

TRABALHO DE GRADUAÇÃO

**Reconhecimento Facial Biométrico
Em Nuvens de Pontos Tridimensionais**

Luís Felipe de Melo Nunes

Brasília, 12 de Dezembro de 2016



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

**Reconhecimento Facial Biométrico
Em Nuvens de Pontos Tridimensionais**

Luís Felipe de Melo Nunes

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Flávio de Barros Vidal, CIC/UnB

Orientador

Prof. Carla M. C. C. Koike, CIC/UnB

Examinador interno

Prof. Cauê Zaghetto, CIC/UnB

Examinador interno

Brasília, 12 de Dezembro de 2016

FICHA CATALOGRÁFICA

LUÍS FELIPE, DE MELO NUNES

Reconhecimento Facial Biométrico Em Nuvens de Pontos Tridimensionais.

[Distrito Federal] 2016.

x, 111p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2016). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Reconhecimento Facial

2. Nuvens de Pontos 3D

3. Visão Computacional

4. Biometria

I. Mecatrônica/FT/UnB

REFERÊNCIA BIBLIOGRÁFICA

NUNES, L.F.M., (2016). Reconhecimento Facial Biométrico em Nuvens de Pontos Tridimensionais. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 36/2016, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 111p.

CESSÃO DE DIREITOS

AUTOR: Luís Felipe de Melo Nunes

TÍTULO DO TRABALHO DE GRADUAÇÃO: Reconhecimento Facial Biométrico em Nuvens de Pontos Tridimensionais.

GRAU: Engenheiro

ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Luís Felipe de Melo Nunes

Setor Noroeste, SQNW 309 Bloco D/E Apartamento 202

70687-120 Brasília – DF – Brasil.

Dedicatória

Dedico este trabalho a minha família, em especial meus pais por todo o apoio a mim dado, a minha namorada e toda ajuda oferecida por ela, ao meu orientador pela oportunidade e pelo conhecimento, e para todos aqueles próximos a mim.

Luís Felipe de Melo Nunes

Agradecimentos

Agradeço primeiramente a Deus por todas as forças concedidas e por guiar meu caminho.

A minha família: meu pai Edilton por todos ensinamentos, apoio e amor, minha mãe Karla por todo amor, carinho e ensinamentos, aos meus irmão e demais familiares.

Agradeço a Luana por toda ajuda, companheirismo, preocupações e amor, principalmente em momentos de dificuldade e por sempre estar ao meu lado.

Agradeço ao Prof. Flávio de Barros Vidal por me orientar neste projeto, mostrando determinação e preocupação com o andamento do mesmo, e instruindo de uma maneira agradável e amistosa.

Agradeço a todos os meus amigos pelos incentivos, ajudas e forças.

Agradeço aos meus amigos da Faculdade de Tecnologia que trilharam/trilham o mesmo caminho de formação profissional. Agradeço a todos os Professores que participaram do meu processo educacional.

Luís Felipe de Melo Nunes

RESUMO

Recentemente, diversos processos de automação fazem uso de conhecimentos relacionados a visão computacional, utilizando-se das informações digitalizadas que auxiliam na tomada de decisões destes processos. O estudo de informações 3D é um assunto que vem sendo recorrente em comunidades de visão computacional e atividades gráficas. Uma gama de métodos vem sendo propostos visando obter melhores resultados de performance, em termos de acurácia e robustez. Neste trabalho realiza-se um processo de reconhecimento facial de posição frontal em uma base de dados contendo 31 sujeitos, em que cada sujeito apresenta 3 imagens de profundidade e 3 imagens de cor (RGB). As imagens de cor são utilizadas para detecção facial por uso de um Haar Cascade, que permite a extração dos pontos da face da imagem de profundidade formando uma nuvem de pontos tridimensional. Da nuvem de pontos foram extraídas a intensidade normal e a intensidade do índice de curvatura de cada ponto permitindo a formação de uma imagem bidimensional, intitulada de mapa de curvatura, a partir da qual extrai-se histogramas utilizados no processo de reconhecimento facial. A métrica utilizada para validar o desempenho do método trata-se da medida de F-Measure.

Palavras Chave: Reconhecimento Facial, Nuvens de Pontos Tridimensional, Haar Cascade, Normal e Curvatura, Mapa de Curvatura, F-Measure.

ABSTRACT

Recently, many automation process make use of knowledge related to computer vision, exploiting digital information in form of images or data that assists the decision-making of these process. 3D data recognition is a trend topic in computer vision and graphics tasks community. A large scale of methods had been proposed for 3D applications, expecting a better performance in accuracy and robustness. In this paper a frontal face recognition process was accomplished in a 31 subject database, which presented 3 colorful images (RGB) and 3 depth images for each subject. The colorful images are utilized for face detection by a Haar Cascade algorithm, allowing the extraction of facial points in the depth image and the generation of a tridimensional point cloud. The point cloud is used to extract the normal intensity and the curvature index intensity of each point allowing the confection of a bidimensional image, entitled curvature map, of which histograms are extracted to perform the facial recognition task. The validation of the performance was fulfilled by the application of a F-Measure.

Keywords: Facial Recognition, Tridimensional Point Clouds, Haar Cascade, Normal and Curvature, Map Curvature, F-Measure.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	OBJETIVOS GERAIS	3
1.2.1	OBJETIVOS ESPECÍFICOS	3
1.3	DIVISÃO DO TRABALHO	3
2	REVISÃO BIBLIOGRÁFICA	4
2.1	NUVENS DE PONTOS	4
2.2	LOOKUP TABLE	6
2.2.1	VANTAGENS E DESVANTAGENS	8
2.3	POINT CLOUD LIBRARY (PCL)	9
2.3.1	DIVISÃO DA PCL	10
2.3.2	VISUALIZAÇÃO	10
2.4	OPENCV	12
2.4.1	VISÃO COMPUTACIONAL	12
2.4.2	ORIGEM DA OPENCV	13
2.4.3	OBJETIVOS DA OPENCV	13
2.4.4	ESTRUTURA DA OPENCV	14
2.5	ALGORITMO HAARCASCADE	15
2.5.1	IMAGENS INTEGRAIS	15
2.5.2	FUNÇÕES DE CLASSIFICAÇÃO DE APRENDIZAGEM	16
2.5.3	CASCATA DE CLASSIFICADORES	17
2.6	NORMAL E CURVATURA DE SUPERFÍCIES	18
2.6.1	NORMAL	18
2.6.2	CURVATURAS	19
3	RECONHECIMENTO FACIAL	21
3.1	O USO DO RECONHECIMENTO FACIAL	21
3.2	APLICAÇÕES	22
3.3	DIFICULDADES GERAIS	23
3.4	MÉTODOS DE RECONHECIMENTO FACIAL	24
3.4.1	RECONHECIMENTO FACIAL DE IMAGENS DE INTENSIDADE	24
3.4.2	BASEADO EM FEATURES	24

3.4.3	HOLÍSTICA	27
3.5	RECONHECIMENTO FACIAL POR OUTRAS ENTRADAS SENSORIAIS	34
3.5.1	MÉTODOS DE MODELO 3D	34
3.5.2	INFRASVERMELHO	36
4	METODOLOGIA	39
4.1	PREPARAÇÃO DA BASE DE DADOS.....	41
4.1.1	BASE DE DADOS	42
4.1.2	SEGMENTAÇÃO FACIAL E CONFECÇÃO DAS NUVENS DE PONTOS	44
4.1.3	MATRIZ DE COVARIÂNCIA DA VIZINHANÇA DOS PONTOS.....	46
4.1.4	ESTIMAÇÃO DE NORMAIS E ÍNDICES DE CURVATURA.....	48
4.1.5	EXTRAÇÃO DOS MAPAS DE CURVATURA	49
4.2	RECONHECIMENTO FACIAL	52
4.2.1	DIVISÃO DO MAPA DE CURVATURA	52
4.2.2	CONSTRUÇÃO DOS HISTOGRAMAS	56
4.2.3	COMPARAÇÃO ENTRE HISTOGRAMAS E SCORES	56
4.2.4	CRITÉRIOS DE REJEIÇÃO E ACEITAÇÃO	57
4.3	VALIDAÇÃO DO ALGORITMO	58
4.3.1	F-MEASURE.....	59
5	RESULTADOS.....	63
5.1	MATERIAIS UTILIZADOS	63
5.2	CENÁRIOS DE TESTE.....	64
5.2.1	CENÁRIO 1: QUALIDADE EM FUNÇÃO DO RAIO.....	64
5.2.2	CENÁRIO 2: QUALIDADE EM FUNÇÃO DO NÚMERO DE BINS.....	70
5.2.3	CENÁRIO 3: QUALIDADE EM FUNÇÃO DA DIVISÃO DE REGIÕES	75
5.2.4	LIMIAR.....	82
5.3	ANÁLISE ESPACIAL DE MEMÓRIA	83
5.3.1	IMAGENS.....	83
5.3.2	NUVENS DE PONTO	83
5.3.3	MATRIZ DE COVARIÂNCIA	83
5.3.4	MAPAS DE CURVATURA	84
5.3.5	HISTOGRAMAS.....	84
5.3.6	ESPAÇO TOTAL DO MÉTODO	84
5.4	DIFICULDADES EVIDENCIADAS.....	84
6	CONCLUSÕES	86
6.1	PERSPECTIVAS FUTURAS	86
	REFERÊNCIAS BIBLIOGRÁFICAS	88

LISTA DE FIGURAS

1.1	Núvens de pontos 3D de Faces Humanas.	2
2.1	Nuvem de pontos.	5
2.2	LUT que correlaciona cores e intensidade normal [1].	7
2.3	3DLUT para cores RGB.	8
2.4	Logo do projeto Point Cloud Library (PCL) [2].	9
2.5	Um manipulador geomérico aplicado: o espaço cartesiano 3D representa os dados X,Y,Z e as flechas representam as normais estimadas para cada ponto da nuvem. [2]	11
2.6	Estrutura Básica do OpenCV [3].	14
2.7	Feature Retângular. Da soma dos pixeis dentro dos retângulos brancos são subtraídos da soma dos pixeis dentro dos retângulos cinzas [4].	16
2.8	Primeiro e Segundo <i>Feature</i> selecionados [4].	17
2.9	Esquemático ilustrativo do sistema de cascata dos classificadores [4].	18
2.10	Normal de uma superfície em um ponto p (Adaptado de [5]).	19
3.1	Features Geométricos utilizados no experimentos de reconhecimento facial [6].	25
3.2	Grades para reconhecimento facial [7].	27
3.3	Treinamento de detecção de features por template e resultado da detecção [8].	29
3.4	Resultado entre autofaces e autofeatures [8].	29
3.5	Variação da aparência da face para diferentes ângulos de iluminação [9].	30
3.6	Amostras retiradas da base em [10].	37
3.7	Primeiras cinco autofaces visíveis [10].	37
3.8	Primeiras cinco LWIR autofaces [10].	38
4.1	Fluxograma da metodologia desenvolvida. Os procedimentos azuis correspondem aos processos de entrada de dados e formação das nuvens de pontos. Os verdes às atividades de extração de informações da nuvem de pontos, os vermelhos à extração de histogramas dos mapas de curvatura, enquanto que os roxos consistem nos processos utilizados para validação.	40
4.2	Região do fluxograma referente a preparação da base de dados segmentada por linha tracejada vermelha.	41
4.4	Processo de entrada das informações da base de dados no fluxograma.	42
4.3	Nuvem de Pontos (à direita) e Imagem de Cores (RGB - à esquerda) correspondente.	43
4.5	Processos de detecção e segmentação no fluxograma de preparação da base de dados.	44

4.6	Segmentação Facial por Haar Cascade.	45
4.7	Nuvem de pontos filtrada antes do Haar Cascade em múltiplas vistas.....	45
4.8	Nuvem de Pontos da região facial em múltiplas vistas. Essa é a nuvem de pontos resultante após o processo de segmentação da Figura 4.7.	46
4.9	Vizinhança ao redor de um ponto.	46
4.10	Processo de formação da matriz de covariância no fluxograma de preparação da base de dados.	47
4.11	Processo de Estimação de Normais e Curvaturas no fluxograma.	48
4.13	Variação da curvatura de acordo com o raio.	49
4.12	Normais e Curvatura.	50
4.14	Processos referentes a extração do mapa de curvatura no fluxograma de preparação da base de dados.	51
4.15	Transposição dos índices de curvatura em Imagem de intensidade.	52
4.16	Região do fluxograma referente ao reconhecimento facial segmentado por linha tracejada vermelha.	53
4.17	Reconhecimento facial por comparação.	54
4.18	Divisão de mapa de curvatura em três regiões.	54
4.19	Divisão de mapa de curvatura em seis regiões.	55
4.20	Divisão de mapa de curvatura em nove regiões.	55
4.21	Processo de divisão do mapa de curvatura e construção dos histogramas segmentado no fluxograma referente ao reconhecimento facial.	56
4.22	Conversão para níveis de cinza.	57
4.24	Processo de reconhecimento facial (função de minimização e avaliação por limiar) no fluxograma.	58
4.25	Região de validação segmentada no fluxograma por linha tracejada vermelha.	59
4.23	Histogramas dos Mapas de Curvatura.	62
5.1	Critérios de F-Measure para o Cenário 1.	65
5.2	Precision para o Cenário 1.	66
5.3	Recall para o Cenário 1.	67
5.4	F-Measure para o Cenário 1.	68
5.5	Tempo Total de preparação dos Mapas de Curvatura.	69
5.6	Tempo Médio de preparação de um Mapa de Curvatura.	70
5.7	Critérios de F-Measure para o Cenário 2.	71
5.8	Precision para o Cenário 2.	72
5.9	Recall para o Cenário 2.	73
5.10	F-Measure para o Cenário 2.	74
5.11	Comparativo entre os resultados de F-Measure par 8 e 16 bins.	75
5.12	Critérios de F-Measure para $n = 3$	76
5.13	Precision para $n = 3$	77
5.14	Recall para $n = 3$	78
5.15	F-Measure para $n = 3$	78

5.16	Cr�terios de F-Measure para $n = 6$	79
5.17	Precision para $n = 6$	80
5.18	Recall para $n = 6$	80
5.19	F-Measure para $n = 6$	81
5.20	Comparativo dos <i>F-Measure</i> para diferentes n�meros de divis�es.....	82
5.21	M�ltiplas faces encontradas em uma imagem.	85

LISTA DE TABELAS

5.1	Tabela de Critérios de F-Measure do Cenário 1	64
5.2	Tabela de valores de Precision do Cenário 1.....	66
5.3	Tabela de Recall para o Cenário 1	66
5.4	Tabela de valores de F-Measure para o Cenário 1.....	68
5.5	Tempo Médio de extração de um mapa de curvatura e Tempo total de extração dos mapas de curvatura da Base	68
5.6	Tabela de Critérios de F-Measure para o Cenário 2	71
5.7	Tabela de Precision para o Cenário 2	71
5.8	Tabela de Recall para o Cenário 2	73
5.9	Tabela de F-Measure do Cenário 2.....	73
5.10	Tabela de Critérios de F-Measure para $n = 3$	76
5.11	Tabela de valores de Precision para $n = 3$	76
5.12	Tabela de valores de Recall para $n = 3$	77
5.13	Tabela de valores de F-Measure para $n = 3$	77
5.14	Tabela de Critérios de F-Measure para $n = 6$	79
5.15	Tabela de valores de Precision para $n = 6$	79
5.16	Tabela de valores e Recall para $n = 6$	81
5.17	Tabela de valores de F-Measure para $n = 6$	81

LISTA DE SÍMBOLOS

Símbolos Latinos

v_j	Autovetores da Matriz de Covariância
x	Número de pontos da vizinhança

Símbolos Gregos

σ	Índice de curvatura
λ_j	Autovalores da Matriz de Covariância

Grupos Adimensionais

n	Número de regiões da face
$bins$	Número de $bins$ dos Histogramas
limiar	Limiar de $score$
P	Índice identificador de pessoa
N	Número da imagem
X, Y, Z	Coordenadas do sistema Cartesiano
M	Matriz de Covariância
$score$	Medida de minimização de histogramas
H_e	Histogramas do sujeito de entrada
$H_{(i,j)}$	Histogramas da imagem j do sujeito i
i, j, k, m	Contadores

Subscritos

e	Sujeito de entrada
i	Ponto i da Vizinhança
j	Índice do Autovalor/Autovetor
(i, j)	Sujeito i e Imagem j

Sobrescritos

$-$	Centroíde
\rightarrow	Vetor
T	Transposto

Siglas

PCL	<i>Point Cloud Library</i>
LUT	<i>Lookup Table</i>
VTK	<i>Visualization Toolkit</i>
CAD	<i>Computer Aided Design</i>
NURBS	<i>Non Uniform Rational Basis Spline</i>
FPGA	<i>Field Programmable Gate Array</i>
LIDAR	<i>Light Detection And Ranging</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
ROS	<i>Robot Operating System</i>
PFH	<i>Point Feature Histograms</i>
FPFH	<i>Fast Point Feature Histograms</i>
NARF	<i>The Normal Aligned Radial Feature</i>
RIFT	<i>Rotation Invariant Feature Transform</i>
RSD	<i>Radius-based Surface Descriptors</i>
VFH	<i>Viewpoint Feature Histogram</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
PCD	<i>Point Cloud Data</i>
ICP	<i>Iterative Closest Point</i>
ML	<i>Machine Learning</i>
EBGM	<i>Elastic Bunch Graph Matching</i>
FBG	<i>Face Bunch Graph</i>
FERET	<i>Face Recognition Technology</i>
HOG	<i>Histogram of Oriented Gradients</i>
IA	<i>Inteligência Artificial</i>
PCA	<i>Principal Components Analysis</i>
LDA	<i>Linear Discriminant Analysis</i>
ICA	<i>Independent Component Analysis</i>
FDA	<i>Fisher Discriminant Analysis</i>
RBF	<i>Radial basis function</i>
LBP	<i>Local Binary Pattern</i>
PWC	<i>Pair Wise Coupling</i>
SVM	<i>Support Vector Machines</i>
HMM	<i>Hidden Markov Model</i>
MCS	<i>Multiple Classifier System</i>
LVQ	<i>Learning Vector Quantization</i>
TP	<i>True Positive</i>
FP	<i>False Positive</i>
TN	<i>True Negative</i>
FN	<i>False Negative</i>

Capítulo 1

Introdução

A engenharia de controle e automação é utilizada em diversas áreas de tecnologia, enquadrando-se inclusive na área de tecnologia da informação, visando a automação de sistemas por meio de computadores. Recentemente, diversos processos de automação fazem uso de conhecimentos relacionados a visão computacional, utilizando-se das informações digitalizadas em formas de imagens que auxiliam na tomada de decisões destes processos.

O estudo de informações 3D é um assunto que vem sendo recorrente em comunidades de visão computacional. Uma gama de métodos vem sendo propostos visando obter melhores resultados de performance, em termos de acurácia e robustez.

Um sistema biométrico é essencialmente um sistema de reconhecimento de padrões, que opera por meio da aquisição de dados biométricos de um indivíduo, extraindo um conjunto de características desses dados e comparando-o com um *template* definido em uma base de dados [11]. Este tipo de aplicação encontra-se empregado em diversos sistemas em dias atuais, e são constantemente estudados em nível acadêmico, desenvolvendo-se aprimoramentos de métodos já utilizados e a criação de novos.

Os avanços tecnológicos e os crescentes estudos na área de reconhecimento biométrico geram diversos métodos inovadores e propícios para diferentes aplicações específicas. Neste contexto insere-se o presente trabalho, que apresenta um método baseado em histogramas extraídos de nuvens de pontos 3D (como as nuvens mostradas na Figura 1.1) para realizar o reconhecimento facial.

1.1 Motivação

O avanço constante da tecnologia e o surgimento de novos equipamentos e sensores na área de visão computacional, favorece a exploração de novas análises para a realização de atividades de reconhecimento, extraindo-se novas informações referentes as características biométricas de um sujeito, características essas que podem ser únicas e, portanto, contribuem para a diferenciação entre pessoas.

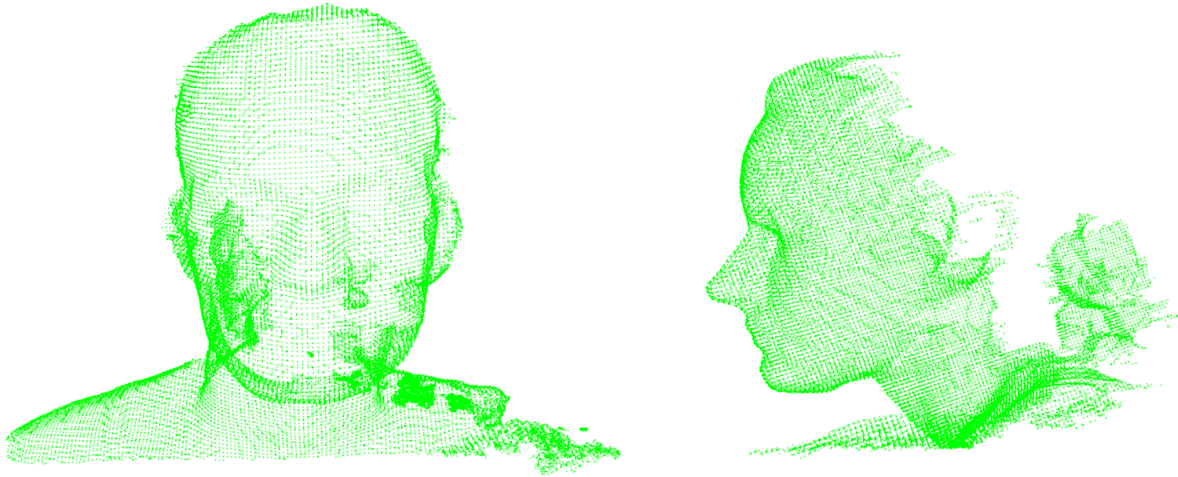


Figura 1.1: Núvens de pontos 3D de Faces Humanas.

Os sistemas de aquisição de dados recentes permitem a captura de informações referentes a descrição do espaço tridimensional (como a profundidade), evidenciado na visão computacional em anos passados, a perda da noção espacial. A noção espacial em visão computacional traz importantes avanços à essa área, como no caso de orientação de robôs, permitindo um melhor tratamento na tomada de decisões de deslocamentos evitando colisões. No quesito biométrico, a informação de profundidade permite a exploração mais aprofundada de características geométricas da face de um humano.

Alguns anos atrás, os sensores e scanners 3D apresentavam um custo proibitivo, sendo assim pouco adquirido em meios acadêmicos, tornando-se o maior empecilho para o desenvolvimento de estudos na área. Entretanto, tecnologias de baixo custo foram lançadas no mercado, permitindo uma maior interação do nível acadêmico com o assunto.

O reconhecimento facial 3D é estudado visando aperfeiçoar métodos de reconhecimento 2D, que já apresentam diversas abordagens com resultados robustos e confiáveis. Por mais robustos que os sistemas de reconhecimento 2D sejam, estes apresentam algumas desvantagens que podem comprometer os seus resultados, tal como a sensibilidade a variações de iluminação e poses [9, 7]. Os sistemas 3D apresentam vantagem sobre esses aspectos negativos dos sistemas 2D, pois a captura de dados independe da iluminação, e quanto a variação de poses, os sistemas 3D podem ser rotacionados, permitindo estabelecer uma posição padrão para comparação.

O método desenvolvido trata-se então do uso de informações tridimensionais para realizar a tarefa de reconhecimento facial. A motivação deste trabalho se resume a possibilidade de desenvolver uma aplicação de reconhecimento facial 3D utilizada para propósitos biométricos.

1.2 Objetivos Gerais

O objetivo do projeto consiste no desenvolvimento de um sistema de reconhecimento facial tridimensional baseado na utilização de nuvens de pontos.

1.2.1 Objetivos específicos

- Realizar o reconhecimento facial utilizando exclusivamente características extraídas de nuvens de pontos tridimensionais.
- Validar o método gerando resultados por meio do uso de medidas de *F-Measure*.
- Implementar estratégias e métodos comparativos que dispensam a necessidade de treinamentos prévios.

1.3 Divisão do Trabalho

O trabalho proposto é dividido nos seguintes capítulos: No Capítulo 2 são apresentados os principais conceitos utilizados para subsidiar o desenvolvimento da Metodologia proposta. O Capítulo 3 aborda em detalhes tópicos de reconhecimento facial, incluindo a sua definição, aplicações, dificuldades, e diversos métodos implementados. No Capítulo 4 é apresentado o método proposto, expondo de forma sequencial os procedimentos realizados. O Capítulo 5 exibe diferentes cenários de resultados, desenvolvidos para avaliar o método proposto em relação à atividade de reconhecimento facial.

Capítulo 2

Revisão Bibliográfica

A implementação do reconhecimento facial biométrico em nuvens de pontos tridimensionais é realizada por intermédio de alguns recursos e conhecimentos, cujo mínimo aprofundamento é importante para um melhor entendimento do processo como um todo. A proposta elaborada para a solução do problema faz uso de conhecimentos básicos de uma nuvem de pontos, do funcionamento de uma *Lookup Table*, dos elementos e estruturas básicas das bibliotecas PCL e OpenCV, do algoritmo de reconhecimento facial Haar Cascade e dos conceitos de aspectos geométricos tais como normal e curvatura de superfícies.

2.1 Nuvens de Pontos

Em computação gráfica, de acordo com [12], as superfícies de um objeto tridimensional são representadas, em geral, por malhas triangulares formadas por uma estimativa matemática conhecida, ou por interpolação, ou ainda por aproximação de um conjunto de pontos não organizados no quesito de reestruturação de uma superfície.

Muitas aplicações em manufatura, medicina, geografia, design, etc, requerem o escaneamento de objetos tridimensionais de certa complexidade, visando incorporá-los a um processamento assistido ou realizado por computador [12]. Então, técnicas de medição foram aperfeiçoadas para facilmente produzir uma enorme quantidade de pontos presentes na superfície de objetos. Esse conjunto de pontos representando um objeto tridimensional, em algum sistema de coordenadas, é chamado de nuvem de pontos (Figura 2.1) [12].

Recentemente, vários métodos foram desenvolvidos para aplicações em nível tridimensional utilizando nuvens de pontos, visando a organização dos dados e uma maior praticidade em formulações e aplicações matemáticas. Algumas dessas aplicações são:

- Reconstrução de superfícies por uso malhas triangulares [12];
- Detecção e Extração de *features* 3D;
- Estimativa de superfícies;

- Estimação de Normais e curvaturas de superfícies;
- Filtragem de dados tridimensionais;
- Criação de imagens de profundidade;
- Reconhecimento de objetos e padrões [13];
- Detecção e Segmentação de objetos [14];
- Métodos de Visualização.

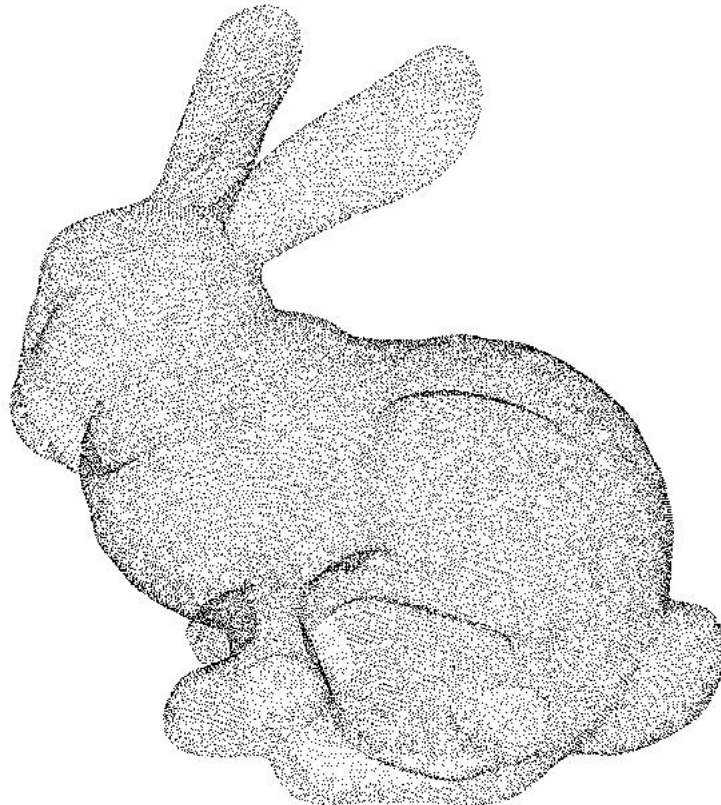


Figura 2.1: Nuvem de pontos.

As nuvens de pontos são representadas por pontos distribuídos em um espaço, normalmente representado por um sistema de coordenadas cartesianas. Em um sistemas de coordenadas tridimensional esses pontos encontram-se usualmente definidos nas coordenadas X , Y e Z de um sistema cartesiano. Os pontos de uma nuvem de pontos podem possuir outras informações além de posição, como valores da componente normal, níveis de intensidade, informações de cor (RGB), valores de profundidade, informações do ponto de vista, intensidade do gradiente, entre outras informações que o construtor da nuvem considere relevante a nível de ponto [2].

Nuvens de pontos são criadas por meio de scanners 3D [15], onde esses aparelhos adquirem vários pontos de uma superfície, presentes em uma grade ou em outros modelos de divisão do espaço no qual os sensores do scanner são projetados para aquisição de pontos, fornecendo como saída a nuvem de pontos dos dados captados em algum arquivo.

Normalmente as nuvens de pontos não são utilizadas diretamente em aplicações tridimensionais. Elas apresentam uma enorme importância quanto a aquisição dos dados, a renderização e inspeção dos mesmos para verificar se a forma representada por seus pontos condiz com a forma esperada, entretanto são frequentemente convertidas para modelos de CAD (*Computer Aided Design* - *softwares* de desenho assistido por computador), polígonos e triângulos, ou superfícies NURBS (*Non Uniform Rational Basis Spline* - modelo matemático utilizado em programas gráficos para gerar e representar curvas e superfícies), por meio de um processo conhecido como reconstrução de superfície.

A próxima seção trata sobre a *Lookup Table*, cuja importância no projeto foi para a elaboração dos mapas de curvatura.

2.2 Lookup Table

As *Lookup Table* [16] surgiram previamente ao computador, sendo utilizadas para agilizar cálculos feitos a mão de funções matemáticas complexas, tais como trigonometria, logaritmo e funções de densidade estatística [17]. Com o surgimento do computador, as operações de entrada e saída eram particularmente lentas, mesmo em comparação com a velocidade dos processadores da época. A redução de operações custosas de leitura tornou-se um método possível graças ao uso de *lookup tables* estáticas (embutidas no programa) ou de arranjos dinâmicos previamente carregados na memória para conter apenas o itens de dados mais recorrentes, criando uma espécie de cache manual.

A *Lookup Table* (LUT) consiste em um arranjo (computacional) que substitui a necessidade de realizar a execução de um procedimento computacional, utilizando uma simples operação de indexação. Essa indexação é utilizada visando a redução do tempo de execução de algum procedimento.

A utilização de LUT, em termos de tempo de processamento, podem ser significantes, levando em consideração que o acesso à uma informação armazenada em memória, normalmente é mais rápido que a realização de operações computacionais. Comumente as tabelas podem ser precalculadas e armazenadas em programas de armazenamento estáticos, calculadas como parte do programa em execução durante uma fase de inicialização, ou até mesmo armazenadas em hardware em plataformas de aplicação específica.

As LUTs são utilizadas extensivamente em algumas linguagens de programação, para validar valores de entrada correlacionando com uma lista de itens válidos (ou inválidos) dentro de um arranjo, e podem incluir funções de ponteiro para processar a correlação de entradas. Outro uso comum de LUT é em FPGAs onde a sua implementação é realizada em hardware e é reconfigurável, sendo utilizada para funcionalidades de hardware programável. As LUTs foram umas das primeiras funcionalidades implementadas em planilhas computacionais.

2.2.0.1 LUT em processamento de imagens

Aplicações de análise de dados [18], tal como processamento de imagens, faz o uso de LUT para transformar os dados de entrada em um formato de saída mais agradável, por exemplo, uma imagem em níveis de cinza de planeta Saturno será transformada em uma imagem colorida para enfatizar as diferenças entre seus anéis.

Um exemplo clássico de redução de execuções computacionais desnecessárias utilizando LUT é a obtenção de resultados de cálculo trigonométrico, como o seno de um valor. O cálculo de funções trigonométricas pode reduzir substancialmente a velocidade de uma aplicação computacional. A mesma aplicação pode terminar muito mais cedo quando ela pre-calcula o seno de alguns valores, por exemplo, a cada x graus (a tabela pode ser definida por variáveis estáticas durante a compilação, reduzindo o custo de tempo de execuções repetidas). Quando o programa requerer o seno de um valor, ele pode utilizar a LUT para obter o valor de seno mais próximo de um endereço de memória, e pode até interpolar o seno do valor desejado, ao contrário de calculá-lo por fórmula matemática. LUT são utilizadas por coprocessadores matemáticos em sistemas computacionais. Um erro em uma LUT foi responsável pelo famoso bug de ponto flutuante da Intel [19].

Funções que utilizam uma única variável (como seno e cosseno) podem ser implementadas com um único arranjo, enquanto funções que envolvem duas ou mais variáveis necessitam de técnicas de indexação de arranjos multidimensionais.

Existem soluções intermediárias que utilizam tabelas em combinação com um pouco de processamento, fazendo o uso frequente de interpolações. O pre-cálculo combinado com interpolações pode produzir resultados de maior acurácia para valores que se encontram entre dois valores pre-calculados. Esta técnica requer um pouco mais de tempo para ser realizada, mas pode aumentar fortemente a acurácia em aplicações que necessitam ser mais acuradas.

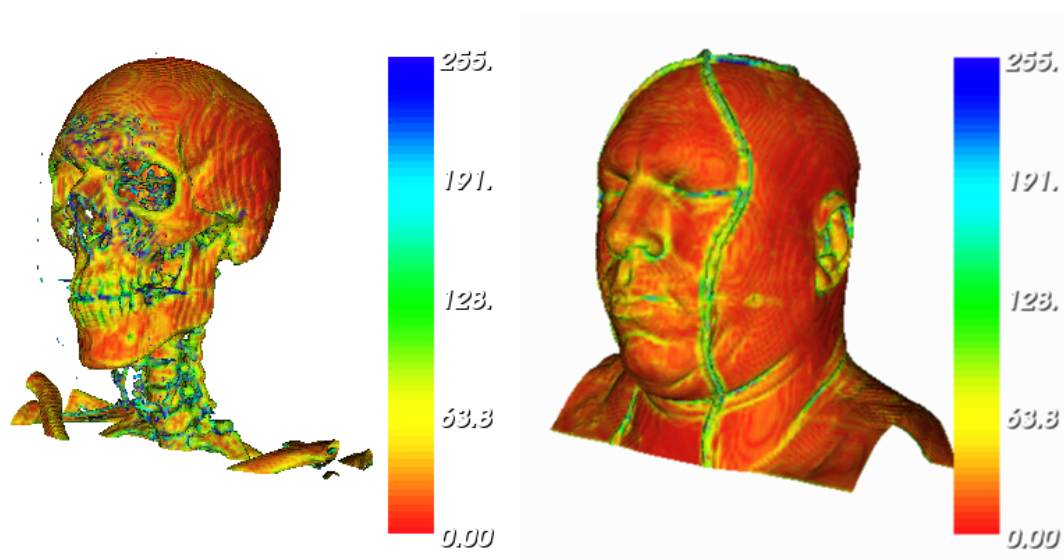


Figura 2.2: LUT que correlaciona cores e intensidade normal [1].

Em processamento de imagens, as *lookup table* são constantemente chamadas de LUTs (ou

3DLUT, Figura 2.3), que retornam um valor de saída para cada intervalo de valores de índices. Uma LUT comum, chamada de *colormap* ou *palette* é utilizada para determinar cores e valores de intensidade nos quais uma imagem será representada [20]. Em tomografia computacional, "*windowing*" se refere a um conceito relacionado para determinar como representar a intensidade de uma radiação medida. Em [18] um método foi desenvolvido para obter meios-tons inversos de uma imagem baseado no uso de LUTs. As *lookup tables* também podem ser utilizadas para relacionar algum valor de intensidade com cores [1] como na Figura 2.2.

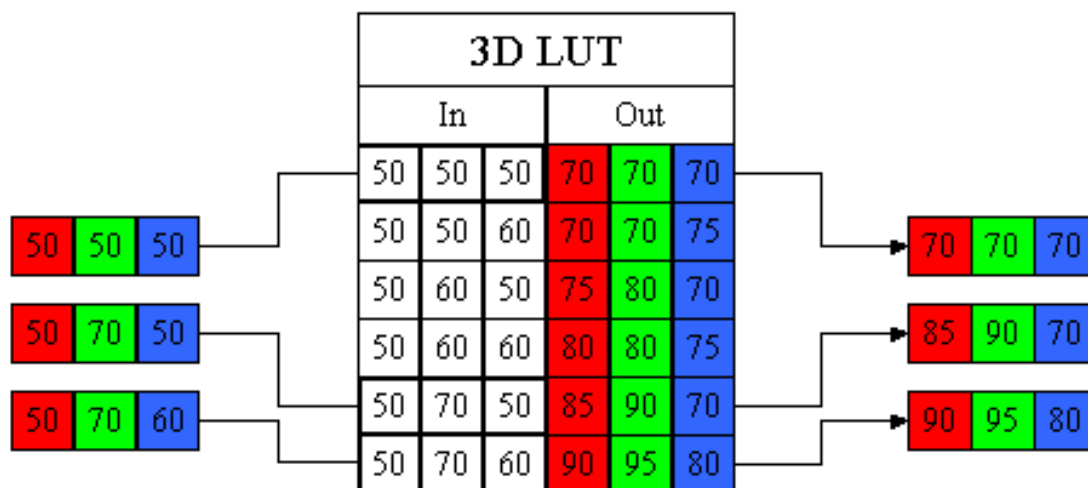


Figura 2.3: 3DLUT para cores RGB.

2.2.1 Vantagens e Desvantagens

As vantagens da LUT já foram previamente mencionadas, sendo a principal delas no quesito de tempo de execução de uma aplicação, evitando a repetição de execuções. Apesar de muitas vezes ser efetivo, a aplicação de uma *lookup table* pode mesmo assim resultar em uma grande penalidade se o cálculo que a LUT substituir for relativamente simples. O tempo de aquisição de dados e complexidade dos requisitos da memória podem aumentar o tempo de operação da aplicação e a complexidade do sistema comparado ao que seria necessário para o cálculo direto por fórmula. A possibilidade de poluir a cache pode também se tornar um problema. Acesso a tabelas muito extensas pode quase sempre causar perda de cache. Esse fenômeno tem se tornado um problema quando processadores extrapolam o espaço de memória. Em alguns ambientes como a linguagem de programação Java, LUTs podem se tornar mais custosas devido as verificações mandatórias de limite, envolvendo comparações adicionais e ramificações para cada verificação.

Existem duas limitações fundamentais que indicam se é possível construir uma *lookup table* para uma determinada operação. A primeira limitação diz respeito a quantidade de memória disponível, dado que uma tabela maior que o espaço disponível não pode ser construída, apesar de ser possível construir tabelas em disco, aumentando o tempo de busca. A outra limitação refere-se

ao tempo necessário para calcular os valores da tabela em primeira instância, apesar de precisar ser realizado apenas uma vez, caso o tempo requerido seja proibitivamente longo, o uso de uma *lookup table* pode ser uma solução inapropriada. No caso da última limitação, as tabelas podem ser definidas estaticamente em muitos casos.

A seção seguinte introduz a biblioteca PCL, utilizada em grande escala no projeto para realização de operações com as nuvens de pontos.

2.3 Point Cloud Library (PCL)

Nos últimos anos, a percepção do mundo, no âmbito computacional, evoluiu de simples sensores de profundidade baseados em sonar ou câmeras infravermelhas (produzindo alguns bytes de informação sobre o mundo) para câmeras ubíquas e scanners a laser. Ultimamente, sensores como o Velodyne *spinning LIDAR* utilizado no DARPA *Urban Challenge* ou o scanner de laser inclinado utilizado no PR2 tem gerado representações tridimensionais de alta qualidade do mundo em nuvens de pontos [2]. Infelizmente esses sistemas possuem alto custo, inviabilizando a sua aquisição por pesquisadores. Entretanto, sensores 3D tem se tornado mais acessíveis, como o sensor Kinect [21] da Microsoft, que provê nuvens de pontos em tempo real assim como imagens 2D. Devido a essas mudanças no cenário dos sensores 3D, espera-se que os computadores e robôs possam interpretar as informações do mundo real, "enxergando-o" em 3D. Todas essas tarefas necessitam de uma ferramenta para manusear estas nuvens de pontos de maneira eficiente, surgindo assim a *Point Cloud Library* (PCL, Figura 2.4).



Figura 2.4: Logo do projeto Point Cloud Library (PCL) [2].

A *Point Cloud Library* (PCL) [2] consiste em uma biblioteca voltada para processamento em nuvens de pontos e geometrias 3D, que apresenta integração com o ROS (*Robot Operating System*) permitindo assim a interação com projetos de robótica. A PCL é voltada para a linguagem de programação C++, sendo completamente baseada em *templates*. A maioria de suas operações matemáticas são implementadas por meio de bibliotecas open-source de álgebra linear. Além desses fatores, a PCL fornece suporte para paralelização de multi-cores.

Em uma perspectiva de algoritmos, a PCL foi desenvolvida visando incorporar uma multidão de algoritmos de processamento tridimensional, operados sobre dados de nuvens de pontos, incluindo: filtragem, estimação de *features*, reconstrução de superfícies, montagem de modelos, segmentação, registro, etc.

A PCL segue uma interface básica para o processamento em *pipeline*, iniciando pela criação de objetos de processamento (e.g. filtros, segmentação, estimadores de *features*), seguido da introdução de nuvens de pontos para o modelo processador da biblioteca, definindo os parâmetros desejados para a operação e finalizando com o computar dos dados para obter a saída.

2.3.1 Divisão da PCL

A PCL visa simplificar o desenvolvimento sendo dividida em uma série de bibliotecas de código menores que podem ser compiladas separadamente. Essas divisões são:

- *libpcl_filters*: implementa filtros de dados tais como *downsampling*, remoção de outlier, extração de índices, projections, etc;
- *libpcl_features*: implementa vários *features* 3D, como normais e curvaturas de superfícies, estimação de pontos de fronteira, momentos invariantes, curvaturas principais, descritores PFH e FPFH, imagens spin, imagens integrais, descritores NARF, RIFT, RSD, VFH, SIFT em dados de intensidade, etc;
- *libpcl_io*: implementa operações I/O como ler e escrever de arquivos PCD (Point Cloud Data);
- *libpcl_segmentation*: implementa extração de *cluster*, montagem de modelo por métodos de consenso de amostra para uma variedade de modelos paramétricos (planos, cilindros, esferas, linhas, etc), extração de prismas poligonais, etc;
- *libpcl_surface*: implementa técnicas de reconstrução de superfície, enredamento, casco convexo, Moving Least Squares, etc;
- *libpcl_registration*: implementa métodos de registro de nuvens de pontos como ICP, etc;
- *libpcl_keypoints*: implementa diferentes métodos de extração de pontos chave, que podem ser utilizados como um passo de pré-processamento para extrair descritores de *features*;
- *libpcl_range_image*: implementa suporte para imagens de profundidade criadas de conjuntos de dados de nuvens de pontos.

As bibliotecas mencionadas acima apresentam testes unitários e de regressão implementados para as classes e métodos, visando garantir a corretude das operações. Além disso, vários tutoriais e exemplos são disponibilizados no site da PCL.

2.3.2 Visualização

A PCL apresenta a sua própria biblioteca de visualização, sendo esta baseada em *Visualization Toolkit* (VTK). VTK oferece diversas ferramentas voltadas para a renderização de elementos tridimensionais, como tensores, textura e métodos volumétricos. Além disso a VTK apresenta um

ótimo suporte multiplataforma, facilitando a sua integração à PCL e favorecendo a renderização de nuvens de pontos e dados de superfícies.

A biblioteca de visualização (*Visualization*) da PCL visa integra-la com a VTK, provendo uma visualização compreensível de estruturas de nuvens de pontos, com o propósito de prototipar e visualizar os resultados de algoritmos operando com dados em seu nível dimensional.

A biblioteca *Visualization* da PCL fornece as seguintes funções:

- Configuração de parâmetros visuais (tamanho e forma dos pontos, cor, opacidade, etc) e métodos de renderização de quaisquer conjuntos de dados de nuvens de pontos n-dimensionais;
- Um módulo de visualização de histogramas (*PCLHistogramVisualizer*) de plots 2D;
- Módulos de visualização de imagens de profundidade;
- Uma gama de manipuladores de cor e geometria, permitindo a especificação de quais dimensões serão utilizadas para o posicionamento dos pontos em um espaço cartesiano 3D (Figura 2.5) ou quais cores devem ser utilizadas para renderizar os pontos;
- Métodos para desenhar formas básicas em 3D na tela (cilindros, esferas, linhas, polígonos, etc) por meio de um conjunto de pontos ou por equações paramétricas;



Figura 2.5: Um manipulador geomérico aplicado: o espaço cartesiano 3D representa os dados X,Y,Z e as flechas representam as normais estimadas para cada ponto da nuvem. [2]

Os interagentes manipuladores são módulos que descrevem como as cores e as geometrias 3D de cada ponto no espaço são computadas e mostradas na tela, e como o usuário interage com os dados. Seu desenvolvimento é feito visando a simplicidade e a facilidade de extensão.

A seção seguinte aborda como a biblioteca OpenCV foi utilizada para a detecção facial por meio do método Haar Cascade.

2.4 OpenCV

A OpenCV [22] é uma biblioteca *open source* de visão computacional, sendo primariamente desenvolvida em C e C++, mas apresentando diversas interfaces para outras linguagens, como Python, Ruby e Matlab[22].

A OpenCV é projetada visando uma melhor eficiência computacional com grande foco em aplicações de tempo real. A OpenCV apresenta suporte para o uso de processadores multi-cores. A proposta da OpenCV é fornecer uma infraestrutura de visão computacional voltada para facilidade e praticidade, auxiliando na implementação rápida de aplicações sofisticadas de visão. A OpenCV apresenta mais de 500 funções implementadas que interagem com diversas outras áreas de visão, como inspeção de produtos de fábrica, calibração de câmeras, interfaces gráficas, segurança, aplicações médicas, visão estereoscópica e robótica, como mencionado em [3].

Normalmente as pessoas apresentam pouco conhecimento sobre a aplicação de visão computacional na resolução de problemas. Aplicações mais comuns são conhecidas (como atividades de vigilância, ou desenvolvimento de vídeos e imagens na internet), entretanto atividades mais rebuscadas (como mapeamento aéreo ou de imagens de ruas) não são associadas a essa área [3].

2.4.1 Visão Computacional

A visão computacional consiste na transformação de dados de câmeras estáticas ou de vídeo em uma decisão ou em uma nova representação. Os dados de entrada podem apresentar informações contextuais, relacionadas ao posicionamento e aquisição das imagens, enquanto uma nova representação está relacionada aos parâmetros de visualização dos dados, como por exemplo a transformação de uma imagem colorida em uma imagem de níveis de cinza.

O ser humano é muito visual, e isso leva a conclusão errônea de que tarefas de visão computacional são simples. O cérebro humano apresenta diversos sistemas responsáveis por reagir ou tomar decisões específicas, dividindo o sinal de visão em múltiplos canais que espalham diferentes tipos de informação ao longo de si mesmo. O cérebro recebe constantes loops de informação que controlam diversos elementos do corpo humano, até mesmo os sensores que captam as informações visuais (os olhos), controlando a quantidade de luz e informação a ser recebida por meio da íris e da retina. As maneiras como o cérebro interpreta as informações visuais e quais ou quantas informações são interpretadas para tomadas de certas decisões ainda são pouco conhecidas [3].

Sistemas de visão de máquina, normalmente apresentam uma máquina que recebe uma grade de números de uma câmera ou disco. Em grande parte, esses sistemas não apresentam nenhuma implementação de reconhecimento de padrões, controle de foco ou associações cruzadas embutidas. Na maioria dos casos os sistemas de visão são considerados como simples, pois a única informação "visível" consiste nas grades de números que representam uma mínima informação visual para o ser humano, e que ainda sofre interferência de ruídos e distorções durante a captura das imagens. O problema de fato é mais complicado devido a diversos fatores, como a percepção do ser humano que é baseada em um espaço tridimensional, enquanto câmeras projetam informações desse espaço

de uma maneira bidimensional, excluindo assim a percepção de profundidade e a noção espacial (apesar de equipamentos recentes contornarem levemente essa situação). Dadas as dificuldades da visão computacional, as aplicações e ações baseadas em dados obtidos de câmera são normalmente executadas em contextos com uma tarefa ou propósito específico.

A OpenCV visa prover as ferramentas básicas necessárias para encontrar a solução de um problema de visão computacional, e mesmo que as suas funcionalidades não sejam suficientes para casos complexos, os seus componentes básicos são suficientemente completos para auxiliar no desenvolvimento de uma solução concreta e completa. Algumas soluções de problemas computacionais conhecidos já se encontram implementadas dentro da biblioteca, permitindo a solução direta de alguns problemas sem exigir uma manipulação de mais baixo nível partindo de operações com imagem.

2.4.2 Origem da OpenCV

A OpenCV nasceu de uma iniciativa da Intel Research para aprimorar aplicações de processamento intensivo. Um dos autores da iniciativa percebeu - por meio de visitas a universidades - que infraestruturas de visão computacional internas a essas instituições eram criadas devido a ausência de uma composição básica. As estruturas produzidas eram passadas de aluno para aluno, onde estes elaboravam aplicações em cima das outras aplicações realizadas por alunos precedentes. Portanto foi concretizada a ideia da biblioteca, visando tornar a infraestrutura de visão computacional uniforme e disponível universalmente.

A OpenCV teve origem então em um laboratório de pesquisas da Intel em colaboração com o grupo *Software Performance Libraries* junto com a implementação e otimização de especialistas russos.

2.4.3 Objetivos da OpenCV

A OpenCV apresenta diversos objetivos, entre os quais se destacam os seguintes:

- Promover pesquisas em visão provendo código aberto e otimizado para infraestruturas básicas de visão;
- Disseminar conhecimentos de visão computacional provendo uma infraestrutura comum que desenvolvedores possam utilizar;
- Promover aplicações comerciais baseadas em visão com uma licença que não apresenta a necessidade da aplicação ser aberta ou gratuita;

Além destes fatores, o crescimento de aplicações de visão computacional aumentam a necessidade de processadores mais rápidos, promovendo o aumento de pesquisa e comércio de processadores.

2.4.4 Estrutura da OpenCV

A OpenCV é estruturada em cinco componentes principais (CXCORE,CV,ML,HighGUI,CvAux) sendo quatro desses mostrados na Figura 2.6. A documentação do OpenCV indica que a biblioteca é dividida em segmentos responsáveis por diversas áreas da visão computacional. Os principais segmentos mencionados são:

- *CXCORE*: Contém as estruturas de dados, álgebra matricial, persistência de objetos, gerenciamento de memória, tratamento de erros, e carregamento dinâmico do código, assim como visualizador, texto e matemática básica.
- *CV*: Contém o processamento de imagem, a análise de estruturas de imagem, rastreamento de movimento, reconhecimento de padrões, e calibração de câmera.
- *Machine Learning (ML)*: Contém funções de agrupamento, classificação e análise de dados.
- *HighGUI*: Contém a interface de usuário (GUI) e armazenamento, além de chamada de vídeos e imagens.
- *CVCAM*: Corresponde a interface de câmera.
- *CVAux*: Contém áreas extintas (como reconhecimento facial por HMM) e algoritmos experimentais, como segmentação de fundo e frente, reconhecimento de gestos por visão estereográfica, correção de formas por regiões de contorno, descritores de textura, rastreamento de boca e olhos, entre outros.

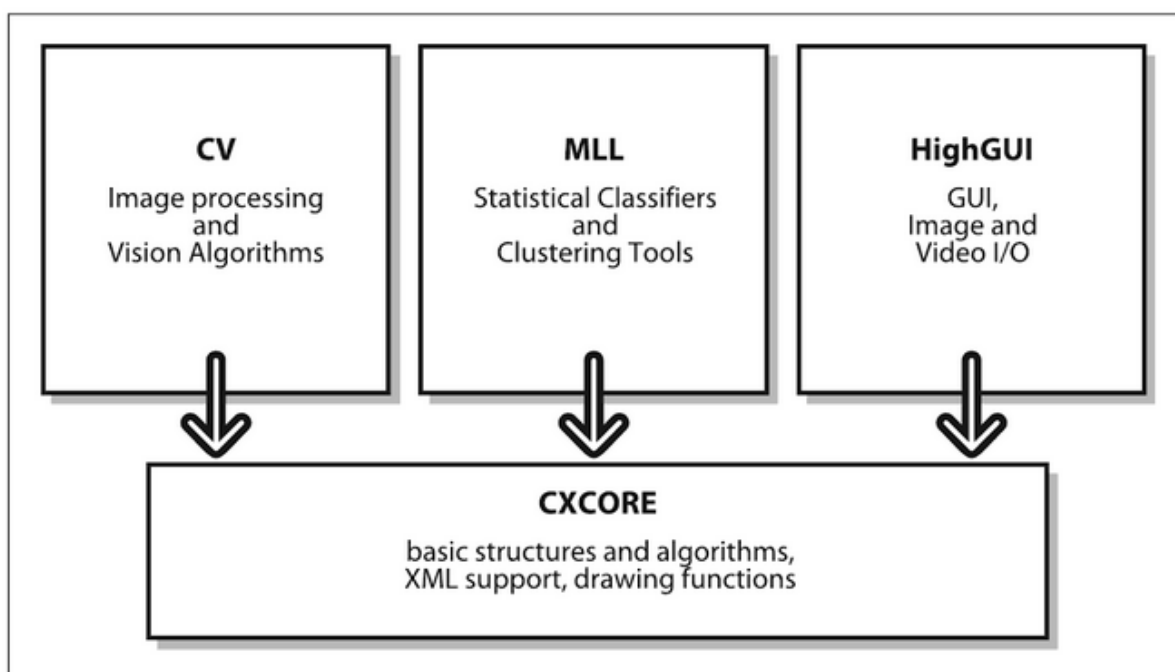


Figura 2.6: Estrutura Básica do OpenCV [3].

A seção a seguir explica de forma sucinta o algoritmo Haar Cascade, utilizado para a detecção da região facial.

2.5 Algoritmo HaarCascade

A detecção de objetos utilizando classificadores em cascata de *features* Haar é um método eficiente de detecção proposto por Paul Viola e Michael Jones [4] baseado em *machine learning*, onde uma função de cascata é treinada por várias imagens negativas e positivas. Posteriormente é utilizado para a detecção de objetos em outras imagens.

Os chamados *Haar-like features* são descritores utilizados em imagens digitais para a detecção de objetos, e foram utilizados no primeiro detector facial em tempo real [23]. Esses descritores foram nomeados assim por sua similaridade intuitiva com as Haar wavelets.

A detecção desenvolvida por Paul Viola e Michael Jones é comumente conhecida como Haar Cascade, e trata-se de um método rápido de detecção que faz uso de imagens integrais, que permitem a avaliação de *features* de maneira mais veloz, além de avaliar de modo invariável a escala. Outro componente de extrema importância para o algoritmo são os métodos de construção de classificadores, que auxiliam na seleção dos *features* mais relevantes, dado que o número de *Haar-like features* em qualquer sub-janela de uma imagem é significativamente grande, muito maior até que o número de pixels. O algoritmo recebe uma terceira contribuição correspondente a um método para combinação sucessiva de classificadores mais complexos em uma estrutura de cascata, que aumenta drasticamente a velocidade da detecção por focar a atenção em regiões promissoras da imagem, reservando processamentos mais complexos apenas para estas regiões.

O algoritmo utiliza classificadores baseado em valores de *features* simples, devido ao fato da sua operação ser muito mais rápida que o caso de uso de pixels. Os *features* simples utilizados são remissivos de funções básicas de Haar, mais especificamente, são utilizados três tipos de *features*. O valor de um *feature* de dois retângulos é a diferença entre as somas dos pixels localizados dentro das regiões retangulares, que apresentam o mesmo tamanho e forma e são adjacentes verticalmente e horizontalmente, como na Figura 2.7. Um *feature* de três retângulos computa a soma de dois retângulos externos subtraída da soma de um retângulo central. Finalmente o *feature* de quatro retângulos computa a diferença entre pares diagonais de retângulo.

2.5.1 Imagens Integrais

Features retangulares podem ser calculados rapidamente fazendo-se uso de uma representação para a imagem chamada de imagem integral. A imagem integral em uma localização (x,y) contém a soma de todos os pixels acima e à esquerda do ponto (x,y) . Utilizando-se a imagem integral, cada soma de retângulos pode ser computada por quatro localizações de referência, portanto a diferença entre o somatório de um *feature* de dois retângulos corresponde a soma de oito referências. Entretanto, dado que os *features* de dois retângulos definidos previamente envolvem a soma de retângulos adjacentes, sua soma pode ser realizada com seis pontos de referência, oito para o caso

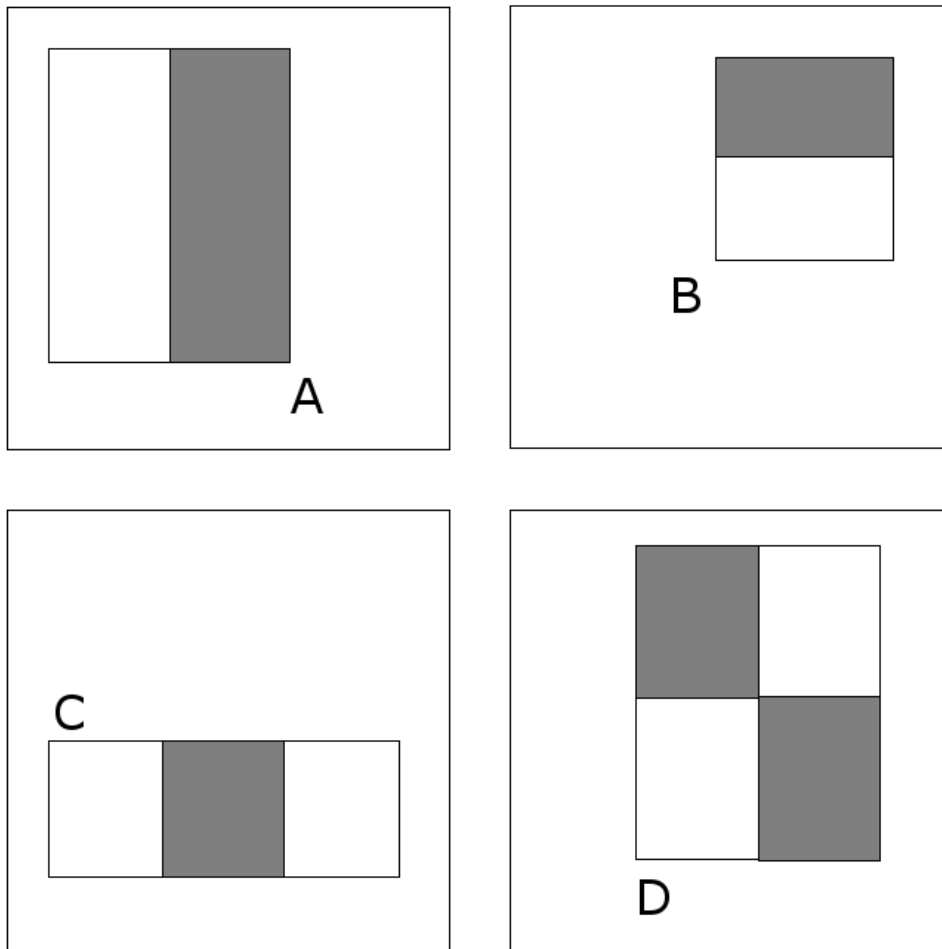


Figura 2.7: Feature Retângular. Da soma dos pixels dentro dos retângulos brancos são subtraídos da soma dos pixels dentro dos retângulos cinzas [4].

de *features* de três retângulos e nove para o caso de *features* de quatro.

2.5.2 Funções de Classificação de Aprendizagem

O algoritmo faz uso de ML para ter uma base de classificação. Dado um conjunto de *features* e um conjunto de treino de imagens positivas e negativas, qualquer número de abordagens de aprendizado de máquina pode ser utilizado para se determinar e aprender uma função de classificação. Um variante do *AdaBoost* é aplicado no algoritmo tanto para a seleção de um pequeno conjunto de *features* como para o treinamento dos classificadores. A seleção de um conjunto de *features* se da em razão de o custo de computar todo o conjunto continuar sendo proibitivo (mesmo que cada *feature* seja computado de maneira muito eficiente).

Experimentos iniciais realizados mostram que um classificador de faces frontais constituído de 200 *features* apresenta uma taxa de detecção de 95% onde a taxa de falsos positivos é de 1 a cada 14084 [4].

Em reconhecimento facial, o primeiro *feature* selecionado pelo *AdaBoost* aparenta focar na propriedade do rosto na qual a região dos olhos é usualmente mais escura do que a região do nariz e das bochechas (Figura 2.8). O segundo *feature* selecionado se da pela propriedade que os olhos são mais escuros que a ponte do nariz.

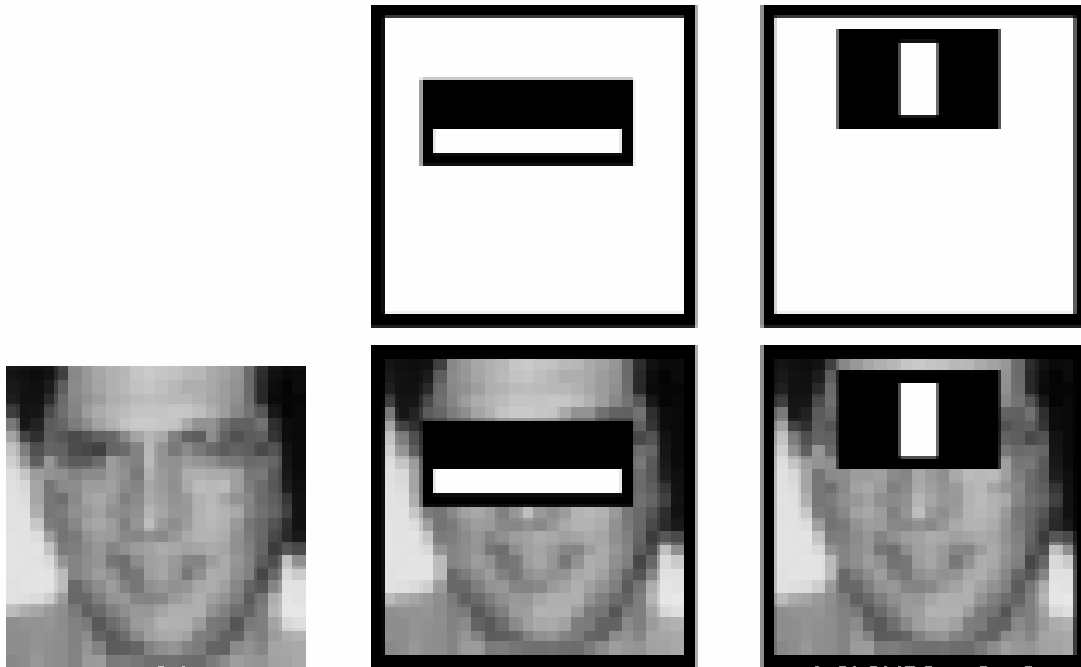


Figura 2.8: Primeiro e Segundo *Feature* selecionados [4].

2.5.3 Cascata de Classificadores

A forma geral do processo de detecção é referente a uma árvore degenerativa de decisão, chamada do termo "cascata" (*cascade*, Figura 2.9). O resultado positivo do primeiro classificador aciona a avaliação de um segundo classificador, que é ajustado para atingir altas taxas de detecção. O resultado positivo do segundo classificador aciona a avaliação de um terceiro, e assim por diante. Uma saída negativa em qualquer ponto leva à rejeição imediata daquela sub imagem.

O processo de treinamento da cascata envolve dois tipos de troca. Normalmente os classificadores que abrangem um número maior de *features* tendem a obter taxas maiores de detecção e taxas menores de falsos positivos, apesar que com um número maior de *features* o custo computacional também vem a ser elevado. Portanto, os principais fatores a serem analisados são os casos de decisão do número de estágios de classificadores, definindo o tamanho da cascata, o número de *features* avaliados e o limiar a ser definido para cada estágio. Logo, há um grande problema em encontrar o número ótimo de todos os elementos descritos que minimizem o número esperado de *features* a serem avaliados e maximizem a eficiência [4].

Na prática, utilizando um *framework* simples foi possível produzir um classificador efetivo e

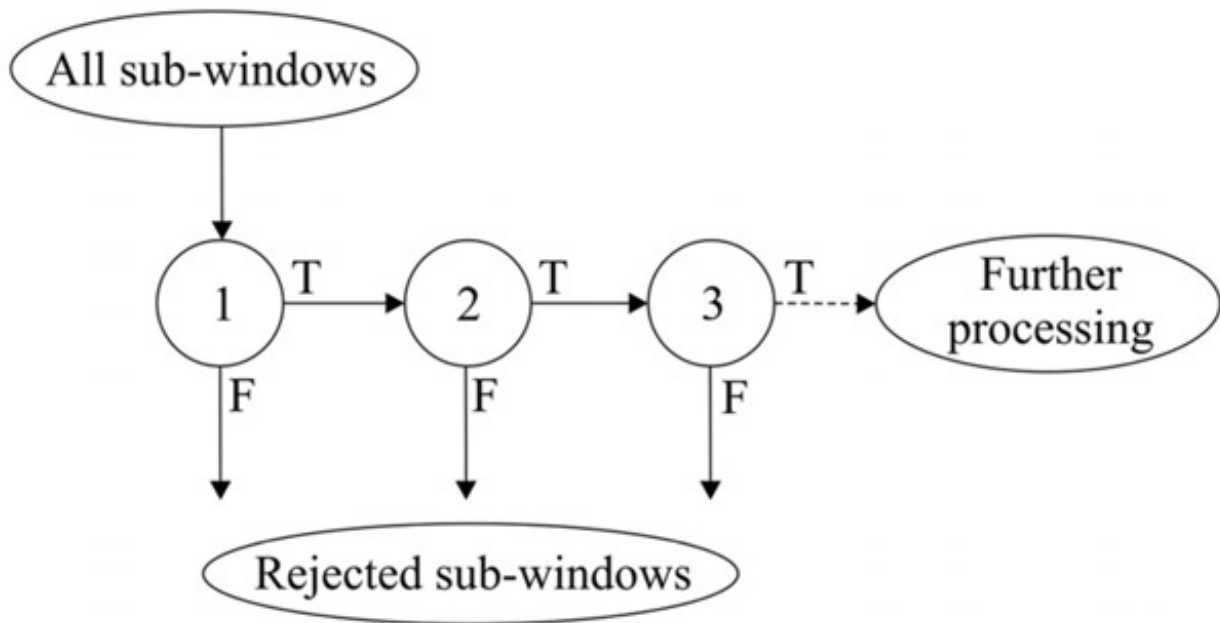


Figura 2.9: Esquemático ilustrativo do sistema de cascata dos classificadores [4].

altamente eficiente. Cada estágio reduz o número de falsos positivos e a taxa de detecção, e um alvo é selecionado para a mínima redução de falsos positivos e máximo decaimento de detecção. Em seguida, cada estágio é treinado pela adição de *features* até que a detecção do alvo e a taxa de falsos positivos se encontrem, obtendo as taxas por meio de um conjunto de validação. Por fim, estágios são adicionados até que o alvo em geral para falsos positivos e a taxa de detecção se encontrem.

Os resultados obtidos em [4, 23] mostram que o algoritmo apresenta uma grande eficiência na detecção de faces e um tempo de execução compatível com múltiplas aplicações de detecção facial.

O último tópico apresentado na próxima seção aborda os conceitos de normal e curvatura de superfícies, cruciais para a confecção dos mapas de curvatura expostos na Metodologia.

2.6 Normal e Curvatura de Superfícies

2.6.1 Normal

Em geometria, a normal se caracteriza como um objeto, sendo este um vetor ou uma linha, perpendicular a outro objeto. No caso bidimensional, a normal a uma curva em determinado ponto consiste na linha perpendicular a tangente neste ponto. No caso tridimensional, a normal se aplica a uma superfície, sendo essa representada por um vetor perpendicular a um plano tangente a superfície em um ponto p (Figura 2.10).

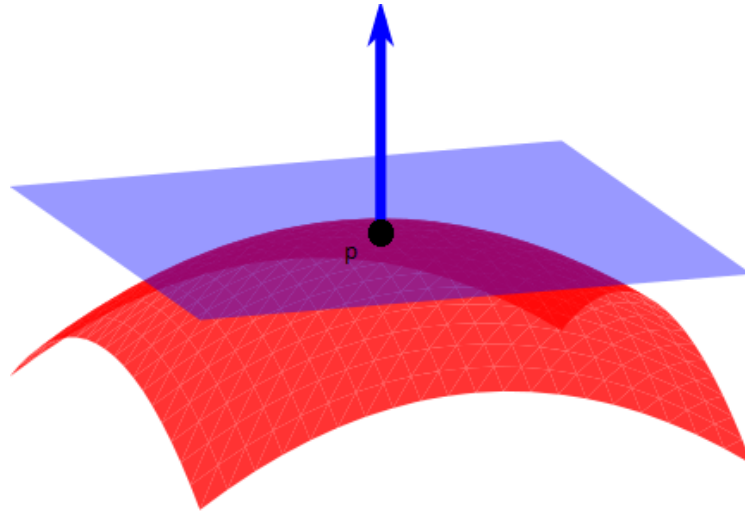


Figura 2.10: Normal de uma superfície em um ponto p (Adaptado de [5]).

2.6.1.1 Unicidade da Normal

A normal de uma superfície não é composta apenas de uma única direção. O vetor que aponta a direção oposta a normal de uma superfície, também é uma normal desta superfície. Em geral, não há uma maneira matemática de resolver o sinal de uma normal, apenas se um referencial for estabelecido, definindo um ponto de vista. No caso de uma superfície orientada, a normal da superfície é usualmente determinada por meio da regra da mão direita.

2.6.1.2 Utilidade da Normal

A normal é uma propriedade geométrica muito utilizada na definição de espaços 3D. Na computação gráfica tridimensional, existe uma grande reincidência do uso das normais, sendo voltadas para aplicações de cálculo de iluminação ou sombreamento, ou até mesmo o mapeamento das normais ao longo de pontos distribuídos na superfície de objetos.

As nuvens de pontos demonstram grande interesse no uso de normais, dado que por meio dos arredores de um ponto, pode-se estimar a composição da forma de um objeto, descrevendo-o para diversos interesses. A aplicação de normais auxilia em processos como triangulação e segmentação em nuvens de pontos.

2.6.2 Curvaturas

Na matemática, a curvatura se refere a uma série de características quantitativas que descrevem o grau em que um objeto geométrico se afasta de um outro objeto considerado plano (tal como uma superfície, se afasta de ser um plano liso, ou curva como no caso de uma linha, quando ela se afasta de ser uma reta), sendo definidas de diferentes maneiras de acordo com o contexto [24, 25].

Dentre os tipos de curvatura, existem as extrínsecas e intrínsecas, onde a primeira é definida para objetos que são embutidos em algum espaço (normalmente um espaço Euclidiano) de uma

maneira que relaciona os raios de curvatura de círculos que tocam o objeto, e a segunda é definida em termos do comprimento de curvas dentro de uma variedade de Riemann [24].

O exemplo canônico de uma curvatura extrínseca é a curvatura de um círculo, que apresenta uma curvatura igual ao recíproco do seu raio por toda parte. Círculos menores tendem a dobrar mais bruscamente e portanto apresentam uma curvatura maior [25].

A curvatura é uma quantia escalar, mas pode ser definida como um vetor, que leva em consideração a direção da dobra assim como a sua agudeza. A curvatura de objetos mais complexos (tal como superfícies ou espaços n -dimensionais curvos) são descritos por elementos mais complexos da álgebra linear, tal como o tensor de curvatura geral de Riemann [24].

2.6.2.1 Curvatura de Espaços

Um espaço de três ou mais dimensões pode ser intrinsecamente curvado. A curvatura é intrínseca no instante que ela é uma propriedade definida em cada ponto do espaço, ao contrário de uma propriedade definida em um espaço maior que o contém. Em geral, um espaço curvo pode ou não ser embutido em um ambiente espacial de maior dimensão, e caso não possa, essa curvatura deve ser definida apenas intrinsecamente [24].

A capacidade da curvatura intrínseca ser definida para cada ponto de um espaço, o torna viável para a utilização em nuvens de pontos, cuja característica visa extrair informações geométricas para cada ponto, independentemente de um espaço maior que o abranja.

Capítulo 3

Reconhecimento Facial

Este capítulo propõe uma descrição detalhada sobre reconhecimento facial, abordando suas aplicações, dificuldades gerais, e diversos métodos já implementados para esta tarefa.

3.1 O uso do reconhecimento facial

O reconhecimento facial consiste em um método composto de técnicas de análise de características biométricas [26]. As características biométricas são um tipo de recurso muito estudado e pesquisado nos dias atuais, envolvido na área de reconhecimento para autenticação.

As tecnologias baseadas em características biométricas incluem identificação baseada em características fisiológicas (tal como face, digitais, geometria dos dedos, geometria das mãos, veias das mãos, palma, íris, retina e orelha) e traços comportamentais (como a maneira de andar, assinaturas e padrão de digitação)[27]. O reconhecimento facial aparenta oferecer várias vantagens em relação aos demais métodos biométricos, iniciando pelo fato que a maioria dessas tecnologias necessitam de uma ação voluntária do usuário, seja de posicionamento da mão ou do dedo para sistemas de análise de geometria de dedos e mãos, ou posicionamento específico em frente a uma câmera para identificação por íris ou retina (entretanto um ambiente descontrolado sem colaboração do usuário prejudica suas taxas de acerto). O reconhecimento facial pode ser realizado de maneira passiva sem ação explícita ou participação por parte do usuário (mas para garantir altas taxas de acerto necessita da sua participação), dado que imagens faciais podem ser adquiridas à certas distâncias por uma câmera . Tal fato é muito benéfico para aplicações de segurança e vigilância. A identificação por íris ou retina exigem equipamentos de alto custo e são muito sensíveis a qualquer movimentação do corpo. Reconhecimento por voz é suscetível a ruídos de ambiente em locais públicos e sofrem variações auditivas em linhas de telefone ou gravações por fita. Assinaturas podem ser modificadas pelo usuário ou forjadas. Bons algoritmos de reconhecimento facial e processamento apropriado das imagens podem compensar ruídos e pequenas variações de orientação, escala e iluminação.

3.2 Aplicações

O reconhecimento facial é primariamente utilizado para duas tarefas

1. Verificação (comparação um para um): Quando a imagem de uma face de um indivíduo desconhecido é apresentada junto com uma afirmação de identidade, verificando se o indivíduo condiz com quem é dito ser; e
2. Identificação (comparação um para muitos): Onde dada a imagem de um indivíduo desconhecido, determina-se a sua identidade comparando (possivelmente após uma codificação) a imagem com a base de dados de imagens de indivíduos conhecidos.

Existem inúmeras áreas de aplicações em que o reconhecimento facial pode ser explorado para esses dois propósitos, como os descritos abaixo:

- Segurança (controle de acesso a prédios, aeroportos/portos marítimos, caixas eletrônicos, pontos de verificações em fronteiras, segurança de computadores/redes, autenticação de e-mails ou estações de trabalho multimídia);
- Vigilância (Inúmeras CCTVs podem ser monitoradas para buscar por criminosos conhecidos, traficantes de drogas, etc; e autoridades podem ser notificadas quando um for localizado. Esse procedimento foi utilizado no jogo do Super Bowl 2001 em Tampa, Flórida [28]);
- Verificação de identidade em geral (registro eleitoral, operações bancárias, comércios eletrônicos, identificação de recém nascidos, identidades nacionais, passaportes, carteiras de motorista, identificação de empregados);
- Sistemas de justiça criminal (sistemas de *booking/mug-shot* policiais, análises pós-evento, forense);
- Investigações por bases de dados de imagens (procurando bases de dados de imagens de motoristas habilitados, beneficiários de algum programa, crianças desaparecidas, imigrantes e *booking* policial);
- Aplicações de “Smart Card” (ao contrário de manter uma base de dados de imagens faciais, a impressão facial pode ser armazenada em um *smart card*, código de barras ou fita magnética, realizando a autenticação por comparação da imagem ao vivo e um *template* armazenado);
- Ambientes multimídia com interfaces adaptativas entre humano e computador (parte de sistema ubíquo ou sistemas conscientes de contexto, monitorando comportamentos em creches ou asilos, reconhecendo o interno e acessando suas necessidades);
- Indexação em vídeos (rotulando faces em vídeo); e
- Reconstrução facial de vítimas.

Em adição a estas aplicações, as técnicas subjacentes na tecnologia de reconhecimento facial atual tem sido modificada e utilizada para aplicações relacionadas (tal como classificação de gênero, reconhecimento de expressões, reconhecimento de *features* faciais e rastreamento) [26]. Cada um desses casos citados possui sua utilidade em diversos domínios, por exemplo, reconhecimento de expressões pode ser utilizado no campo da medicina para monitoramento de tratamentos intensivos, enquanto reconhecimento e detecção de *features* faciais pode ser utilizada para rastreamento dos olhos de um motorista, monitorando e avaliando seu nível de fadiga, ou detecção de estresse.

O reconhecimento facial também vem sendo utilizado em conjunto com outros sistemas biométricos como reconhecimento de fala, íris, digitais, orelha e modo de andar com o intuito de aperfeiçoar a performance de reconhecimento desses outros métodos.

3.3 Dificuldades Gerais

Reconhecimento facial é um trabalhoso e específico caso de reconhecimento de objeto. A dificuldade desse processo é sustentada pelo fato que em sua forma mais comum (visão frontal), faces aparentam ser pouco semelhantes e suas diferenças visam ser bastante sutis. Consequentemente, imagens de faces frontais formam um grupo bem denso em espaços de imagem, que torna virtualmente impossível para técnicas de reconhecimento de padrões distingui-las entre si de maneira acurada e com alto índice de sucesso[29].

Ademais, a face humana não é um objeto rígido e único. De fato, existem inúmeros fatores que causam variações na aparência de uma face. A fonte da variação na aparência da face pode ser categorizado em dois grupos, sendo esses fatores intrínsecos e extrínsecos [30].

Fatores intrínsecos são decorrentes, puramente, da natureza física de uma face, e são independentes do observador. Esses fatores podem ser divididos em duas classes, intrapessoal e interpessoal. Fatores intrapessoais são responsáveis pela variação da aparência da face da mesma pessoa, sendo exemplos dessa classe a idade, expressões faciais, parafernálias faciais (pelos faciais, óculos, cosméticos, etc). Fatores interpessoais, entretanto, são responsáveis pelas diferenças na aparência facial de pessoas diferentes, por exemplo etnia e gênero.

Fatores extrínsecos geram alterações na aparência facial por meio da interação da luz com a face e o observador. Esses fatores incluem iluminação, postura, escala e parâmetros de imagem (resolução, foco, ruído, etc).

Apesar da maioria dos sistemas de reconhecimento facial funcionarem bem em condições restritivas, ou seja, cenários em que ao menos alguns dos fatores que contribuem para uma variação entre as imagens faciais são controladas, a performance da maioria desses sistemas rapidamente decresce quando são postos em operação sob condições onde nenhum desses fatores são controlados.

3.4 Métodos de reconhecimento facial

Os métodos usados para detecção facial dependem da finalidade da aplicação. Por exemplo, para aplicações de vigilância as imagens usualmente são retiradas de câmeras de segurança, enquanto que investigações por meio de pesquisa em base de dados podem requerer imagens de intensidade estáticas obtidas por câmeras padrões. Algumas aplicações como certificação de identidade por reconhecimento 3D necessitam que o usuário se submeta a vários scanners 3D ou sensores infravermelho.

Dependendo do tipo de aquisição de dados faciais, os métodos de detecção facial podem ser divididos em três categorias: métodos que operam sobre imagens de intensidade, métodos que lidam com sequências de vídeos, e aquelas que requerem outros dados sensoriais como imagens 3D ou imagens no infravermelho. Em [26] são listados diversos métodos desenvolvidos que se classificam nesses tipos.

As sessões subsequentes tratam brevemente de alguns dos métodos de cada uma das três categorias, dando uma visão dos prós e cons de cada esquema mencionado.

3.4.1 Reconhecimento facial de imagens de intensidade

Os métodos de reconhecimento facial por imagens de intensidade recaem em duas categorias principais: baseado em *features* e holísticas. Uma visão geral dos métodos mais conhecidos dentro dessas categorias é dada abaixo.

3.4.2 Baseado em features

Abordagens baseadas em *features* primeiramente processam a imagem de entrada a fim de identificar, extrair e qualificar características faciais distintas como olhos, bocas, nariz, assim como outros pontos marcantes da face. A partir desses dados, são computadas as relações geométricas entre os pontos dessas regiões, reduzindo assim a imagem de entrada a um vetor de *features*. Técnicas estatísticas de reconhecimento são então empregadas para procurar correspondências entre faces usando essas medidas.

Os primeiros trabalhos realizados sobre reconhecimento facial automatizado foram baseados nessa técnica, como o feito por Kanade [31], que empregou métodos simples de processamento de imagens para extrair um vetor de 16 parâmetros faciais – proporções entre distâncias, áreas e ângulos – e usou distâncias euclidianas para comparar, chegando a uma ótima performance de 75% numa base de dados de 20 pessoas diferentes, usando 2 imagens por pessoas.

Brunelli e Poggio [6], baseando-se na abordagem de Kanade, computaram um vetor de 35 características geométricas (Figura 3.1) de uma base de dados de 47 pessoas (cada um com 4 imagens) e chegaram a uma taxa de 90 % de reconhecimento. Entretanto, eles também relataram uma taxa de 100% de acurácia para a mesma base, usando um simples modelo de correspondência.

Algumas técnicas de extração de *features* mais sofisticadas envolvem modelos deformáveis,

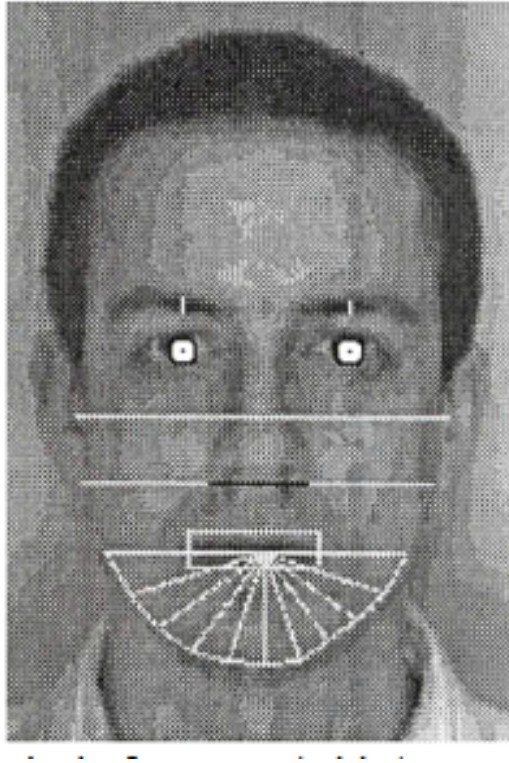


Figura 3.1: Features Geométricos utilizados no experimentos de reconhecimento facial [6].

transformadas Hough [32], operador de simetria de Reisfeld [33], e filtros e operadores morfológicos de Graf [34]. Todavia, todas essas técnicas se baseiam fortemente em heurísticas, assim como restringe o espaço de busca com limitações geométricas. Além disso, uma certa tolerância deve ser dada aos modelos, já que eles nunca poderiam encaixar as estruturas nas imagens. O uso de um valor de tolerância muito grande tende a destruir a precisão necessária para reconhecer indivíduos da base dos melhores modelos finais de parâmetros e faz dessas técnicas insensíveis a variações no tempo necessárias para o reconhecimento. Cox et al. [35] relatou uma performance de reconhecimento de 95% numa base de dados de 685 imagens (com uma imagem para cada indivíduo), usando um vetor de *features* de 30 dimensões derivadas de 35 características faciais. No entanto, as *features* foram extraídas manualmente, então é razoável assumir que se fossem automatizadas, as performances seriam piores, portanto menos precisas. Em geral, os algoritmos mais atuais para extração de *features* automáticas não provêm um alto grau de acurácia e demandam grande poder computacional.

Outra técnica bem conhecida que possui abordagem baseada em *features* é Elastic Bunch Graph Matching (EBGM). Proposta por Wiskott et al. [36], baseia-se em Estruturas de dados dinâmicas. Um grafo para um indivíduo é gerado da seguinte forma: um conjunto de pontos chave do rosto são escolhidos, e associados a um nó de um grafo completo, que é rotulado com um dos filtros de Gabor aplicado a uma janela ao redor dos pontos chave. À cada aresta é atribuído um peso que é a distância entre os pontos chave. Um conjunto representativo desses grafos é combinado numa estrutura do tipo pilha, chamado Face Bunch Graph (FBG). Uma vez que o sistema cria um FBG,

grafos para as novas faces podem ser automaticamente geradas pelo EBGM. O reconhecimento de uma nova imagem facial é feito comparando o grafo gerado com aqueles do banco de dados e escolhendo aquele com maior valor de similaridade.

Usando essa arquitetura, a taxa de reconhecimento pode chegar a 98% para o primeiro classificado, e até 99% para os 10 primeiros classificados utilizando uma galeria de 250 indivíduos. O sistema foi aperfeiçoado para permitir trabalhar com diferentes posturas (Figura 3.2), mas o desempenho do reconhecimento em faces na mesma orientação permanece o mesmo. Apesar deste método estar entre os métodos de melhor performance na mais recente avaliação da FERET, o método apresenta uma grande desvantagem de requerer o posicionamento manual dos grafos para as primeiras 70 faces antes da correspondência elástica de grafos se tornar adequadamente confiável. Campadelli e Lanzarotti [37] têm realizado experimentos recentemente com esta técnica, onde eles eliminam a necessidade do posicionamento manual do grafo utilizando modelos paramétricos, baseado em *templates* deformáveis propostos em [38], para localizar pontos fiduciais automaticamente. Outra variação recente deste método substitui os *features* de Gabor por uma estratégia de correspondência de grafos e HOGs (Histogram of Oriented Gradients [39]).

Esforços consideráveis têm sido dedicados em reconhecimento facial por meio do perfil das faces, dado que a extração de *features* se torna de certa maneira um problema unidimensional mais simples. Kaufman e Breeding [40] reportaram uma taxa de reconhecimento de 90% utilizando perfil de faces, entretanto eles utilizaram uma base de apenas 10 indivíduos. Harmon et al. [41] obteve taxas de reconhecimento de 96% em uma base de dados de 112 indivíduos, utilizando um vetor de *features* de dezessete dimensões para descrever o perfil das faces, e utilizando uma medida de distância Euclidiana para comparação e correspondência. Recentemente Liposcak e Loncaric [42] reportaram uma taxa de acerto de 90% em uma base de dados de 30 pessoas, utilizando um filtro de subespaço para extrair um vetor de *features* de vinte e uma dimensões para descrever os perfis e assumindo uma distância euclidiana para correlacioná-los.

3.4.2.1 Vantagens e desvantagens

A principal vantagem oferecida pelas técnicas baseadas em *features* é decorrente do fato que a extração de pontos precede a análise feita para comparação de uma imagem com a de um indivíduo conhecido, tais métodos são robustos a variações de posição na imagem de entrada. A princípio, esquemas baseados em *features* podem ser invariantes em relação ao tamanho, orientação ou brilho. Outros benefícios de tais esquemas incluem representação compacta das imagens faciais e velocidade alta de comparação.

As maiores desvantagens dessas abordagens são a dificuldade de automatização da detecção das *features* e o fato de que o implementador de qualquer uma dessas técnicas tem de tomar decisões arbitrárias em relação a quais *features* são importantes. Afinal, se ao conjunto de *features* faltasse habilidade de distinção, por mais que o tempo de processamento fosse imenso, isso não compensaria a deficiência intrínseca do método.

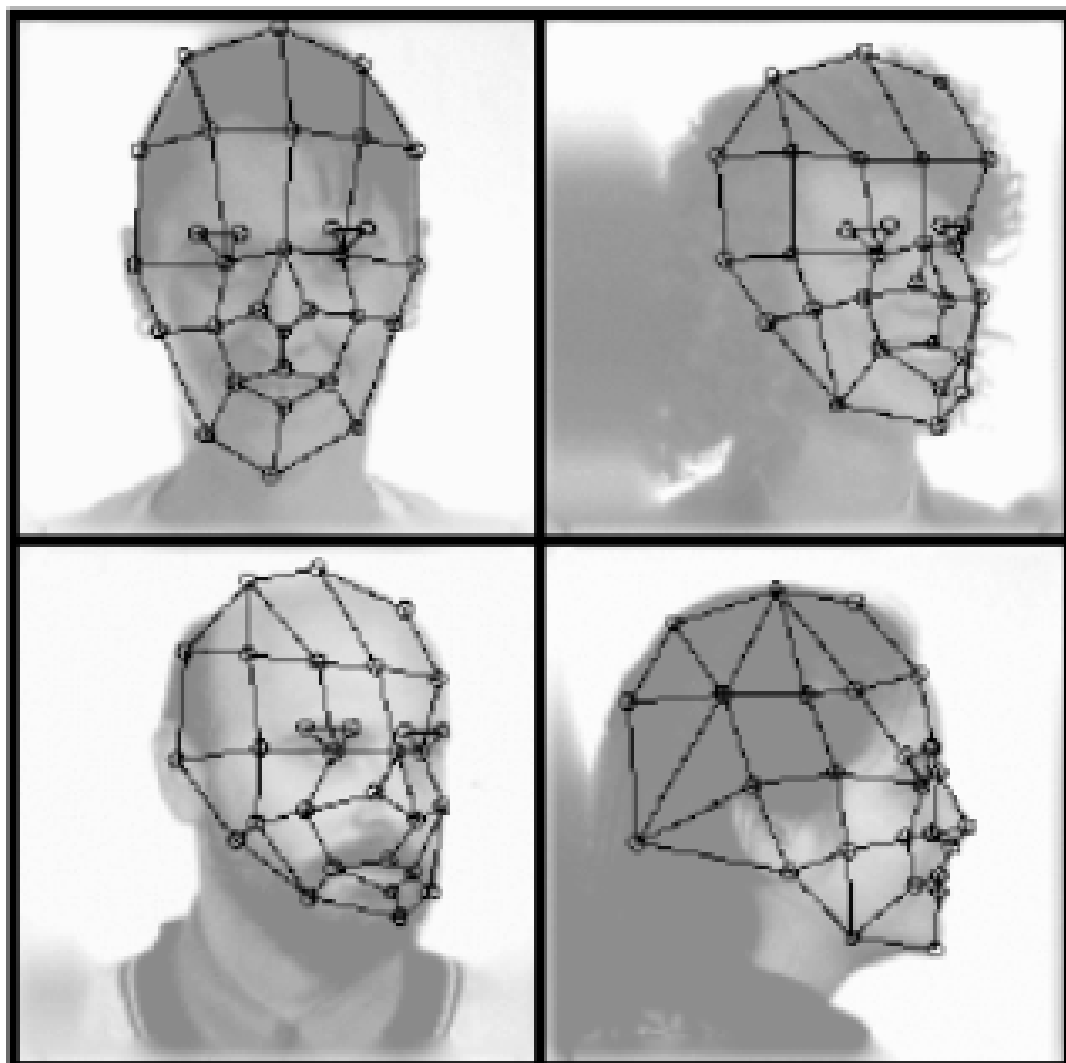


Figura 3.2: Grades para reconhecimento facial [7].

3.4.3 Holística

Abordagens holísticas tem por objetivo identificar faces usando representações globais, ou seja, descrições baseadas na imagem como um todo e não em regiões específicas do rosto. Esses esquemas são subdivididos em dois grupos: estatísticas e abordagens IA.

3.4.3.1 Estatística

Na estatística, a imagem é representada como um vetor 2D de valores de intensidade, e o reconhecimento é feito por comparação direta entre a imagem de entrada e as faces da base de dados. Embora essa abordagem tenha se mostrado funcional em circunstâncias limitadas (i.e. iluminação, escala e poses idênticas), é muito custosa computacionalmente e sofre das deficiências usuais de abordagens baseadas em comparações diretas, como sensibilidade a orientação da face,

tamanho e variações de condições de iluminação, desordem de fundo da imagem e ruídos. Os maiores obstáculos no uso de métodos de reconhecimento fundamentados em comparação direta, é que eles realizam a classificação num espaço de alta dimensionalidade. Em contrapartida, vários outros esquemas empregam reduções de dimensionalidade estatísticas para obter e conservar as dimensões das *features* mais significativas antes do reconhecimento. Alguns desses esquemas estão mencionados nos parágrafos a seguir.

Sirovich and Kirby [43] foram os primeiros a utilizar análise de componentes principais (PCA) [44, 45] para representar economicamente imagens de rosto. Eles demonstraram que qualquer imagem pode ser eficientemente representada ao longo de espaços de coordenadas de auto-figuras, e que qualquer face pode ser construída praticamente usando apenas uma pequena coleção de auto-figuras e as projeções correspondentes ao longo de cada uma.

Turk e Pentland [46, 47] realizaram, com base nas descobertas de Sirovich e Pentland, que projeções ao longo de auto-figuras poderiam ser usadas na classificação e reconhecimento de *features*. Eles empregaram esse raciocínio para desenvolver um sistema de reconhecimento facial que gera autofaces, que correspondem à autovetores associados aos autovalores dominantes da matriz de covariância de rostos conhecidos, e então reconhece outros rostos, comparando as projeções deles ao longo das autofaces com as de imagens de rostos já conhecidos. As autofaces definem um espaço de *features* que reduzem drasticamente a dimensionalidade do espaço original, e a identificação facial é realizada nesse espaço reduzido.

A capacidade dos sistema de Turk e Pentland tem sido expandidas em diversas maneiras em [8] e testado em uma base de dados de 7562 imagens de aproximadamente 3000 pessoas. Um método de “observador múltiplo” foi sugerido para lidar com grandes mudanças de postura: Dado N indivíduos em M vistas diferentes, um observador pode realizar tanto reconhecimento quanto estimação de postura em um autoespaço universal calculado pela combinação de NM imagens (aproximação paramétrica) ou, alternativamente, pode construir um conjunto de M autoespaços separados, um para cada N vistas (a abordagem baseada em vistas). Relata-se que a abordagem baseada em vistas apresentou um resultado mais condizente do que a aproximação paramétrica. Uma abordagem de “*autofeatures*” modular também foi proposta para lidar com variações localizadas na aparência facial onde uma descrição de baixa resolução de toda face é aumentada por adição de detalhes em maior resolução em termos dos *features* faciais salientes (Figura 3.3). Esse sistema indica ter produzido resultados ligeiramente melhores do que a abordagem básica de autofaces (Figura 3.4). PCA aparenta funcionar corretamente quando apenas uma imagem de cada indivíduo está presente, então Belhumeur et al. [9] argumentou que escolhendo a projeção que maximiza a dispersão total, PCA conserva variações indesejadas devido a iluminação e expressões faciais.

Como declarado por Moses et al. [48], “As variações entre imagens da mesma face devido a direção da iluminação são quase sempre maiores do que variações de imagem por trocar a identidade da face” (ver Figura 3.5). Portanto, é proposto um dispersor interclasse que é supostamente melhor para classificação do que o PCA. Conduzindo vários testes em 330 imagens de 5 pessoas (66 de cada), reportou-se que o método deles, chamado de Fisherfaces, que utiliza uma projeção de subespaço anterior a projeção LDA (para prevenir que o dispersor interclasse se torne degenerativo)

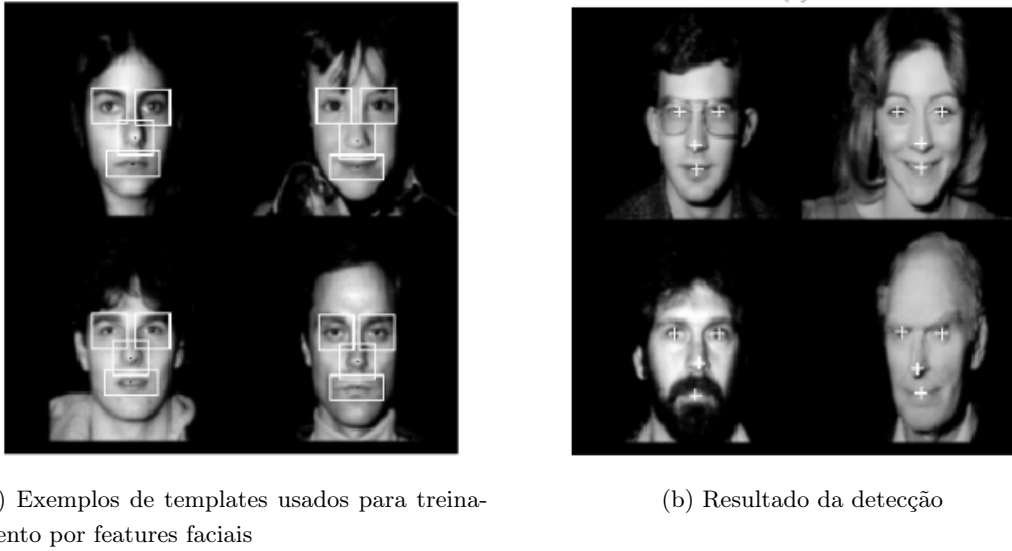


Figura 3.3: Treinamento de detecção de features por template e resultado da detecção [8].

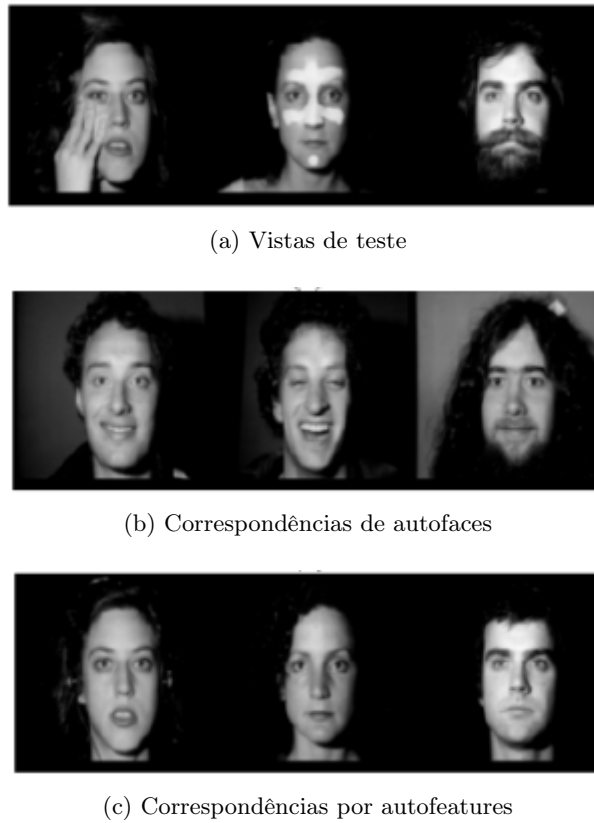


Figura 3.4: Resultado entre autofaces e autofeatures [8].

é melhor em variações simultâneas manipuladas de iluminação e expressão. Swets e Wang [49] previamente anunciaram resultados semelhantes quando empregaram o mesmo procedimento não apenas para faces, mas para objetos gerais também (90% de acurácia em uma base de dados de 1316+298 imagens de 504 classes). Deve-se ressaltar que trabalhos recentes mostram que quando

o conjunto de dados de teste é pequeno, PCA pode superar LDA e que o PCA é menos sensível a conjuntos de teste diferentes.



Figura 3.5: Variação da aparência da face para diferentes ângulos de iluminação [9].

Uma das principais desvantagens do PCA e do LDA é que essas técnicas efetivamente analisam apenas a estrutura Euclidiana e falham em descobrir a estrutura subjacente se as imagens faciais se encontram em uma subvariedade não linear no espaço da imagem. Como foi mostrado que imagens faciais possivelmente se encontram em subvariedades não lineares [50] (especialmente se houver uma variação perceptível do ponto de vista, iluminação ou expressão facial), algumas técnicas não lineares tem, conseqüentemente, sido propostas para descobrir estruturas não lineares das variedades.

Os autovetores encontrados por PCA dependem apenas de relações emparelhadas entre pixels na base de dados. Entretanto outros métodos existentes podem encontrar vetores base que dependem de relações de maior ordem entre os pixels, e é razoável esperar que a utilização dessas técnicas trariam resultados melhores de reconhecimento. *Independent Component Analysis* (ICA) [51], uma generalização do PCA, é um destes métodos empregados na tarefa de reconhecimento facial. ICA visa encontrar uma decomposição e representação da imagem independente, ao contrário de uma não correlacionada. Bartlett et al. [52] aplicou ICA em imagens da base de dados da FERET em duas arquiteturas diferentes: uma tratou as imagens como variáveis aleatórias e os pixels como saída, e inversamente, a segunda tratou pixels como variáveis aleatórias e as imagens como saída. Ambas representações ICA superaram as representações PCA em reconhecimento facial ao longo do dia e em mudanças de expressão. Um classificador que combinou ambas as representações ICA apresentou o melhor desempenho. Outros pesquisaram com o ICA [53, 54] e alegaram que esta técnica, e variações dela, aparentam ter uma melhor performance do que PCA na maioria das circunstâncias.

Outros métodos de subespaço vem sendo explorados para a tarefa de reconhecimento facial: Foon et al. [55] integrou várias transformadas de wavelet e fatorações de matrizes não negativas e alegaram ter obtido taxas de verificação melhores quando comparado a abordagens básicas de autofaces. Em [56], um subespaço intraclasses é construído, e a classificação é baseada na menor distância ponderada entre a face consultada e cada subespaço intraclasses. Resultados experimentais são apresentados para demonstrar que este método apresenta uma performance melhor que outras técnicas de *feature* mais próximo.

Um estudo de comparação de quatro representações de subespaços para reconhecimento facial, i.e., PCA, ICA, Fisher Discriminant Analysis (FDA), e autofaces probabilísticas e suas versões “kernalizadas” (quando disponíveis), é apresentado em [53]. Uma análise compreensiva de avanços em análises de subespaço para reconhecimento facial pode ser encontrado em [54].

3.4.3.2 Inteligência Artificial

Abordagens de inteligência artificial (IA) utilizam ferramentas como redes neurais e técnicas de ML para reconhecer faces. Alguns exemplos de métodos pertencentes a essa categoria são citados nos parágrafos subsequentes.

Em um método existente, 50 dos principais componentes da imagem foram extraídos e uma rede neural auto-associativa foi usada para reduzir os componentes a cinco dimensões. Um perceptron foi aproveitado para classificar a representação final, e ainda que resultados favoráveis tenham sido recebidos, a base de dados usada para treinos e testes foi bem simples: as figuras foram manualmente alinhadas, não havia variação de luminosidade, inclinação ou rotação e haviam apenas 20 pessoas na base de dados [57].

Weng et al. em [58] fez uso de uma rede neural hierárquica que crescia automaticamente, mesmo não treinada no método tradicional de gradiente descendente. Eles reportaram bons resultados numa base de dados de 10 sujeitos.

Lawrence et al. em [59] reportou uma taxa de reconhecimento de 96.2% numa base de dados da ORL (uma base de dados de 400 imagens de 40 indivíduos) usando uma solução de rede neural híbrida que combinava amostragem de imagens locais, um mapa de redes neurais auto ajustáveis e uma rede neural convolucional. O método das auto-faces [46, 47] produziu uma acurácia de 89,5% no reconhecimento nos mesmos dados. Substituindo o mapa auto ajustável pela transformada de Karhunen-Loeve e a rede neural convolucional por um perceptron multi-layer resultou em uma taxa de reconhecimento de 94.7% e 60%, respectivamente.

Elyan e Demeriel [60] usaram a análise de componentes principais para obter vetores de projeção de *features* de imagens faciais que na sequência foram classificadas utilizando redes neurais de *feedforward*. Alguns testes na base de dados da ORL utilizando diversos números de treinamentos e imagens de teste apresentaram que o desempenho desse sistema foi melhor que o método das auto-faces [46, 47], onde um classificador de vizinho mais próximo foi utilizado para classificação.

Li e Yin [61] introduziram um sistema em que a imagem facial é primeiramente decomposta utilizando transformadas wavelet, em três níveis. O método de Fisherfaces [9] é aplicado para

cada uma das três sub-imagens de baixa frequência. Em seguida, os classificadores individuais são fundidos utilizando uma rede neural RBF. O sistema resultante foi testado em imagens de 40 indivíduos da base de dados da FERET e demonstrou superar os classificadores individuais e o método de Fisherfaces.

Melin et al. [62] dividiu a face em três regiões (olhos, boca e nariz) e atribuiu cada região à um módulo da rede neural. Um integral *fuzzy* de Sugeno foi utilizada para combinar as saídas dos três módulos para tomar uma decisão final da face reconhecida. Os testes foram realizados em uma base de dados pequena composta por 20 pessoas e foi relatado que a rede modular rendeu melhores resultados que a rede monolítica.

Zhang et. al [63] propôs uma solução onde uma função de similaridade é aprendida descrevendo o nível de confiabilidade que duas imagens pertencem à mesma pessoa, similar a [64]. Os *features* faciais são selecionados obtendo-se histogramas LBP (*Local Binary Pattern*) de sub-regiões da imagem facial e a distância qui-quadrado entre os histogramas LBP correspondentes são escolhidos como os *features* característicos. O algoritmo de aprendizagem AdaBoost, apresentado por Freund e Schapire [65], é aplicado para selecionar os *features* LBP mais eficientes assim como é aplicado para obter a função de similaridade na forma de uma combinação linear de aprendizes mais fracos baseados em *features* LBP. Resultados experimentais adquiridos pela aplicação do método ao conjuntos de imagens faciais frontais da FERET mostrou que esse método proporcionou uma taxa de reconhecimento maior que 97.9% utilizando menos *features* do que alguns métodos similares propostos por Ahonen et al. [66].

Alguns pesquisadores utilizaram a comparação de um para um por decomposição do problema de reconhecimento facial de múltiplas classes em um número de problemas de classificação binários. Neste método um classificador é treinado para cada par de classes, ignorando todas as outras. As saídas de todos os classificadores binários são combinados para construir um resultado global. Para classificadores binários com saídas probabilísticas, *pair-wise coupling* (PWC) pode ser utilizado para parear essas saídas em um conjunto de probabilidades a posteriori. Na sequência, um exemplo de teste é atribuído para a classe com a máxima probabilidade a posteriori. Uma das principais desvantagens da PWC é quando um exemplo de teste não pertence a ambas as classes relacionadas ao classificador binário, então a saída daquele classificar é irrelevante e pode interferir de maneira negativa no resultado global.

Um novo algoritmo chamado PWC-CC [67](onde CC se refere à *correcting classifier*) é apresentado para resolver este problema: para cada classificador binário separando a classe c_i da classe c_j , um novo classificador separando as duas classes de todas as outras classes é treinado. Apesar do PWC-CC ser mais eficiente que o PWC, ele apresenta suas próprias desvantagens. Em [68], um novo método PWC-CC (NPWC-CC) é proposto para resolver o problema do reconhecimento facial e os resultados dos testes na base de dados da ORL sustentam a alegação de que sua performance supera a do PWC-CC. In [69], o método PWC ótimo (O-PWC) é apresentado e demonstra ter taxas de reconhecimento melhores que a do PWC. A extração de *features* é feita utilizando uma análise de componentes principais em [60] e pelas transformadas de wavelet em [69]. Em ambos os casos ([68] e [69]), SVMs (*Support Vector Machines*) foram empregadas como classificadores

binários e as suas saídas foram mapeadas em probabilidades utilizando o método sugerido por Platt [70].

Deve-se levar em consideração que a SVM é considerado um dos algoritmos mais efetivos para problemas de classificação de padrões. Em geral, ele funciona da seguinte maneira para problemas binários: Primeiramente, os exemplos de treino são mapeados para um *feature* espacial de alta dimensão H . Em seguida o hiperplano ótimo em H é solicitado para separar o melhor possível exemplos de classes diferentes, enquanto maximiza a distância de ambas as classes em relação ao hiperplano. SVM foi utilizada para o reconhecimento facial por vários outros pesquisadores [71, 72, 73, 74].

Modelos Ocultos de Markov (HMM) [75] também vem sendo aplicados para tarefas de reconhecimento facial. Samaria e Harter [76] utilizaram um HMM unidimensional para obter um pico de acurácia de reconhecimento de 87% na base de dados da ORL. Posteriormente, o HMM unidimensional foi aprimorado para um pseudo HMM bidimensional [77], adquirindo uma performance de reconhecimento melhor de 95% na mesma base de dados utilizando metade das imagens para treinamento e a outra metade para testes. Nefian e Hayes III [78] apresentaram uma taxa de reconhecimento melhor de cerca (98%) com o mesmo treinamento e o mesmo teste utilizando HMM de modelos faciais incorporados, e afirmaram que esse sistema é muito mais rápido que o implementado por Samaria [77] e invariante para a escala das imagens faciais.

3.4.3.3 Sistemas de Múltiplos Classificadores

Dado que a performance de qualquer classificador é mais sensível a alguns fatores e relativamente invariantes para outros, a tendência tem sido combinar classificadores individuais com o intuito de integrar suas informações complementares e então criar um sistema que é mais robusto que qualquer classificador individual para variáveis que dificultam a tarefa de reconhecimento. Esses sistemas foram denominados como MCSs (*Multiple Classifier System*) [79] e são uma área de pesquisa muito ativa recentemente. Um exemplo dessa abordagem inserida no reconhecimento facial pode ser encontrada em Lu et al. [80], que fundiu os resultados do PCA, ICA e LDA utilizando a estratégia de integração da regra da soma e uma rede baseada em RBF.

Marcialis e Roli [81, 82, 83] combinaram os resultados dos algoritmos PCA e LDA. Achermann e Bunke [84] utilizaram regras de fusão simples (maioria dos votos, somatório de classificação, regra de combinação de Bayes) para integrar a saída ponderada dos três classificadores, baseado em vistas faciais frontais e de perfil. Tolba e Abu-Rezq [85] empregou uma simples regra de combinação para fundir decisões de redes RBF e LVQ. Wan et al. [86] usaram modelo híbrido de SVM e HMM. Kwak e Pedrycz [87] dividiram a face em três regiões, aplicaram o método de Fisherfaces para as regiões assim como para toda a face, e integrou o resultado de classificação utilizando a integral *fuzzy* de Choquet.

3.4.3.4 Vantagens e Desvantagens

A principal vantagem das abordagens holísticas é que elas não destroem nenhuma informação das imagens por concentrar apenas em regiões limitadas de pontos de interesse [88]. Entretanto, essa mesma propriedade é a sua principal desvantagem também, dado que a maioria dessas abordagens iniciam com a hipótese básica que todos os pixels na imagem são igualmente importantes. Conseqüentemente, essas técnicas não são apenas computacionalmente custosas como requerem um alto grau de correlação entre a imagem de teste e as imagens de treino, não funcionando de maneira efetiva para grandes variações de postura, escala e iluminação, etc. Mesmo assim, muitos desses algoritmos foram modificados e/ou aprimorados para compensar essas variações, e técnicas de redução de dimensionalidade têm sido exploradas (considerando que apesar destas técnicas aumentar a capacidade de generalização, o efeito colateral é a potencial perda de informações descritivas [89]). com o fato de que essas abordagens aparentam apresentar resultados melhores de reconhecimento em relação às técnicas baseadas em *features* no geral. Em uma avaliação da FERET [90, 91], os métodos de autofaces probabilísticas [64], Fisherface [9] e EBGM [36] foram classificados como as três melhores técnicas para reconhecimento facial (mesmo que o método EBGM seja baseado em *features* em geral e seu sucesso depende na aplicação de métodos de redes neurais holísticas em nível de *features*).

3.5 Reconhecimento Facial por outras entradas sensoriais

A pesquisa de reconhecimento facial vem sendo focada na identificação de indivíduos de imagens de intensidade 2D, porém atualmente os estudos vêm sendo direcionados a exploração de outras modalidades, tais como imagens 3D, de profundidade, ou infravermelho para o mesmo propósito.

3.5.1 Métodos de modelo 3D

O principal argumento a favor do uso de informações 3D para o reconhecimento facial aparenta ser o fato que ela permite a exploração de *features* baseados no formato e na curvatura da face (como o formato da testa, a linha da mandíbula, e bochechas) sem serem atormentadas por variações de iluminação, orientação e confusão de fundo que afeta sistemas 2D [88, 92, 93]. Outro argumento para o uso de dados de profundidade é que no nível atual de tecnologia, essa é a maneira mais direta para introduzir ou registrar informações complexas de formato para análises de máquinas. A desvantagem óbvia destas abordagens se encontra na sua complexidade e custo computacional.

As seguintes técnicas são utilizadas atualmente para a aquisição de informações 3D:

- Sistemas de escaneamento: Scanners a base de laser vem produzindo resultados bastante acurados, entretanto o custo desses serviços comerciais de escaneamento são elevados.
- Sistemas de luz estruturada: Esses sistemas fazem uso dos princípios de visão estereográfica para obter dados de profundidade. Sua principal vantagem é que os únicos equipamentos necessário são câmeras e algum tipo de sistema de projeção.

- Sistemas de visão estéreo: Esses sistemas tentam extrair a informação 3D de duas ou mais imagens 2D do mesmo objeto tiradas de diferentes ângulos. Eles são limitados a objetos que irão “gerar um número suficiente de *features* de imagem” que permitirão uma correspondência conclusiva. No caso de tentar estabelecer o formato de um objeto razoavelmente suave, tal como a face humana, esses sistemas seriam incapazes de gerar um formato de superfície acurado. (Superfícies suaves podem ser “enrijecidas” pela projeção de um padrão de textura na face).
- Renderização/Formação reversa por sombreado: O esforço dessas técnicas é construir o formato de um objeto utilizando conhecimento sobre iluminação e as propriedades físicas do objeto.

Há alguns anos, haviam poucos *papers* que descrevem tentativas de reconhecer faces baseado sobretudo em dados de profundidade ou dados 3D sobre as faces dos sujeitos. Entretanto, recentemente houve um grande interesse nesta área e diversos novos esquemas deste tipo têm sido propostos nos últimos anos. Uma das primeiras abordagens é descrita em [94], onde as principais curvaturas da superfície da face são calculadas por dados de profundidade, posteriormente esses dados - complementado por uma informação posterior sobre a estrutura da face - é utilizada para localizar vários *features* faciais (i.e. nariz, olhos, testa, pescoço, queixo, etc). As faces são normalizadas para uma posição padrão e re-interpoladas em uma grade cilíndrica regular. O volume de espaço entre duas superfícies normalizadas é utilizado como uma medida de similaridade. O sistema foi testado utilizando imagens faciais de 8 pessoas (3 imagens por pessoa). *Features* foram detectados adequadamente para todas as faces. Taxas de reconhecimento foram de 97% e 100% para *features* individuais e toda a face respectivamente.

Outra abordagem descrita em [95] utiliza perfis (que apresenta contornos externos consistindo em partes bastante rígidas) ao invés de imagens frontais, capturando dados 3D por triangulação, e então realizar uma comparação 3D dos dados do perfil. Esse método requer uma boa cooperação do usuário e restrições no cenário de fundo para funcionar.

[88] utiliza dados 3D para normalizar os resultados obtidos de um algoritmo de detecção facial para formar uma forma mais apropriada para a ferramenta de reconhecimento, i.e., nesse caso, os dados 3D estão sendo utilizados apenas para complementar ao contrário de substituir a detecção facial e os algoritmos de reconhecimento.

Alguns exemplos de reconhecimento facial recentes baseados em dados 3D incluem os seguintes itens: Castellani et al. [96] aproxima as imagens de profundidade das faces obtidas por análise estereoscópica utilizando B-Splines Multi-level [97] e SVMs então são utilizadas para classificar os coeficientes de aproximação resultantes. Algumas outras técnicas [93, 98, 99] primeiro projetam os dados 3D para imagens de intensidade 2D, e posteriormente as imagens 2D projetadas são processadas como imagens de intensidade padrões. Apesar de outros métodos terem sido propostos para reconhecimento facial 3D baseado em *features* locais [100], sinais geométricos locais e globais [101], perfis [102, 103, 104, 105], e a decisão baseada em ranks, fusão de múltiplos classificadores baseados em formas.

Diversas abordagens também propuseram a integração de informações de textura 2D e formas 3D. Estes métodos fazem uso de PCA de imagens de intensidade [106, 107, 108], intensidade de perfis faciais [109], *Iterative Closest Points* (ICP [110]) [111, 112], Gabor wavelets [113], e *Local Feature Analysis* [105], etc. Por exemplo, Wang et al. [113] extraiu *templates* de formas 3D de imagens de profundidade e *templates* de textura de imagens em escala de cinza das faces, aplicando PCA separadamente para ambos os tipos de *template* para reduzi-los para vetores de dimensões menores, concatenando os vetores de textura e forma e, finalmente, utilizando uma SVM nos vetores resultantes para classificação. Em geral, experimentos com estes sistemas indicam que a combinação de informações de textura e forma reduzem taxas de classificação errada do reconhecedor de faces.

3.5.2 Infravermelho

Como imagens térmicas de infravermelho são relativamente pouco sensíveis a variação de iluminação, essas imagens podem então ser utilizadas como uma opção para detecção e reconhecimento de faces. Além disso, [114] argumenta que como imagens faciais de infravermelho revela as veias e a estrutura do tecido da face, que é única para cada indivíduo (assim como digitais), algumas técnicas de reconhecimento facial para o espectro visível deve então fornecer resultados favoráveis quando aplicados a essas imagens. Entretanto, existe uma multidão de fatores que desencorajam a exploração deste tipo de imagem para tarefas de reconhecimento facial, entre esses o custo substancial de sensores térmicos, a baixa resolução e o alto nível de ruído nas imagens, a ausência de conjuntos de dados de imagens infravermelhas, o fato de de radiação infravermelho ser opaca a vidro (sendo possível a oclusão de partes da face utilizando óculos) [115] e , por último mas não menos importante, o fato de imagens infravermelhas serem sensíveis a mudanças de temperatura de ambiente, vento e processos metabólicos no sujeito [116].

Em [114], a técnica da auto-face [46, 47] foi aplicada a uma base de dados de 288 imagens de baixa resolução alinhadas a mão (160x120) de 24 sujeitos, tiradas de 3 angulações diferentes. As seguintes taxas de reconhecimento foram reportadas: 96% para visões frontais, 96% para visões a 45 graus e 100% para visões de perfil.

Wilder et al. [117] comparou o desempenho de três algoritmos de reconhecimento facial em uma base de dados de imagens visíveis e infravermelhas de 101 sujeitos e concluiu que os resultados do reconhecimento de uma modalidade não foram significativamente melhores que os algoritmos da outra modalidade.

Socolinsky et al. [10] testou os algoritmos de autoface [46, 47] e ARENA [118] em uma base de dados de imagens visíveis e infravermelhas de 91 sujeitos distintos (capturadas sob variadas condições de iluminação, com variações de expressões faciais, e com ou sem óculos, utilizando um sensor capaz de obter imagens de ambas as modalidades simultaneamente(Figuras 3.6, 3.7 e 3.8)), e alegaram que as imagens infravermelhas superaram significativamente as imagens visíveis em todos os experimentos de classificação conduzidos sob variações das condições mencionadas acima.

Selinger e Socolinsky [116] usaram a mesma base de dados de 91 sujeitos e testaram a perfor-

mance de quatro algoritmos de reconhecimento (PCA, LDA, LFA e ICA) sob as mesmas condições anteriores. Os resultados a que chegaram foram os mesmos, embora eles tenham admitido que a aparente superioridade da abordagem do infravermelho pode ter sido causada pelo fato de que os dados não continham informações acerca de situações mais rebuscadas (como mudanças de temperatura, ventilação.. etc), ao passo que as visíveis continham.



Figura 3.6: Amostras retiradas da base em [10].



Figura 3.7: Primeiras cinco autofaces visíveis [10].

Chen et al. [119] coletou diversos *datasets* de imagens (de ambos espectros, visível e infravermelho) de 240 sujeitos distintos sob variações de expressões e condições de iluminação em tempos diferentes (algumas imagens foram tiradas na mesma sessão enquanto outras foram tiradas em um período de dez semanas). Foram estudados os efeitos de mudanças na aparência facial ao longo do tempo no desempenho de algoritmos de autofaces em ambas as modalidades e concluiu que, apesar da acurácia de reconhecimento ter sido semelhantes para imagens obtidas na mesma sessão, ima-



Figura 3.8: Primeiras cinco LWIR autofaces [10].

gens visíveis superaram as infravermelhas quando havia uma espaço de tempo significativo entre quando as imagens de treino e de teste foram adquiridas. Eles atribuíram a baixa performance das imagens infravermelhas às variações nos padrões de temperatura num mesmo sujeito e a sensibilidade das imagens infravermelhas à localização manual dos olhos. Eles também descobriram que o software FACEIT [120] obtiveram resultados melhores que as autofaces em ambas modalidades. Entretanto, a combinação de dois classificadores utilizando a regra da soma [121] superaram os classificadores individuais, assim como o FACEIT. Experimentos posteriores conduzidos por Chen et al. [122] em um conjunto de dados maiores confirmaram novamente este resultado.

Outras abordagens têm sido propostas recentemente, onde a fusão de imagens no espectro visível e infravermelho é explorada para aprimorar o desempenho do reconhecimento facial [115, 123, 124, 125].

Capítulo 4

Metodologia

A metodologia desenvolvida para a resolução do problema previamente descrito é dividida em pequenos processos sequenciais desde a preparação da base de dados até o reconhecimento de indivíduos e validação da solução desenvolvida.

Durante toda a definição da solução do problema, evitou-se o uso de análises de características no âmbito de imagens bidimensionais para auxiliar o desempenho e resultados, focando apenas em atividades e características extraídas do composto tridimensional formado por imagem e profundidade, ou em transformações do espaço tridimensional para o bidimensional.

A preparação da base de dados condiz com a segmentação da face dos indivíduos e montagem do espaço tridimensional, representado por nuvens de pontos, além de operações aplicadas em espaços tridimensionais, para extrair características matemáticas para descrição do espaço e, conseqüentemente, do sujeito representado por este. A preparação da base de dados é finalizada após a concepção dos mapas de curvatura de cada indivíduo da base.

O reconhecimento do indivíduo é realizado por meio da comparação entre histogramas de intensidade de níveis de cinza extraídos das imagens montadas sobre o domínio tridimensional durante a preparação da base de dados.

Finalizando a metodologia, a validação do algoritmo é implementada utilizando métodos estatísticos envolvendo a recusa de indivíduos, caso os mesmos aparentemente não estejam presentes na base de dados.

Visando tornar os procedimentos discursados e a sequência dos mesmos mais inteligível, um fluxograma (Figura 4.1) foi desenvolvido, apresentando as entradas de cada processo, sendo esses produtos oriundos da base de dados ou variáveis dinâmicas, que regulam o comportamento e as saídas dos processos.

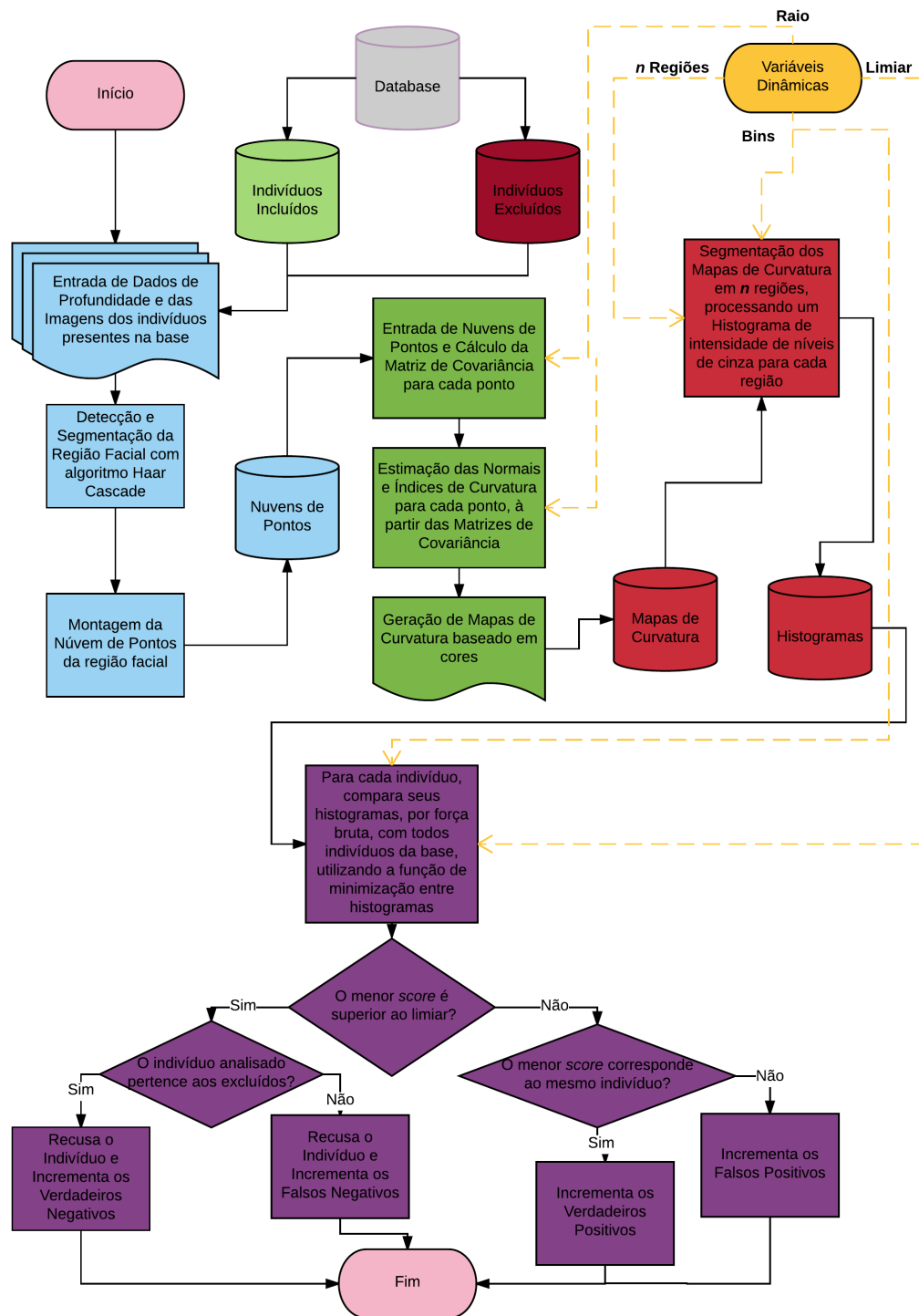


Figura 4.1: Fluxograma da metodologia desenvolvida. Os procedimentos azuis correspondem aos processos de entrada de dados e formação das nuvens de pontos. Os verdes às atividades de extração de informações da nuvem de pontos, os vermelhos à extração de histogramas dos mapas de curvatura, enquanto que os roxos consistem nos processos utilizados para validação.

4.1 Preparação da Base de Dados

Essa seção da metodologia é formada pelos passos iniciais referentes a solução proposta. Essa seção aborda as estruturas de dados utilizadas para armazenar as informações da base de dados, e como essas informações estão primariamente organizadas, de forma que possam ser transformadas em dados e em nuvens de pontos, desejadas para realizar as operações seguintes de reconhecimento em um espaço tridimensional.

A preparação da base de dados segue até à formação dos mapas de curvatura, que posteriormente serão utilizados para realizar o processo de reconhecimento facial biométrico de indivíduos. A região do fluxograma referente a preparação da base de dados pode ser vista na Figura 4.2

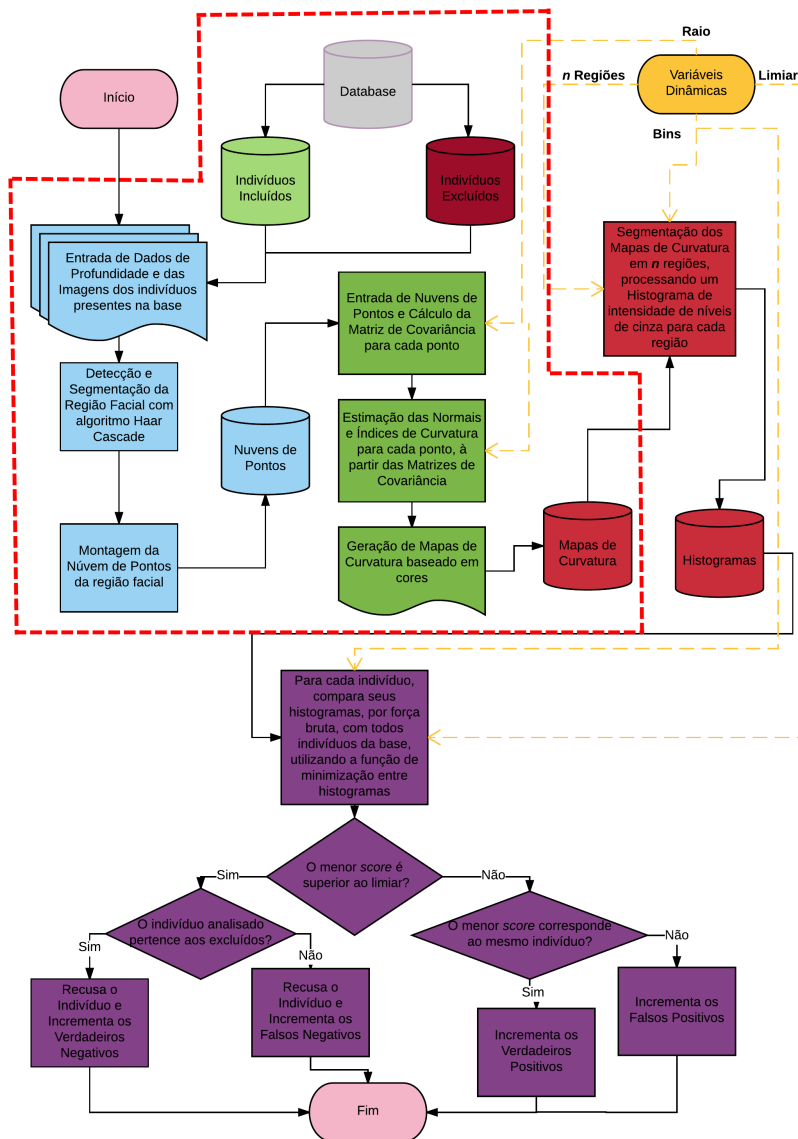


Figura 4.2: Região do fluxograma referente a preparação da base de dados segmentada por linha tracejada vermelha.

4.1.1 Base de Dados

A Base de Dados utilizada durante todo o trabalho foi fornecida por [126], sendo está composta por trinta e uma pessoas, cada uma apresentando treze posicionamentos diferentes para a face e quatro emoções diferentes, e para cada posição e emoção foram obtidas três amostras, permitindo a comparação da pessoa consigo mesma em uma mesma posição. Toda a base de dados foi montada utilizando o dispositivo Kinect [21] da Microsoft, onde os resultados finais foram suavizados pelo próprio autor visando um melhor aproveitamento da base.

As amostras obtidas das posições descritas acima são compostas por uma imagem e informações de profundidade extraídas do Kinect, sendo que essas informações de profundidade estão armazenadas em um arquivo de extensão *dat*, indicando os valores de profundidade do Kinect que variam no intervalo $[-1,4095]$, sendo o -1 representado como distância indefinida, 0 como muito perto e 4095 como muito longe. Os valores representam a distância em milímetros em relação ao sensor. As imagens de profundidade apresentam uma dimensão de 640x480 pixels, apresentando um valor de profundidade para cada pixel, e as imagens capturadas pela câmera apresentam dimensão de 1280x960 pixels, correlacionando a profundidade à imagem por uma razão de dobro.

A entrada dos componentes da base de dados corresponde ao primeiro processo do fluxograma referente a preparação da base de dados visto na Figura 4.4.

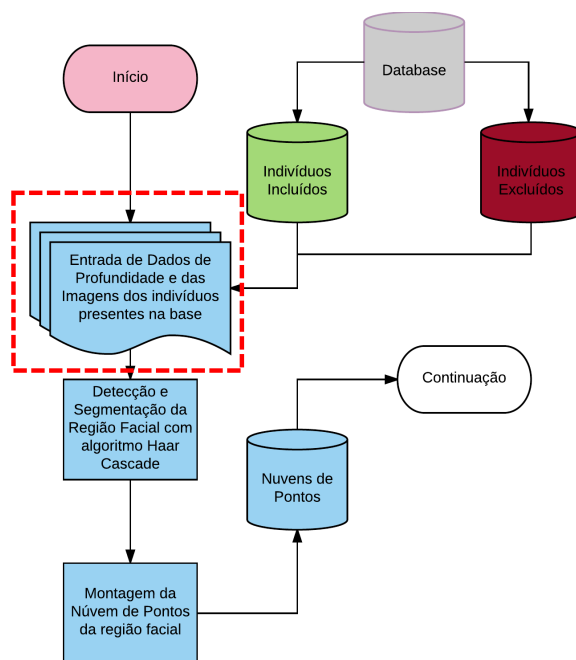


Figura 4.4: Processo de entrada das informações da base de dados no fluxograma.

A nomenclatura original adotada por [126] é definida como $POS_N_T.TTT$, na qual POS define o índice referente a posição, N representa o número da imagem, dado que para cada posição há três (3) imagens, e $T.TTT$ indica se a imagem corresponde a uma imagem de profundidade ou de cor, sendo $d.dat$ o equivalente a profundidade e $c.bmp$ equivalente a cor. Em trabalhos

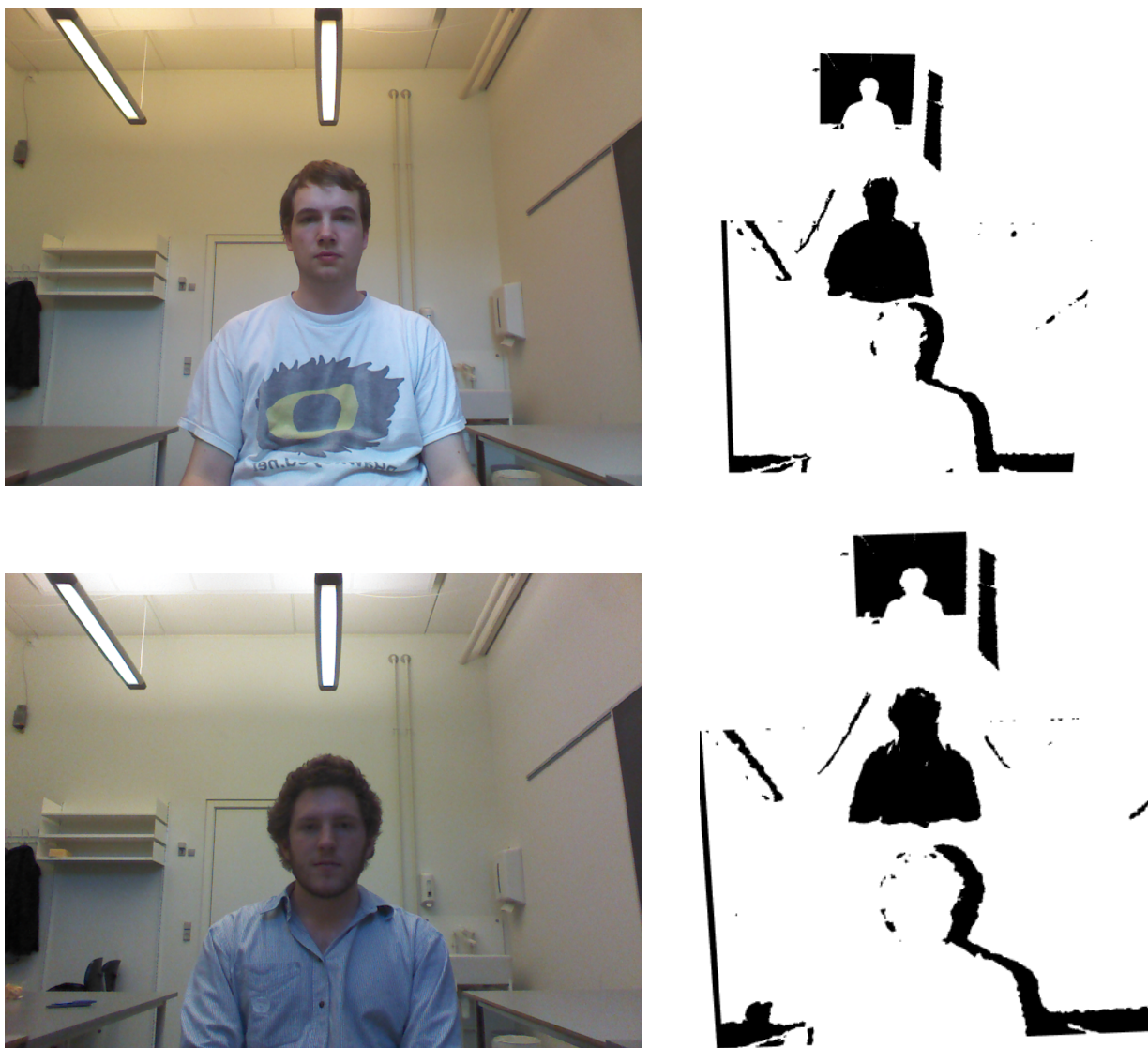


Figura 4.3: Nuvem de Pontos (à direita) e Imagem de Cores (RGB - à esquerda) correspondente.

com sistemas de reconhecimentos, é necessário conhecer com qual sujeito se está trabalhando, adotando uma nova nomenclatura para os arquivos, adicionando um índice a mais para identificar cada sujeito, resultando em $P_POS_N_T.TTT$, onde P corresponde a um índice identificador de cada pessoa da base.

A Figura 4.3 demonstra, um exemplo de visualização de uma nuvem de ponto montada por meio de uma imagem de profundidade, e a sua imagem de cor correspondente, que são os elementos que compõem a base de dados.

Esta base de dados foi desenvolvida para detecção de faces por análises de espaços tridimensionais, entretanto foi utilizada visando reconhecimento facial e biometria, desenvolvendo e utilizando um cenário onde apenas as faces frontais (correspondentes a posição sete da nomenclatura da base de dados) estão incluídas.

4.1.2 Segmentação Facial e Confeção das Nuvens de Pontos

Descrita a formação e composição da base de dados, é possível prosseguir com o processo de confecção do espaço tridimensional em nuvens de pontos. Entretanto a base de dados, por ser voltada para detecção de faces, apresenta outros elementos no cenário (variando entre outras partes do corpo do sujeito a objetos e materiais), necessitando realizar uma segmentação da região facial, para que apenas a face seja tratada durante os demais processos. A relação entre a imagem de profundidade e a imagem de cor é conhecida e o algoritmo Haar Cascade, como descrito em [4], é usado para detectar a região da face na imagem de níveis de cinza (extraída da imagem de cor - Figura 4.6), e em seguida ligar a região obtida com a região referente a imagem de profundidade. Esses processos podem ser vistos no fluxograma na Figura 4.5.

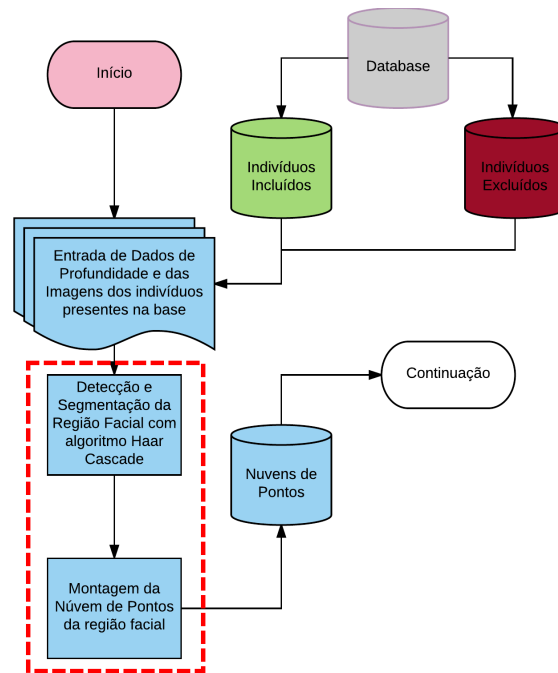


Figura 4.5: Processos de detecção e segmentação no fluxograma de preparação da base de dados.

Descoberta a região facial na imagem de profundidade, é possível prosseguir para a confecção da estrutura de dado (nuvem de ponto) referente a face. As informações de profundidade e pixel são armazenadas na classe e estrutura de dados *PointCloud* da *Point Cloud Library*, utilizando pontos do tipo *PointXYZ*, que armazenam valores de ponto flutuante referentes à localização dos pontos no espaço cartesiano tridimensional, composto pelas coordenadas X , Y e Z . A imagem de profundidade é constituída por uma matriz de 640×480 pixels, como mencionado anteriormente, e cada pixel representa um ponto, portanto os pontos apresentam três coordenadas possíveis, justificando o uso dos pontos do tipo *PointXYZ*, sendo X o número da coluna do pixel que o representa, Y o número da linha do pixel que o representa, e Z o valor de profundidade armazenado nesse pixel.

A nuvem de pontos é montada ponto à ponto, ou seja, uma *stream* lê o valor de profundidade de um ponto e armazena, em variáveis locais, os números da sua coluna e linha, atribuindo em

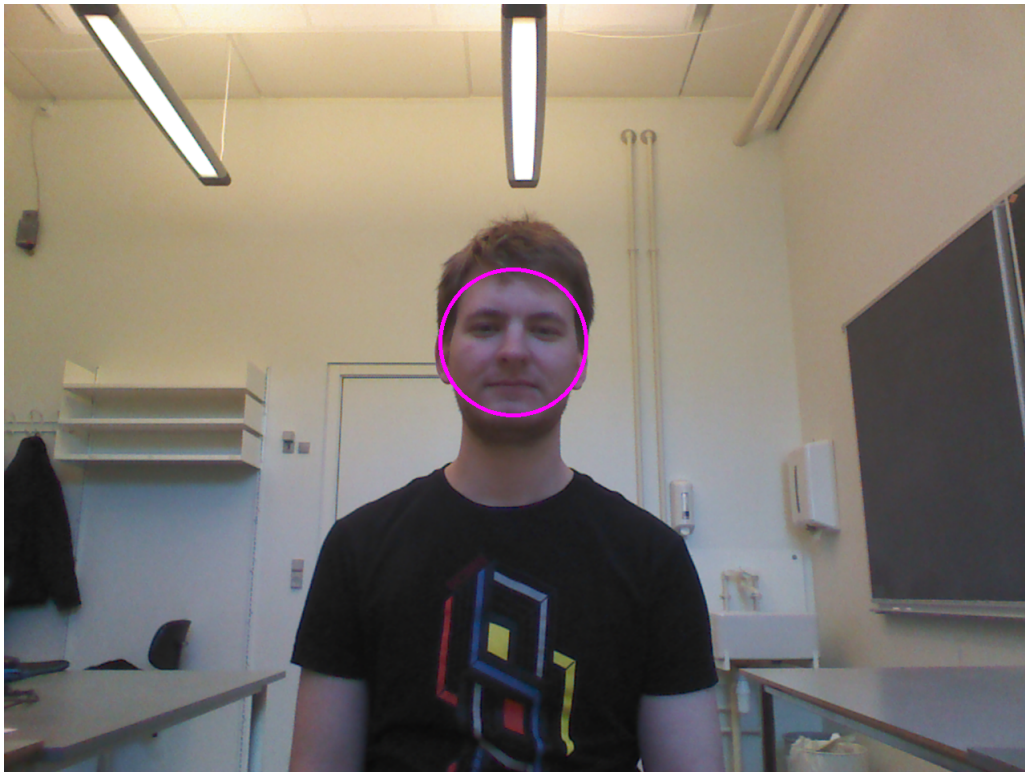


Figura 4.6: Segmentação Facial por Haar Cascade.

seguida esses valores às respectivas variáveis do *PointXYZ*, gerando um ponto e anexando-o a nuvem. O processo é repetido até que todos os pontos tenham sido integrados a nuvem. Os pontos correspondentes a erros de leitura, muito perto (leitura 0 do Kinect) e muito longe (leitura 4095 do Kinect) são descartados na construção da nuvem de pontos, por serem desnecessários para os procedimentos adiantes, entretanto estes causam omissões de pontos que comprometem a qualidade das nuvens de pontos.

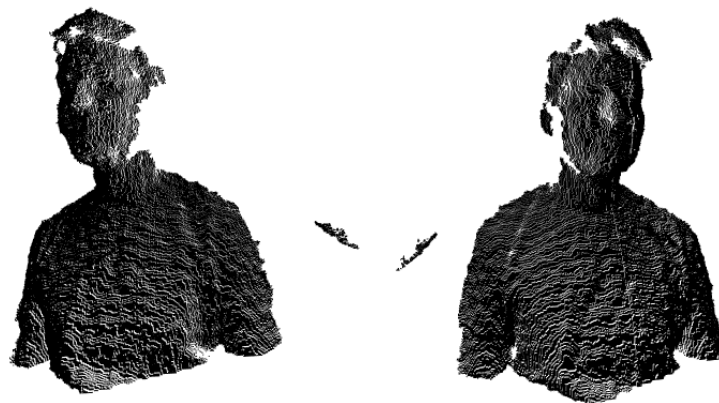


Figura 4.7: Nuvem de pontos filtrada antes do Haar Cascade em múltiplas vistas.

Ao construir completamente o objeto que define a nuvem, este é armazenado em outro arquivo, apresentando a extensão *pcd* - (*Point Cloud Data*), para que a nuvem resultante possa ser observada, por meio de outras classes da PCL, e para que seja carregada com mais facilidade em outros módulos

do projeto.

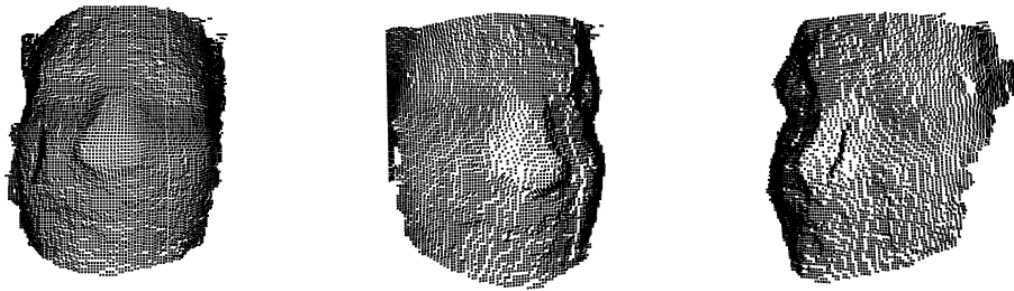


Figura 4.8: Nuvem de Pontos da região facial em múltiplas vistas. Essa é a nuvem de pontos resultante após o processo de segmentação da Figura 4.7.

Um módulo de visualização das nuvens foi desenvolvido, utilizando a classe *visualization* da PCL, para se averiguar visualmente a corretude das nuvens de pontos geradas. As nuvens de pontos originais e resultantes da segmentação e suas visualizações podem ser constatadas nas Figura 4.7 e 4.8 que se referem, respectivamente, a nuvem de ponto antes do Haar Cascade e depois.

4.1.3 Matriz de Covariância da Vizinhança dos Pontos

A análise desenvolvida sobre o espaço tridimensional é direcionada às características geométricas das faces das pessoas, aspirando comparar os sujeitos por meio das características faciais, como o formato do nariz, a espessura do rosto, a forma do cabelo, entre outros aspectos que se destacam de maneira geométrica.

Para definir as propriedades geométricas do espaço, é utilizada uma técnica de estudo e exploração da vizinhança de cada ponto presente neste espaço, e para tal é preciso definir essa vizinhança.

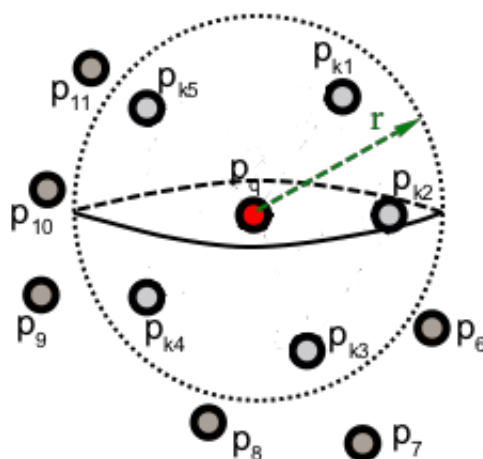


Figura 4.9: Vizinhança ao redor de um ponto.

A vizinhança, ou entorno de um ponto, é definida por uma esfera cujo centro é o próprio ponto, composta pelos pontos que sejam internos ao volume dessa esfera. O estudo da vizinhança é realizado por meio da confecção de uma matriz de covariância (referente ao processo do fluxograma da Figura 4.10), que faz uso das componentes de posição dos pontos para definir um centróide e os valores internos a matriz. Os pontos são considerados vetores no espaço que podem ser representado no formato (X, Y, Z) .

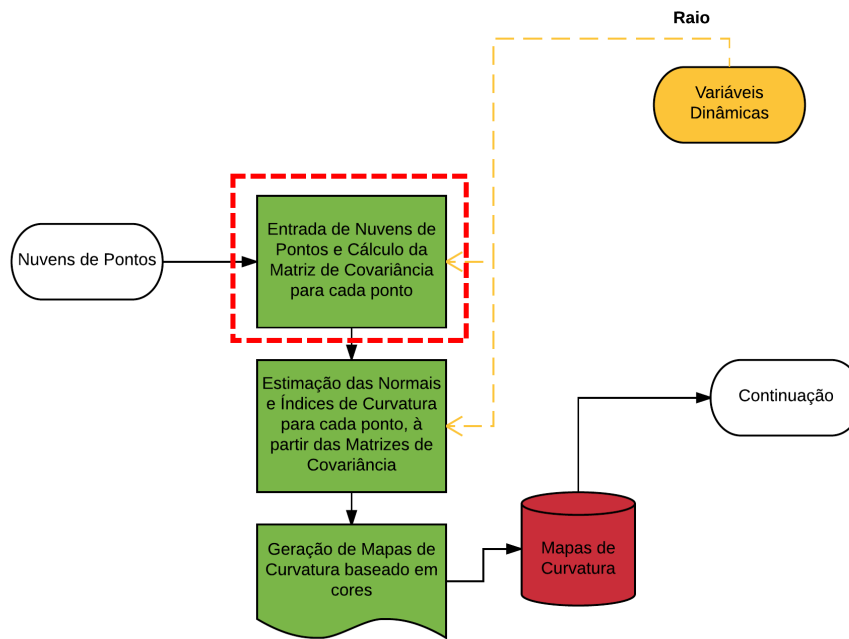


Figura 4.10: Processo de formação da matriz de covariância no fluxograma de preparação da base de dados.

A matriz de covariância de um ponto é expressa matematicamente por meio da Equação 4.1, obtida pela média das matrizes de covariância (de cada ponto) dentro da vizinhança desse ponto (calculadas à partir da multiplicação do vetor diferença - entre o vetor posição de um ponto e do centroide da região - e o seu transposto), representando a distribuição dos pontos na vizinhança para estimar a superfície formada.

$$M = \frac{1}{x} \sum_{i=1}^x (p_i - \bar{p}) \cdot (p_i - \bar{p})^T, \quad (4.1)$$

onde:

M é a matriz de covariância;

x é o número de pontos presentes na vizinhança

\bar{p} representa vetor posição correspondente ao centroide para a região definida.

p_i representa o vetor posição de um ponto presente na vizinhança.

A matriz tem crítica importância para a caracterização das feições geométricas das pessoas, dado que por meio de seus autovalores é possível estimar componentes espaciais que divergem de acordo com a formação do espaço. Os autovalores da matriz são definidos como descrito na Equação 4.2.

$$M \cdot \vec{v}_j = \lambda_j \cdot \vec{v}_j, \quad j \in \{0, 1, 2\} \quad (4.2)$$

sendo:

\vec{v}_j os autovetores da matriz de covariância M.

λ_j os autovalores da matriz de covariância M.

j se refere ao índice do autovalor/autovetor. As matrizes apresentam três autovalores/autovetores, dado que o espaço é tridimensional (logo a matriz é 3x3), portanto os valores que j pode assumir é definido pelo conjunto 0,1,2, atribuindo o início do vetor de autovalores/autovetores ao índice 0.

Os autovalores serão utilizados para prosseguir com o processo de preparação da base, estimando-se as normais e índices de curvatura para obter o mapa de curvatura.

4.1.4 Estimação de Normais e Índices de Curvatura

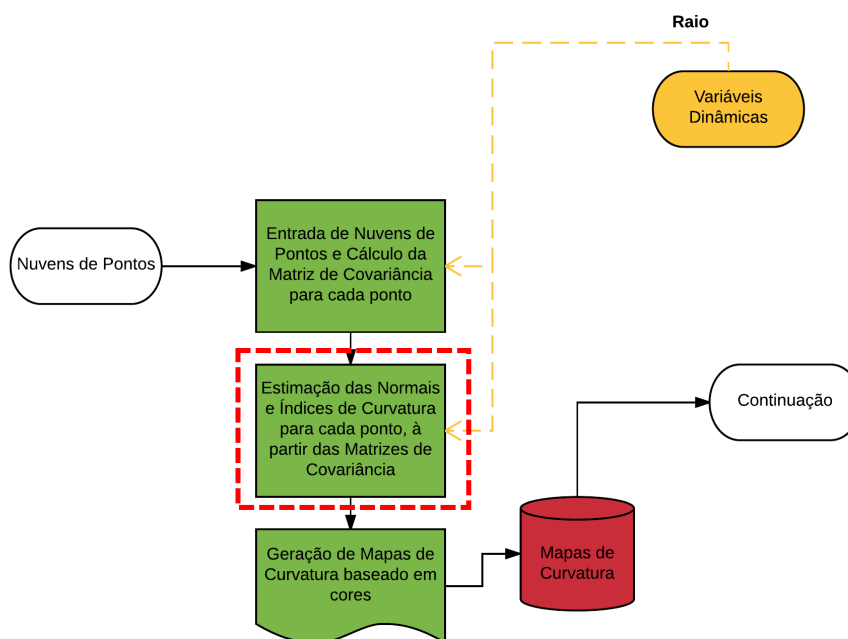


Figura 4.11: Processo de Estimação de Normais e Curvaturas no fluxograma.

As normais de uma superfície são propriedades importantes para a geometria da superfície. O método adotado para se calcular as normais é semelhante a estimação da mesma em um plano

tangente a superfície, transformando-se em um problema de estimação de encaixe de planos por mínimos quadrados, como descrito em [127].

Os índices de curvatura da superfície aqui mencionados, são representados por uma estimação modelada como uma correlação entre os autovalores da matriz de covariância, cuja característica é informar a distribuição dos pontos da região de vizinhança. A expressão que define o cálculo do índice de curvatura é representado pela Equação 4.3:

$$\sigma = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \quad (4.3)$$

em que σ é o índice de curvatura daquele ponto e os λ 's representam os três autovalores da sua matriz de covariância. O processo de estimação de normais e curvaturas pode ser visto no fluxograma na Figura 4.11.

As normais e curvaturas estimadas são armazenadas em uma *PointCloud* de pontos do tipo *PointNormal*, que armazenam o valor das componentes normais em cada coordenada daquele ponto e a curvatura.

É importante ressaltar que a variação do raio que define a vizinhança de cada ponto, impacta diretamente nos valores das normais e índices de curvatura, portanto o estudo de variação do raio é importante na obtenção de melhores resultados nos procedimentos de reconhecimento (ver Figura 4.13).

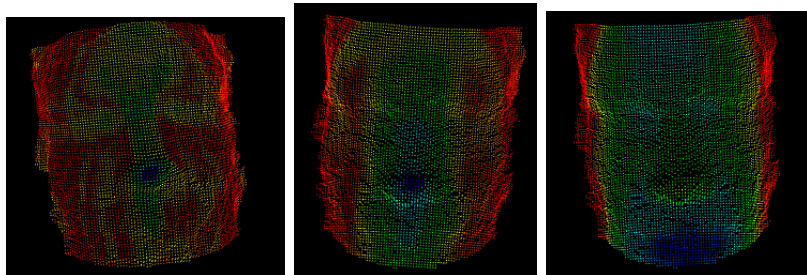
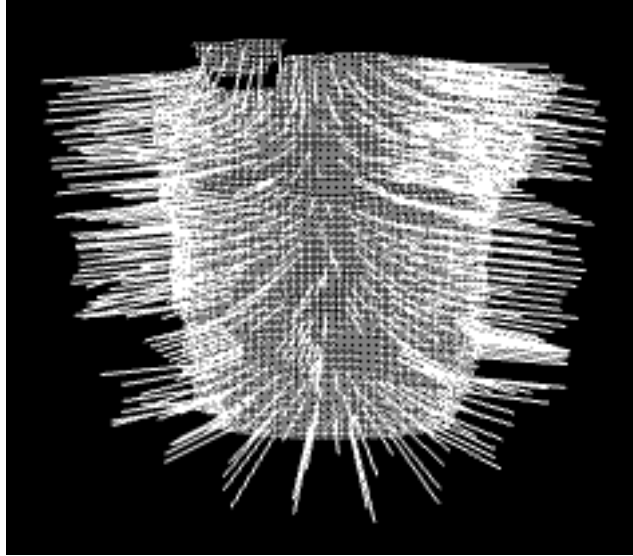


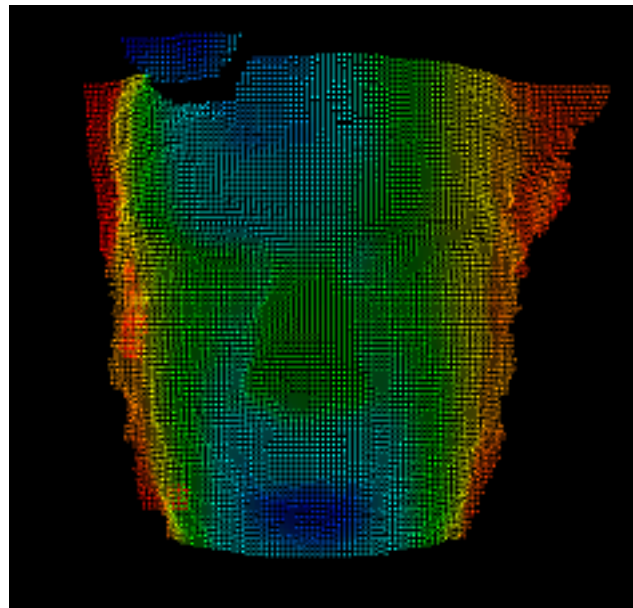
Figura 4.13: Variação da curvatura de acordo com o raio.

4.1.5 Extração dos Mapas de Curvatura

Os mapas de curvatura resumem-se em uma transformação do espaço retratado pela nuvem de pontos, para imagens bidimensionais cuja intensidade de cada pixel é representada pelos valores dos índices de curvatura. A intensidade dos índices de curvatura encontram-se em um intervalo retratado entre 0 (zero) e 1 (um), em que zero representa uma região plana e um representa uma região esférica. O processo referente a extração dos mapas de curvatura no fluxograma pode ser visto na Figura 4.14.



(a) Normais



(b) Curvatura

Figura 4.12: Normais e Curvatura.

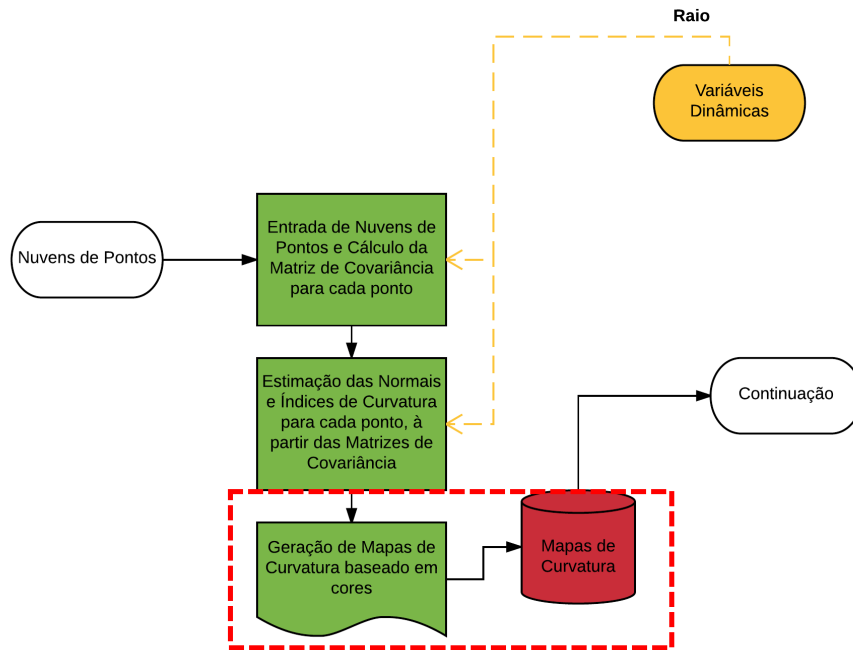


Figura 4.14: Processos referentes a extração do mapa de curvatura no fluxograma de preparação da base de dados.

A ideia dos mapas de curvatura surgiu da análise da classe visualizadora da PCL, em que os índices de curvatura são representados por cores, e para cada ponto é atribuída uma cor, que é apresentada em um tela como representado na Figura 4.12.

A PCL utiliza um *data array* da biblioteca VTK para armazenar os valores dos índices de curvatura de cada ponto das nuvens, e por meio desse *data array* é possível criar uma tabela de visualização conhecida como *LookUpTable*, que normaliza os valores de curvatura em relação ao máximo e mínimo, e retorna valores de intensidade de cor para os espectros vermelho, verde e azul (RGB).

Através dessa função de retorno de intensidade de cor é possível, de um ponto de vista perfeitamente frontal a face, planificar a nuvem de pontos de curvatura e obter como resultado um mapa de curvatura, como o mostrado na Figura 4.15, por meio da atribuição dos valores a cada pixel de uma imagem criada com as mesmas dimensões da nuvem de pontos.

Posteriormente os mapas de curvatura são convertidos para imagens em nível de cinza, utilizadas para a extração de histogramas e realização do reconhecimento facial.

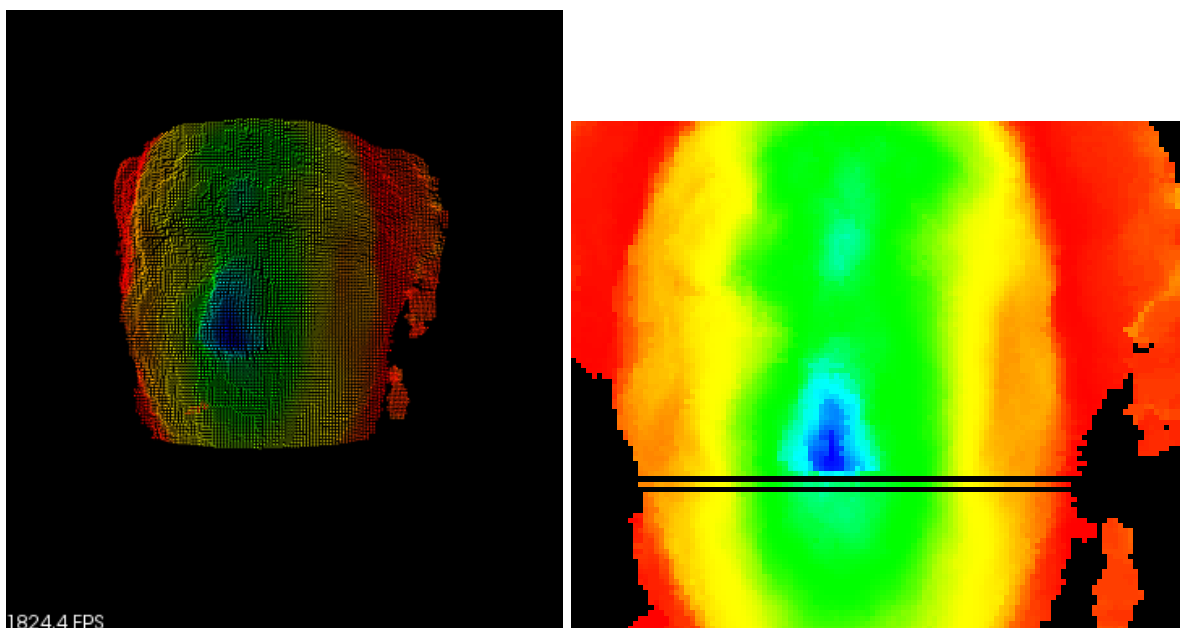


Figura 4.15: Transposição dos índices de curvatura em Imagem de intensidade.

Finalizando a criação dos mapas de curvatura, é possível dar procedência ao processo de reconhecimento, concretizando comparações entre os sujeitos da base de dados e sujeitos de entrada no sistema, por meio de técnicas de inspeção de imagens.

4.2 Reconhecimento Facial

Essa seção da metodologia engloba os procedimentos relacionados a comparação entre os indivíduos da base de dados (Figura 4.17), por meio da extração de recursos dos mapas de curvatura e formação de histogramas de nível de cinza por meios desses (correspondendo a região segmentada do fluxograma na Figura 4.16). Os histogramas são divididos por um número de *bins*, que representam um intervalo da escala de níveis de cinza. O tamanho desse intervalo varia de acordo com o número de *bins*.

4.2.1 Divisão do Mapa de Curvatura

Ao fim do processo de preparação da base de dados, são gerados mapas de curvatura para cada nuvem de pontos pertencente aos membros da base, e por meio desses mapas deseja-se, para qualquer mapa utilizado como entrada do sistema de reconhecimento, comparar a entrada com todos os outros mapas por força bruta e avaliar se o sujeito mais semelhante é de fato o mesmo sujeito que o da entrada, ou mesmo se esse sujeito de entrada se encontra presente na base de dados.

Objetivando evitar que uma região da face de uma entidade se assemelhe a outra região da face de outra entidade, o mapa de curvatura é dividido em diferentes regiões, cujo número é fornecido pelo usuário por meio da variável dinâmica n .

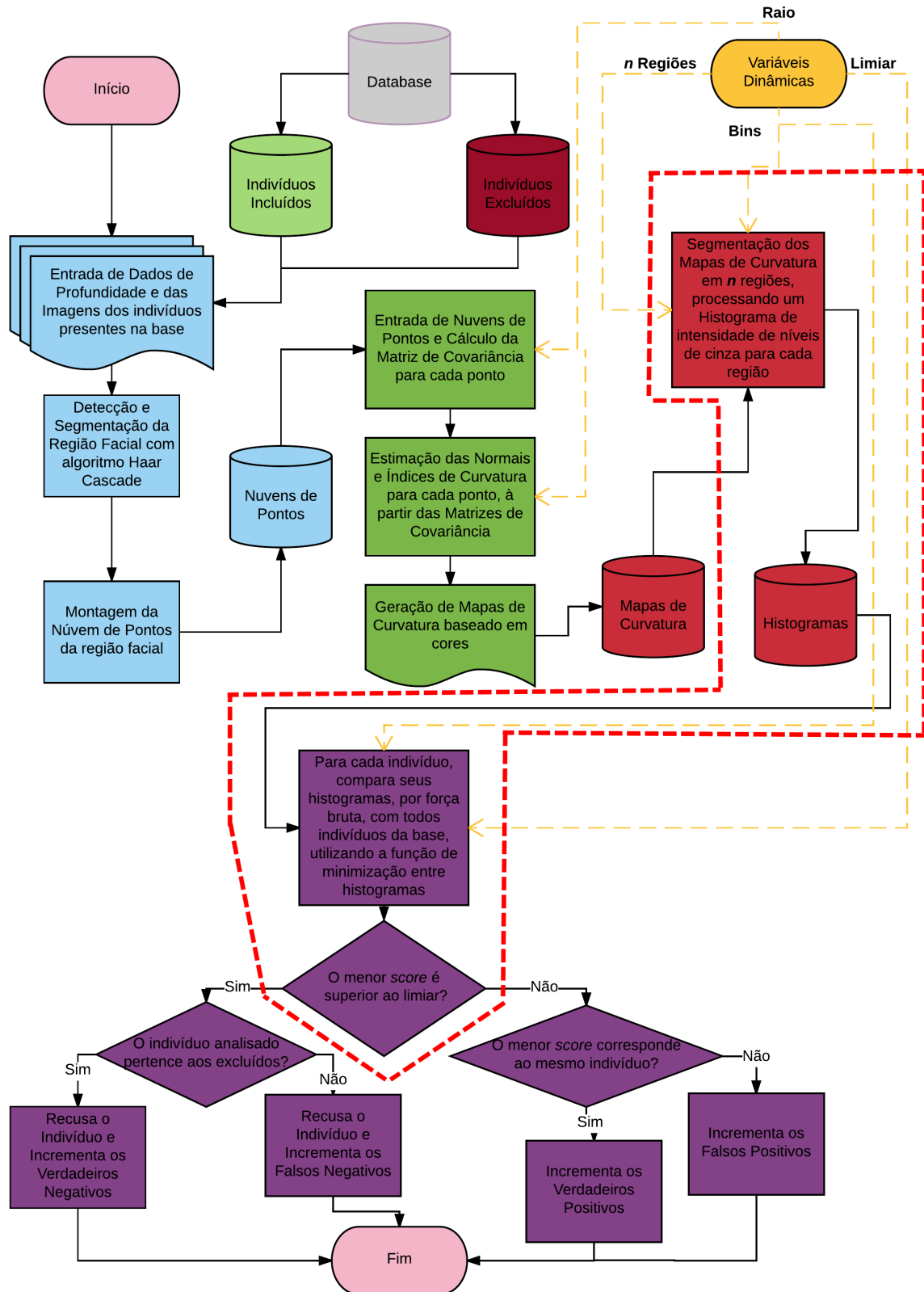


Figura 4.16: Região do fluxograma referente ao reconhecimento facial segmentado por linha tracejada vermelha.

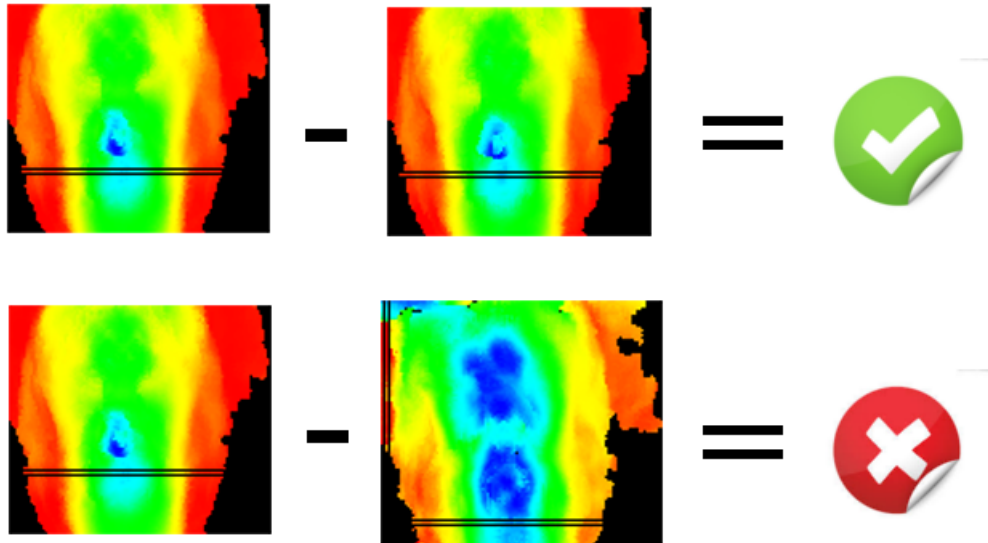


Figura 4.17: Reconhecimento facial por comparação.

As divisões propostas e avaliadas durante os estudos foram de três, seis e nove regiões. A divisão em três evidencia três secções horizontais, enquanto a divisão em seis estabelece uma repartição de três por dois (3x2), e a divisão em nove estipula uma divisão uniforme de três por três (3x3). Essas divisões são evidenciadas, respectivamente, pelas Figuras 4.18, 4.19 e 4.20

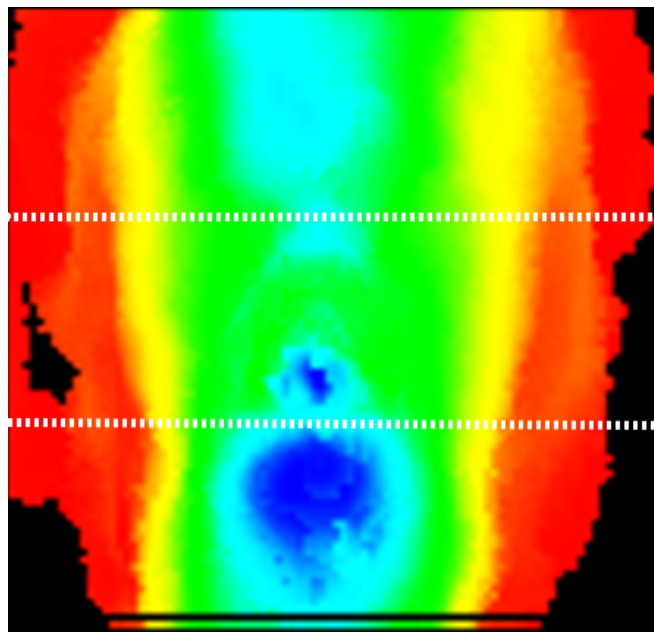


Figura 4.18: Divisão de mapa de curvatura em três regiões.

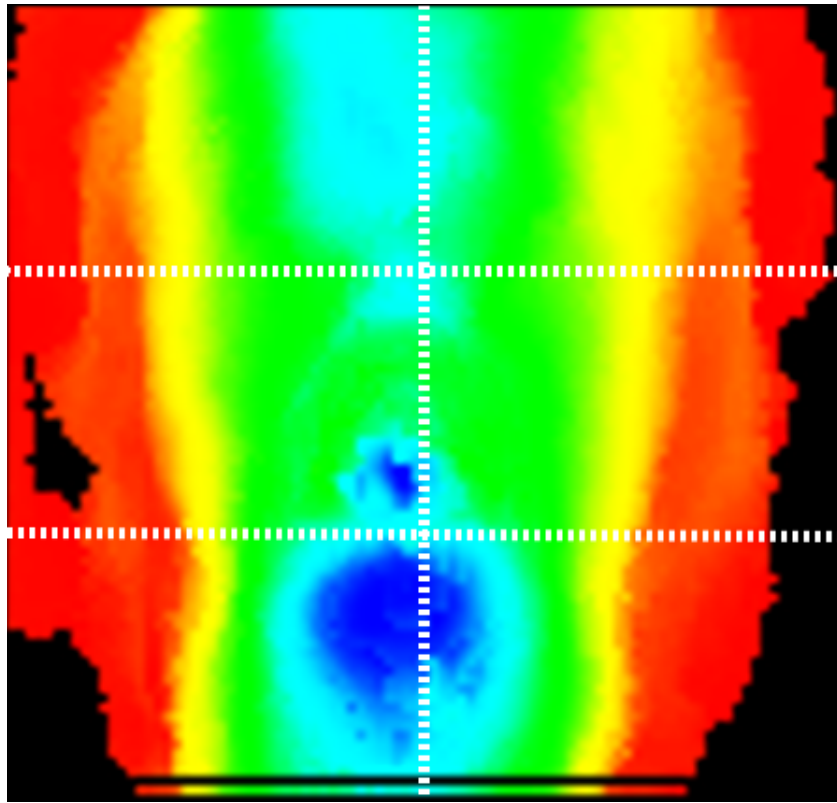


Figura 4.19: Divisão de mapa de curvatura em seis regiões.

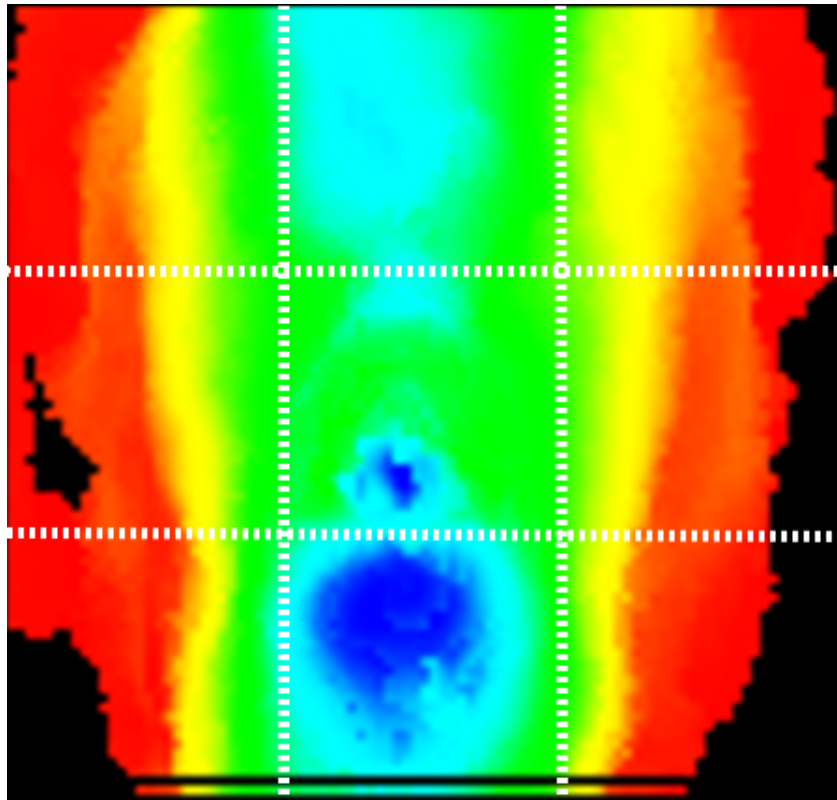


Figura 4.20: Divisão de mapa de curvatura em nove regiões.

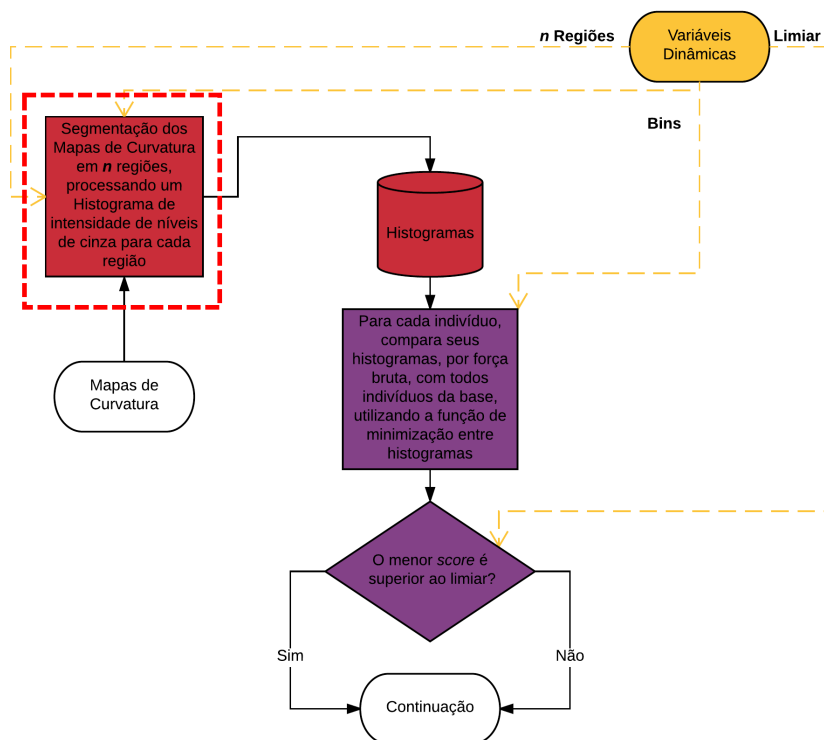


Figura 4.21: Processo de divisão do mapa de curvatura e construção dos histogramas segmentado no fluxograma referente ao reconhecimento facial.

4.2.2 Construção dos Histogramas

Os histogramas são montados sobre os níveis de cinza dos pixels dos mapas de curvatura (Figura 4.22), onde cada *bin* é correlacionado a um intervalo de nível de cinza, que varia de 0 a 255, e portanto os histogramas contabilizam a frequência de ocorrência dos níveis de cinza em diferentes intervalos. Para fim de equalização da comparação entre os histogramas, após a varredura de todos os pixels da imagem e concretização da construção do histograma, ocorre uma normalização do histograma, dividindo a frequência de cada *bin* pelo número total de pixels. O valor zero, referente à cor preta, foi desprezado durante a confecção dos histogramas. Todo processo desde a divisão do mapa de curvatura até a construção dos histogramas pode ser vista no fluxograma de reconhecimento facial na Figura 4.21.

4.2.3 Comparação entre Histogramas e Scores

Formados os histogramas de todos os membros da base de dados e do sujeito de entrada, inicia-se o processo de comparação visando reconhecer o sujeito de entrada como um dos membros da base de dados. Essa comparação é feita por meio do cálculo de um *score*, em que o histograma do sujeito de entrada é contraposto com todos os histogramas referentes a base de dados mediante uma função de minimização entre as frequências normalizadas dos histogramas. A função de

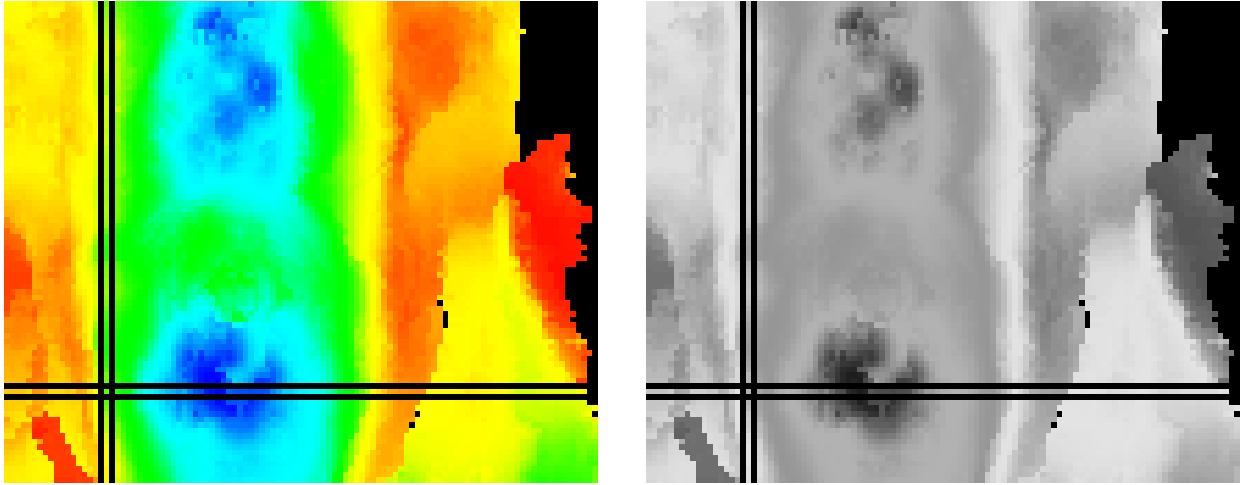


Figura 4.22: Conversão para níveis de cinza.

minimização é descrita de acordo com a Equação 4.4.

$$score = \min \sum_{m=1}^n \sqrt{\sum_{k=1}^{bins} (H_e^k[m] - H_{(i,j)}^k[m])^2} \quad (4.4)$$

em que:

H_e refere-se aos histogramas do sujeito de entrada;

$H_{(i,j)}$ refere-se aos histogramas do sujeito i da base de dados e da imagem j desse sujeito, lembrando que cada sujeito tem 3 imagens;

k representa os *bins* dos histogramas;

bins número de bins dos histogramas;

m representa o número do histograma, dado que há um histograma pra cada uma das n regiões;

e

n representa o número de regiões em que o mapa de curvatura foi dividido.

A Figura 4.23 apresenta exemplos de histogramas extraídos de membros da base de dados (sendo a Figura 4.23a referente a histogramas de 8 *bins* e 3 regiões, e a Figura 4.23b referente a histogramas de 16 *bins* e 3 regiões).

4.2.4 Critérios de Rejeição e Aceitação

Para verificar se um sujeito de entrada é rejeitado ou reconhecido como um dos sujeitos da base de dados, foi utilizado um limiar variado dinamicamente, visando rejeitar o mínimo de membros que pertençam a base de dados e ao máximo pessoas que não estão presentes nela. O princípio de rejeição se baseia nos *scores* calculados para o indivíduo de entrada, em que caso o menor *score* obtido por este seja maior que o limiar, o indivíduo deve ser rejeitado, e caso contrário, aceite e identificado como um membro da base. O limiar ideal é obtido observando os valores dos

scores mínimos de cada membro da base com outros membros que não seja o mesmo. O limiar é dinâmico, pois ele muda de acordo com os valores de *score* obtidos da função de minimização, visando estabelecer melhores resultados para o reconhecimento facial.

Para cada sujeito de entrada não rejeitado, o sistema gera uma saída que corresponde a um membro da base de dados com o qual apresentou maior semelhança. A Figura 4.24 mostra o processo de reconhecimento facial utilizando a função de minimização e sua comparação ao limiar no fluxograma.

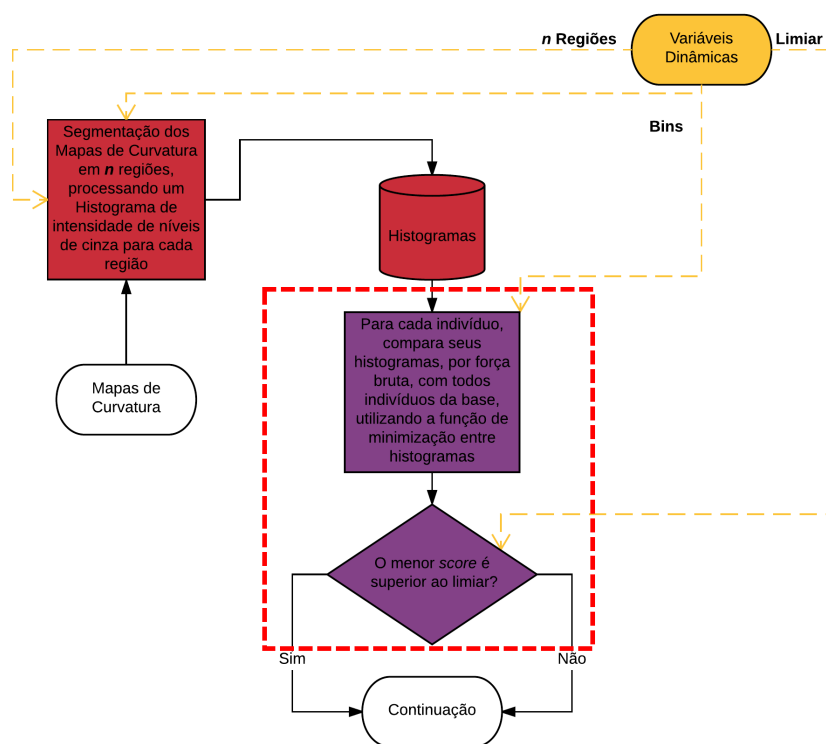


Figura 4.24: Processo de reconhecimento facial (função de minimização e avaliação por limiar) no fluxograma.

4.3 Validação do Algoritmo

O processo de validação do algoritmo tem por objetivo mensurar a qualidade e eficiência do método de reconhecimento facial proposto. Essa mensuração é obtida por meio do método estatístico F-Measure, avaliando todos os reconhecimentos e rejeições, realizados corretamente e erroneamente.

A validação é realizada utilizando os processos de reconhecimento facial descritos previamente na metodologia em todos os indivíduos da base de dados. A validação pode ser visualizada segmentada no fluxograma na Figura 4.25.

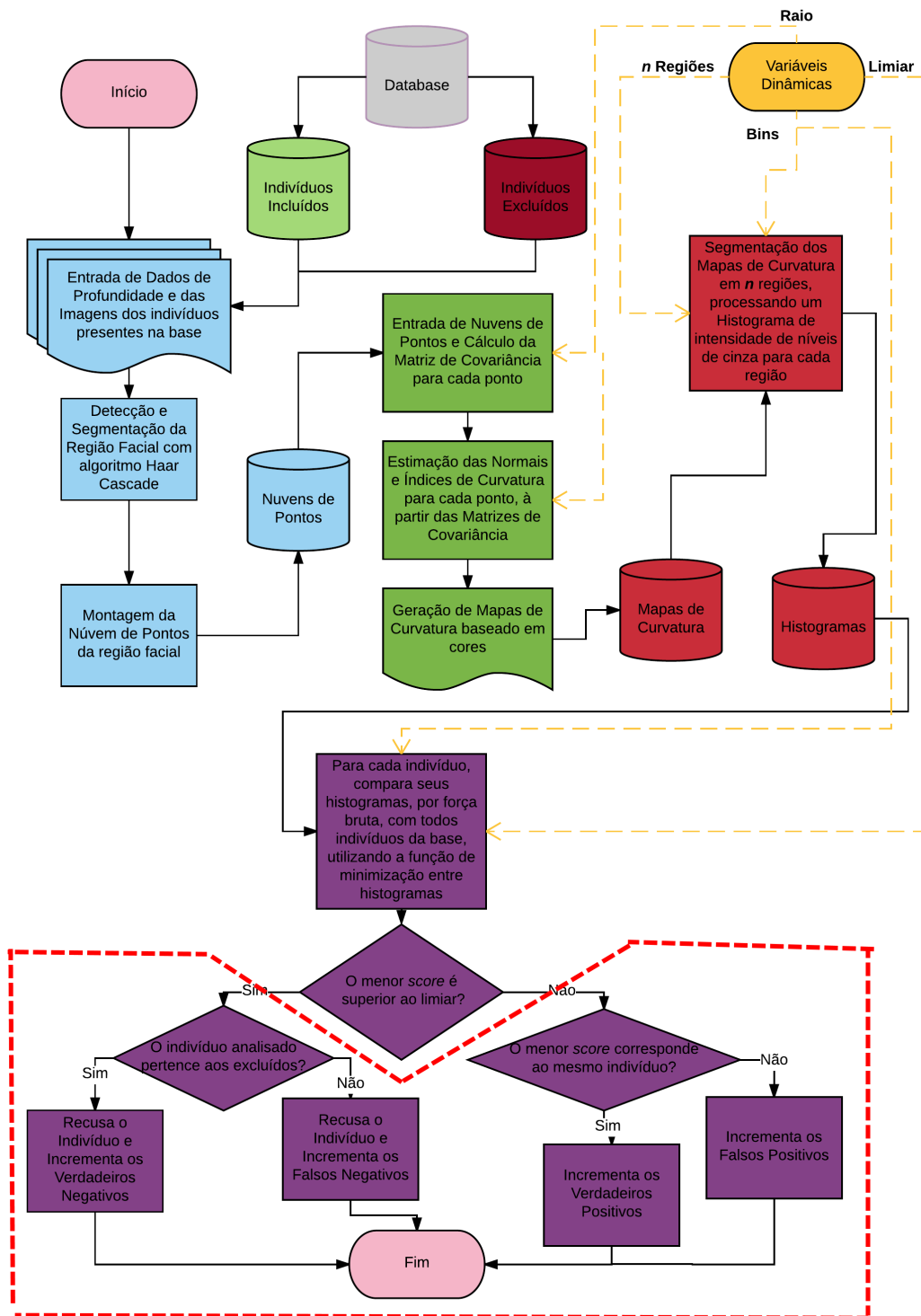


Figura 4.25: Região de validação segmentada no fluxograma por linha tracejada vermelha.

4.3.1 F-Measure

O F-Measure é a medida utilizada para avaliar a qualidade do sistema, e o principal meio de verificar o seu comportamento ao variar as variáveis previamente descritas. O F-Measure se baseia

na relação entre o número de acertos e erros do sistema, sendo esses erros por reconhecimento errôneo ou por recusar um indivíduo que não deveria ser recusado. Os acertos e erros são computados pelo sistema por verificação dos índices de identificação (descritos na nomenclatura) dos sujeitos da base de dados.

A validação do processo de reconhecimento se inicia pelo uso da *F-Measure*, estipulando critérios de verdade e falsidade, e positividade e negatividade.

Os critérios para gerar a avaliação do processo de reconhecimento seguem as seguintes definições:

- Verdadeiro Positivo (True Positive) - Sujeito de entrada não é rejeitado e o reconhecimento é realizado de maneira correta (sujeito de maior semelhança é igual ao sujeito de entrada);
- Falso Positivo (False Positive) - Sujeito de entrada não é rejeitado e o reconhecimento é realizado de maneira errônea (sujeito de maior semelhança não é igual ao sujeito de entrada);
- Verdadeiro Negativo (True Negative) - Sujeito de entrada é rejeitado e não está presente na base de dados de referência;
- Falso Negativo (False Negative) - Sujeito de entrada é rejeitado e está presente na base de dados de referência.

O levantamento dos dados necessários para a validação é realizado por força bruta do sistema, em que todos os indivíduos da base de dados são utilizados como entrada do sistema, e 15% da base de dados é excluída da base de referência para poder testar a recusa, permanecendo ainda no conjunto de imagens candidatas. Os indivíduos excluídos são selecionados aleatoriamente a cada execução da validação, variando portanto os indivíduos a serem recusados e obtendo possíveis resultados diferentes para a mesma configuração das variáveis do sistema. Devido a este fator, foi realizada uma média dos parâmetros do *F-Measure* obtidos em 10 execuções com as mesmas configurações.

Com os dados obtidos, é possível realizar o cálculo do *F-Measure*, como demonstrado nas expressões 4.5, 4.6 e 4.7.

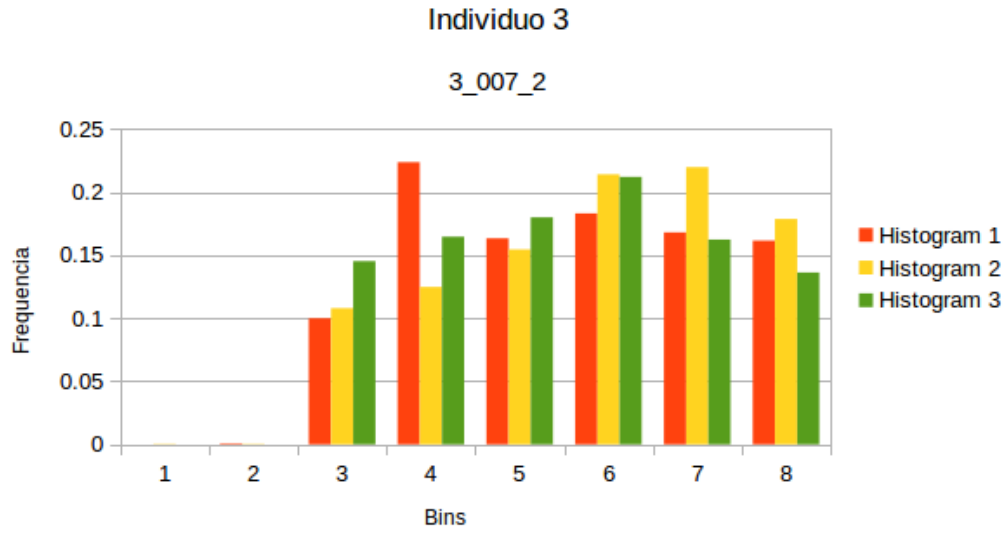
Utilizando os critérios da metodologia declarados para o *F1 Score* é possível prosseguir aos cálculos do *Precision* e *Recall*, representados pelas Equações 4.5 e 4.6, respectivamente. No caso do sistema de reconhecimento desenvolvido, o *Precision* demonstra a capacidade do sistema em relação ao reconhecimento, relacionando os verdadeiros positivos e falsos positivos, enquanto o *Recall* demonstra a capacidade do sistema em relação à recusa, relacionando os verdadeiros positivos e falsos negativos.

$$Precision = \frac{VerdadeiroPositivo}{VerdadeiroPositivo + FalsoPositivo} \quad (4.5)$$

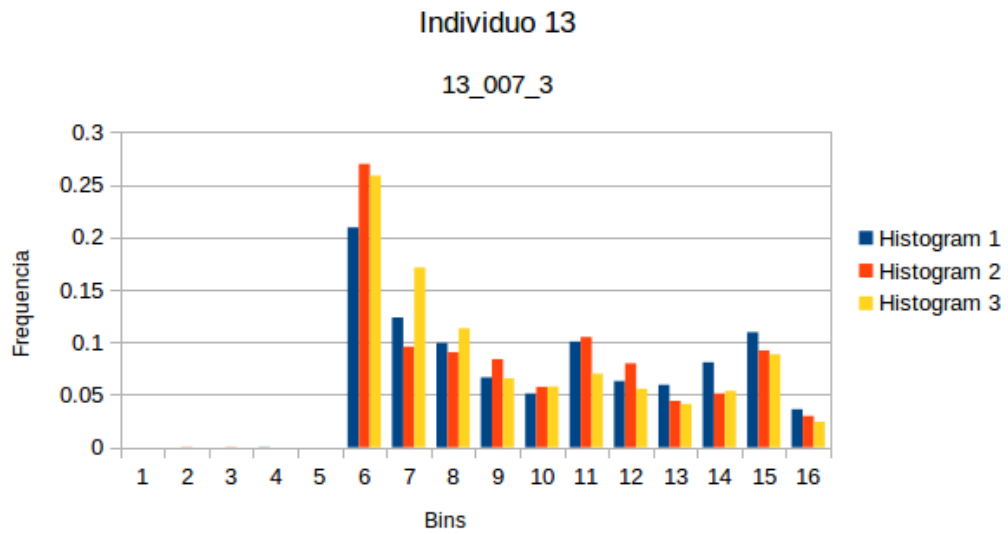
$$Recall = \frac{VerdadeiroPositivo}{VerdadeiroPositivo + FalsoNegativo} \quad (4.6)$$

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4.7)$$

Por fim, o método proposto realiza um processo de reconhecimento facial, preparando e extraindo características de uma base de dados, e provê uma medida de qualidade que descreve a capacidade do sistema para realizar reconhecimento facial.



(a) Histograma de 8 bins para divisão em 3 regiões



(b) Histograma de 16 bins para divisão em 3 regiões

Figura 4.23: Histogramas dos Mapas de Curvatura.

Capítulo 5

Resultados

Este capítulo abrange o comportamento do sistema desenvolvido para diferentes configurações iniciais, representadas pelos valores das variáveis dinâmicas do sistema. As variáveis dinâmicas são responsáveis por controlar as características das saídas de processos definidos na metodologia.

Os resultados são expostos por meio de tabelas e figuras, expondo o comportamento do sistema na coleta de dados em diferentes cenários de teste, onde diferentes valores de entrada são dados às variáveis dinâmicas (parâmetros do método).

O raio (r) define a vizinhança em torno de um ponto, que por sua vez influencia nos valores estimados de normal e índice curvatura, que por consequência impacta no mapa de curvatura. O valor do raio, portanto, altera as características do mapa de curvatura e, em consequência disso, os valores e saídas dos processos conseguintes.

A variável *bins* é responsável pela delimitação do número de intervalos em que os níveis de cinza são divididos, alterando a estrutura dos histogramas e os resultados de comparação.

A variável n apresenta a função de definir o número de divisões realizadas nos mapas de curvatura, e portanto o número de histogramas que cada mapa de curvatura possui, alterando os *scores* de comparação entre diferentes indivíduos.

A última variável consiste no limiar, responsável por estabelecer um teto de *score* que classifica se um indivíduo deve ser rejeitado ou aceito pelo sistema, alterando os valores de F-Measure no final.

Em todos os casos foram realizadas mais de uma medida e tomada a média dessas medidas, devido a variação do comportamento do sistema de acordo com os indivíduos aleatoriamente excluídos da base de referência, aproximando-se a um modelo *leave-one-out* de verificação cruzada.

5.1 Materiais Utilizados

A implementação do sistema foi desenvolvida em linguagem de programação C++, fazendo-se uso das seguintes bibliotecas:

- Point Cloud Library (PCL)
- OpenCV

Todos os procedimentos foram realizados em um computador que apresenta um processador Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz e uma memória RAM de 4GB, sendo todos executados no sistema operacional Linux Mint Cinnamon (64-bit).

5.2 Cenários de Teste

A avaliação do algoritmo se dá por meio da definição de cenários de teste, onde uma das variáveis dinâmicas é testada, ou diferentes composições das possibilidades de combinação de entrada, analisando o impacto que isso gera na aceitação, recusa e conjunto como um todo por meio da F-Measure.

5.2.1 Cenário 1: Qualidade em Função do Raio

Esse cenário aborda o impacto do raio no comportamento do sistema, avaliando as alterações sofridas pelas métricas do *F-Measure* a medida que o raio varia, e consequentemente, o mapa de curvatura varia. O limiar é variado visando o melhor caso para cada raio, dado que o foco desse cenário é avaliar o desempenho com a variação do raio. O número de *bins* e das **n** regiões da face foram permanecidos constantes, com os valores de 8 *bins* e 9 divisões.

Tabela 5.1: Tabela de Critérios de F-Measure do Cenário 1

raio	TP	FP	TN	FN
5	27	9	9	48
10	40	3	12	38
15	46	4	11	32
20	52	4	11	26
25	52	3	11	27
30	60	1	11	21
35	61	0	10	22
40	62	0	12	19
45	68	0	12	13
50	71	2	10	10
55	67	4	11	11
60	69	1	10	13
65	64	4	12	13

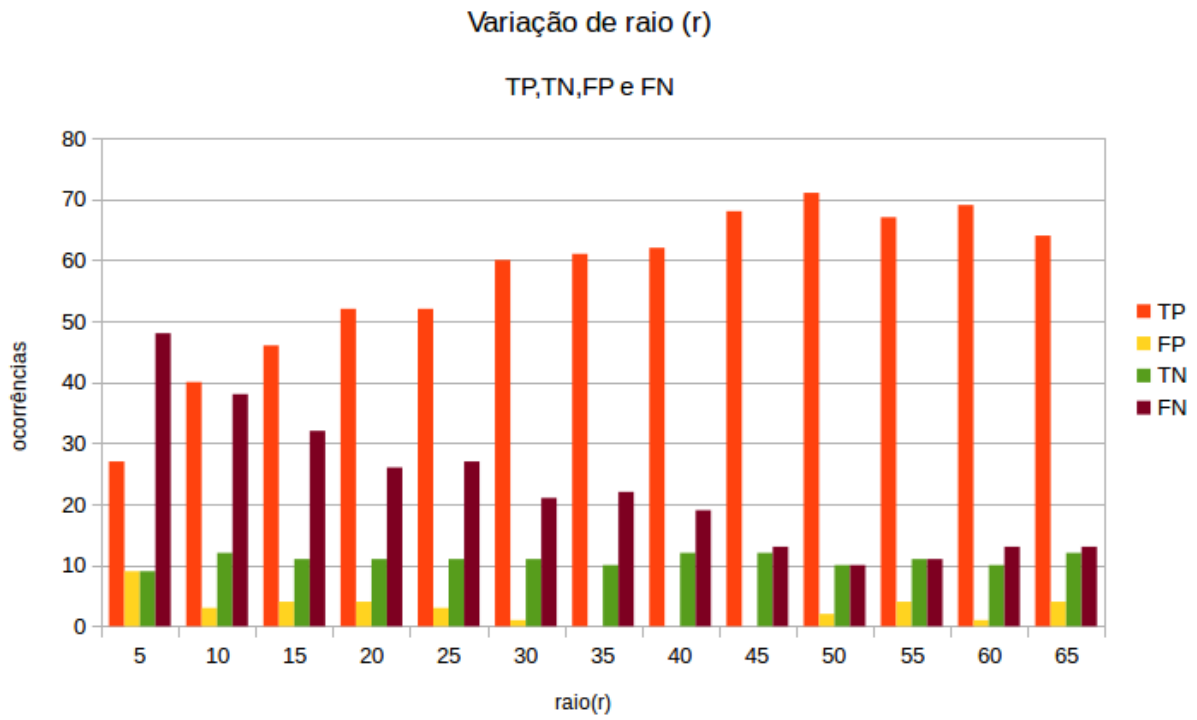


Figura 5.1: Critérios de F-Measure para o Cenário 1.

Pelo comportamento deste cenário, demonstrado na Figura 5.1, verifica-se que o número de verdadeiros positivos tendem a aumentar ao longo do aumento do raio (até atingir um ponto ótimo de operação onde eles começam a decair), e que o número de falsos positivos e negativos tendem a diminuir. Os verdadeiros positivos costumam variar pouco, mas normalmente rejeitando os membros que foram selecionados como excluídos da base de dados. Os valores obtidos para cada um dos critérios constam na Tabela 5.1.

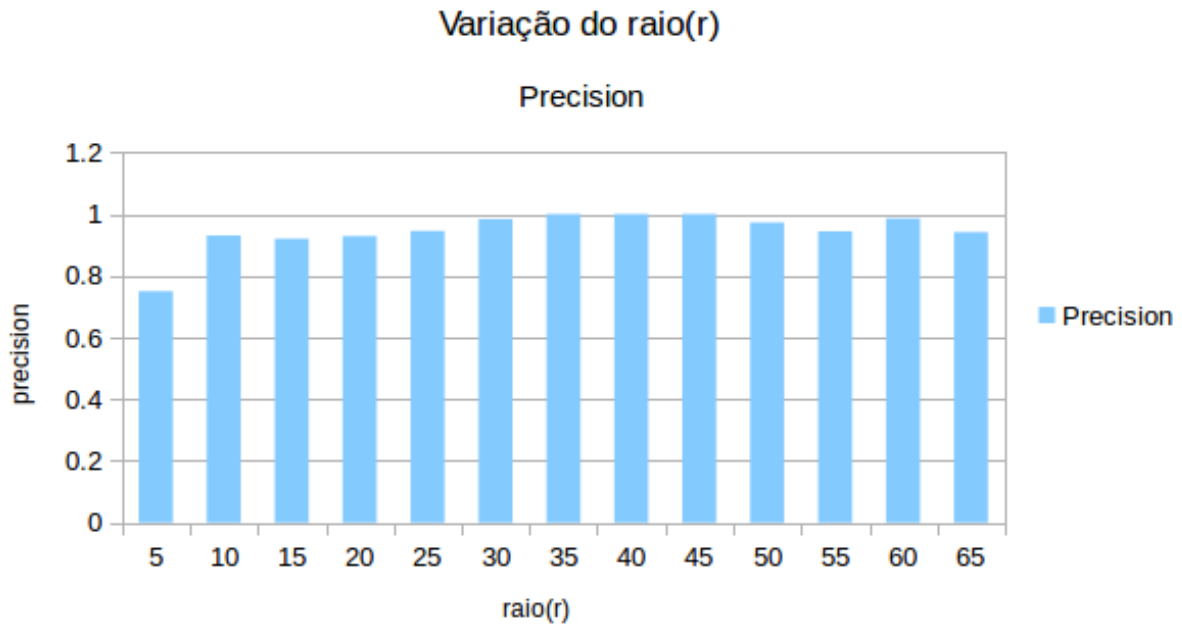


Figura 5.2: Precision para o Cenário 1.

Tabela 5.2: Tabela de valores de Precision do Cenário 1

raio	Precision
5	0.75
10	0.930232
15	0.92
20	0.928571
25	0.945454
30	0.983607
35	1
40	1
45	1
50	0.972603
55	0.943662
60	0.985714
65	0.941176

Tabela 5.3: Tabela de Recall para o Cenário 1

raio	Recall
5	0.36
10	0.512821
15	0.589744
20	0.666667
25	0.658228
30	0.740741
35	0.734940
40	0.765432
45	0.839506
50	0.876543
55	0.858974
60	0.841463
65	0.831169

Observa-se por meio da Figura 5.2 que o *Precision* aumenta a medida que o raio aumenta, atingindo um ponto ótimo de operação onde seu valor é maximizado, e voltando a decair após este ponto. Tal efeito se vale da hipótese que a partir de certos valores de raio, a vizinhança definida para cada ponto é suficiente para começar a descrever a geometria da superfície do rosto de uma

pessoa, divergindo-a da geometria da superfície do rosto de outra pessoa, entretanto existem valores de raio em que a vizinhança dos pontos engloba muita informação e (como as cabeças humanas como um todo apresentam formatos não muito discriminantes) dificulta a divergência entre os indivíduos . Os valores de *Precision* obtidos podem ser conferidos na Tabela 5.2.

Pode-se observar pela Tabela 5.2 que o valor de *Precision* atinge o máximo, indicando que a partir de um certo raio o número de falsos positivos é nulo, e portanto nenhum indivíduo está sendo reconhecido de maneira errônea, mas não impossibilita que muitos indivíduos estejam sendo recusados de maneira errônea.

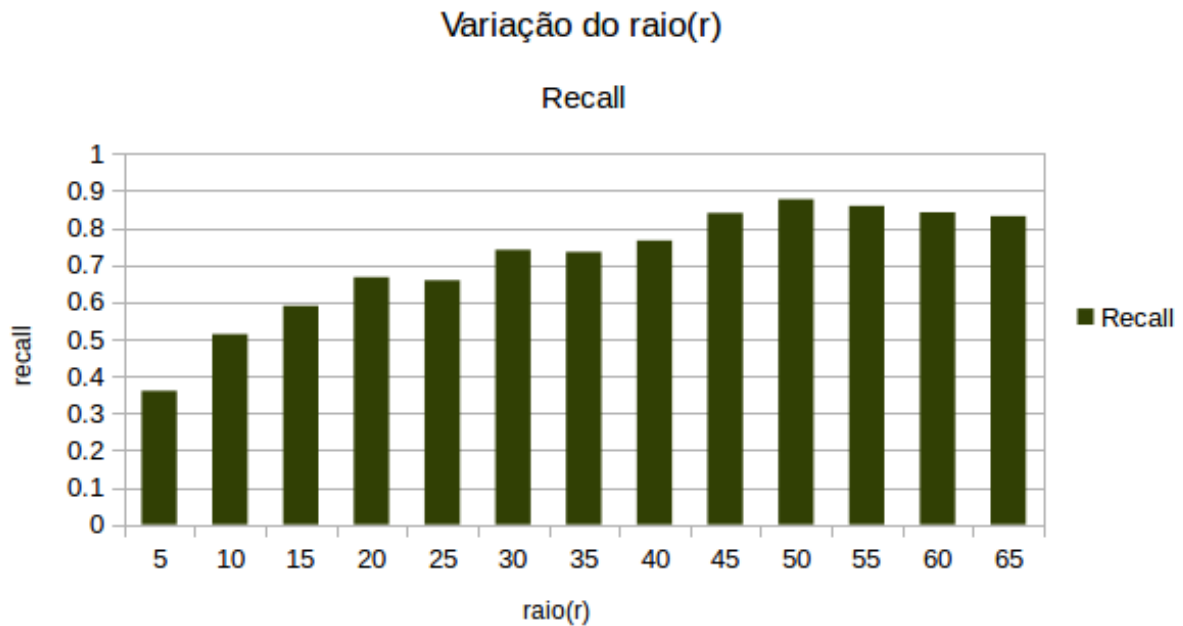


Figura 5.3: Recall para o Cenário 1

O *Recall* também apresenta um crescimento ao longo do aumento do raio, evidenciado na Figura 5.3, entretanto seu crescimento não atinge o valor máximo, como visto nos valores da Tabela 5.3, dado que os falsos negativos estão sempre presentes para os raios avaliados.

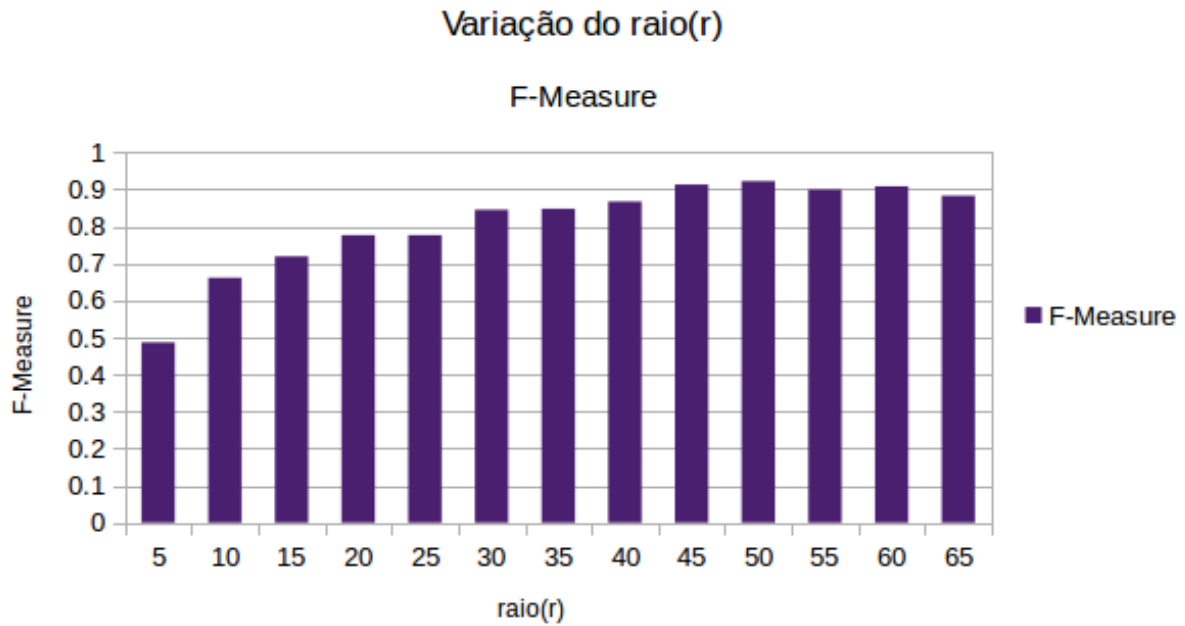


Figura 5.4: F-Measure para o Cenário 1

Tabela 5.4: Tabela de valores de F-Measure para o Cenário 1

raio	F-Measure
5	0.486486
10	0.661157
15	0.71875
20	0.776119
25	0.776119
30	0.845070
35	0.847222
40	0.867133
45	0.912752
50	0.922078
55	0.899329
60	0.907894
65	0.882759

Tabela 5.5: Tempo Médio de extração de um mapa de curvatura e Tempo total de extração dos mapas de curvatura da Base

raio	Tempo Medio(s)	Tempo Total(s)
5	2.392	212.929
10	2.952	262.813
15	3.541	315.174
20	4.396	391.331
25	6.02	535.825
30	8.209	730.676
35	9.341	831.372
40	11.315	1007.082
45	13.11	1166.854
50	14.945	1330.179
55	15.444	1374.566
60	17.807	1584.886
65	20.988	1868.002

O *F-Measure*, por ser dependente dos valores de *Precision* e *Recall*, também apresenta um crescimento, como pode ser notado na Figura 5.4 e Tabela 5.4, portanto o algoritmo começa a apresentar resultados mais confiáveis ao longo desse cenário. O *F-Measure* desenvolve um com-

portamento crescente ao longo do crescimento do raio, porém a partir de um ponto ele começa a decair, surgindo a hipótese que para esses raios as curvaturas estimadas se assemelham, pois em uma escala maior os rostos humanos apresentam formatos parecidos.

Uma desvantagem em relação ao aumento do raio se da no desempenho do algoritmo, dado que o conteúdo abrangido em cada vizinhança tende a ser maior, apresentando um maior custo computacional para desenvolver o cálculo da matriz de covariância. Isso acarreta em um tempo maior necessário para se extrair o mapa de curvatura da nuvem de pontos, como visto nas Tabela 5.5, demonstrando um crescimento acentuado em relação ao crescimento do raio, como visto nas Figuras 5.5 e 5.6. A avaliação de desempenho em relação ao tempo de execução é realizada apenas sobre o raio, por ser a atividade de maior tempo de execução do processo. Essa análise foi deduzida devido a maneira como o módulo foi implementado (em pequenos módulos executados separadamente em uma ordem), observando-se um tempo maior para execução na estimação de normais e curvaturas (realizada por métodos implementados pela PCL). A complexidade em função do raio é desconhecida devido aos métodos de estimação de curvatura e normal utilizados serem provenientes da biblioteca PCL, desconhecendo a maneira como foram implementados. Os tempos de execução para formação de histogramas, reconhecimento e validação também foram analisados e serão posteriormente discutidos.

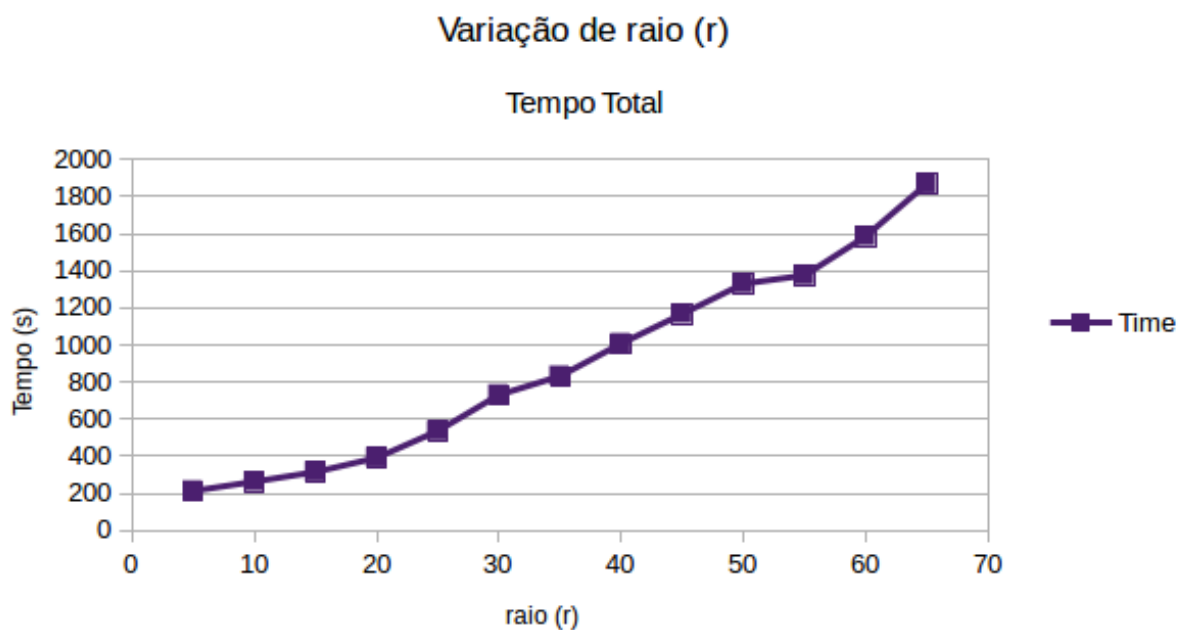


Figura 5.5: Tempo Total de preparação dos Mapas de Curvatura

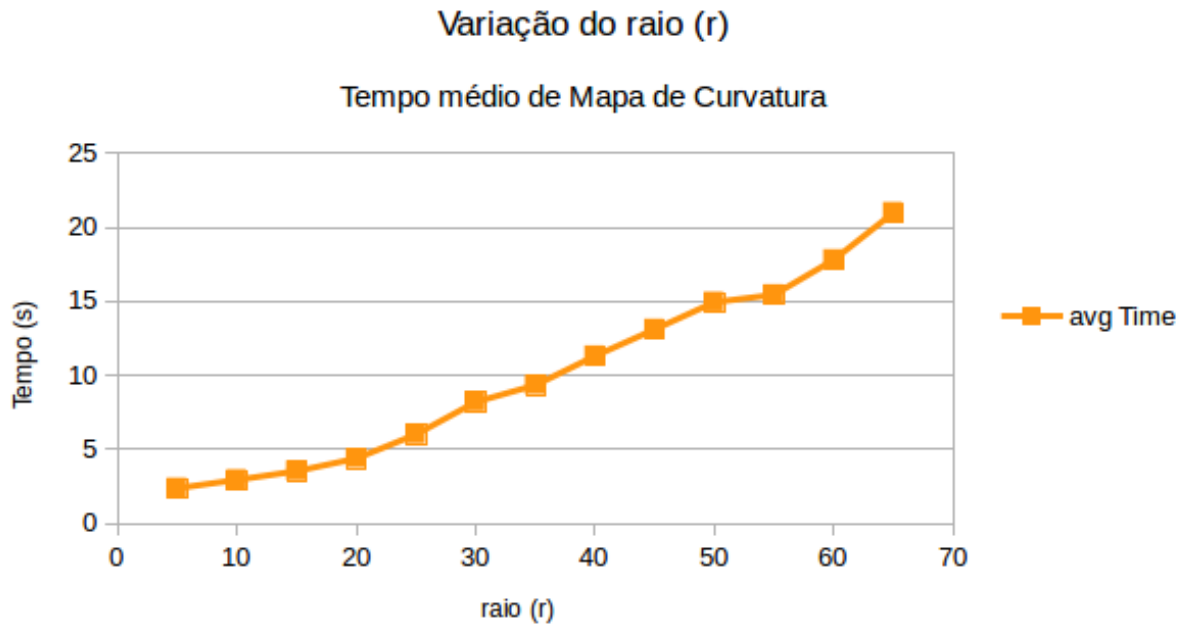


Figura 5.6: Tempo Médio de preparação de um Mapa de Curvatura

No Cenário 1 portanto, verifica-se um pico de desempenho no *F-Measure* evidenciado na Tabela 5.4 e Figura 5.4, indicando o ponto ótimo de operação do sistema para as especificações aplicadas ao método. O próximo cenário avalia o desempenho em relação ao número de *bins* definidos para cada histograma.

5.2.2 Cenário 2: Qualidade em Função do Número de bins

Esse cenário visa a análise do comportamento do sistema para histogramas compostos por diferentes números de *bins*. Esse cenário foi demandado para estudar o resultado que o número de *bins* proporciona ao reconhecimento proposto, com o intuito de estipular o melhor número de *bins* para a base de dados utilizada.

As configurações do cenário foram idênticas as configurações do cenário anterior (variando-se o raio para os mesmos valores e verificando os resultados de *F-Measure*), entretanto o número de *bins* foi alterado para 16, visando apresentar um comparativo entre os resultados obtidos para 8 *bins*.

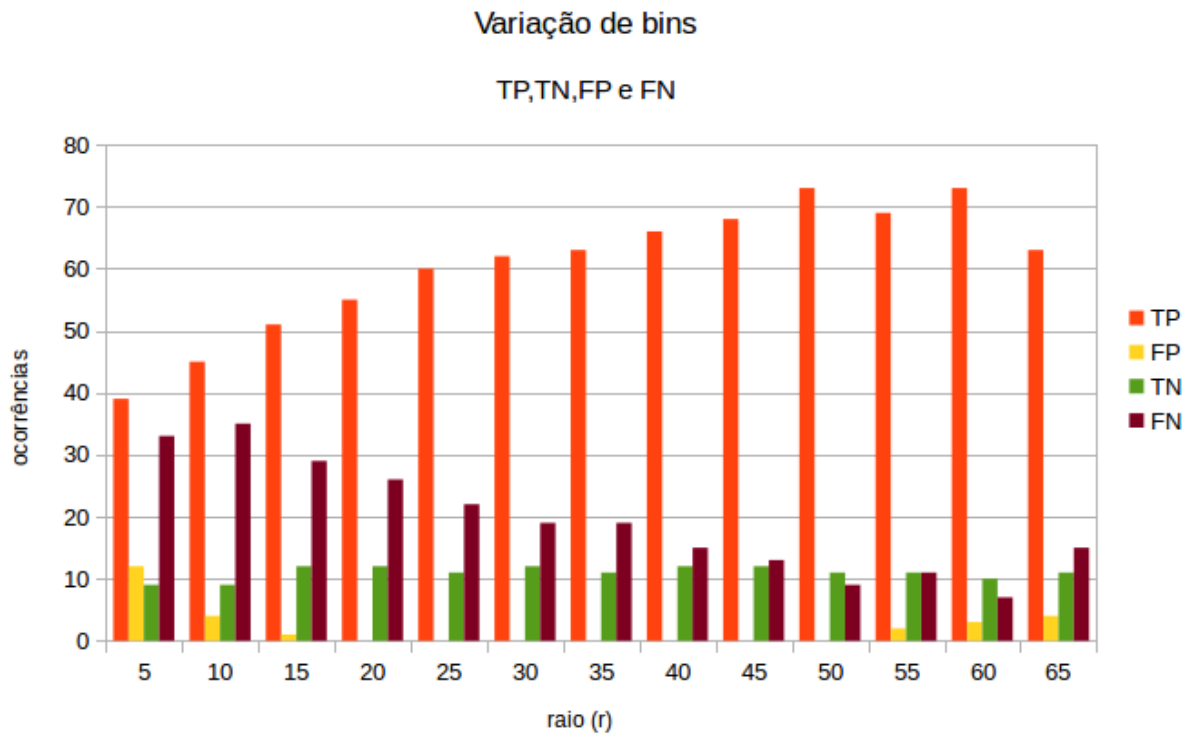


Figura 5.7: Critérios de F-Measure para o Cenário 2

Tabela 5.6: Tabela de Critérios de F-Measure para o Cenário 2

raio	TP	FP	TN	FN
5	39	12	9	33
10	45	4	9	35
15	51	1	12	29
20	55	0	12	26
25	60	0	11	22
30	62	0	12	19
35	63	0	11	19
40	66	0	12	15
45	68	0	12	13
50	73	0	11	9
55	69	2	11	11
60	73	3	10	7
65	63	4	11	15

Tabela 5.7: Tabela de Precision para o Cenário 2

raio	Precision
5	0.764706
10	0.918367
15	0.980769
20	1
25	1
30	1
35	1
40	1
45	1
50	1
55	0.971831
60	0.960526
65	0.940299

Comparando as Figuras 5.1, e 5.7 e as Tabelas 5.1 e 5.6, observa-se que o número de verdadeiros

positivos tende a ser maior para raios menores, e convergem a medida que o raio aumenta para 16 *bins*. Os falsos positivos tendem a serem nulos mais rapidamente utilizando 16 *bins*, enquanto os falsos negativos costumam ser bem próximos, assim como os verdadeiros negativos.

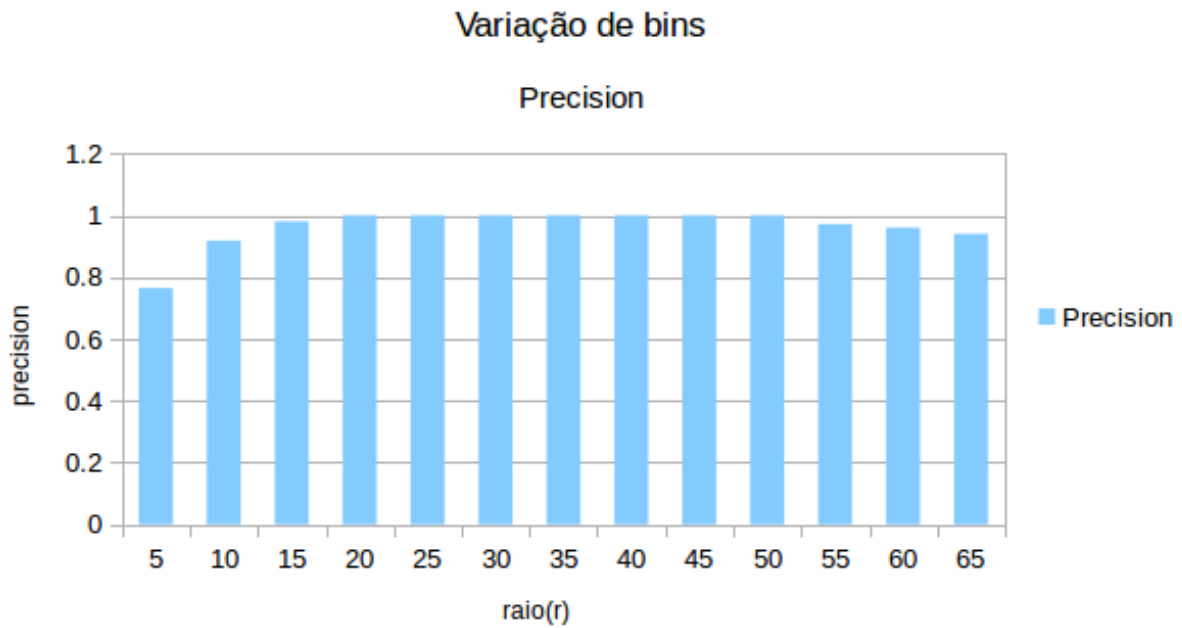


Figura 5.8: Precision para o Cenário 2

O *Precision* entre 8 e 16 *bins* é comparado através das Figuras 5.2 e 5.8 e Tabelas 5.2 e 5.7. Verifica-se que ambos os casos chegam ao valor máximo do *Precision*, entretanto para 16 *bins* esse efeito ocorre mais rapidamente, devido ao número de falsos positivos decair mais à medida que o raio cresce. Tal fato indica que com 16 *bins* o reconhecimento apresenta uma taxa de acerto melhor em comparação com 8 *bins*.

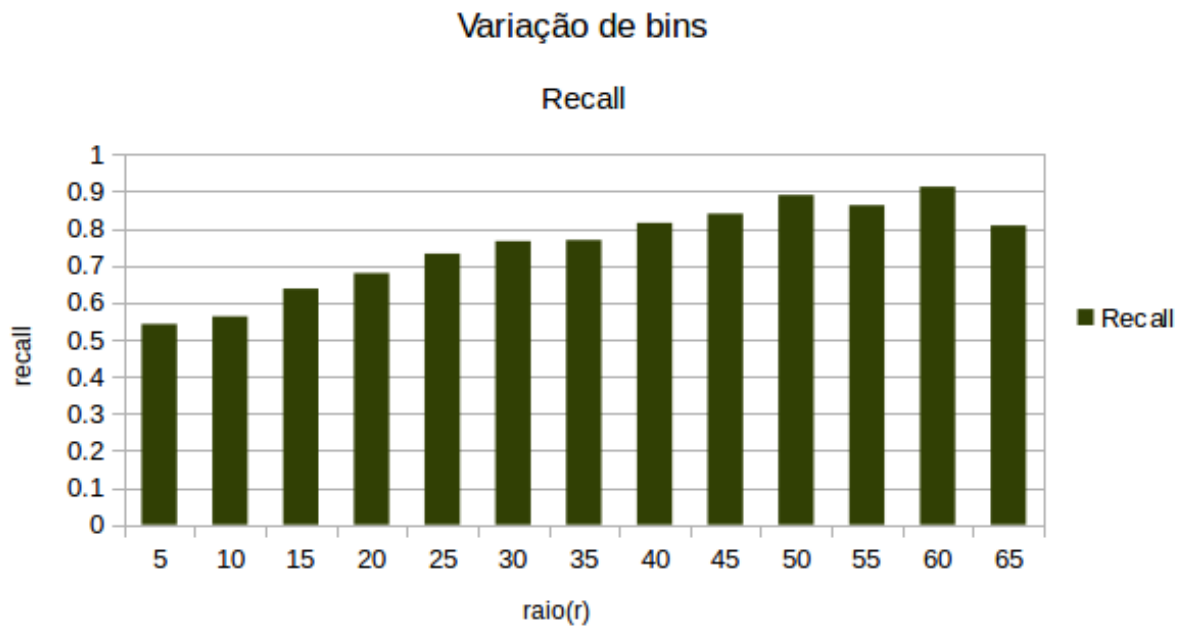


Figura 5.9: Recall para o Cenário 2

Em relação ao *Recall*, não há uma variação muito notável entre os valores vistos nas Tabelas 5.8 e 5.3 e nas Figuras 5.3 e 5.9, a medida que o raio cresce, devido ao fato de falsos negativos serem muito próximos a partir de raios maiores. Portanto não se evidencia um impacto considerável na capacidade de recusa do sistema por meio da variação do número de *bins*.

Tabela 5.8: Tabela de Recall para o Cenário 2

raio	Recall
5	0.541667
10	0.5625
15	0.6375
20	0.679012
25	0.731707
30	0.765432
35	0.768293
40	0.814815
45	0.839506
50	0.890244
55	0.8625
60	0.9125
65	0.807692

Tabela 5.9: Tabela de F-Measure do Cenário 2

raio	F-Measure
5	0.634146
10	0.697674
15	0.772727
20	0.808824
25	0.845070
30	0.867113
35	0.868966
40	0.897959
45	0.912752
50	0.941935
55	0.913907
60	0.935897
65	0.868966

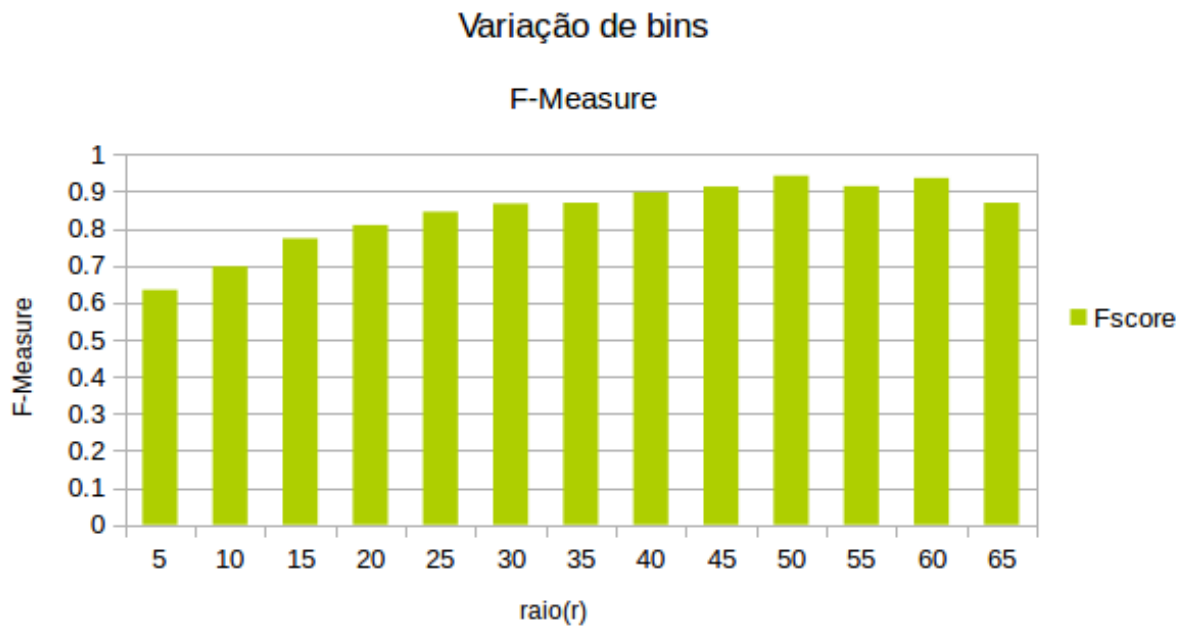


Figura 5.10: F-Measure para o Cenário 2

Por fim, novamente levando em concordância o *Precision* e o *Recall*, obtêm-se o resultado final do *F-Measure*, representado na Figura 5.10 e Tabela 5.9, cujo valor máximo obtido é maior que o valor máximo observado para 8 *bins*, portanto apresentando uma pequena melhoria quantitativa em relação ao cenário anteriormente articulado. Um comparativo entre os valores de F-Measure obtidos para 8 bins e 16 bins pode ser visto na Figura 5.11

O crescimento do número de *bins* causou um impacto significativamente menor no tempo de execução referente a formação de histogramas, comparação entre eles e validação. Para 8 *bins* o tempo médio de execução para formação de todos os histogramas foi de 76ms, enquanto para 16 *bins* o tempo médio de execução foi de 92ms. A comparação entre os histogramas e validação do algoritmo com 8 *bins* levaram 472ms em média para serem executadas, enquanto para 16 *bins* foi notado um tempo médio de execução de 783ms. O tempo de execução em relação ao número de regiões é tratado no próximo cenário.

Pela elaboração deste cenário é possível observar que há uma divergência das comparações e saídas do sistema de acordo com a variação do número de *bins* que compõem os histogramas.

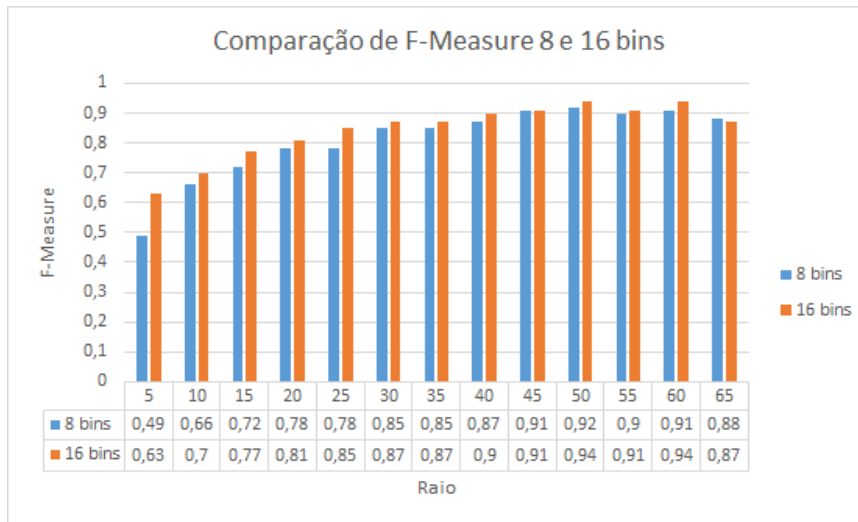


Figura 5.11: Comparativo entre os resultados de F-Measure por 8 e 16 bins.

5.2.3 Cenário 3: Qualidade em Função da divisão de Regiões

O Terceiro cenário visa avaliar a qualidade do sistema em relação ao número de n regiões o qual o mapa de curvaturas é dividido, ou seja, o número de histogramas que será gerado a partir de cada nuvem de pontos. O cenário compartilha as mesmas configurações do Cenário 1, apenas alterando o número de regiões (n) em que os mapas de curvatura são divididos.

A comparação nesse cenário se dá pelos valores referentes ao F-Measure, utilizando-se os mesmos valores de raio para as diferentes divisões, e portanto estabelecendo uma relação quantitativa com a qualidade do algoritmo e o número de divisões.

As Figuras 5.12, 5.13, 5.14, 5.15 e Tabelas 5.10, 5.11, 5.12, 5.13 representam os valores observados para os componentes do *F-Measure* em relação a divisão dos mapas de curvatura em 3 regiões, portanto para $n = 3$. A partir dos valores, é possível observar que o sistema apresentou mais erros em relação aos dados do Cenário 1, verificando que para este caso um número maior de regiões foi favorável na diferenciação de sujeitos.

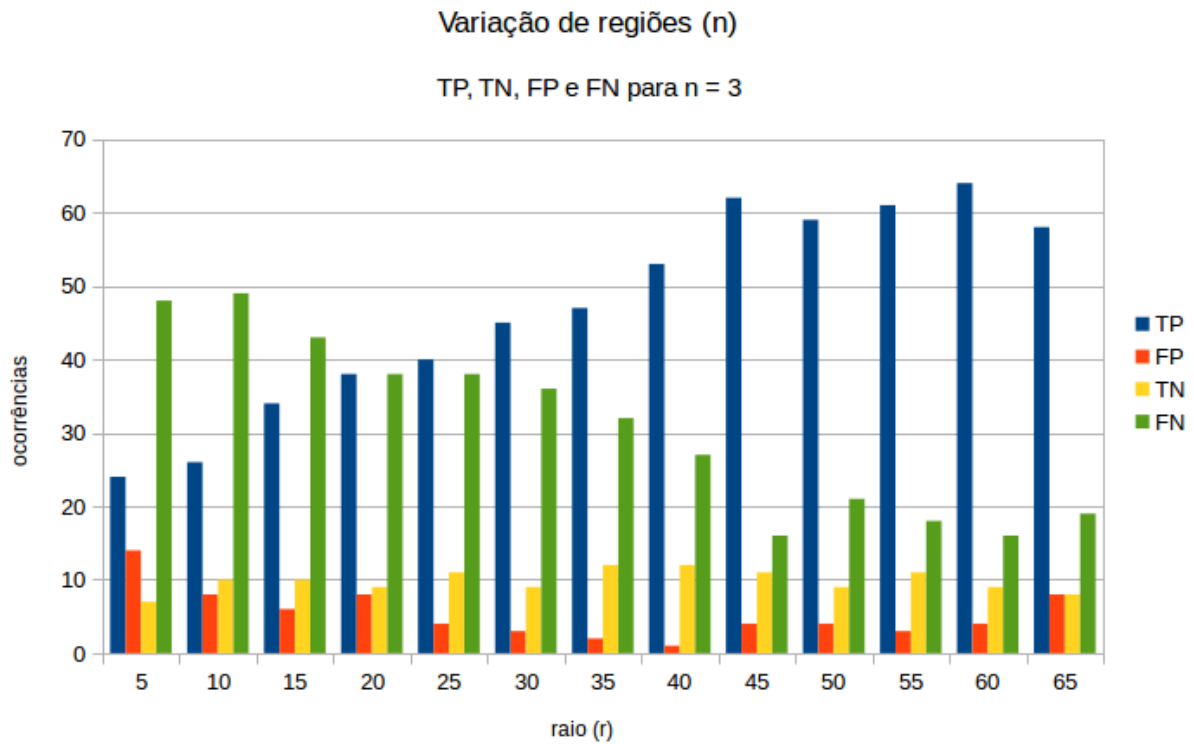


Figura 5.12: Critérios de F-Measure para n = 3

Tabela 5.10: Tabela de Critérios de F-Measure para n = 3

raio	TP	FP	TN	FN
5	24	14	7	48
10	26	8	10	49
15	34	6	10	43
20	38	8	9	38
25	40	4	11	38
30	45	3	9	36
35	47	2	12	32
40	53	1	12	27
45	62	4	11	16
50	59	4	9	21
55	61	3	11	18
60	64	4	9	16
65	58	8	8	19

Tabela 5.11: Tabela de valores de Precision para n = 3

raio	Precision
5	0.631579
10	0.764706
15	0.85
20	0.826087
25	0.909091
30	0.9375
35	0.959184
40	0.981481
45	0.939394
50	0.936508
55	0.953125
60	0.941176
65	0.878788

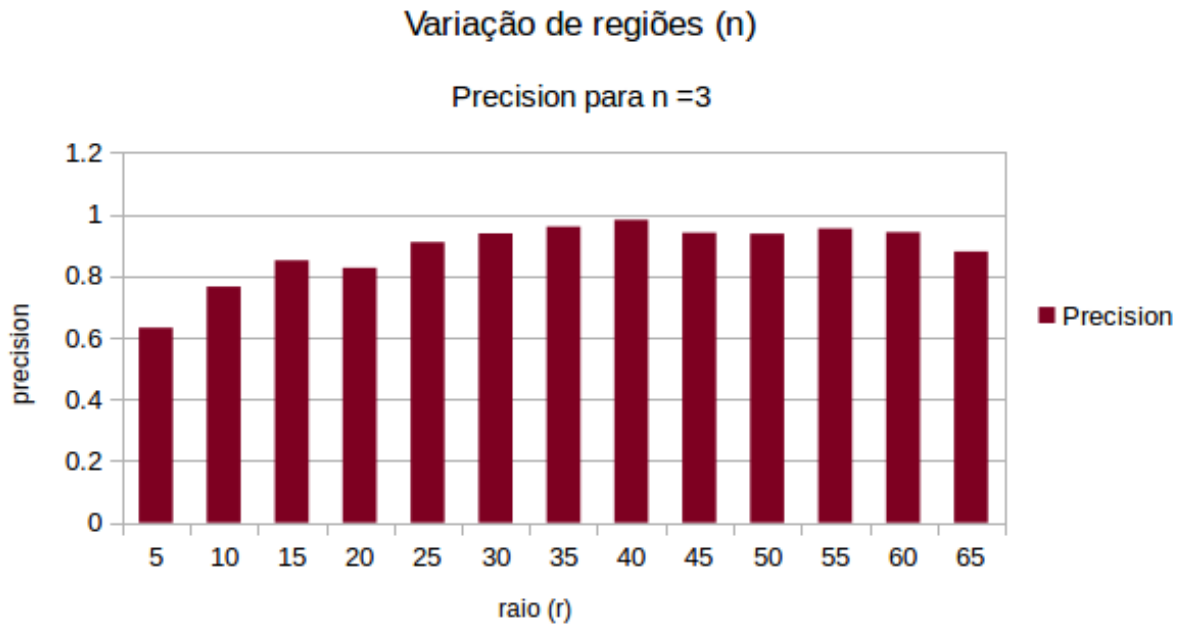


Figura 5.13: Precision para n = 3

Tabela 5.12: Tabela de valores de Recall para n = 3

raio	Recall
5	0.333333
10	0.346667
15	0.441558
20	0.5
25	0.512821
30	0.555556
35	0.594937
40	0.6625
45	0.794872
50	0.7375
55	0.772152
60	0.8
65	0.753247

Tabela 5.13: Tabela de valores de F-Measure para n = 3

raio	F-Measure
5	0.436363
10	0.477064
15	0.581197
20	0.622951
25	0.655738
30	0.697674
35	0.734375
40	0.791045
45	0.861111
50	0.825175
55	0.853147
60	0.864865
65	0.811189

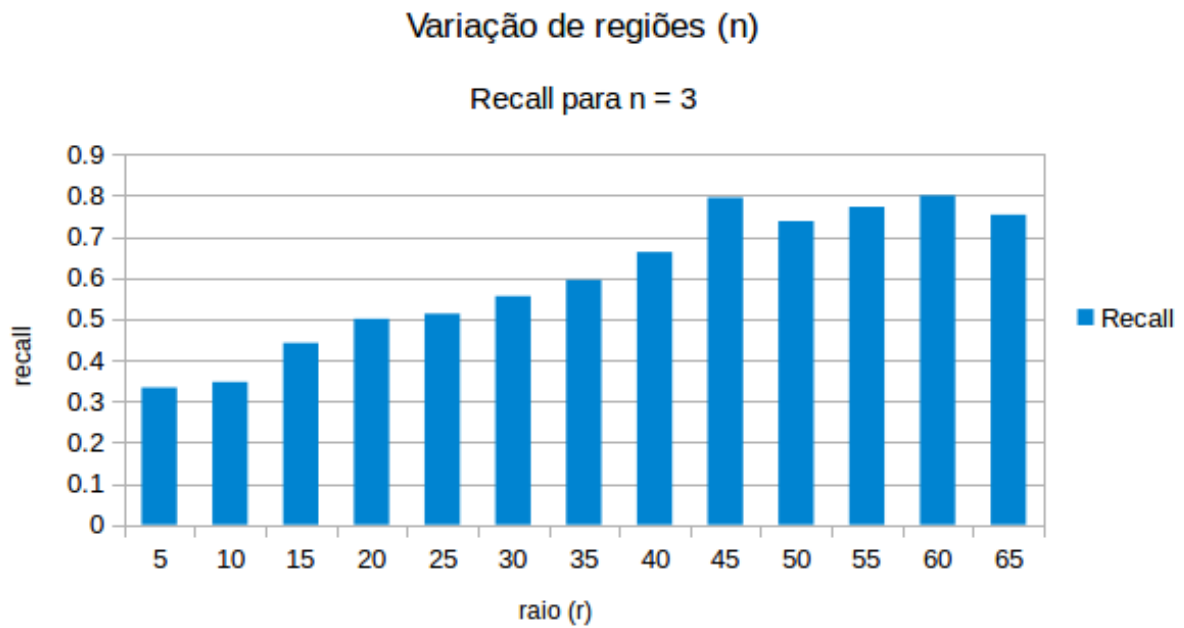


Figura 5.14: Recall para n = 3

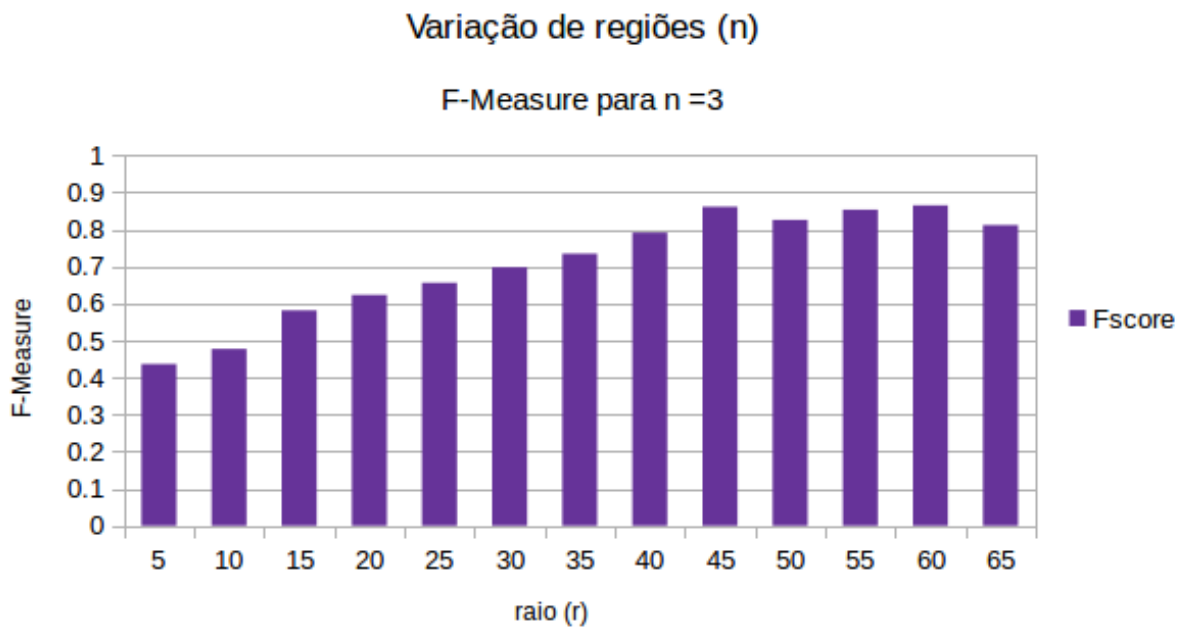


Figura 5.15: F-Measure para n = 3

As Figuras 5.16, 5.17, 5.18, 5.19 e Tabelas 5.14, 5.15, 5.16, 5.17 representam os dados referentes a divisão dos mapas de curvatura em 6 regiões, portanto para $n = 6$. Novamente observa-se que o comportamento do algoritmo para reconhecimento foi mais favorável nos dados do Cenário 1, indicando que um número maior de regiões dos mapas de curvatura e a análise em regiões mais

focadas da face favoreceu no processo de reconhecimento.

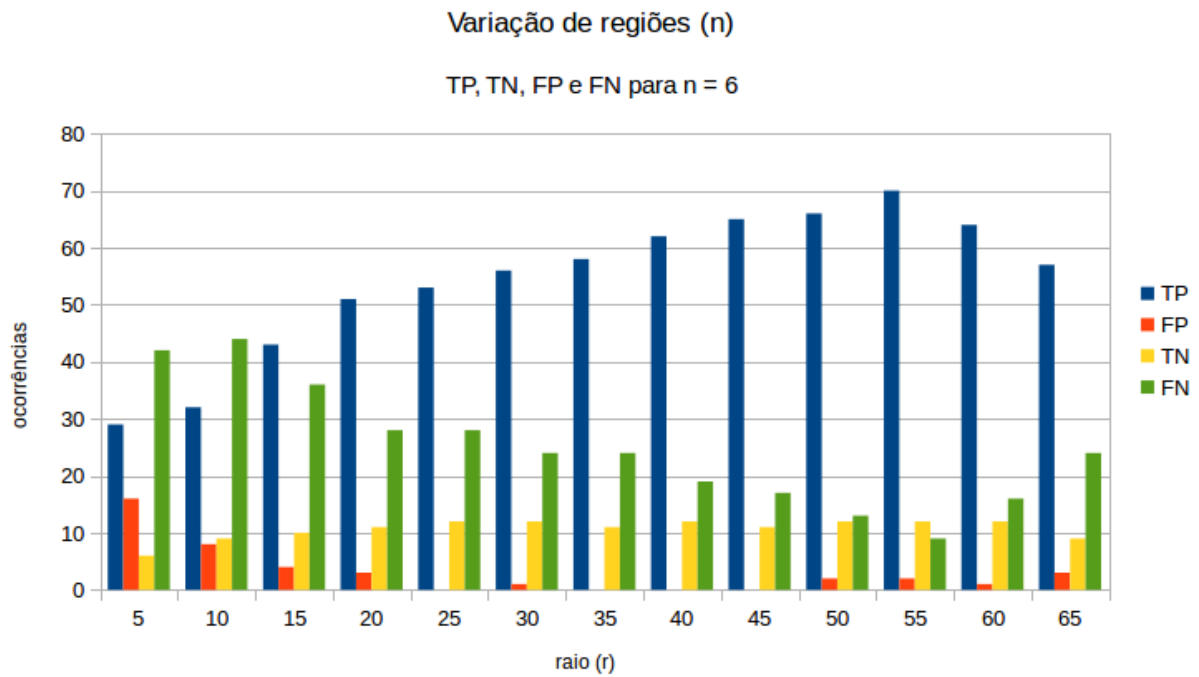


Figura 5.16: Critérios de F-Measure para n = 6

Tabela 5.14: Tabela de Critérios de F-Measure para n = 6

raio	TP	FP	TN	FN
5	29	16	6	42
10	32	8	9	44
15	43	4	10	36
20	51	3	11	28
25	53	0	12	28
30	56	1	12	24
35	58	0	11	24
40	62	0	12	19
45	65	0	11	17
50	66	2	12	13
55	70	2	12	9
60	64	1	12	16
65	57	3	9	24

Tabela 5.15: Tabela de valores de Precision para n = 6

raio	Precision
5	0.644444
10	0.8
15	0.914894
20	0.944444
25	1
30	0.982456
35	1
40	1
45	1
50	0.970588
55	0.972222
60	0.984615
65	0.95

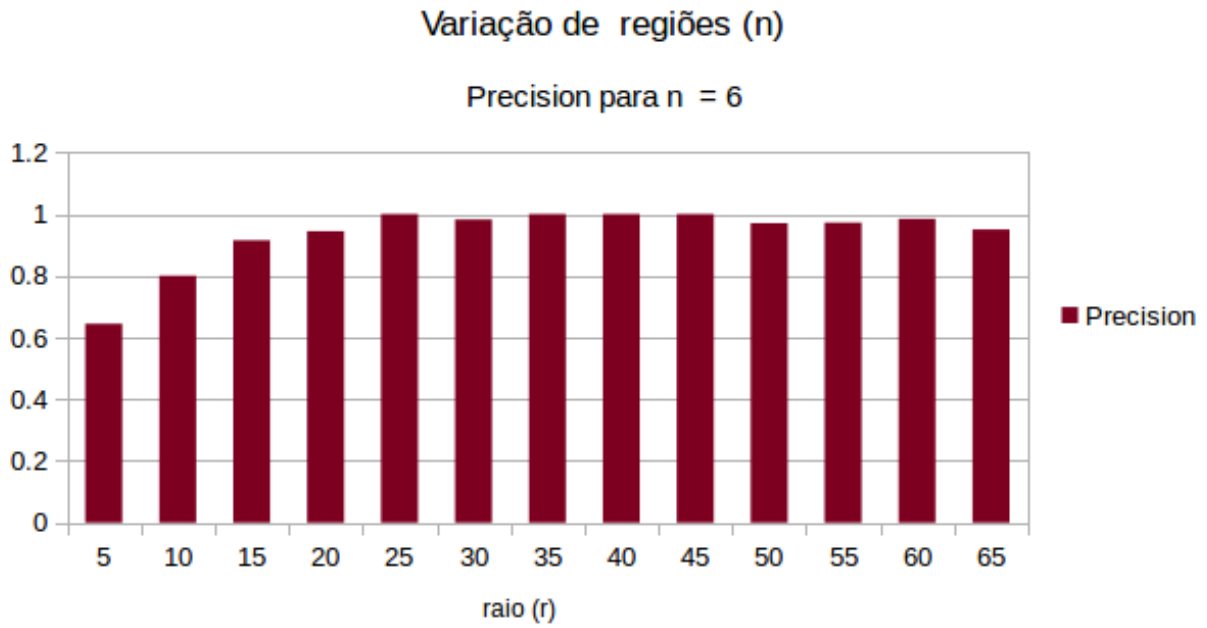


Figura 5.17: Precision para n = 6

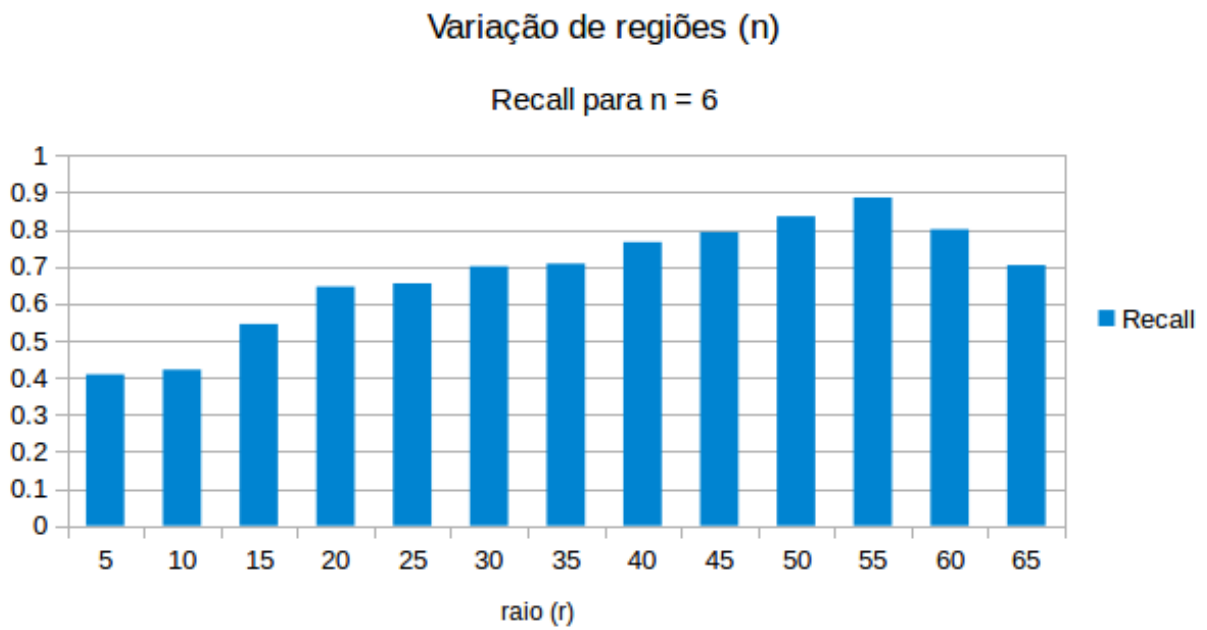


Figura 5.18: Recall para n = 6

Tabela 5.16: Tabela de valores e Recall para $n = 6$

raio	Recall
5	0.408451
10	0.421053
15	0.544304
20	0.645570
25	0.654321
30	0.7
35	0.707317
40	0.765432
45	0.792683
50	0.835443
55	0.886076
60	0.8
65	0.703704

Tabela 5.17: Tabela de valores de F-Measure para $n = 6$

raio	F-Measure
5	0.5
10	0.571724
15	0.682540
20	0.766917
25	0.791045
30	0.817518
35	0.828571
40	0.867133
45	0.884354
50	0.897959
55	0.927152
60	0.882759
65	0.808511

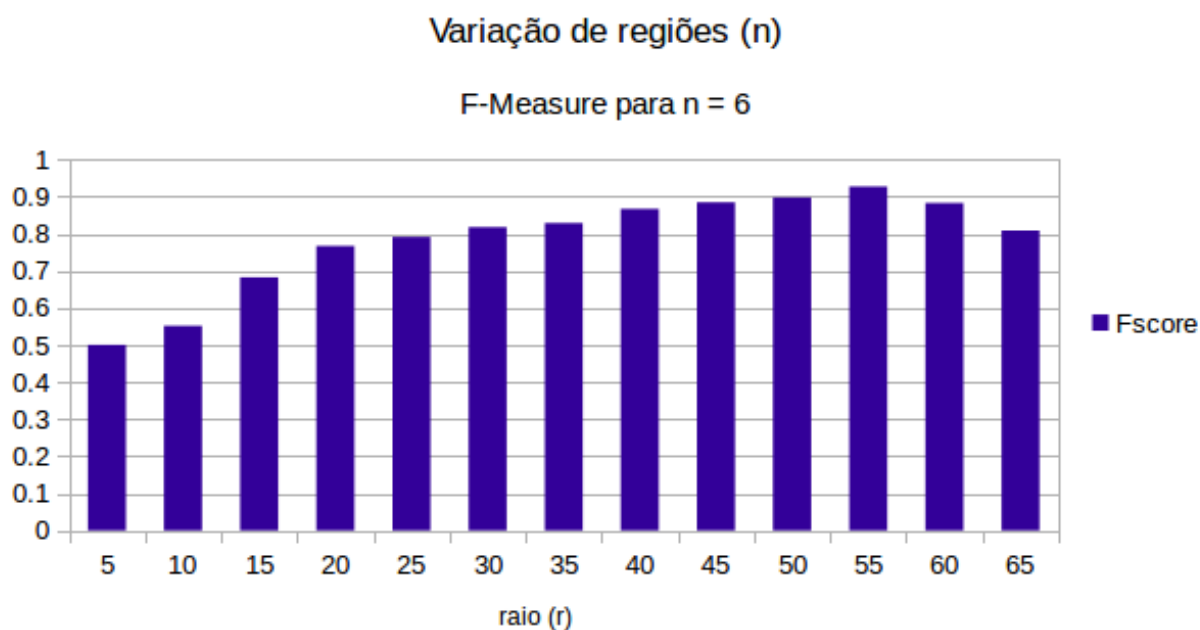


Figura 5.19: F-Measure para $n = 6$

O maior número de divisões realizadas no mapa de curvatura aparentou influenciar uma maior distinção entre indivíduos diferentes. Esse comportamento supostamente ocorre devido ao fato de que um baixo número de regiões está correlacionando diferentes geometrias da face que se assemelham em diferentes sujeitos. Essa comparação descrita acaba minimizando o *score* entre

sujeitos diferentes, e incentiva a ocorrência de erros de reconhecimento. A medida que o número de divisões da face aumenta, as comparações começam a se tornar mais focadas, e conseqüentemente um indivíduo tende a divergir dos demais a menos que suas feições faciais sejam parecidas. Um comparativo dos *F-Measure* obtidos para diferentes números de divisões podem ser observados na Figura 5.20

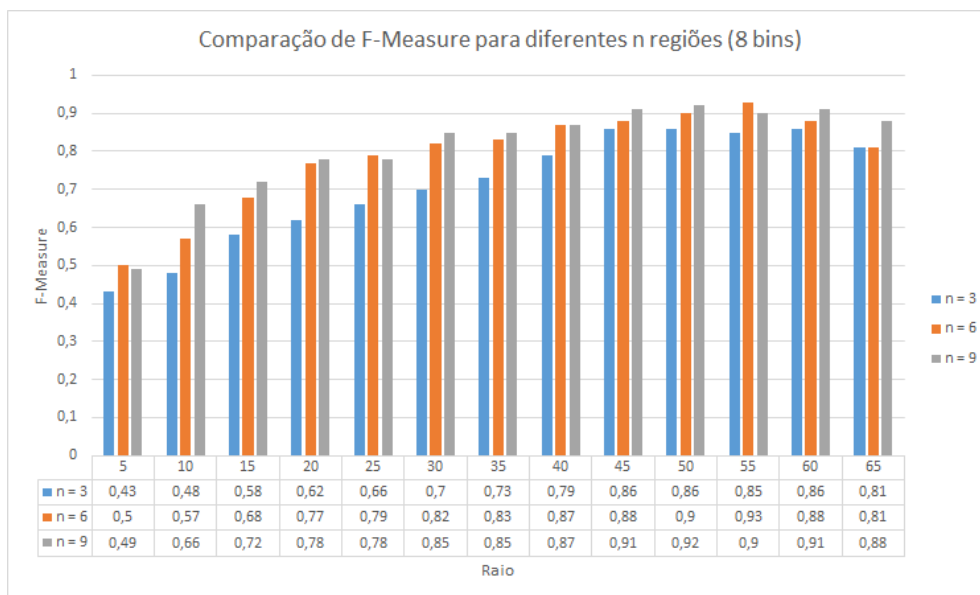


Figura 5.20: Comparativo dos *F-Measure* para diferentes números de divisões.

O número de regiões apresentou uma baixa interferência no tempo de execução da formação dos histogramas. Mantendo o raio (50) e número de *bins* (8) fixos, foram observados tempos médios de execução de 60ms para $n = 3$, 68ms para $n = 6$, e 76ms para $n = 9$. Para as mesmas configurações, os tempos de execução de comparação e validação foram de 204ms para $n = 3$, 296ms para $n = 6$ e 472ms para $n = 9$. O tempo necessário para a comparação (utilizando a função de minimização) tende a crescer devido a complexidade da sua implementação ser de $O(n^3)$, havendo um número maior de iterações em relação ao número de histogramas, definido pelo número de divisões do mapa de curvatura.

Por fim, estima-se com o estudo deste cenário que um maior número de regiões proporcionaria um melhor resultado na comparação de indivíduos, atingindo um número ótimo de regiões que garante o melhor desempenho do sistema.

5.2.4 Limiar

O Limiar foi selecionado para cada caso específico priorizando a redução máxima de falsos positivos, dado que para maioria dos sistemas de reconhecimento um falso positivo é muito mais impactante que um falso negativo, como exemplo um sistema de reconhecimento que controla a entrada de sujeitos à um local apresenta um comportamento melhor quando se nega a entrada de um conhecido do que quando se autoriza a entrada de um estranho.

5.3 Análise Espacial de Memória

A análise referente ao uso de memória é relevante para o método desenvolvido, devido a grande quantidade de dados (Imagens, matrizes, nuvens, pontos, etc) necessária para sua implementação e execução.

5.3.1 Imagens

A princípio, a base de dados é formada por diversas imagens, sendo estas divididas em imagens de cor e imagens de profundidade. As imagens são organizadas em matrizes, nas quais a quantidade de dados armazenados varia de acordo com seus parâmetros (tamanho, número de canais, etc).

As imagens de cor da base de dados apresentam um tamanho de 1280x960 pixels e são carregadas utilizando a OpenCV em modo RGB, apresentando três canais de intensidade (cujo intervalo de valores de intensidade é definido por um byte, assumindo valores do intervalo [0,255]). Cada pixel da imagem apresenta 3 bytes de informação de intensidade referentes as cores. O espaço de memória necessário apenas para imagens de cor corresponde ao número de bytes por pixel, multiplicado pelo número de pixels de cada imagem, multiplicado pelo número de imagens por sujeito da base de dados, multiplicado pelo número de sujeitos da base de dados. Essas imagens de cor posteriormente são transformadas em escala de cinza para realizar procedimentos como o Haar Cascade, apresentando apenas 1 byte de informação de intensidade por pixel, referente a intensidade de nível de cinza.

As imagens de profundidade apresentam um tamanho de 690x480 pixels, onde cada pixel apresenta um valor inteiro (4 bytes) referente a leitura de profundidade do Kinect. O espaço de memória necessário é calculado da mesma maneira que para imagens de cor, apresentando o uso cerca de 123206400 bytes.

5.3.2 Nuvens de Ponto

As nuvens de pontos são formadas a partir das imagens de profundidade, gerando pontos a partir das informações de pixel.

O número de pontos da nuvem de pontos de cada sujeito varia, sendo observados valores entre 4000 e 5500 pontos por face. Para cada ponto é necessário um espaço de 12 bytes, correspondentes aos pontos flutuantes referentes as posições em coordenadas cartesianas. Portanto a memória necessária para uma nuvem de pontos corresponde a multiplicação do número de pontos pelo número de bytes por ponto.

5.3.3 Matriz de Covariância

As matrizes de covariância do método consistem em matrizes de dimensão 3x3, apresentando 9 valores de ponto flutuante por matriz (e conseqüentemente 36 bytes por matriz). As matrizes

são calculadas para cada ponto das nuvens de pontos. O espaço de memória necessário consiste na multiplicação do número das nuvens de pontos pelo número de bytes de cada matriz.

5.3.4 Mapas de Curvatura

Os mapas de curvatura são extraídos de cada nuvem de ponto da base de dados. A versão final dos mapas de curvatura consistem em imagens em escala de nível de cinza, apresentando apenas 1 byte por pixel. O tamanho da imagem é relativo ao número de pontos da nuvem de pontos da qual ela foi extraída, sendo observados números de linhas e colunas entre 70 e 90.

5.3.5 Histogramas

Os histogramas consistem em vetores de pontos flutuantes que armazenam a frequência normalizada dos níveis de cinza dos mapas de curvatura. O tamanho do vetor corresponde ao número de *bins* do histograma. O espaço de memória necessário para cada histograma consiste no número de *bins* multiplicado pelo tamanho de um ponto flutuante (float - 4 bytes).

5.3.6 Espaço total do método

Por meio dos custos de memória definidos posteriormente, é possível observar que a memória necessária para a execução do método varia com o número de sujeitos da base dados. Com a base de dados descrita neste trabalho, é necessário cerca de 1 GigaByte para armazenar toda a informação processada.

5.4 Dificuldades Evidenciadas

Durante a execução da metodologia proposta, foram evidenciados problemas que comprometeram o aproveitamento completo da base de dados, como a falha na detecção facial do Haar Cascade (utilizado visando a automação do processo de detecção facial), não encontrando uma face em região diferente da face ou encontrando múltiplas faces, como exemplo mostrado na Figura 5.21. Os casos de múltiplas faces foi corrigido utilizando a dedução que a face mais próxima do centro da imagem corresponde a face correta, dada a maneira como a base de dados foi desenvolvida.

Outros problemas foram constatados quanto a base de dados, como distorções ocorridas durante a captura das imagens de profundidade ou problemas de precisão do Kinect, dado que a distância na qual os indivíduos se encontram da câmera fica em torno de 2 metros do sensor, o que propicia uma baixa captação de detalhes e uma maior probabilidade de distorções. Por fim, ainda relacionado ao Kinect, em algumas faces houve um erro de leitura da grade, normalmente em uma coluna ou linha, gerando nuvens de pontos com algumas oclusões, que resultavam em menos informações para os mapas de curvatura, como as duas colunas e linhas vistas na Figura 4.22.



Figura 5.21: Múltiplas faces encontradas em uma imagem.

Capítulo 6

Conclusões

O algoritmo proposto e desenvolvido visou realizar o reconhecimento facial 3D de faces frontais, explorando apenas características extraídas da nuvem de pontos tridimensional representando os dados da face. Para tal feito, foram utilizadas técnicas de estimação de curvaturas e normais para nuvens de pontos, transpondo essas informações para imagens coloridas que são transformadas em imagens de escala de cinza, e realizando a comparação entre os indivíduos por uma função de minimização entre essas imagens de níveis de cinza.

O algoritmo apresentou resultados satisfatórios, apresentando um valor de *F-Measure* de até 94%, contendo uma baixa taxa de falsos reconhecimentos. A análise tomada ainda permitiu uma verificação de desempenho quanto ao tempo de execução necessário para preparação dos dados a serem utilizados, observando uma certa relação entre a qualidade de identificação do algoritmo e o tempo necessário para esta identificação.

Apesar da base de dados utilizada não ser específica para reconhecimento facial e suas imagens apresentarem baixa densidade de informações, a sua aplicação em sistemas de segurança de controle de acesso poderia ser viável, apesar da taxa de rejeição obtida do sistema poder causar interferências e transtornos nesse tipo de aplicação. O método proposto seria inviável para aplicações como vigilância, dado que o ambiente seria descontrolado e que o sensor apresenta uma grande restrição de alcance.

A proposta apresentada pode ser requintada por meio da implementação de aprimoramentos, amplificando a taxa de reconhecimento e reduzindo rejeições incorretas, através da exploração de outras características da nuvem de pontos tridimensional, e por modificações ao processo de reconhecimento.

6.1 Perspectivas Futuras

A solução proposta para realizar o reconhecimento facial em 3D deixa de explorar todo o potencial da base de dados fornecida, testando-se apenas a implementação em uma das poses. Um estudo futuro permite a integração das outras posições da face aos elementos analisados,

desenvolvendo um sistema invariante a poses.

Em relação a solução, a implementação de diferentes técnicas de comparação seria um processo interessante para o método, considerando que a comparação por histogramas pode não ser a técnica mais apropriada.

Outra realização que pode ser integrada ao método desenvolvido consiste na exploração de mais características extraídas das nuvens de pontos, além de métodos computacionais auxiliares para classificação de melhores decisões durante o reconhecimento.

Visando evitar ou reduzir os problemas enfrentados neste trabalho, a confecção e integração de um sistema de detecção facial em nuvens de pontos 3D favoreceria o sistema como um todo, necessitando-se um número menor de dados para realização do método e tornando o sistema independente de aplicações em nível de imagens bidimensionais.

Por fim, uma proposta que permitiria um incentivo ao estudo na área seria o desenvolvimento de uma base de dados mais rica em detalhes e feições da face, provendo a elaboração de hipóteses melhores sobre o impacto de algumas características geométricas do rosto.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ROBERTSON, C. *CS 6630 - Scientific Visualization: Project 3*. The University of Utah. Disponível em: <<http://www.cs.utah.edu/~croberts/courses/cs6630/project3/>>.
- [2] RUSU, R. B.; COUSINS, S. 3d is here: Point cloud library (pcl). In: IEEE. *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. [S.l.], 2011. p. 1–4.
- [3] BRADSKI, G.; KAEHLER, A. *Learning OpenCV: Computer vision with the OpenCV library*. [S.l.]: "O'Reilly Media, Inc.", 2008.
- [4] VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.], 2001. v. 1, p. I–511.
- [5] PATRICK. *Surface Normal*. Acessado em 15/12/2016. Disponível em: <[https://en.wikipedia.org/wiki/Normal\(geometry\)#/media/File:Surface_normal_illustration.svg](https://en.wikipedia.org/wiki/Normal(geometry)#/media/File:Surface_normal_illustration.svg)>.
- [6] BRUNELLI, R.; POGGIO, T. Face recognition: Features versus templates. *IEEE transactions on pattern analysis and machine intelligence*, v. 15, n. 10, p. 1042–1052, 1993.
- [7] WISKOTT, L. et al. Intelligent biometric techniques in fingerprint and face recognition. In: JAIN, L. C. et al. (Ed.). Boca Raton, FL, USA: CRC Press, Inc., 1999. cap. Face Recognition by Elastic Bunch Graph Matching, p. 357–396. ISBN 0-8493-2055-0. Disponível em: <<http://dl.acm.org/citation.cfm?id=342658.342697>>.
- [8] PENTLAND, A.; MOGHADDAM, B.; STARNER, T. View-based and modular eigenspaces for face recognition. In: IEEE. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. [S.l.], 1994. p. 84–91.
- [9] BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 1996. p. 43–58.
- [10] SOCOLINSKY, D. A. et al. Illumination invariant face recognition using thermal infrared imagery. In: IEEE. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.], 2001. v. 1, p. I–527.

- [11] JAIN, A. K.; ROSS, A.; PRABHAKAR, S. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, IEEE, v. 14, n. 1, p. 4–20, 2004.
- [12] LINSEN, L. *Point cloud representation*. [S.l.]: Univ., Fak. für Informatik, Bibliothek, 2001.
- [13] LAI, K.; FOX, D. Object recognition in 3d point clouds using web data and domain adaptation. *The International Journal of Robotics Research*, SAGE Publications, v. 29, n. 8, p. 1019–1037, 2010.
- [14] SITHOLE, G.; VOSSelman, G. Automatic structure detection in a point-cloud of an urban landscape. In: IEEE. *Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on*. [S.l.], 2003. p. 67–71.
- [15] BOEHLER, W.; MARBS, A. 3d scanning instruments. *Proceedings of the CIPA WG*, v. 6, p. 9–18, 2002.
- [16] FRIESEN, J. *What is a Look Up Table (LUT), Anyway?* May 2011. Disponível em: <<http://nofilmschool.com/2011/05/what-is-a-look-up-table-lut-anyway>>.
- [17] CAMPBELL-KELLY, M. *The history of mathematical tables: from Sumer to spreadsheets*. [S.l.]: Oxford University Press, 2003.
- [18] MESE, M.; VAIDYANATHAN, P. P. Look-up table (lut) method for inverse halftoning. *IEEE Transactions on Image Processing*, IEEE, v. 10, n. 10, p. 1566–1578, 2001.
- [19] HALFHILL, T. R. An error in a lookup table created the infamous bug in intel’s latest processor. *BYTE*, March, 1995.
- [20] INMOS. *Graphics Book*. 1990. Second Edition.
- [21] ZHANG, Z. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 19, n. 2, p. 4–10, abr. 2012. ISSN 1070-986X. Disponível em: <<http://dx.doi.org/10.1109/MMUL.2012.24>>.
- [22] BRADSKI, G. *Dr. Dobb’s Journal of Software Tools*.
- [23] VIOLA, P.; JONES, M. J. Robust real-time face detection. *International journal of computer vision*, Springer, v. 57, n. 2, p. 137–154, 2004.
- [24] KLINE, M. *Calculus: an intuitive and physical approach*. [S.l.]: Courier Corporation, 1998.
- [25] KLAF, A. A. *Calculus Refresher*. [S.l.]: Courier Corporation, 1956.
- [26] JAFRI, R.; ARABNIA, H. R. A survey of face recognition techniques. *Jips*, v. 5, n. 2, p. 41–68, 2009.
- [27] JAIN, A.; BOLLE, R.; PANKANTI, S. *Biometrics: personal identification in networked society*. [S.l.]: Springer Science & Business Media, 2006.
- [28] MCCULLAGH, D. Call it super bowl face scan i. *Wired*, Feb, v. 2, 2001.

- [29] NASTAR, C.; MITSCHKE, M. Real-time face recognition using feature combination. In: IEEE. *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on.* [S.l.], 1998. p. 312–317.
- [30] GONG, S.; MCKENNA, S. J.; PSARROU, A. *Dynamic vision: from images to face recognition.* [S.l.]: Imperial College Press, 2000.
- [31] KANADE, T. Picture processing system by computer complex and recognition of human faces. *Doctoral dissertation, Kyoto University,* Eurographics Association, v. 3952, p. 83–97, 1973.
- [32] NIXON, M. Eye spacing measurement for facial recognition. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *29th Annual Technical Symposium.* [S.l.], 1985. p. 279–285.
- [33] REISFELD, D. *Generalized symmetry transforms: attentional mechanisms and face recognition.* Tese (Doutorado) — Tel Aviv University, 1994.
- [34] GRAF, H. P. et al. Locating faces and facial parts. In: *Proc. First Int'l Workshop Automatic Face and Gesture Recognition.* [S.l.: s.n.], 1995. p. 41–46.
- [35] COX, I. J.; GHOSN, J.; YIANILOS, P. N. Feature-based face recognition using mixture-distance. In: IEEE. *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on.* [S.l.], 1996. p. 209–216.
- [36] WISKOTT, L. et al. Face recognition by elastic bunch graph matching. *IEEE Transactions on pattern analysis and machine intelligence,* IEEE, v. 19, n. 7, p. 775–779, 1997.
- [37] CAMPADELLI, P.; LANZAROTTI, R. A face recognition system based on local feature characterization. In: *Advanced Studies in Biometrics.* [S.l.]: Springer, 2005. p. 147–152.
- [38] YUILLE, A.; COHEN, D.; HALLINAN, P. Feature extraction from faces using deformable templates. In: IEEE. *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89., IEEE Computer Society Conference on.* [S.l.], 1989. p. 104–109.
- [39] ALBIOL, A. et al. Face recognition using hog–ebgm. *Pattern Recognition Letters,* Elsevier, v. 29, n. 10, p. 1537–1543, 2008.
- [40] KAUFMAN, G. J.; BREEDING, K. J. The automatic recognition of human faces from profile silhouettes. *IEEE Transactions on Systems, Man, and Cybernetics,* IEEE, n. 2, p. 113–121, 1976.
- [41] HARMON, L. et al. Machine identification of human faces. *Pattern Recognition,* Elsevier, v. 13, n. 2, p. 97–110, 1981.
- [42] LIPOŠČAK, Z.; LONCARIC, S. A scale-space approach to face recognition from profiles. In: SPRINGER. *International Conference on Computer Analysis of Images and Patterns.* [S.l.], 1999. p. 243–250.
- [43] SIROVICH, L.; KIRBY, M. Low-dimensional procedure for the characterization of human faces. *Josa a,* Optical Society of America, v. 4, n. 3, p. 519–524, 1987.

- [44] JAIN, A. K.; DUBES, R. C. *Algorithms for clustering data*. [S.l.]: Prentice-Hall, Inc., 1988.
- [45] FUKUNAGA, K. *Introduction to statistical pattern recognition*. [S.l.]: Academic press, 2013.
- [46] TURK, M. A.; PENTLAND, A. P. Face recognition using eigenfaces. In: IEEE. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. [S.l.], 1991. p. 586–591.
- [47] TURK, M.; PENTLAND, A. Eigenfaces for recognition. *Journal of cognitive neuroscience*, MIT Press, v. 3, n. 1, p. 71–86, 1991.
- [48] MOSES, Y.; ADINI, Y.; ULLMAN, S. Face recognition: The problem of compensating for changes in illumination direction. In: SPRINGER. *European conference on computer vision*. [S.l.], 1994. p. 286–296.
- [49] L., D.; WENG, J. Using Discriminant Eigenfeatures for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 18, n. 8, p. 831–836, 1996.
- [50] CHANG, Y.; HU, C.; TURK, M. Manifold of facial expression. In: IEEE COMPUTER SOCIETY. *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*. [S.l.], 2003. p. 28.
- [51] COMON, P. Independent component analysis, a new concept? *Signal processing*, Elsevier, v. 36, n. 3, p. 287–314, 1994.
- [52] BARTLETT, M. S.; MOVELLAN, J. R.; SEJNOWSKI, T. J. Face recognition by independent component analysis. *IEEE Transactions on neural networks*, IEEE, v. 13, n. 6, p. 1450–1464, 2002.
- [53] DRAPER, B. A. et al. Recognizing faces with pca and ica. *Computer vision and image understanding*, Elsevier, v. 91, n. 1, p. 115–137, 2003.
- [54] LIU, C.; WECHSLER, H. Comparative assessment of independent component analysis (ica) for face recognition. In: CITESEER. *International conference on audio and video based biometric person authentication*. [S.l.], 1999.
- [55] FOON, N. H.; JIN, A. T. B.; LING, D. N. C. Face recognition using wavelet transform and non-negative matrix factorization. In: SPRINGER. *Australasian Joint Conference on Artificial Intelligence*. [S.l.], 2004. p. 192–202.
- [56] LIU, W. et al. Nearest intra-class space classifier for face recognition. In: IEEE. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. [S.l.], 2004. v. 4, p. 495–498.
- [57] KIM, J.; CHOI, J.; YI, J. Face recognition based on locally salient ica information. In: SPRINGER. *International Workshop on Biometric Authentication*. [S.l.], 2004. p. 1–9.

- [58] YI, J. et al. Face recognition based on ica combined with fld. In: SPRINGER. *International Workshop on Biometric Authentication*. [S.l.], 2002. p. 10–18.
- [59] LAWRENCE, S. et al. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, IEEE, v. 8, n. 1, p. 98–113, 1997.
- [60] ELEYAN, A.; DEMIREL, H. Face recognition system based on pca and feedforward neural networks. In: SPRINGER. *International Work-Conference on Artificial Neural Networks*. [S.l.], 2005. p. 935–942.
- [61] LI, B.; YIN, H. Face recognition using rbf neural networks and wavelet transform. In: SPRINGER. *International Symposium on Neural Networks*. [S.l.], 2005. p. 105–111.
- [62] MELIN, P.; FELIX, C.; CASTILLO, O. Face recognition using modular neural networks and the fuzzy sugeno integral for response integration. *International journal of intelligent systems*, Wiley Online Library, v. 20, n. 2, p. 275–291, 2005.
- [63] ZHANG, G. et al. Boosting local binary pattern (lbp)-based face recognition. In: *Advances in biometric person authentication*. [S.l.]: Springer, 2004. p. 179–186.
- [64] MOGHADDAM, B.; NASTAR, C.; PENTLAND, A. A bayesian similarity measure for direct image matching. In: IEEE. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. [S.l.], 1996. v. 2, p. 350–358.
- [65] FREUND, Y.; SCHAPIRE, R. E. A desicion-theoretic generalization of on-line learning and an application to boosting. In: SPRINGER. *European conference on computational learning theory*. [S.l.], 1995. p. 23–37.
- [66] AHONEN, T.; HADID, A.; PIETIKÄINEN, M. Face recognition with local binary patterns. In: SPRINGER. *European conference on computer vision*. [S.l.], 2004. p. 469–481.
- [67] MOREIRA, M.; MAYORAZ, E. Improved pairwise coupling classification with correcting classifiers. In: SPRINGER. *European conference on machine learning*. [S.l.], 1998. p. 160–171.
- [68] LI, H.; QI, F.; WANG, S. Face recognition with improved pairwise coupling support vector machines. In: SPRINGER. *International Work-Conference on Artificial Neural Networks*. [S.l.], 2005. p. 927–934.
- [69] LI, Z.; TANG, S. Face recognition using improved pairwise coupling support vector machines. In: IEEE. *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on*. [S.l.], 2002. v. 2, p. 876–880.
- [70] PLATT, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, Cambridge, MA, v. 10, n. 3, p. 61–74, 1999.
- [71] LI, H.; WANG, S.; QI, F. Automatic face recognition by support vector machines. In: SPRINGER. *International Workshop on Combinatorial Image Analysis*. [S.l.], 2004. p. 716–725.

- [72] DAI, G.; ZHOU, C. Face recognition using support vector machines with the robust feature. In: IEEE. *Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003. The 12th IEEE International Workshop on*. [S.l.], 2003. p. 49–53.
- [73] DÉNIZ, O.; CASTRILLON, M.; HERNÁNDEZ, M. Face recognition using independent component analysis and support vector machines. *Pattern recognition letters*, Elsevier, v. 24, n. 13, p. 2153–2157, 2003.
- [74] JONSSON, K. et al. Learning support vectors for face verification and recognition. In: IEEE. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. [S.l.], 2000. p. 208–213.
- [75] RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, IEEE, v. 77, n. 2, p. 257–286, 1989.
- [76] SAMARIA, F. S.; HARTER, A. C. Parameterisation of a stochastic model for human face identification. In: IEEE. *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*. [S.l.], 1994. p. 138–142.
- [77] SAMARIA, F. S. *Face recognition using hidden Markov models*. Tese (Doutorado) — University of Cambridge, 1994.
- [78] NEFIAN, A.; HAYES, M. Face recognition using an embedded hmm. In: *IEEE Conference on Audio and Video-based Biometric Person Authentication*. [S.l.: s.n.], 1999. p. 19–24.
- [79] ROLI, F.; KITTLER, J. *Multiple Classifier Systems: Third International Workshop, MCS 2002, Cagliari, Italy, June 24-26, 2002. Proceedings*. [S.l.]: Springer Science & Business Media, 2002.
- [80] LU, X.; WANG, Y.; JAIN, A. K. Combining classifiers for face recognition. In: IEEE. *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*. [S.l.], 2003. v. 3, p. III–13.
- [81] MARCIALIS, G. L.; ROLI, F. Fusion of lda and pca for face recognition. *Department of Electrical and Electronic Engineering, University of Cagliari, Piazza diArmi*, Citeseer, 2002.
- [82] MARCIALIS, G. L.; ROLI, F. Fusion of lda and pca for face verification. In: *Proceedings of the International ECCV 2002 Workshop Copenhagen on Biometric Authentication*. London, UK, UK: Springer-Verlag, 2002. (ECCV '02), p. 30–38. ISBN 3-540-43723-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=645307.649127>>.
- [83] MARCIALIS, G. L.; ROLI, F. Fusion of appearance-based face recognition algorithms. *Pattern Analysis and Applications*, Springer, v. 7, n. 2, p. 151–163, 2004.
- [84] ACHERMANN, B.; BUNKE, H. *Combination of classifiers on the decision level for face recognition*. [S.l.]: Citeseer, 1996.
- [85] TOLBA, A. S.; ABU-REZQ, A. Combined classifiers for invariant face recognition. *Pattern Analysis & Applications*, Springer, v. 3, n. 4, p. 289–302, 2000.

- [86] WAN, Y. et al. Video program clustering indexing based on face recognition hybrid model of hidden markov model and support vector machine. In: SPRINGER. *International Workshop on Combinatorial Image Analysis*. [S.l.], 2004. p. 739–749.
- [87] KWAK, K.-C.; PEDRYCZ, W. Face recognition: A study in information fusion using fuzzy integral. *Pattern Recognition Letters*, Elsevier, v. 26, n. 6, p. 719–733, 2005.
- [88] JEBARA, T. S. *3D pose estimation and normalization for face recognition*. Tese (Doutorado) — McGill University, 1995.
- [89] YANG, Q.; TANG, X. Recent advances in subspace analysis for face recognition. In: *Advances in Biometric Person Authentication*. [S.l.]: Springer, 2004. p. 275–287.
- [90] PHILLIPS, P. J. et al. *FERET (face recognition technology) recognition algorithm development and test results*. [S.l.]: Army Research Laboratory Adelphi, MD, 1996.
- [91] PHILLIPS, P. J. et al. The feret evaluation methodology for face-recognition algorithms. In: IEEE. *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. [S.l.], 1997. p. 137–143.
- [92] TIBBALDS, A. D. *Three dimensional human face acquisition for recognition*. Tese (Doutorado) — University of Cambridge, 1998.
- [93] HESHER, C.; SRIVASTAVA, A.; ERLEBACHER, G. A novel technique for face recognition using range imaging. In: IEEE. *Signal processing and its applications, 2003. Proceedings. Seventh international symposium on*. [S.l.], 2003. v. 2, p. 201–204.
- [94] GORDON, G. G. Face recognition based on depth maps and surface curvature. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *San Diego, '91, San Diego, CA*. [S.l.], 1991. p. 234–247.
- [95] BEUMIER, C.; ACHEROY, M. Automatic 3d face authentication. *Image and Vision Computing*, Elsevier, v. 18, n. 4, p. 315–321, 2000.
- [96] CASTELLANI, U. et al. 3d face recognition using stereoscopic vision. In: *Advanced Studies in Biometrics*. [S.l.]: Springer, 2005. p. 126–137.
- [97] LEE, S.; WOLBERG, G.; SHIN, S. Y. Scattered data interpolation with multilevel b-splines. *IEEE transactions on visualization and computer graphics*, IEEE, v. 3, n. 3, p. 228–244, 1997.
- [98] PAN, G.; WU, Z.; PAN, Y. Automatic 3d face verification from range data. In: IEEE. *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). 2003 IEEE International Conference on*. [S.l.], 2003. v. 3, p. III–193.
- [99] XU, C. et al. Automatic 3d face recognition combining global geometric features with local shape variation information. In: IEEE. *Automatic face and gesture recognition, 2004. Proceedings. Sixth IEEE international conference on*. [S.l.], 2004. p. 308–313.

- [100] LEE, Y. et al. Local feature based 3d face recognition. In: SPRINGER. *International Conference on Audio-and Video-based Biometric Person Authentication*. [S.l.], 2005. p. 909–918.
- [101] AL-OSAIMI, F. R.; BENNAMOUN, M.; MIAN, A. Integration of local and global geometrical cues for 3d face recognition. *Pattern Recognition*, Elsevier, v. 41, n. 3, p. 1030–1040, 2008.
- [102] CARTOUX, J.-Y.; LAPRESTÉ, J.-T.; RICHTIN, M. Face authentication or recognition by profile extraction from range images. In: IEEE. *Interpretation of 3D Scenes, 1989. Proceedings., Workshop on*. [S.l.], 1989. p. 194–199.
- [103] NAGAMINE, T.; UEMURA, T.; MASUDA, I. 3d facial image analysis for human identification. In: IEEE. *Pattern Recognition, 1992. Vol. I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*. [S.l.], 1992. p. 324–327.
- [104] LI, C.; BARRETO, A. Profile-based 3d face registration and recognition. In: SPRINGER. *International Conference on Information Security and Cryptology*. [S.l.], 2004. p. 478–488.
- [105] WU, Y.; PAN, G.; WU, Z. Face authentication based on multiple profiles extracted from range data. In: SPRINGER. *International Conference on Audio-and Video-Based Biometric Person Authentication*. [S.l.], 2003. p. 515–522.
- [106] WANG, J.-G.; TOH, K.-A.; VENKATESWARLU, R. Fusion of appearance and depth information for face recognition. In: SPRINGER. *International Conference on Audio-and Video-Based Biometric Person Authentication*. [S.l.], 2005. p. 919–928.
- [107] TSALAKANIDOU, F.; TZOVARAS, D.; STRINTZIS, M. G. Use of depth and colour eigenfaces for face recognition. *Pattern Recognition Letters*, Elsevier, v. 24, n. 9, p. 1427–1435, 2003.
- [108] CHANG, K.; BOWYER, K.; FLYNN, P. Face recognition using 2d and 3d facial data. In: CITESEER. *ACM Workshop on Multimodal User Authentication*. [S.l.], 2003. p. 25–32.
- [109] BEUMIER, C.; ACHEROY, M. Face verification from 3d and grey level clues. *Pattern recognition letters*, Elsevier, v. 22, n. 12, p. 1321–1329, 2001.
- [110] BESL, P. J.; MCKAY, N. D. Method for registration of 3-d shapes. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Robotics-DL tentative*. [S.l.], 1992. p. 586–606.
- [111] PAPTAEODOROU, T.; RUECKERT, D. Evaluation of automatic 4d face recognition using surface and texture registration. In: IEEE. *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*. [S.l.], 2004. p. 321–326.
- [112] LU, X.; JAIN, A. K. Integrating range and texture information for 3d face recognition. In: IEEE. *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*. [S.l.], 2005. v. 1, p. 156–163.
- [113] WANG, Y.; CHUA, C.-S.; HO, Y.-K. Facial feature detection and face recognition from 2d and 3d images. *Pattern Recognition Letters*, Elsevier, v. 23, n. 10, p. 1191–1202, 2002.

- [114] CUTLER, R. G. *Face recognition using infrared images and eigenfaces*. [S.l.]: Citeseer, 1996.
- [115] GYAOUROVA, A.; BEBIS, G.; PAVLIDIS, I. Fusion of infrared and visible images for face recognition. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2004. p. 456–468.
- [116] SELINGER, A.; SOCOLINSKY, D. A. *Appearance-based facial recognition using visible and thermal imagery: a comparative study*. [S.l.], 2006.
- [117] WILDER, J. et al. Comparison of visible and infra-red imagery for face recognition. In: IEEE. *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*. [S.l.], 1996. p. 182–187.
- [118] SIM, T. et al. Memory-based face recognition for visitor identification. In: IEEE. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. [S.l.], 2000. p. 214–220.
- [119] KEVIN, X. C. P. J. F.; BOWYER, W. Visible-light and infrared face recognition. In: CITESEER. *Workshop on Multimodal User Authentication*. [S.l.], 2003. p. 48.
- [120] IDENTIX. *Identix Inc.: Electronic Documentation*.
- [121] KITTLER, J. et al. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 20, n. 3, p. 226–239, 1998.
- [122] CHEN, X.; FLYNN, P. J.; BOWYER, K. W. Ir and visible light face recognition. *Computer Vision and Image Understanding*, Elsevier, v. 99, n. 3, p. 332–358, 2005.
- [123] SINGH, R.; VATSA, M.; NOORE, A. Hierarchical fusion of multi-spectral face images for improved recognition performance. *Information Fusion*, Elsevier, v. 9, n. 2, p. 200–210, 2008.
- [124] HEO, J. et al. Fusion of visual and thermal signatures with eyeglass removal for robust face recognition. In: IEEE. *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*. [S.l.], 2004. p. 122–122.
- [125] SINGH, R.; VATSA, M.; NOORE, A. Integrated multilevel image fusion and match score fusion of visible and infrared face images for robust face recognition. *Pattern Recognition*, Elsevier, v. 41, n. 3, p. 880–893, 2008.
- [126] HG, R. et al. An rgb-d database using microsoft's kinect for windows for face detection. In: IEEE. *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*. [S.l.], 2012. p. 42–46.
- [127] RUSU, R. B. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, Springer, v. 24, n. 4, p. 345–348, 2010.