, 2000.
- [74] MOSCI, S., ROSASCO, L., VERRI, A., “Dimensionality Reduction and Generalization”. In: *Proceedings of the International Conference on Machine Learning*, pp. 657–664, Corvalis, Oregon, USA, 2007.
- [75] WASHIZAWA, Y., YAMASHITA, Y., “Kernel Projection Classifiers with Suppressing Features of Other Classes”, *Neural Computing*, v. 18, n. 8, pp. 1932–1950, 2006.
- [76] CHANG, L., CHENG, C.-M., CHEN, T.-C., “An Efficient Adaptive KLT for Multispectral Image Compression”. In: *Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, p. 252, Washington, DC, 2000.
- [77] CASTELLS, F., LAGUNA, P., SÖRNMO, L., et al., “Principal Component Analysis in ECG Signal Processing”, *Journal Applied Signal Processing*, v. 2007, pp. 98–98, 2007.
- [78] SU, Z., LI, S., ZHANG, H., “Extraction of feature subspaces for content-based retrieval using relevance feedback”. In: *Proceedings of the ACM International Conference on Multimedia*, pp. 98–106, Ottawa, Canada, 2001.
- [79] XU, D., WANG, Y., “An Automated Feature Extraction and Emboli Detection System Based on the Pca and Fuzzy Sets”, *Computers in Biology and Medicine*, v. 37, n. 6, pp. 861–871, 2007.
- [80] WANG, H., LENG, Y., WANG, Z., et al., “Application of Image Correction and Bit-Plane Fusion in Generalized PCA Based Face Recognition”, *Elsevier Pattern Recognition Letter*, v. 28, pp. 2352–2358, 2007.

- [81] HOTTA, K., “View-Invariant Face Detection Method Based on Local PCA Cells”. In: *Proceedings of the International Conference on Image Analysis and Processing*, p. 57, Washington, DC, 2003.
- [82] MELO, G., RODRIGO, S., BAUCHSPIESS, A., “Estudo da Eficiência do Reconhecimento Facial Utilizando PCA Incremental”. In: *Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, Curitiba, PR, Brasil, Novembro de 2004.
- [83] LI, S. Z., JAIN, A. K., *Handbook of Face Recognition*. 1st ed. Springer-Verlag: Secaucus, NJ, USA, 2004.
- [84] LOPES, L. F. D., *Análise de Componentes Principais Aplicada à Confiabilidade de Sistemas Complexos*, Tese de D.Sc., UFSC, Florianópolis, SC, Brasil, Agosto de 2001.
- [85] JONATHON, S., “A Tutorial on Principal Component Analysis”, Disponível em: <http://www.snل.salk.edu/shlens/pub/notes/pca.pdf>. Acesso em: Outubro de 2008.
- [86] KLEMA, V.; LAUB, A., “The Singular Value Decomposition: Its Computation and Some Applications”, *IEEE Transactions on Automatic Control*, v. 25, pp. 164–176, April 1980.
- [87] KALMAN, D., “A Singularly Valuable Decomposition: The SVD of a Matrix”, *The College Mathematics Journal*, v. 27, pp. 2–23, 1996.
- [88] Disponível em: <http://www.uwlax.edu/faculty/will/svd/index.html>. Acesso em: Dezembro de 2007.
- [89] LAI, H., LI, F., WECHSLER, H., “Robust Face Recognition Strategies using Feed-Foward Architectures and Parts”. In: *Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 290–304, 2007.
- [90] ASHIMINE, J., DE CAMPOS, T., CESAR, R., et al., “Person Recognition Through Eigeneye Selection Based on Fuzzy Distance to Multiple Class

- Prototypes”. In: *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pp. 407–440, 2002.
- [91] Disponível em: <http://www.bioid.com/support/downloads/software/bioid-face-database.html>. Acesso em: Abril de 2010.
- [92] Disponível em: <http://face.nist.gov/colorferet/>. Acesso em: Abril de 2010.
- [93] PHILLIPS, P. J., WECHSLER, H., HUANG, J., et al., “The FERET database and evaluation procedure for face recognition algorithms”, *Journal of Image and Vision Computing*, v. 16, n. 5, pp. 295–306, 1998.
- [94] PHILLIPS, P. J., MOON, H., RIZVI, S. A., et al., “The FERET Evaluation Methodology for Face Recognition Algorithms”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, v. 22, pp. 1090–1104, 2000.
- [95] VIOLA, P., JONES, M., “Robust Real-Time Object Detection”, *International Journal of Computer Vision*, v. 57, n. 2, pp. 137–154, July 2001.
- [96] VIOLA, P., JONES, M., “Rapid Object Detection Using a Boosted Cascade of Simple Features”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 511–518, Kauai Marriott, Hawaii, December 2001.
- [97] TAN, X., TRIGGS, B., “Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions”. In: *Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 168–182, Rio de Janeiro, Brazil, October 2007.
- [98] TAN, X., TRIGGS, B., “Fusing Gabor and LBP Feature Sets for Kernel-Based Face Recognition”. In: *Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 235–249, Rio de Janeiro, BR, October 2007.
- [99] VIJAYA KUMAR, B. V. K., MAHALANOBIS, A., JUDAY, R. D., *Correlation Pattern Recognition*. 1st ed. Cambridge University: Cambridge, UK, 2005.

- [100] KEREEKES, R. A., VIJAYA KUMAR, B. V. K., “Correlation Filters With Controlled Scale Response”, *IEEE Transactions on Image Processing*, v. 15, pp. 1794–1802, 2006.
- [101] HYVARINEN, A., KARHUNEN, J., OJA, E., *Independent Component Analysis*. 2nd ed. Wiley-Interscience, 2001.
- [102] HYVÄRINEN, A., OJA, E., “Independent Component Analysis: Algorithms and Applications”, In: Neural Networks Research Centre. Disponível em: <http://www.cs.helsinki.fi/u/ahyvarin/papers/NN00new.pdf>. Acesso em: Novembro de 2008.
- [103] STONE, J., *Independent Component Analysis: A Tutorial Introduction*. 1st ed. Cambridge University: New York, NY, USA, 2004.
- [104] CAMPBELL, S. L., MEYER, C. D., *Generalized Inverses of Linear Transformations*. 1st ed. Dover: Mineola, NY, USA, 1991.
- [105] ANTUNES, E. J. B., *Detecção de Frequências através de Predição Linear*, Dissertação de M.Sc., UNICAMP, Campinas, SP, Brasil, Maio de 1992.
- [106] SAMAD, S. A., RAMLI, D. A., HUSSAIN, A., “Lower Face Verification Centered on Lips using Correlation Filters”, *Information Technology Journal*, v. 6, pp. 1146–1151, 2007.
- [107] VIJAYA KUMAR, B. V. K., SAVVIDES, M., VENKATARAMANI, K., et al., “Spatial Frequency Domain Image Processing for Biometric Recognition”. In: *Proceedings of the International Conference on Image Processing*, pp. 97–100, Rochester, New York, USA, September 2002.
- [108] JEONG, K. H., HAN, S., PRINCIPE, J. C., “The Fast Correntropy Mace Filter”. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 613–616, Honolulu, Hawai’i, USA, April 2007.
- [109] ABIANTUN, R., SAVVIDES, M., VIJAYA KUMAR, B. V. K., “How Low Can You Go? Low Resolution Face Recognition Study using Kernel Correlation Feature Analysis on the FRGCv2 Dataset”. In: *Proceedings of*

the Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference, pp. 1–6, Baltimore, Maryland, USA, September 2006.

- [110] SAO, A. K., YEGNANARAYANA, B., “Face Verification using Correlation Filters and Autoassociative Neural Networks”. In: *Proceedings of the International Conference on Intelligent Sensing and Information Processing*, pp. 364–367, Melbourne, Australia, December 2004.
- [111] CONDE, C., RUIZ, A., CABELLO, E., “PCA vs Low Resolution Images in Face Verification”. In: *Proceedings of the International Conference on Image Analysis and Processing*, pp. 63–67, Mantova, Italy, September 2003.
- [112] ARAUJO, G. M., *Algoritmos para Reconhecimento de Características Faciais baseado em Filtros de Correlação*, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Março de 2010.
- [113] ROSEN, J., SHAMIR, J., “Circular harmonic phase filters for efficient rotationinvariant pattern recognition”, *Applied Optics*, v. 27, pp. 2895–2899, 1998.
- [114] KUMAR, B. V. K. V., MAHALANOBIS, A., TAKESSIAN, A., “Optimal Tradeoff Circular Harmonic Function Correlation Filter Methods Providing Controlled In-Plane Rotation Response”, *IEEE Transaction on Image Processing*, v. 9, n. 6, pp. 1025–1034, June 2000.
- [115] WISKOTT, L., FELLOUS, J.-M., KRÜGER, N., et al., “Face Recognition by Elastic Bunch Graph Matching”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, v. 19, pp. 775–779, 1997.
- [116] ZHAO, W., CHELLAPPA, R., PHILLIPS, P. J., et al., “Face Recognition: A Literature Survey”, 2000.
- [117] FREUND, Y., SCHAPIRE, R. E., “A Short Introduction to Boosting”, *Journal of Japanese Society for Artificial Intelligence*, v. 14, pp. 771–780, September 1999.

- [118] FREUND, Y., SCHAPIRE, R. E., “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”. In: *Proceedings of the European Conference on Computational Learning Theory*, pp. 23–37, Barcelona, Spain, March 1995.
- [119] FREUND, Y., SCHAPIRE, R. E., “A Brief Introduction to Boosting”. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, July 1999.
- [120] SOUZA, L. V. D., *Programação Genética e Combinação de Preditores para Previsão de Séries Temporais*, Tese de D.Sc., UFPR, Curitiba, PR, Brasil, Março de 2006.
- [121] LIENHART, R., MAYDT, J., “An Extended Set of Haar-Like Features for Rapid Object Detection”. In: *Proceedings of the International Conference on Image Processing*, pp. 900–903, Rochester, New York, USA, September 2002.
- [122] TORRALBA, A., MURPHY, K. P., FREEMAN, W. T., “Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 762–769, Washington, DC, June 2004.
- [123] RAMANA, I., RUSSELL, G., ELGAMMA, A., “Learning a Dynamic Classification Method to Detect Faces and Identify Facial Expression”. In: *Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 69–83, Beijing, China, October 2005.
- [124] VIOLA, P., JONES, M., “Fast Multi-view Face Detection”, In: Report TR2003-096, Mitsubishi Electric Research Laboratories, August 2003.
- [125] ZHANG, H., GAO, W., CHEN, X., et al., “Object detection using spatial histogram features”, *Image and Vision Computing*, v. 24, n. 4, pp. 327–341, 2006.

- [126] GARY, B., ADRIAM, K., VADIM, P., “Learning-Based Computer Vision with Intel’s Open Source Computer Vision Library”, *Intel Technology Journal*, v. 09, n. 2, pp. 118–131, May 2005.
- [127] PAPAGEORGIOU, C. P., OREN, M., POGGIO, T., “A General Framework for Object Detection”. In: *Proceedings of the International Conference on Computer Vision*, p. 555, Bombay, Índia, January 1998.
- [128] OREN, M., PAPAGEORGIOU, C., SINHA, P., et al., “Pedestrian Detection Using Wavelet Templates”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, p. 193, San Juan, Puerto Rico, June 1997.
- [129] SARKAR, P., “Image Classification: Classifying Distributions of Visual Features”. In: *Proceedings of the International Conference on Pattern Recognition*, pp. 472–475, Hong Kong, China, August 2006.
- [130] ZORMAN, M., PODGORELEC, V., KOKOL, P., et al., “Decision Tree’s Induction Strategies Evaluated on a Hard Real World Problem”. In: *Proceedings of the Symposium on Computer-Based Medical Systems*, pp. 19–24, Houston, Texas, USA, June 2000.
- [131] BENBRAHIM, H.; BENSALD, A., “A Comparative Study of Pruned Decision Trees and Fuzzy Decision Trees”. In: *Proceeding of the International Conference on Fuzzy Information Processing Society*, pp. 227–231, 2000.
- [132] MAHALANOBIS, A., VIJAYA KUMAR, B. V. K., SONG, S., et al., “Unconstrained Correlation Filters”, *Applied Optics*, v. 33, pp. 3751–3759, 1994.
- [133] EFRON, B., TIBSHIRANI, R., *An Introduction to the Bootstrap*. 1st ed. CRC Press, 1993.
- [134] KOHAVI, R., “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1137–1143, Montréal, Canada, August 1996.

- [135] STONE, M., “Cross-validatory Choice and Assessment of Statistical Predictions”, *Journal of the Royal Statistical Society B (Methodological)*, v. 36, pp. 111–147, 1974.
- [136] EVERINGHAM, M., SIVIC, J., ZISSERMAN, A., “Hello! My name is... Buffy – Automatic Naming of Characters in TV Video”. In: *Proceedings of the British Machine Vision Conference*, 2006.

Apêndice A

Imagens bases de dados *BioID* e *Feret*

Neste apêndice, apresentamos exemplos de imagens para a base de dados *BioID* nas Figuras A.1(a), A.1(b), A.1(c) e A.1(d) e *Feret* nas Figuras A.2(a), A.2(b), A.2(c) e A.2(d).



Figura A.1: Quatro exemplos de imagens da base de dados *BioID*.



(a)



(b)



(c)



(d)

Figura A.2: Quatro exemplos de imagens da base de dados *Feret*.

Apêndice B

Artigos Desenvolvidos nesta Tese

Neste apêndice, apresentamos os artigos desenvolvidos nesta Tese.

- [1] JÚNIOR, W. S. S., ARAUJO, G. M., SILVA, E. A. B., et al., Facial Fiducial Points Detection using Discriminative Filtering on Principal Components. In: Proceedings of the International Conference on Image Processing, Hong Kong, China, September 2010.
- [2] ARAUJO, G. M., JÚNIOR, W. S. S., SILVA, E. A. B., et al., Facial Landmarks Detection Based on Correlation Filters. In: Proceedings of the IEEE International Telecommunication Symposium, Manaus, Brazil, October 2010.