



# Modelos de Processamento de Imagem, com Múltiplas Fontes de Aquisição, para Manipulação Aplicada à Domótica

---

*Frederico José Lapa Grilo*

Tese apresentada à Universidade de Évora  
para obtenção do Grau de Doutor em Engenharia Mecatrónica e Energia  
Especialidade: Engenharia Mecatrónica

ORIENTADOR : *Professor Doutor João M. Gouveia Figueiredo*

ÉVORA, Julho 2019





# **Modelos de Processamento de Imagem, com Múltiplas Fontes de Aquisição, para Manipulação Aplicada à Domótica**

*Frederico José Lapa Grilo*

Tese apresentada à Universidade de Évora  
para obtenção do Grau de Doutor em Engenharia Mecatrónica e Energia  
Especialidade: Engenharia Mecatrónica

ORIENTADOR : *Professor Doutor João M. Gouveia Figueiredo*

ÉVORA, JULHO 2019



## JÚRI

**Presidente:** (por delegação de competências do Diretor do Instituto de Investigação e Formação Avançada)

Doutor **António Domingos Heitor da Silva Reis**,  
Professor Catedrático, da Universidade de Évora.

## VOGAIS

Doutor **João Miguel da Costa Sousa**,  
Professor Catedrático, Instituto Superior Técnico, Universidade de Lisboa;

Doutor **Miguel Afonso Dias de Ayala Botto**,  
Professor Catedrático, Instituto Superior Técnico, Universidade de Lisboa;

Doutor **João Manuel Gouveia de Figueiredo**,  
Professor Associado c/Agregação, da Universidade de Évora, orientador;

Doutor **João Francisco Alves Martins**,  
Professor Associado, FCT da Universidade Nova de Lisboa;

Doutor **Fernando Manuel Tim Tim Janeiro**,  
Professor Auxiliar, da Universidade de Évora.



## Resumo

Este trabalho foca-se em modelos de processamento de imagem para utilização na visão por computador. Modelos de processamento de imagem com multi-aquisição e/ou em multi-perspectiva, para um conhecimento do meio circundante, com possibilidade de comando e controlo na área da domótica e/ou robótica móvel. Os algoritmos desenvolvidos têm a capacidade de serem implementados em blocos de *software* ou *hardware*, de forma independente (autónomos), ou integrados como componentes de um sistema mais complexo. O desenvolvimento dos algoritmos privilegiou o seu elevado desempenho, constringido pela minimização da carga computacional.

Nos modelos de processamento de imagem desenvolvidos foram focados 4 tópicos fundamentais de investigação: a) detecção de movimento de objectos e seres humano em ambiente não controlado; b) detecção da face humana, a ser usada como variável de controlo (entre outras aplicações); c) capacidade de utilização de multi-fontes de aquisição e processamento de imagem, com diferentes condições de iluminação não controladas, integradas num sistema complexo com diversas topologias; d) capacidade de funcionamento de forma autónoma ou em rede distribuída, apenas comunicando resultados finais, ou integrados modularmente na solução final de sistemas complexos de aquisição de imagem.

A implementação laboratorial, com teste em protótipos, foi ferramenta decisiva no melhoramento de todos os algoritmos desenvolvidos neste trabalho.

## Palavras-Chave

Visão por computador

Processamento digital de imagem

Aquisição de imagem

Domotica

Robótica



## **IMAGE PROCESSING MODELS, WITH MULTIPLE ACQUISITION SOURCES, FOR MANIPULATION IN DOMOTICS**

### **Abstract**

This work focuses on image processing models for computer vision. Image processing models with multi-acquisition and/or multi-perspective models were developed to acquire knowledge over the surrounding environment, allowing system control in the field of domotics and/or mobile robotics. The developed algorithms have the capacity to be implemented in software or hardware blocks, independently (autonomous), or integrated as a component in more complex systems. The development of the algorithms was focused on high performance constrained by the computational burden minimization.

In the developed image processing models it were addressed 4 main research topics: a) movement detection of objects and human beings in an uncontrolled environment; b) detection of the human face to be used as a control variable (among other applications); c) possibility of using multi-sources of acquisition and image processing, with different uncontrolled lighting conditions, integrated into a complex system with different topologies; d) ability to work as an autonomous entity or as a node integrated on a distributed network, only transmitting final results, or integrated as a link in a complex image processing system.

The laboratorial implementation, with prototype tests, was the main tool for the improvement of all developed algorithms, discussed in the present work.

### **Keywords**

Computer vision

Digital image processing

Image acquisition

Domotics

Robotics





## Agradecimentos

Tendo em conta todo o processo que foi o desenvolvimento da tese, todo o tempo, todo o trabalho realizado, torna-se de grande importância expressar os meus agradecimentos.

Ao Professor Doutor João Figueiredo que foi mais que um orientador, que me apontou sempre o melhor caminho, que a sua capacidade e orientação científica foi fundamental, que em todo o trabalho desenvolvido a sua orientação e ajuda foram muito importantes e como pessoa tem uma forma de comunicação e uma facilidade de intersecção excepcionais.

Ao Professor Doutor Octávio Páscoa Dias agradeço a sua colaboração e orientação e nos primeiros trabalhos que contribuíram de forma importante para a tese final.

A todos os meus colegas do DEE da Escola Superior de Tecnologia do Instituto Politécnico de Setúbal, pelos conselhos, sugestões e apoio.

Aos colegas do Departamento de Física da Escola de Ciências e Tecnologia da Universidade de Évora, pelo apoio e acolhimento para o desenvolvimento dos trabalhos.

Agradeço à minha Família:

Ao meu avô Hermínio que entre muitas coisas, me transmitiu a vontade e o prazer de fazermos coisas com as nossas mãos e que elas funcionem.

À minha avó Maria, a qual sempre teve paciência de me ouvir nos temas que me interessavam e que sempre me fez ver o quanto o saber é importante.

Aos meus pais por ser tão importantes na minha vida, ao meu pai, que desde de miúdo me mostrou o quanto é importante saber escolher o que queremos ser e trabalhar de forma a sermos sempre melhores, à minha mãe que sempre me apoiou nas minhas escolhas, sempre acreditou que eu fazia bem em querer ser engenheiro e que na década de 80 do século passado me comprou o meu primeiro computador o ZX Spectrum Sinclair.

À minha mulher e companheira Daniele por todo o apoio, paciência e carinho que me dá.

À Anabela a mãe dos meus filhos e que foi minha esposa durante 14 anos pelo apoio e por tudo em que me ajudou.

Aos meus filhos, Gonçalo e Luís, nos quais tenho grande orgulho e que tiveram que por vezes que “partilhar” o seu tempo com o meu trabalho, espero que me perdoem e que pelo menos os tenha motivado...



## Índice

Capítulo 1 Introdução .....	27
1.1 Objectivos .....	28
1.2 Algoritmos propostos e Contribuições científicas .....	29
1.3 Publicações .....	30
1.3.1 Capítulo em Livro .....	30
1.3.2 Artigo em Revistas Internacionais com arbitragem .....	30
1.3.3 Artigos em “Proceedings” de Congressos Internacionais com arbitragem .....	30
1.4 Estado da Arte.....	31
1.4.1 A visão por computador.....	31
1.4.2 O volume de dados no processamento de imagem .....	33
1.4.3 A visão artificial na domótica .....	35
1.4.3.1 A detecção de movimento.....	37
1.4.3.2 A face humana como elemento de comando .....	38
1.4.3.3 Detecção de pele em imagens .....	39
1.4.3.4 Detecção da face em imagens .....	40
1.5 Estrutura da tese .....	41
Capítulo 2 Visão por computador e Processamento de imagem .....	43
2.1 Visão por computador .....	44
2.2 Processamento de imagem.....	45
2.2.1 As origens históricas do processamento digital de imagem .....	45
2.2.2 Processamento digital de imagem .....	50
2.2.2.1 Sistemas .....	50
2.2.2.1.2 Sistemas lineares .....	50
2.2.2.1.3 Sistemas invariante no tempo.....	51
2.2.2.1.4 Sistemas Linear e Invariante no Tempo (SLIT).....	51
2.2.2.1.5 Resposta de um SLIT.....	51
2.2.2.2 Digitalização .....	53
2.2.2.3 Segmentação .....	54
2.2.2.4 Imagens binárias.....	55
2.2.2.5 Espaços de Cor.....	55
2.2.2.5.1 RGB .....	56

2.2.2.5.2 RGB Normalizado .....	57
2.2.2.5.3 YCbCr .....	58
2.2.2.5.4 YIQ .....	58
2.2.2.5.5 HSV (HSL).....	59
2.2.2.5.6 YUV .....	60
2.2.2.5.7 YDbDr.....	62
2.2.2.5.8 CIE L * a * b .....	63
2.2.2.6 Binarização (segmentação monocromática) .....	64
2.2.2.7 Definição de pixel vizinho .....	66
2.2.2.8 Definição de Caminhos.....	67
2.2.2.9 Definição de Fronteira .....	67
2.2.2.10 Definição de Distância.....	68
2.3 Algoritmos de processamento de imagem.....	69
2.3.1 Algoritmo de filtro de tamanho.....	69
2.3.2 Algoritmo de filtro morfológico Erosão e dilatação .....	71
2.3.3 Algoritmo de Igualização do histograma.....	72
2.3.4 Algoritmo segmentação dos níveis de cinzento.....	73
2.3.5 Algoritmos de filtro com elemento estruturante .....	73
2.3.5.1 Algoritmo de filtro de mediana .....	74
2.3.5.2 Algoritmo de filtro de média.....	75
2.3.5.3 Algoritmo de filtro de mínimo .....	75
2.3.5.3 Algoritmo de filtro de Máximo .....	75
2.3.5.4 Algoritmo de filtro de Sobel .....	75
2.3.6 Algoritmo para detecção de arestas.....	76
2.3.6.1 Algoritmo Gradiente.....	77
Capítulo 3 Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células .....	81
3.1 Introdução .....	82
3.2 Algoritmo de Processamento de Imagem.....	83
3.2.1 Descrição do algoritmo.....	83
3.2.2 Redução do algoritmo.....	84
3.3 Implementação do sistema.....	85
3.4 Análise e conclusões .....	88
3.4.1 Testes realizados .....	88

3.4.2 Conclusões .....	93
Capítulo 4 Detecção de pele em Imagens a cores por Método adaptativo .....	97
4.1 Introdução .....	98
4.2 Algoritmo de Processamento de Imagem.....	100
4.2.1 Algoritmo Base de detecção de pele.....	100
4.2.2 Algoritmo adaptativo .....	101
4.3 Implementação do sistema.....	105
4.3.1 Funcionalidade de variação do brilho da imagem.....	108
4.3.2 Algoritmo da funcionalidade de validação de pixel de pele para detecção de face .....	110
4.4. Testes realizados .....	112
4.5 Conclusões .....	119
Capítulo 5 Detecção da variação de posição da face humana por enquadramento de rectângulo áureo com Método adaptativo.....	123
5.1 Introdução .....	124
5.1.1 O número de ouro ou razão divina .....	124
5.2 Algoritmo de Processamento de Imagem.....	126
5.2.1 Detecção de pele.....	126
5.2.1.1 Algoritmo de detecção de pele com Método adaptativo.....	126
5.2.2 Detecção da variação da posição da face.....	126
5.2.2.1 Detecção da posição da face.....	126
5.2.2.2 Detecção da variação da posição da face .....	127
5.2.2.3 Área de detecção adaptativa da pele (Face) .....	127
5.3 Implementação do sistema.....	129
5.3.1 Algoritmo A .....	131
5.3.1.1 Área de imagens do Sistema A.....	131
5.3.1.2 Detecção e validação dos pixels de pele e sua detecção no bloco de face	133
5.3.2 Algoritmo B .....	133
5.3.2.1 Área de imagens do Sistema B .....	133
5.3.3 Sistema de utilização prática de posicionamento 2D .....	136
5.4 Análise e conclusões .....	136
5.4.1 Testes realizados .....	136
5.4.2 Análise com o factor Af.....	140
5.4.3 Conclusões .....	140

Capítulo 6	Visão por computador e robôs móveis	143
6.1	Introdução	144
6.1.1	Introdução histórica dos AGVs	144
6.1.2	Métodos de navegação dos AGVs	145
6.1.2.1	Navegação fixa	146
6.1.2.1.1	Orientação indutiva	146
6.1.2.1.2	Orientação óptica	147
6.1.2.2	Navegação aberta	148
6.1.2.2.1	Navegação por marcas ou pontos de ancoragem	148
6.1.2.2.1.1	Navegação por marcas magnéticas	148
6.1.2.2.1.2	Navegação por marcas de radiofrequência	149
6.1.2.2.1.2	Navegação por marcas visuais	149
6.1.2.2.2	Navegação por laser	149
6.1.2.2.2.1	Navegação por laser artificial (laser com marcas artificiais)	150
6.1.2.2.2.2	Navegação por contorno (navegação a laser com marcas naturais)	150
6.1.2.2.3	Navegação por sonar	152
6.1.2.2.4	Navegação por visão com câmara	152
6.1.2.2.5	Navegação GPS	152
6.2	Implementação do sistema	153
6.2.1	Caracterização do AGV	153
6.2.2	A Estratégia de Controlo do AGV - <i>Master/ Slave</i>	157
6.2.2.1	Programa LabView do <i>PC-Master</i>	158
6.2.2.2	Programa LabView do <i>PC-Slave</i>	160
6.2.2.3	Programa do Arduino	162
6.2.3	Processamento de imagem para seguimento automático de trajectória	164
6.2.4	Processamento de imagem para controlo do AGV por movimento da face humana	166
6.3	Testes e Conclusões	169
Capítulo 7	Conclusões e Trabalho Futuro	171
7.1	Conclusões	172
7.1.1	Redução de Volume de dados - Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células	173
7.1.2	Redução de Complexidade - Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células	173

7.1.3 Avaliação da robustez - Detecção de pele em Imagens a cores por Método adaptativo .....	174
7.1.4 Redução de Volume de dados e complexidade - Detecção de pele em Imagens a cores por Método adaptativo .....	174
7.1.5 Utilização da face como elemento de comando.....	175
7.2 Trabalho Futuro.....	175
Referências Bibliograficas.....	181



## Lista de figuras

<i>Fig. 1.1 Comparação da capacidade de armazenamento, projecto digitalização do cérebro humano, (Jeff Lichtman, da Universidade de Harvard &amp; Daniel Berge MIT)</i> .....	35
<i>Fig. 2.1 Modelo de Visão por Computador</i> .....	45
<i>Fig. 2.2 Imagem produzida em 1921 a partir de uma fita codificada por um telégrafo impressora com fontes de caracteres especiais (IEEE)</i> .....	46
<i>Fig. 2.3 A primeira imagem da Lua captada pela Sonda Ranger 7 em 31 de Julho de 1964 às 9:09 A.M. hora da costa Leste, as marcas na imagem, são marcas de referência usados para correcções geométricas (NASA)</i> .....	48
<i>Fig. 2.4 Sistemas de tempo contínuo e tempo discreto</i> .....	50
<i>Fig. 2.5 Resposta do SLIT em tempo contínuo e discreto</i> .....	52
<i>Fig. 2.6 Resposta do SLIT em frequência</i> .....	52
<i>Fig. 2.7 Digitalização de uma imagem</i> .....	54
<i>Fig. 2.8 Imagem a cores, imagens em tons de cinzento e imagem binarizada</i> .....	55
<i>Fig. 2.9 Espaço de cor RGB (<a href="http://www.graficacriasett.com.br/o-que-e-cromia-cmyk/rgb/">www.graficacriasett.com.br/o-que-e-cromia-cmyk/rgb/</a>, acedido em Maio 2019)</i> .....	56
<i>Fig. 2.10 Espaço de cor RGB (<a href="https://www.researchgate.net/figure/RGB-left-and-HSV-right-color-spaces_fig1_310474598">https://www.researchgate.net/figure/RGB-left-and-HSV-right-color-spaces_fig1_310474598</a>, acedido em Maio de 2019)</i> .....	57
<i>Fig. 2.11 Espaço de cor YCbCr (<a href="http://en.wikipedia.org/wiki/YCbCr">en.wikipedia.org/wiki/YCbCr</a>, acedido em Maio de 2018)</i> .....	58
<i>Fig. 2.12 Espaço de cor YIQ (<a href="http://en.wikipedia.org/wiki/YIQ">en.wikipedia.org/wiki/YIQ</a>, acedido em Maio de 2018)</i> .....	59
<i>Fig. 2.13 Espaço de cor HSV (<a href="http://www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html">www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html</a>, acedido em Maio de 2019)</i> .....	60
<i>Fig. 2.14 Espaço de cor YUV (<a href="http://en.wikipedia.org/wiki/YUV">en.wikipedia.org/wiki/YUV</a>, acedio em Maio de 2018)</i> .....	61
<i>Fig. 2.15 Espaço de cor YDbDr (<a href="http://en.wikipedia.org/wiki/YDbDr">en.wikipedia.org/wiki/YDbDr</a>, acedido em Maio de 2018)</i> .....	62
<i>Fig. 2.16 Espaço de cor CIE <math>L^*a^*b</math> (<a href="http://www.researchgate.net/figure/Figura-26-Espaco-de-cor-CIELab_fig4_235438609">www.researchgate.net/figure/Figura-26-Espaco-de-cor-CIELab_fig4_235438609</a>, acedido em Maio de 2019)</i> .....	64

<i>Fig. 2.17 a. Matriz original, b. Matriz depois da Binarização.</i> .....	65
<i>Fig. 2.18 Matriz no formado binário (0,1)</i> .....	65
<i>Fig. 2.19 Binarização.</i> .....	66
<i>Fig. 2.20 Pixel vizinho</i> .....	66
<i>Fig. 2.21 Caminhos: 4-Caminho e 8-Caminhos</i> .....	67
<i>Fig. 2.22 Pixeis Fronteira</i> .....	67
<i>Fig. 2.23.a Distância euclidiana</i> .....	68
<i>Fig. 2.23.b Distância quarteirão</i> .....	68
<i>Fig. 2.23.c Distância tabuleiro de xadrez</i> .....	69
<i>Fig. 2.24 Imagem original com ruído.</i> .....	70
<i>Fig. 2.25 Imagem sem ruído</i> .....	70
<i>Fig. 2.26 Filtro morfológico erosão – dilatação.</i> .....	71
<i>Fig. 2.27 Imagem original com ruído.</i> .....	71
<i>Fig. 2.27.a Aplicação da erosão-dilatação</i> .....	72
<i>Fig. 2.27.b Aplicação da dilatação-erosão</i> .....	72
<i>Fig. 2.28 Igualização do histograma</i> .....	73
<i>Fig. 2.29 Convolução discreta no domínio espacial.</i> .....	74
<i>Fig. 2.30 Descontinuidade na intensidade da imagem</i> .....	76
<i>Fig. 2.31 Detecção de arestas</i> .....	78
<i>Fig. 3.1 Células de trabalho com a dimensão <math>P/2 \times Q/2</math>.</i> .....	83
<i>Fig. 3.2 Linhas de detecção 1 – 4.</i> .....	84
<i>Fig. 3.3 Algoritmo de processamento de imagem.</i> .....	85
<i>Fig. 3.4 Area 1</i> .....	86
<i>Fig. 3.5 Area 2</i> .....	87
<i>Fig. 3.6 Resultado dos testes.</i> .....	89
<i>Fig. 3.7 Teste dos frames de 1 a 6.</i> .....	90
<i>Fig. 3.8 Numero de pixeis testados com e sem o algoritmo adaptativo</i> .....	91
<i>Fig. 3.9 Valor do factor <math>a_i</math>.</i> .....	91
<i>Fig. 3.10 Variação do limiar dinâmico (threshold) <math>\tau_i</math>.</i> .....	92
<i>Fig. 3.11 Total de pixeis a processar vs Total de pixeis para cada resolução.</i> ..	94
<i>Fig. 4.1 Topologia de malha fechada.</i> .....	99

<i>Fig. 4.2 Topologia de malha aberta.....</i>	<i>99</i>
<i>Fig. 4.3 Utilização do algoritmo Base de detecção de pixel de pele.....</i>	<i>101</i>
<i>Fig. 4.4 Integração do algoritmo adaptativo no algoritmo base.....</i>	<i>102</i>
<i>Fig. 4.5 Calculo do valor médio em cada componente: de MnVIR, MnVIG e MnVIB.....</i>	<i>103</i>
<i>Fig. 4.6 Os 3 níveis dos factores adaptativos Nr, Ng, Nb com <math>\alpha = 0,06</math>.....</i>	<i>105</i>
<i>Fig. 4.7 Interface do utilizador. ....</i>	<i>106</i>
<i>Fig. 4.8 Interface do utilizador (Área 1). ....</i>	<i>107</i>
<i>Fig. 4.9 Interface do utilizador (Área 2). ....</i>	<i>108</i>
<i>Fig. 4.10 O fluxograma da funcionalidade de variação do brilho da imagem. ....</i>	<i>109</i>
<i>Fig. 4.11 A implementação da funcionalidade de variação do brilho da imagem (desenvl. em LabVIEW).....</i>	<i>109</i>
<i>Fig. 4.12 Funcionalidade de validação de pixel de pele e detecção de face..</i>	<i>111</i>
<i>Fig. 4.13 Validação de pixel verdadeiro de pele (passos 1 a 7). ....</i>	<i>111</i>
<i>Fig. 4.14 Funcionalidade de validação de pixel de pele e detecção de face (implementado em LabVIEW). ....</i>	<i>112</i>
<i>Fig. 4.15 Detecção de pele sem (esquerda) e com (direita) algoritmo adaptativo, com Lum = 0.45, Pixeis de pele-detectados (total skin pxs) e <b>Pixeis de pele-verdadeiros-positivos-da-face</b> (true skin pixel), para uma imagem com o numero de <b>Pixeis total-de pele-da-face</b> de 19.800.....</i>	<i>113</i>
<i>Fig. 4.16 Valores médios de R, G e B; valores NR, NG e NB. ....</i>	<i>114</i>
<i>Fig. 4.17 Detecção de pele e detecção de face sem (cima) e com (baixo) algoritmo adaptativo, Total Pxs pele (Adp on/Adp off) = 114% e Total Pxs pele-verdadeiros-post-face (Adp on/Adp off) = 120%. Fonte de iluminação: lâmpada fluorescente no tecto. ....</i>	<i>115</i>
<i>Fig. 4.18 Detecção de pele e detecção de face sem (cima) e com (baixo) algoritmo adaptativo, Total Pxs pele (Adp on/Adp off) = 215% e Total Pxs pele-verdadeiros-post-face (Adp on/Adp off) = 291%. Fonte de iluminação: lâmpada de halogéneo de trás.....</i>	<i>115</i>
<i>Fig. 4.19 Detecção de pele e detecção de face sem (cima) e com (baixo) algoritmo adaptativo, Total Pxs pele (Adp on/Adp off) = 2230% e Total Pxs pele</i>	

<i>verdadeiros-post-face (Adp on/Adp off) = n.c.. Fonte de iluminação: iluminação directa da luz do monitor do computador. ....</i>	116
<i>Fig. 4.20 Resultado dos testes com Lum = 1,1. ....</i>	117
<i>Fig. 4.21 Resultado dos teste com Lum = 0,75. ....</i>	117
<i>Fig. 4.22 Resultado dos testes com Lum = 0,35. ....</i>	118
<i>Fig. 4.23 Resultado dos testes com variação R = G = B = 1,1 de forma independente, razão dos pixels pele-verdadeiros-positivos-da-face com e sem algoritmo (Adp On /Apt Off). ....</i>	119
<i>Fig. 4.24 Factores Nr, Ng e Nb com R = 0,50, G = 1,0 e B = 1,0. ....</i>	120
<i>Fig. 4.25 Factores Nr, Ng e Nb com R = G = B = 1,0. ....</i>	121
<i>Fig. 4.26 Desempenho do algoritmo adaptativo nas três condições de iluminação. ....</i>	122
<i>Fig. 5.1 Rectângulo áureo na face humana (webeduc.mec.gov.br/portaldoprofessor/matematica/condigital2/guias/Guia_aud iovisual_i.pdf, acedido em Maio de 2019). ....</i>	125
<i>Fig. 5.2 Detecção da posição da face com o rectângulo áureo. ....</i>	127
<i>Fig. 5.3 Detecção variação da posição da face em relação à origem de um referencial do centro da imagem, <math>\vec{P} = \vec{P}_x + \vec{P}_y</math>. ....</i>	127
<i>Fig. 5.4 Variação da área de detecção de pixels de pele Ad em função da variação de posição da área da face detectada nas iterações anteriores dx. ....</i>	128
<i>Fig. 5.5 Processo adaptativo da variação da área de detecção de pixels de pele Ad. ....</i>	129
<i>Fig. 5.6 Implementação do Sistema. ....</i>	129
<i>Fig. 5.7 Painel do utilizador da aplicação do sistema. ....</i>	130
<i>Fig. 5.8 A Área das imagens dos sistemas 1, 2A e 3A. ....</i>	131
<i>Fig. 5.9 O código LabVIEW desenvolvido que implementa a função que enquadra a face com o rectângulo áureo para detecção da face. ....</i>	132
<i>Fig. 5.10 O código LabVIEW desenvolvido que implementa o rectângulo áureo da área de busca de detecção de movimento a qual irá variar tendo em conta a “dinâmica” da variação da posição da face enquadrada. ....</i>	132
<i>Fig. 5.11 A imagem 4B. ....</i>	134

<i>Fig. 5.12 Posicionamento 2D.</i> .....	134
<i>Fig. 5.13 A imagem 5b.</i> .....	135
<i>Fig. 5.14 O gráfico do factor Af.</i> .....	135
<i>Fig. 5.15 Valor de Rect 1 e Rect 2 (area de busca).</i> .....	137
<i>Fig. 5.16 Valor do factor Af.</i> .....	137
<i>Fig. 5.17 Valor de Rect 1 e Rect 2 (area de busca).</i> .....	138
<i>Fig. 5.18 Valor do factor Af.</i> .....	138
<i>Fig. 5.19 Valor de Rect 1 e Rect 2 (area de busca).</i> .....	139
<i>Fig. 5.20 Valor do factor Af.</i> .....	139
<i>Fig. 5.21 A variação de Af nos três cenários de teste e o seu Valor médio....</i>	140
<i>Fig. 6.1 - AGV com orientação indutiva (www.Goetting-agv.com, acedido em Maio de 2019)</i> .....	147
<i>Fig. 6.2 - AGV com orientação optica (www.Goetting-agv.com, acedido em Maio de 2019)</i> .....	148
<i>Fig. 6.3 - Navegação por laser artificial (NDC Solutions)</i> .....	150
<i>Fig. 6.4 - Navegação por contorno - navegação por laser com marcas naturais (NDC Solutions)</i> .....	151
<i>Fig. 6.5 Caracterização do Sistema Motor: 1-Estrutura metálica; 2-Placa controladora dos motores; 3-Motores DC; 4-Bateria 24V DC; 5-Controlador Arduino.</i> .....	154
<i>Fig. 6.6 Placa controladora dos motores</i> .....	154
<i>(www.robot-electronics.co.uk/htm/md49tech.htm, acedido em Maio de 2019).</i>	
.....	154
<i>Fig. 6.7 Conjunto de comandos para a placa controladora dos motores (www.robot-electronics.co.uk/htm/md49tech.htm,acedido em Maio 2019)....</i>	155
<i>Fig. 6.8 Caracterização do sistema processamento</i> .....	156
<i>1-PC-Master; 2-PC-Slave com câmara de vídeo; 3-Rede sem fios Xbee.</i> .....	156
<i>Fig. 6.9 Estratégia de controlo do AGV.</i> .....	157
<i>Fig. 6.10 Fluxo de comunicação entre o PC-Master e o PC-Slave.</i> .....	158
<i>Fig. 6.11 Interface gráfica do utilizador do PC-Master (implementação em LabVIEW).</i> .....	158

<i>Fig. 6.12 LabView Master: selecção de modo manual / automático.....</i>	<i>159</i>
<i>Fig. 6.13 LabView Master: função xbee para envio dos comandos .....</i>	<i>160</i>
<i>de movimento ao AGV. ....</i>	<i>160</i>
<i>Fig. 6.14 Interface gráfica do utilizador do PC-Slave (implementado em LabVIEW).....</i>	<i>161</i>
<i>Fig. 6.15 Programa LabView - inicialização de comunicações PC-Slave.....</i>	<i>162</i>
<i>Fig. 6.16 Programa do Arduino: Definição de variáveis.....</i>	<i>163</i>
<i>Fig. 6.17 Programa do Arduino: Movimento para frente.....</i>	<i>163</i>
<i>Fig. 6.18 Trajectória marcada no chão com uma faixa preta.....</i>	<i>164</i>
<i>Fig. 6.19 Linhas de fronteira para seguimento de caminho AGV.....</i>	<i>165</i>
<i>(T = tolerância). ....</i>	<i>165</i>
<i>Fig. 6.20 - Processamento de morfologia da imagem. ....</i>	<i>165</i>
<i>Fig. 6.21 AGV executando seguimento de trajectória. ....</i>	<i>166</i>
<i>Fig. 6.22 Interface gráfica da aplicação do PC-Master para o controlo do AGV por movimento da face humana. ....</i>	<i>167</i>
<i>Fig. 6.22a. Pormenor do Valor do vector da distância calculada (detalhe da GUI da Fig.6.22). ....</i>	<i>167</i>
<i>Fig. 6.23 Algoritmo para controlo do AGV (comparação com limites Tx e Ty) 168</i>	
<i>Fig. 6.24 Implementação do algoritmo de navegação (Tx=10 e Ty=8). ....</i>	<i>168</i>
<i>Fig. 6.25 Sentidos possíveis da condução na navegação.....</i>	<i>169</i>
<i>Fig. 7.1 O Raspberry Pi Zero e câmara (www.raspberrypi.org, acedido em Maio de 2018). ....</i>	<i>176</i>
<i>Fig. 7.2 Aplicação em Python, versão 2.7.9. ....</i>	<i>177</i>
<i>Fig. 7.3 Aplicação a correr no protótipo, com acesso via VNCviewer na rede-WIFI.....</i>	<i>177</i>

## **Lista de tabelas**

Tabela 3.1 Resoluções de vídeo.....	93
Tabela 3.2 Redução dos pixels nas várias resoluções de vídeo.....	94

## **Siglas e acrónimos**

**2D** 2 Dimensões

**3D** 3 Dimensões

**4G** 4ª Geração

**5G** 5ª Geração

**AC** *Alternating Current*

**AGV** *Automated Guided Vehicle*

**ASCII** *American Standard Code for Information Interchange*

**AVI** *Audio Video Interleave*

**BMP** *Bitmap image*

**CIE** *Commission International de L'Éclairage*

**CNN** *Convolutional Neural Network*

**COBOL** *Common Business Oriented Language*

**CPU** *Central Process Unit*

**DC** *Direct Current*

**ECCOMAS** *European Community on Computational Methods in Applied Sciences*

**EDT** *Eastern Daylight Time*

**EKF** *Extended Kalman Filter*

**ENIGMA** *Enhancing Neuroimaging Genetics through Meta-analysis*

**EOL** *End of Line*

**FIR** *Finite Impulse Response*

**Fps** *Frames per second*

**FORTRAN** *Formula Translator*

**GPS** *Global Positioning System*

**GUI** *Graphical User Interface*

**HCI** *Human Computer Interaction*

**HD** *High Definition*

**HSL** *Hue Saturation Lightness*

**HSV** *Hue Saturation Value*

**HTML** *Hyper Text Markup Language*

**IA** *Inteligência Artificial*

**IBM** *International Business Machines*

**IC** *Integrated Circuit*

**ICM** *International Conference on Mechatronics*

**IEEE** *Institute of Electrical and Electronics Engineers*

**IMAQ** *Image Acquisition*

**IoT** *Internet of things*

**JPEG** *Joint Photographic Experts Group*

**JPL** *Jet Propulsion Laboratory*

**LPR** *Local Positioning Radar*

**LSI** *Large Scale Integration*

**MPI** *Multi Point Interface*

**MIT** *Massachusetts Institute of Technology*

**NASA** *National Aeronautics and Space Administration*

**NI** *National Instruments*

**NTSC** *National Television System Committee*

**OP** *Operating Panel*



**PAL** *Phase Alternating Line*

**Pixel** *Picture (Pix) element*

**PC** *Personal Computer*

**PCA** *Principal Component Analysis*

**QQVGA** *Quarter Quarter Video Graphics Array*

**QVGA** *Quarter Video Graphics Array*

**RFID** *Radio Frequency IDentification*

**RGB** *Red Green Blue*

**RSS** *Received Signal Strength*

**SECAM** *Séquentiel Couleur à Mémoire*

**SLAM** *Simultaneous Localization and Mapping*

**SLIT** *Sistema Linear e Invariante no Tempo*

**SVGA** *Super Video Graphics Array*

**TAC** *Tomografia Axial Computorizada*

**ULSI** *Ultra Large Scale Integration*

**UXGA** *Ultra Extended Graphics Array*

**VGA** *Video Graphics Array*

**VLSI** *Very Large Scale Integration*

**VNC** *Virtual Network Computing*

**WISP** *Workshop on Intelligent Signal Processing*

**WIFI** *Wireless Fidelity*

**WXGA** *Wide Extended Graphics Array*

**XGA** *Extended Graphics Array*





# Capítulo 1

## Introdução

*Neste capítulo são apresentados os objectivos, os algoritmos propostos, as contribuições científicas, o estado da arte e por fim a estrutura da tese.*

## 1.1 Objectivos

No desenvolvimento dos trabalhos teve-se como orientação principal o desenvolvimento na área de coordenação de várias fontes de informação obtidas por visão artificial. O objectivo do desenvolvimento de algoritmos com atenção reforçada no melhoramento do desempenho assim como na redução da carga computacional.

A construção e desenvolvimento de modelos de processamento de imagem, multi-dimensionais e ou em multi-perspectiva, para um conhecimento do meio circundante com possibilidade de comando e controlo na área da domótica e ou robótica móvel.

O objectivo foi desenvolver algoritmos que mais tarde possam ser implementados em blocos de software ou hardware, de forma independente, como sistemas isolados, ou integrados como componente de um sistema. Daí, a importância da carga computacional, ou seja, não colocarmos à partida a premissa de que dispomos de uma máquina com todos os recursos necessários disponíveis.

Para além disso, também a possibilidade de no campo do hardware, o desenvolvimento destes sistemas, na área dos sistemas embebidos, poder tomar vários caminhos: i) a utilização de sistemas dedicados com microprocessadores; ii) a utilização de sistemas com lógica reconfigurável (com a vantagem do poderoso recurso do paralelismo).

No campo do software o seu potencial de utilização também é significativo: i) em aplicações desenvolvidas para sistemas operativos de uso geral, em máquinas com várias configurações, de forma independente; ii) em blocos de software integrados em bibliotecas para uso em aplicações de desenvolvimento em variadas linguagens ou ambientes.

Também com aplicação nos dois campos, em sistemas de hardware do tipo sistemas embebidos, com sistemas operativos dedicados de tempo real, para aplicações específicas

Finalmente, o desenvolvimento de uma solução focada na vertente da comunicação entre sistemas, com a possibilidade de funcionarem de forma isolada. Apresentando o potencial da utilização da visão por computador com processamento de imagens na utilização da detecção e avaliação da posição da face humana como elemento de comando.

Tendo em conta a nova área da Internet das coisas (Iot), o campo da comunicação e interacção dos sistemas, reforça ainda mais o objectivo de redução da carga computacional, assim como da versatilidade na implementação dos vários algoritmos, tendo em conta a heterogeneidade dos equipamentos ligados, em vários níveis.

## 1.2 Algoritmos propostos e Contribuições científicas

Nos modelos de processamento de imagem aplicados à domótica identificaram-se como pontos importantes: i) a detecção de movimento de objectos e seres humano em ambiente não controlado; ii) a detecção da face humana como elemento de comando, entre outras aplicações; iii) a capacidade de utilização de multi-fontes de aquisição e processamento de imagem, com diferentes condições de iluminação não controladas, integradas num sistema completo com diversas topologias na domótica e na robótica

No decorrer do trabalho realizado desenvolveram-se fundamentalmente três novas metodologias:

- **Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células** (Grilo et al, 2007a; Grilo et al, 2007b)

Nesta contribuição foi desenvolvido um novo método de detecção do movimento em imagens 2-D, baseado na variação adaptativa da dimensão das células de busca rectangulares. A dimensão célula é definida automaticamente com base num método de adaptação estatística, que tem em conta os resultados de pesquisa anteriores. A principal contribuição do presente trabalho em relação aos métodos usuais de pesquisa de movimento de imagem é a metodologia que auto adapta e reduz significativamente o trabalho computacional em comparação com células de imagem de dimensão fixa;

- **Detecção de pele em Imagens a cores por Método adaptativo** (Grilo et al, 2009; Grilo et al, 2014)

Nesta contribuição foi desenvolvido um novo método que utiliza a metodologia adaptativa incorporada no algoritmo classificador da pele para a classificação do pixel. O algoritmo adaptativo varia o brilho da imagem em cada um dos componentes do espaço de cor RGB, com uma nova abordagem para a segmentação de pele, detecção de rosto e um novo método na localização da face na imagem. Esta metodologia melhora significativamente a detecção dos pixels de pele, reduzindo simultaneamente a carga computacional;

- **Detecção da variação de posição da face humana por enquadramento de rectângulo áureo** (artigo em processo de submissão a Revista Internacional)

Este método tem como novas abordagens a utilização do rectângulo áureo como validação e detecção da face humana, num sistema de classificação de pele, em imagens a cores para posterior avaliação da variação da posição da face em imagens consecutivas, com o objectivo de controlar sistemas automáticos através do movimento da face. A

nova abordagem utiliza uma metodologia adaptativa embutida no algoritmo classificador de pele, por segmentação, melhorando a detecção da pele e da variação de posição do rosto por enquadramento no rectângulo áureo.

## **1.3 Publicações**

### **1.3.1 Capítulo em Livro**

GRILO, F., FIGUEIREDO, J. [2018] " Computer-Vision in Industrial Automation and Mobile Robots", in Introduction to Mechanical Engineering, Ed. P. Davim, SPRINGER Intl. Publishing, ISBN 978-3-319-78488-5, 2018, pp. 241-266  
ON LINE available: <http://dx.doi.org/10.1007/978-3-319-78488-5>

### **1.3.2 Artigo em Revistas Internacionais com arbitragem**

GRILO, F., FIGUEIREDO, J., DIAS, O. (2014). Adaptive Method for Improvement of Human Skin Detection in Colour Images, Intl. Journal Computer Applications in Technology, Vol.49-No.1 (2014), pp.1-11, - Inderscience Publishers.  
ON LINE available : <http://www.inderscience.com/info/inarticle.php?artid=59091>  
(Indexado às bases de Dados SCOPUS.)

### **1.3.3 Artigos em “Proceedings” de Congressos Internacionais com arbitragem**

GRILO, F., FIGUEIREDO, J., DIAS, O., AMARAL, T. (2009) "Adaptive Method for Skin Detection in Coloured Images", Proc. IEEE Intl. Conf. on Mechatronics – ICM 2009, April 2009, Málaga – Spain

GRILO, F., FIGUEIREDO, J. (2008) "An Industrial Vision System for Quality Control Based on a Distributed Strategy", Proc. 8th Intl. Conf. on Automatic Control - Controlo 2008, July 2008, Vila Real – Portugal

GRILO, F., FIGUEIREDO, J., DIAS, O., AMARAL, T. (2007) "Adaptive Topologies for Dimension of Cells in 2D-Images for Movement Detection", Proc. IEEE Intl. Symposium on Intelligent Signal Processing, WISP 2007, October 2007, Madrid – Spain

GRILO, F., FIGUEIREDO, J., DIAS, O., AMARAL, T. (2007) "Movement Detection in 2-D Images with Variable Dimension Cells", Proc. ECCOMAS Thematic Conf. On Computational Vision and Medical Image Processing, VIP IMAGE, October 2007, Porto – Portugal

## 1.4 Estado da Arte

### 1.4.1 A visão por computador

A visão artificial (visão por computador) é uma ferramenta robusta e de grande utilização em campos tecnológicos diversos: na indústria, na segurança, na robótica, nos edifícios inteligentes (Domótica) e também, não menos importante, na medicina. O reconhecimento de padrões, a restauração da geometria, o servo-controlo visual, a inteligência artificial, são domínios em que a visão artificial e o processamento digital de imagem fornecem contribuições importantes. Bons exemplos do potencial da visão por computador e processamento digital de imagem, diversificados no tempo e no objecto de estudo, podem ser encontrados em Nelson, 2003; Hlou et. al, 2003; Umbaugh, 2010; Tekalp, 2015.

Actualmente, na visão por computador, devido ao grande volume de dados (*Big Data*) gerados, manipulados e processados, todas as metodologias relacionadas com a sua redução nos algoritmos e nos processos, ganham uma importância crucial (Hensman et al, 2013; Huang, 2014; Najafabadi e tal, 2015).

O tema da Inteligência artificial (IA - *Artificial Intelligence*) tem tido grande evolução, com níveis mais complexos de coordenação e aprendizagem em processos e decisões. A sua aplicabilidade a contextos cada vez mais elaborados e complexos permite a sua aplicação a várias áreas tecnológicas actuais, nomeadamente à automação industrial, à robótica e à domótica (Chen and Lin, 2014; Tekalp, 2015).

A análise do contexto para a determinação de acções a tomar, para a captura de referências locais e temporais, para o registo da actividade de humanos/objectos adquire uma importância cada vez maior nos sistemas de decisão actuais (Moeslund et al, 2016). Este enorme volume de informação a ser incorporado no processo de aprendizagem, para ser utilizado na tomada de decisão e acções futuras poderá atingir volumes tais que dificultem o seu processamento em tempo útil. A visão por computador adquire grande importância na detecção e sinalização de parâmetros relevantes, por exemplo, na detecção de seres humanos e sua movimentação em ambientes complexos (Toshev and Szegedy, 2014; Szegedy et al, 2017; Kale and Patil, 2018). Um exemplo recorrente é a tecnologia de IA da IBM, denominado *Watson*, um motor de inteligência artificial, que pode ser utilizado nos mais variados cenários, desde logo em plataformas petrolíferas, no mar, com a capacidade de integração do elemento humano, e das suas acções, com todo o equipamento envolvente, do próprio sistema, desenvolvendo funcionalidades de previsão e tomando acções de decisão e controlo. Outra aplicação relevante é a sua utilização em redes de elevadores, em várias cidades do mundo, executando manutenções preventivas (a vários níveis) e incrementando os níveis de



segurança verificados (Chen and Weber, 2016; Obermeyer and Emanuel, 2016).

A análise automática de imagens, tomadas por satélites militares ou civis é ferramenta essencial em várias áreas, nomeadamente: na meteorologia, na ecologia, na caracterização dos solos e do ambiente urbano e no reconhecimento militar, onde a detecção de objectos (partículas) (Szegedy et al, 2015) é muito importante, tal como o seu registo e as suas possíveis previsões de movimento (Erhan et al, 2014; Chang et al, 2016; Prakash and Kak, 2018).

Na medicina, a análise automática de imagens com a integração de inteligência artificial, para a coordenação de vários meios de diagnóstico, ou mesmo, em diagnóstico “automático”, têm tido uma evolução muito acentuada (Jain et al, 2015; Santy and Joseph, 2015; Agarwal et al, 2017). O processamento digital de imagem tem revelado a sua enorme importância, quer em imagens no espectro do visível da visão humana, quer em radiografias ou outras fontes de imagem para recolha de dados médicos (Chen and Weber, 2016; Obermeyer and Emanuel, 2016).

A visão por computador na condução automática de veículos (*AGV Automated Guided Vehicle*), é uma realidade incontornável, nos dias de hoje, e terá certamente, no futuro, uma aplicação crescente, a nível civil e militar, nos veículos terrestres, aquáticos e aéreos (Miljkovic et al 2013; Xing, et al 2014; Vignesh et al, 2017). Uma referência de topo, a este nível, é o caso da aeronave militar dos Estados Unidos que, em 2012, levantou vôo e aterrou num porta-aviões, de forma autónoma, sem intervenção humana. Também as colunas militares de veículos de transporte pesado, sem condutor, que de forma experimental estão a ser utilizadas pelo exercito dos Estados Unidos em zonas de conflito, nomeadamente no Afeganistão e no Iraque.

A condução automática integra um grande conjunto de sensores e sistemas de análise e decisão onde estão integrados vários sistemas de visão artificial, fundamentalmente nos espectros visíveis e infravermelho, com detecção de obstáculos, detecção de seres humanos, detecção de movimentos, etc (Han and Wang, 2015; Brasil and Machado, 2017).

Na área da aquisição e processamento de imagem em multi-perspectiva existem aplicações importantes, nomeadamente no desporto: no ténis, já há alguns anos com o sistema “Olho de falcão” (*Hawk Eye*), ou mais recentemente no futebol, com o sistema *GoalControl-4D*. Ambos estes sistemas integram informação de imagens, em tempo real, tomadas de vários pontos, com a aplicação de algoritmos de processamento, para determinar a posição exacta da bola nas várias áreas de jogo, no caso do ténis, ou no caso do futebol, para determinar se a bola entrou na baliza (Ahmadi and Sobhani, 2014).

Outros sistemas e aplicações existem onde a detecção da posição, do movimento e a determinação de trajectórias, em ambientes complexos e não controlados, apresentam grande actualidade.

Na área da *Internet* das coisas (IoT - *Internet of things*), temos a visão por computador com a integração de vários sensores de aquisição de imagem com maior ou menor capacidade em termos de: processamento de imagem e actuação no meio (utilização na domótica).

Neste caso, temos uma enorme capacidade disponível com a utilização dos vários recursos acessíveis na rede, na nuvem (*cloud*), com a utilização distribuída de algoritmos de IA, e com um diversificado conjunto de bibliotecas e ferramentas, nomeadamente de Aprendizagem automática, *Machine learning*, aprendizagem profunda - *deep learning* (Erhan et al, 2014; Loos et al, 2017; Szegedy et al, 2017; Mohammadi et al, 2018),

Finalmente, temos a referir as várias plataformas e linguagens de desenvolvimento de aplicações de processamento digital de imagem para aplicação, ou não, na visão por computador. Há plataformas que correm nos vários sistemas operativos suportados por equipamentos com arquitecturas tipo PC, ou dedicadas, nomeadamente: LabVIEW da NI; Processing da Processing Foundation/MIT; MatLab e Simulink da Mathworks, Mathematics da Wolfram; recorrendo a várias linguagens de programação: i) de tipo gráfico; ii) C; iii) Java; iv) Python e v) linguagem *scripts* próprias. Também relevantes são as bibliotecas multiplataformas como por exemplo OpenCV, desenvolvida pela Intel em 2000, de uso livre.

### 1.4.2 O volume de dados no processamento de imagem

No processamento digital de imagem o volume de dados manipulados tornou-se um ponto muito importante e em certas aplicações até crítico. O aumento das resoluções das imagens e o aumento das taxas de transmissão nas diferentes redes digitais de interligação de sistemas, nomeadamente nas redes de comunicações móveis apresentam um gigantesco desafio (Manogaran G et al, 2018).

O *Big Data* surgiu com novas oportunidades de pesquisa, de desenvolvimento, de inovação e de negócios. A transformação dos sistemas de 4G para em 5G é um grande desafio para a capacidade de processamento. A computação em nuvem emerge como um novo paradigma que apresenta a computação como um serviço de utilidade para atender a diferentes necessidades de processamento (Bhimani et al, 2017; Yang et al, 2017; Hossain and Muhammad, 2018)

A utilização da Internet das coisas (IoT), como já foi referido, fornece novos tipos de serviços, alguns dos quais se podem aplicar à domótica, no campo das cidades inteligentes, com sistemas a colectar e a gerir dados de vários tipos de sensores, presentes nos edifícios inteligentes, incluído a aquisição e processamento de imagens, gerando grandes volumes de dados.

Tecnologias baseadas em serviços em nuvem, com a aquisição, a gestão e o controlo remoto dos múltiplos dados provenientes de várias fontes (meio ambiente, socioeconómicos, energia, transportes, etc), devem ser estruturados numa perspectiva integrada de gestão e análise de grande volume de dados para responder às várias questões da sociedade, num contexto mais inteligente. Torna-se assim crucial a questão do volume dos dados (Khan et al, 2015; Plageras et al, 2018).

As diversas áreas da visão por computador onde o processamento da imagem e a gestão do volume de dados tomam um papel muito importante é, por exemplo, a utilização de grande quantidade de imagens heterogéneas adquiridas por satélites, nomeadamente no desafio do *Big Data* em plataformas como o *Google Earth Engine* (GEE) ao classificar imagens de satélite multi-níveis (Shelestov et al, 2017).

Também nas ciências da saúde, como a neurociência clínica e básica, a imagem cerebral estrutural e funcional tornou-se actualmente numa ferramenta poderosa em diagnósticos. No entanto só recentemente alguns estudos de imagens cerebrais começaram a trabalhar na ciência do *Big Data* (Bearden and Thompson, 2017; Smith and Nichols, 2018). Neste domínio, refira-se o trabalho desenvolvido pelo neurocientista Jeff Lichtman, da Universidade de Harvard, na criação de imagens 3D, extremamente pormenorizadas, dos neurónios de ratos. A sua equipa introduziu pedaços de cérebro de rato, num fatiador neuroanatômico, que corta fatias de tecido com espessura inferior a um milésimo de um cabelo humano, utilizando um microscópio electrónico para captar cada imagem e, finalmente, processar e construir a imagem 3D final.

A capacidade necessária para armazenar o cérebro de um rato, com a resolução de 0,01  $\mu\text{m}$ , é de 450 mil terabytes. A capacidade necessária para armazenar o cérebro humano, mantendo a mesma resolução, seria de 1300 milhões terabytes, ou seja, cerca de metade da capacidade mundial de armazenamento digital disponível em 2012 (2700 milhões terabytes – ver figura 1.1).

Surgem por isso novas necessidades de desenvolver: i) tecnologias avançadas e algoritmos para lidar com grandes volumes de dados; ii) ferramentas de redução e análise de dados que utilizem novas formas com capacidades de inteligência artificial com recurso a soluções inovadoras, por exemplo: métodos geométricos e topológicos; iii) metodologias que permitam a análise de dados altamente complexos; iv) técnicas que criem uma representação resumida, ou compactada, de todos os recursos de dados, para ajudar na rápida detecção de padrões e relação entre dados (Bhimani et al, 2017; Snášel et al, 2017; Manogaran G et al, 2018)

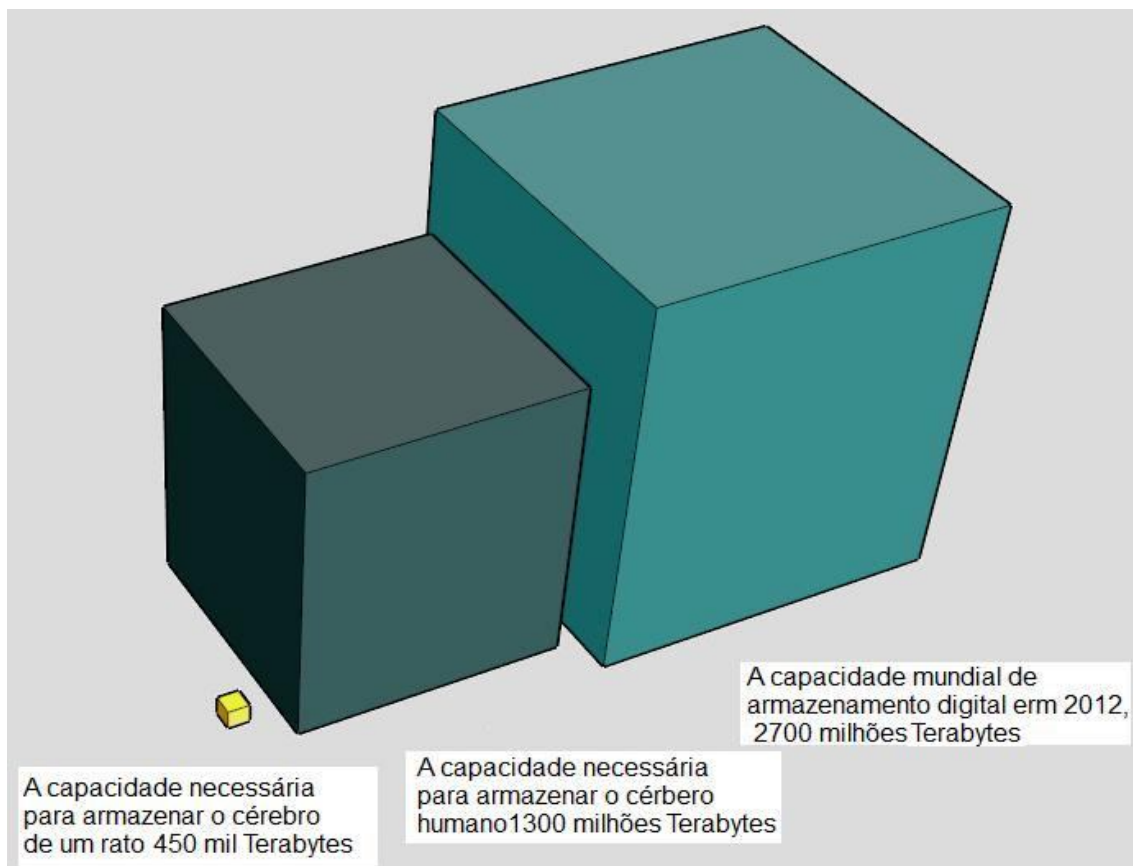


Fig. 1.1 Comparação da capacidade de armazenamento, projecto digitalização do cérebro humano, (Jeff Lichtman, da Universidade de Harvard & Daniel Berge MIT)

### 1.4.3 A visão artificial na domótica

Na domótica torna-se particularmente relevante a análise automática de imagens, nomeadamente: i) na determinação da posição, movimento e previsão do movimento de seres humanos em edifícios (Huang, 2014; Toshev and Szegedy, 2014; Szegedy et al, 2015; Liu et al 2016), com diversos níveis de utilização, com várias etapas de inteligência; ii) no comando dos vários níveis de conforto – climatização, iluminação, controlo e comando de acessos, etc.; iii) na hierarquização de níveis na segurança - detecção de intrusão, detecção de acidentes, previsão e coordenação das trajectórias de evacuação em caso de acidentes e catástrofes (Girshick et al, 2014; Long, et al, 2015). Estamos na presença, mais uma vez, de uma área onde a visão por computador com o processamento digital de imagem, em conjunto com a IA, adquirem uma importância determinante (Szegedy et al, 2016).

Os sistemas e aplicações para controlo e automação de ambientes residenciais (domótica) são uma área emergente de estudo no contexto de *Human-Computer Interaction* (HCI) (Correia, A. et al, 2013). Um dos grandes desafios está relacionado com o projecto de sistemas de interacção do ser humano, como elemento de controlo, nas aplicações de automação. A capacidade de interacção em ambientes inteligentes domóticos na vertente social e doméstica revela a importância crescente dos aspectos relacionados com a interacção humano-computador (HCI).

As Tecnologias emergentes de apoio, como a domótica assistida ou os robôs de apoio e assistência, na vertente social, têm um potencial considerável na melhoria das condições de vida de muitos idosos e indivíduos com limitações físicas, em todo o mundo. Seres humanos, com ou sem limitações visuais, podem usar estas tecnologias, para diferentes tarefas, nomeadamente no reconhecimento de objectos, no tratamento e gestão de várias tarefas domésticas e na navegação em ambientes interiores ou exteriores (Giizel and Zakaria, 2013; Santos et al, 2013; Shalaby et al, 2014).

Os equipamentos portáteis, com boas capacidades de visualização e com baixo custo, conjuntamente com os dispositivos de detecção de movimento, sem contacto, proporcionam-nos o desenvolvimento de ambientes virtuais imersivos e altamente interactivos. Nestes ambientes virtuais, a interface homem-computador, e particularmente a tecnologia de navegação, desempenham um papel crucial (Santos et al, 2013; Caggianese et al 2015).

Os aspectos sócio-técnicos na interacção gestual em ambientes inteligentes domóticos apresentam novos desafios nomeadamente no projecto de aplicações de interacção gestual, que minimizem as ambiguidades e considerem os aspectos culturais e sociais. O objectivo último será o projecto de uma interacção significativa que seja intuitiva e fácil de aprender (Correia et al, 2013)

Grandes oportunidades também se colocam no desenvolvimento de i) novas abordagens com capacidade de exploração 3D, em tempo real e interacção gráfica, usando ambientes de realidade virtual imersiva e interfaces de utilizador natural; ii) Abordagens de desenvolvimento em ambientes globais, com estudo de avaliação empírico, agrupando pessoas em configurações diferentes, para explorar e interagir com ambientes dinâmicos, projectando configurações inovadoras, explorando as interfaces naturais, com a utilização de ambientes de realidade aumentada ou realidade virtual.

A evolução actual sugere que estas tecnologias, no futuro próximo, serão um desafio face às tradicionais, permitindo ao utilizador envolver-se cada vez mais nas tarefas de interacção e visualização, tornando as experiências mais amigáveis, ao combinar interfaces baseadas em gestos e realidade virtual. (Erra et al, 2017)

No contexto específico da presente investigação, foram identificadas 4 áreas com relevância particular na utilização da visão artificial na Domótica, concretamente: a) a detecção de movimento de objectos e seres humano em ambiente não controlado; b) a face humana como elemento de comando entre outras aplicações; c) a detecção de pele em imagens; d) a detecção de faces em imagens.

Apresenta-se seguidamente o estado da Arte nestas 4 importantes áreas.

### 1.4.3.1 A detecção de movimento

Entre os diferentes métodos de detecção de movimento existem duas principais classes básicas (Alexandre and Campilho, 1998; Gonzalez and Woods, 2002):

- Métodos baseados em características;
- Métodos baseados no fluxo óptico.

Nos métodos baseados no fluxo óptico, duas abordagens são possíveis:

- Diferenciação entre imagens (*interframe differencing*) ;
- Diferenciação à imagem de referência (*reference frame differencing*).

O desempenho de ambos os métodos é drasticamente influenciado pela área de pixels seleccionada e pela sua subsequente utilização nos resultados obtidos. Os resultados do processamento são normalmente usados na monitorização em tempo real (Fathy and Siyal, 1995), e aplicações de compressão de vídeo (Jain, 1981), (Mitchell et al, 1996).

No método da diferenciação entre imagens (*interframe differencing*) (Jain et al, 1995) é definido se existe diferença para cada pixel (x,y) entre a imagem (*frame*) obtida no instante t1 (*frame t1*) e a imagem obtida no instante t2 (*frame t2*):

$$DP(x, y) = \begin{cases} 1, & |F(x, y, t_1) - F(x, y, t_2)| \geq \tau \\ 0, & otherwise \end{cases} \quad (1.1)$$

Em que  $D_p(x,y)$  é definida como a função booleana que sinaliza se há, ou não, diferença (0 ou 1) em função da diferença de intensidade do pixel (x,y), relativamente à referência  $\tau$  (limiar de diferenciação)

No método da diferenciação à imagem de referencia (*reference frame differencing*) (Jain et. al, 1995) compara-se a diferença, para cada pixel (x,y), entre a Imagem (*frame*) de referência,  $F_r$  e a imagem (*frame*) n,  $F_n$ , onde  $t_r$  e  $t_n$

representam os instantes em que as imagens (*frames*)  $F_r$  e  $F_n$  são adquiridos, respectivamente:

$$DP(x, y) = \begin{cases} 1, & |F_r(x, y, t_r) - F_n(x, y, t_n)| \geq \tau \\ 0, & \textit{otherwise} \end{cases} \quad (1.2)$$

Nos métodos baseados em características, os algoritmos de análise do movimento, são métodos com um nível de análise superior ao pixel e com um nível de interpretação da imagem mais complexo que os anteriores. Recorrem à detecção das características de forma a aproximar-se da forma como o ser humano detecta o movimento (Kale and Patil, 2018), com recurso a segmentação semântica e algoritmos com componente de “inteligência” por exemplo a redes de convolução (*convolutional networks for semantic segmentation*) (Girshick et al, 2014; Long e tal, 2015). São algoritmos que por norma necessitam de maior carga computacional.

### 1.4.3.2 A face humana como elemento de comando

O enorme aumento da interactividade entre pessoas e máquinas, o crescente aumento da comunicação entre ambos, a diversificação dos vários níveis e velocidades de comunicação, associados ao gradual desaparecimento dos teclados e outros meios tradicionais de interacção com os equipamentos, têm como consequência directa a crescente utilização da face humana, entre outras referências fisiológicas, como elemento de controlo e comando.

O uso do rosto humano como elemento de comando e controlo, sofreu assim um crescimento em vários campos de aplicação, especialmente relacionados com questões de segurança:

- Doméstico;
- Industrial;
- Militar.

Em alguns sistemas interactivos, uma abordagem ao controlo de movimento é realizada através da detecção do rosto do operador. Outro campo importante de detecção de rosto humano está relacionado com a localização de pessoas em áreas especificadas. A visão artificial e o processamento de imagem já têm um papel importante em vários domínios tecnológicos, nomeadamente:

- ✓ **Reconhecimento de padrões**, com múltiplas aplicações em controlo de qualidade industrial (Grilo and Figueiredo, 2008; Figueiredo, 2008; Du and Sun, 2008; Kim et al, 2010; Zhang et al, 2010, 2012); diagnóstico médico (Gouttière and Coninck, 2007; Brown et al, 2010; Xiaoping and Bui, 2012; Jain and Pise, 2015; Santy and Joseph, 2015) Agarwal et al, 2017); detecção de rosto (Grilo et al, 2009 ; Pai et al, 2010 ) , a restauração da geometria (Hlou et al , 2003; Lim et al, 2011; Chen et al, 2012);
- ✓ **Visual servoing** (Nelson, 2003; Fei et al, 2012; Vignesh et al , 2017);
- ✓ **Vigilância, Rastreamento de objectos e Detecção de movimento** (Lee and Shinozuka, 2006; Grilo et al, 2007; .Cui et al, 2008; Yang et al, 2010; Messon, 2010; Ahmed and Terada, 2012; Li and Wang, 2012 ; Toshev and Szegedy, 2014; Kale and Patil, 2018).

### 1.4.3.3 Detecção de pele em imagens

Relativamente à detecção de pele em imagens a cores são usualmente utilizados dois tipos de abordagens (Zhao et al, 2003; Soares and Tavares, 2005; Mahmoodi and Sayedi, 2016; Agarwal et al, 2017):

- Métodos que utilizam processos de aprendizagem (Jones and Rehg, 2002; Yang et al, 2002; Zhao et al, 2003; Wu et al, 1999; Rein-Lien and Abdel- Mottaleb, 2002; Mahmoodi and Sayedi, 2015);
- Métodos que utilizam a relação entre os componentes de cor em cada pixel (Vezhnevets et ai, 2003; Rein-Lien and Abdel-Mottaleb, 2002; Chai and Ngan, 1999; KaKumanu et al, 2007; Shaik e tal, 2015).

Os métodos que utilizam processos de aprendizagem são baseados na análise de um conjunto de imagens de teste, de modo a gerar uma função estatística de distribuição das variáveis, que usualmente são as componentes do espaço de cor correspondente - RGB, rgb, HSI, HSV, HSL ou YCbCr - (Vezhnevets et al, 2003; Kim e tal, 2017). A função estatística é então aplicada a cada pixel a fim de ser, ou não, identificado como um pixel de pele (Wang and Yuan, 2002; Vadakkepat et al, 2008; Angelopoulou et al, 2001; Phung et al, 2005; KaKumanu et al, 2007, Agarwal et al, 2017).

Por outro lado, os métodos que utilizam a relação entre as componentes do espaço de cor para construir um classificador de pele, comparam, para cada pixel, o grau de semelhança entre os seus componentes de cor (Peer et al, 2003; KaKumanu et al, 2007; Shaik e tal, 2015). Um algoritmo de decisão é então responsável por identificar se o pixel corresponde a um pixel de pele, ou



não (segmentação por cor). A simplicidade deste método aliada aos bons resultados fornecidos, tem motivado alguns investigadores no seu melhoramento. A principal vantagem deste método é a simplicidade das regras de detecção de pele que proporcionam o desenvolvimento de rápidos classificadores (Vezhnevets et al, 2003; Par et al, 2003).

Este tipo de abordagem tem fornecido bons resultados, desde que as condições do brilho da imagem não sofram variações significativas. No entanto, havendo variações significativas das condições de iluminação enquanto as imagens são adquiridas, poderão ocorrer erros graves na identificação dos pixels (Rein-lien e Abdel-Mottaleb, 2002). A ocorrência destes erros deve-se fundamentalmente ao facto deste método quantificar a contribuição de cada uma das cores primárias, do espaço de cor (ex.:RGB), para cada pixel e verifica-se usualmente que o atributo da cor varia significativamente com o brilho da imagem. Para superar este problema, existem algumas abordagens, baseadas fundamentalmente em algoritmos adaptativos com processos de aprendizagem, para a reconstrução das regras do classificador. Estas abordagens necessitam previamente de algumas imagens de teste para iniciar o processo de aprendizagem (Rein-Lien and Abdel-Mottaleb, 2002; Mahmoodi and Sayedi, 2015).

#### **1.4.3.4 Detecção da face em imagens**

Actualmente, os vários métodos utilizados na detecção da face em imagens, podem dividir-se nas seguintes 4 classes (Hjelmås and Kee Low, 2001; Qi Xiaojun, 2015; Zafeiriou e tal, 2015):

- Detecção de faces em imagens com controlo de fundo;
- Detecção de faces por cor;
- Detecção de faces por movimento;
- Detecção de faces em cenas sem restrições.

A metodologia de detecção de faces em imagens com controlo de fundo usa um segundo plano monocromático, ou um segundo plano estático, com uma característica bem definida, de modo a detectar as fronteiras da face, para a sua posterior segmentação (Qi Xiaojun, 2015).

A metodologia de detecção de faces por cor usa a detecção da pele em imagens a cores para a segmentação da face. Esta metodologia inclui também o método designado de PCA - *Principal Component Analysis* (Qi Xiaojun, 2015; Barlett et al, 2002; Rein-Lien and Abdel-Mottaleb, 2002).

A metodologia de detecção de faces por movimento é utilizada em vídeo, em tempo real, calculando as áreas em movimento e classificando aquelas que, pela sua dimensão, sejam possíveis faces. Adicionalmente, poderão

considerar-se outras características da face para validação do processo (ex.: a posição dos olhos, etc.) (Haro et al, 2000; Qi Xiaojun, 2015).

Finalmente, na metodologia de detecção de faces em cenas sem restrições encontram-se os métodos que contemplam os processamentos mais complexos, recorrendo a diversos tipos de aproximações e algoritmos, nomeadamente com base em redes neuronais (Hjelmås. and Kee Low, 2001; Jain and Learned-Miller, 2010; Qi Xiaojun, 2015; e aprendizagem com inteligência artificial (Chuan-Xian et al., 2016). São dominantes as seguintes abordagens:

- Aproximação por rede neuronal (Rowley et al, 1998; Feraund et al, 2001; Zafeiriou et al, 2015);
- Redes Neuronais utilizando informações de conjunto estatístico (Rowley et al, 1998; Feraund et al, 2001; Chen et al, 2008);
- Modelo baseado em Rastreamento da Face (Gokturk et al, 2002)
- Classificador fraco em cascata (Huang et al, 2004);
- Métodos de aprendizagem profunda (*Deep Learning*) (Sun Y. et al, 2014);
- Redes neurais convolucionais profundas (*CNN Convolutional Neural network*) (Krizhevsky et al, 2012; Schroff et al, 2015).

## 1.5 Estrutura da tese

A estrutura do documento apresentado reflecte o desenvolvimento temporal dos estudos e dos trabalhos realizados:

- Cp.1 Introdução
  - 1.1 Objectivos
  - 1.2 Algoritmos propostos e Contribuições científicas
  - 1.3 Publicações
  - 1.4 Estado da Arte
  - 1.5 Estrutura da tese
- Cp.2 Visão por computador e Processamento de imagem
  - 2.1 Visão por computador
  - 2.2 Processamento de imagem
  - 2.3 Algoritmos de processamento de imagem
- Cp.3 Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células
  - 3.1 Introdução
  - 3.2 Algoritmo de Processamento de Imagem
  - 3.3 Implementação do sistema

- 3.4 Análise e conclusões
- Cp.4 Detecção de pele em Imagens a cores por Método adaptativo
  - 4.1 Introdução
  - 4.2 Algoritmo de Processamento de Imagem
  - 4.3 Implementação do sistema
  - 4.4 Análise e conclusões
- Cp.5 Detecção da variação de posição da face humana por enquadramento de rectângulo áureo com Método adaptativo
  - 5.1 Introdução
  - 5.2 Algoritmo de Processamento de Imagem
  - 5.3 Implementação do sistema
  - 5.4 Análise e conclusões
- Cp.6 Visão por computador e robôs móveis
  - 6.1 Visão por computador em robôs móveis
  - 6.2 Implementação do sistema
  - 6.3 Análise e conclusões
- Cp.7 Conclusões e Trabalho futuro
  - 7.1 Conclusões
  - 7.2 Trabalho Futuro

Refere-se finalmente que no desenvolvimento dos trabalhos e na implementação dos sistemas e testes realizados foi utilizado o ambiente de desenvolvimento LabView da *National Instruments* para o desenvolvimento das aplicações de *software*.

Deste modo, nos capítulos onde são explicadas as respectivas aplicações recorreu-se à utilização de figuras tomadas dos painéis de utilizador desenvolvidos - *Graphical User interface* -, técnica vulgarmente designada por *screen captures*, de forma a tornar mais clara a caracterização da aplicação desenvolvida.

Também em certas situações, para tornar mais elucidativa a interpretação dos resultados dos testes, foram apresentados os diversos gráficos e fontes retiradas, directamente, das imagens dos painéis desenvolvidos.

# Capítulo 2

## Visão por computador e Processamento de imagem

*Neste capítulo é apresentado o que é a visão por computador e o processamento digital de imagem, com a apresentação das origens históricas do processamento de imagem, assim como os conceitos, métodos e algoritmos usados no processamento digital.*

## 2.1 Visão por computador

A visão por computador é uma área científica multidisciplinar que engloba a ciência de computação, a matemática e a engenharia electrotécnica. Inclui formas de adquirir, processar, analisar e compreender imagens e vídeos do mundo real para reproduzir a visão humana e posteriormente utilizar a informação extraída para o comando e controlo de sistemas automáticos. Além disso, ao contrário da visão humana, a visão por computador também pode ser usada para analisar e processar imagens de profundidade a vários níveis do espectro da luz visível e não visível, como por exemplo o infravermelho. A visão por computador também está preocupada com a teoria da extracção de informações de imagens e vídeo. Um sistema de visão por computador pode aceitar diferentes formas de dados como entrada, incluindo imagens, sequências de imagens e vídeos que podem ser adquiridos de várias fontes para posterior processamento e extracção de informações, úteis para a tomada de decisões. A Inteligência artificial e a visão por computador compartilham áreas comuns, como o processamento de imagem, o reconhecimento de padrões e as técnicas de aprendizagem automáticas.

O grande aumento da utilização da visão por computador, fundamentalmente na indústria, na robótica (móvel e de manipulação) e na domótica, tem tido como consequência directa a maior exigência na qualidade das aplicações e na sua utilização, seja no meio industrial como no meio doméstico. É claro que não bastaria a maior exigência de qualidade, caso não se estivesse perante uma grande revolução tecnológica dos sistemas de informação/informáticos e dos sistemas electrónicos, cada vez mais rápidos e compactos e acima de tudo por questões de custos, cada vez mais acessíveis. Aliás, a conjugação destes factores favoráveis, têm sido determinantes nos enormes desenvolvimentos verificados tanto na robótica como na domótica, permitindo uma maior penetração de automatismos e robots num cada vez mais vasto campo de aplicação, substituindo as tarefas executadas manualmente.

De uma forma genérica a aplicação de visão por computador na Automação, Robótica e Domótica tem o seguinte modelo (Fig.2.1):

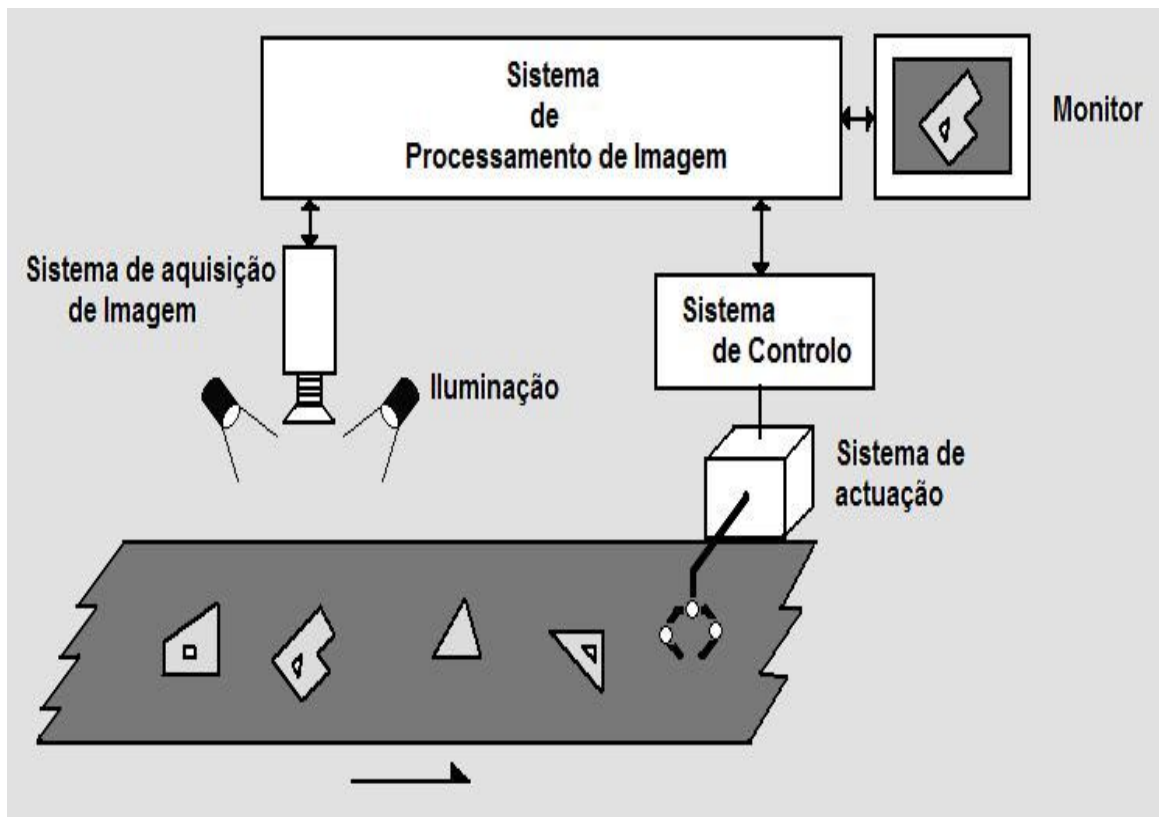


Fig. 2.1 Modelo de Visão por Computador.

## 2.2 Processamento de imagem

### 2.2.1 As origens históricas do processamento digital de imagem

Em termos históricos o processamento de imagem surge primeiro que o processamento digital de imagem. Uma das primeiras aplicações de processamento de imagens ocorreu na indústria dos “média”, o jornalismo. Pela primeira vez, fotografias foram enviadas por cabo submarino entre Londres e Nova Iorque. O sistema de transmissão de imagens por cabo Bartlane introduzido no início da década de 1920 reduziu o tempo necessário para “transmissão” de uma imagem ao longo do Atlântico de mais de uma semana para menos de três horas.

Na Figura 2.2 é apresentada a imagem processada pelo referido equipamento para transmissão por cabo e posterior reconstrução e impressão na recepção. A reprodução da imagem foi feita numa impressora de telégrafo, preenchida com fontes de letras simulando um padrão de meio-tom.



Fig. 2.2 Imagem produzida em 1921 a partir de uma fita codificada por um telégrafo impressora com fontes de caracteres especiais (IEEE).

O método de impressão inicial foi abandonado no final do ano de 1921 a favor do sistema Bartlane, uma técnica baseada na reprodução fotográfica feita a partir de fitas perfuradas no terminal telegráfico na recepção. O sistema era capaz de codificar imagens com 5 níveis de cinzento. Esta capacidade foi aumentada para 15 níveis em 1929.

Neste período, foi introduzido um sistema de desenvolvimento com uma placa de filme estimulada por via de feixes de luz modulados pela fita de codificação da imagem, melhorando consideravelmente o processo de reprodução.

Embora estes sistemas envolvam imagens digitais, eles não são considerados de verdadeiro processamento de imagem digital, porque, resultando da definição, não existiram computadores envolvidos na sua criação.

Assim, o processamento digital de imagens está intimamente associado ao desenvolvimento do computador digital. De facto, as imagens digitais são tão exigentes em termos de armazenamento de dados e em capacidade computacional, que o progresso no campo do processamento digital de imagens tem estado dependente do desenvolvimento dos computadores digitais e das tecnologias de suporte que incluem o armazenamento de dados, exibição e transmissão.

A ideia da computação no cálculo remonta à invenção do ábaco na Ásia Menor, há mais de 5000 anos. Mais recentemente, nos dois últimos séculos, houve desenvolvimentos que viriam a culminar naquilo a que hoje chamamos de computador. No entanto, a base para o computador digital moderno remonta apenas à década de 40 do século passado com a introdução, por parte de John von Neumann, de dois conceitos-chave:

- Uma memória para armazenar um programa e para armazenar dados;
- Um fluxo condicional.

Estas duas ideias são a base de uma unidade central de processamento (CPU), que é hoje o coração dos computadores. Começando com von Neumann, houve uma série de avanços tecnológicos importantes que levaram a que os computadores tivessem uma contínua e significativa melhoria do seu

desempenho computacional permitindo assim a sua utilização no processamento de imagem digital. Resumidamente, esses avanços podem ser apresentados da seguinte forma:

- (1) a invenção do transistor de Bell Laboratories em 1949;
- (2) o desenvolvimento, nas décadas de 1950 e 1960 das linguagens de programação de alto nível COBOL (*Common Business-Oriented Language*) e FORTRAN (*Fórmula Translator*);
- (3) a invenção do circuito integrado (IC) pela Texas Instruments em 1958;
- (4) o desenvolvimento de sistemas operativos no início dos anos 1960;
- (5) o desenvolvimento microprocessador (com integração num único circuito da unidade central de processamento, memória e controlos de entrada e saída) pela Intell, no início da década de 1970;
- (6) a introdução, pela IBM, do computador pessoal em 1981;
- (7) e a progressiva miniaturização dos componentes, começando pela integração em grande escala (LSI) no final dos anos 1970, a integração em muito grande escala (VLSI) na década de 1980, até à data presente da integração em ultra grande escala (ULSI).

Paralelamente a estes avanços tecnológicos houve um grande desenvolvimento nas áreas do armazenamento em massa e nos sistemas de reprodução (*displays*), sendo ambos requisitos fundamentais para o processamento digital de imagem.

Os primeiros computadores com capacidade suficiente para realizar as tarefas necessárias ao processamento de imagem apareceram no início de 1960.

O nascimento do que se chama processamento digital de imagem de hoje pode ser atribuído à capacidade dessas máquinas e ao início do programa espacial norte-americano. Nesse período, foi necessária a combinação destes dois desenvolvimentos para colocar em foco o potencial do conceito do processamento digital de imagem. O Desenvolvimento de técnicas de computação para melhorar imagens recebidas de uma sonda espacial teve o seu início no Jet Propulsion Laboratory (JPL, Pasadena, Califórnia) em 1964, quando as imagens da Lua, transmitidas pelo Ranger 7, foram processadas por um computador para corrigir vários tipos de distorção de imagem inerentes às limitações da câmara de televisão a bordo.

A Figura 2.3 mostra a primeira imagem da Lua captada pela sonda Ranger 7, em 31 de Julho de 1964, cerca de 17 minutos antes do impacto na superfície lunar. Esta foi a primeira imagem da Lua feita por um veículo espacial norte-americano. A experiência aprendida no tratamento das imagens com a sonda Ranger 7 serviu de base para os métodos de processamento utilizados para restaurar e melhorar as imagens nas missões seguintes à Lua, nas missões Mariner de voo a Marte, nos vôos tripulados à Lua (missões Apollo) e outros.





Fig. 2.3 A primeira imagem da Lua captada pela Sonda Ranger 7 em 31 de Julho de 1964 às 9:09 A.M. hora da costa Leste, as marcas na imagem, são marcas de referência usados para correcções geométricas (NASA).

Em paralelo com as aplicações espaciais, as técnicas de processamento de imagens digitais começaram também nos finais da década de 1960 e início da década de 1970 a ser utilizadas em imagens médicas, observação e monitorização remota dos recursos do planeta e na astronomia. A invenção no início da década de 1970 da tomografia axial computadorizada (TAC) foi um dos eventos mais relevante na aplicação de imagens para diagnóstico médico. A tomografia axial computadorizada é um processo em que um anel de detectores circunda um objecto (ou paciente) e uma fonte de raios-X, concêntrica com o Anel detector, gira sobre o objecto. Os raios-X passam pelo objecto e são colectados na extremidade oposta pelos detectores correspondentes do anel, enquanto a fonte gira. Este procedimento é repetido. A tomografia consiste em algoritmos que usam os dados detectados para construir uma imagem que represente um corte (secção) através do objecto.

O movimento do objecto numa direcção perpendicular ao anel de detectores produz um conjunto dessas secções (“fatias”), que constituem uma interpretação tridimensional (3D) do interior do objecto. A Tomografia foi inventada de forma independente por Godfrey N. Hounsfield e Allan M. Cormack, que compartilharam o Prêmio Nobel da Medicina em 1979 pela sua invenção. É interessante notar que os raios-X foram descobertos em 1895 por Wilhelm Conrad Roentgen, o qual recebeu o Prêmio Nobel (Física) em 1901. Estas duas invenções que têm quase 100 anos de intervalo, conduziam a uma

das áreas de aplicação mais activas no processamento digital de imagens da actualidade.

Da década de 1960 até ao presente, a área do processamento de imagem cresceu vigorosamente, ainda bastante para além das aplicações em medicina e no programa espacial. As técnicas de processamento digital de imagem são actualmente utilizadas numa ampla gama de aplicações.

Os procedimentos são usados fundamentalmente para aumentar o contraste ou codificar os níveis de intensidade de cor para facilitar a interpretação de raios-X, outras imagens usadas na indústria farmacêutica e ciências biológicas. No domínio da Geografia, técnicas semelhantes são utilizadas para estudar padrões de poluição em imagens aéreas e de satélite. O melhoramento da imagem e os processos de restauração usados para processar imagens degradadas de objectos irrecuperáveis ou resultados experimentais, são técnicas amplamente utilizadas. Na área da arqueologia, métodos de processamento de imagem são utilizados com significado sucesso na restauração de imagens desvanecidas que por vezes são os únicos registos disponíveis de artefactos raros, perdidos ou danificados, após terem sido fotografados.

Na Física e áreas afins, técnicas computacionais processam imagens nos domínios de plasmas de alta energia e microscopia electrónica. Finalmente aplicações adicionais de processamento de imagem, podem ser encontradas na Astronomia, Biologia, Direito, Defesa e Automação Industrial.

A segunda maior área de aplicação das técnicas de processamento digital de imagem relaciona-se com a resolução de problemas de percepção da máquina, aplicada na visão por computador. Neste caso, o foco incide nos procedimentos de extracção de informação da imagem, de forma adequada, para o posterior processamento em sistemas computadorizados. Normalmente esta informação tem pouca semelhança com as características visuais que os seres humanos usam para interpretação do conteúdo de uma imagem. Alguns dos tipos de informação utilizados na percepção da máquina são: propriedades estatísticas, coeficientes da transformada de Fourier (análise em frequência), medidas de distância multidimensionais, etc. Problemas típicos na percepção da máquina que normalmente utilizam técnicas de processamento de imagem são o reconhecimento automático de caracteres, visão industrial para a montagem e inspecção de produtos, reconhecimento militar, processamento automático de impressões digitais, rastreio de raios-X, processamento de imagens aéreas e de satélite para previsão climática e avaliação ambiental, robótica móvel e condução autónoma de veículos, etc.

Finalmente, a diminuição contínua na razão entre o preço dos equipamentos e o seu desempenho, o grande incremento dos processos distribuídos em rede, a banda larga na Internet, a Nuvem (*Cloud*) e os novos sistemas de equipamentos moveis, abrem oportunidades de crescimento sem precedentes, no futuro próximo, área do processamento digital de imagens.

## 2.2.2 Processamento digital de imagem

No processamento de imagem pretende-se a partir de uma imagem de entrada obter a imagem de saída, que se relacione com a primeira, e que seja de alguma forma mais fácil de ser tratada. Para além disso pretende-se, a partir de uma imagem de entrada, obter vários níveis de informação para posterior utilização, em vários domínios, na visão por computador.

### 2.2.2.1 Sistemas

Um sistema é um modelo matemático de um processo físico que relaciona o sinal de entrada,  $x(t)$  ou  $x[n]$ , com o sinal de saída  $y(t)$  ou  $y[n]$ , (figura 2.4), tal que:

$$y = G\{x\} \quad (2.1)$$

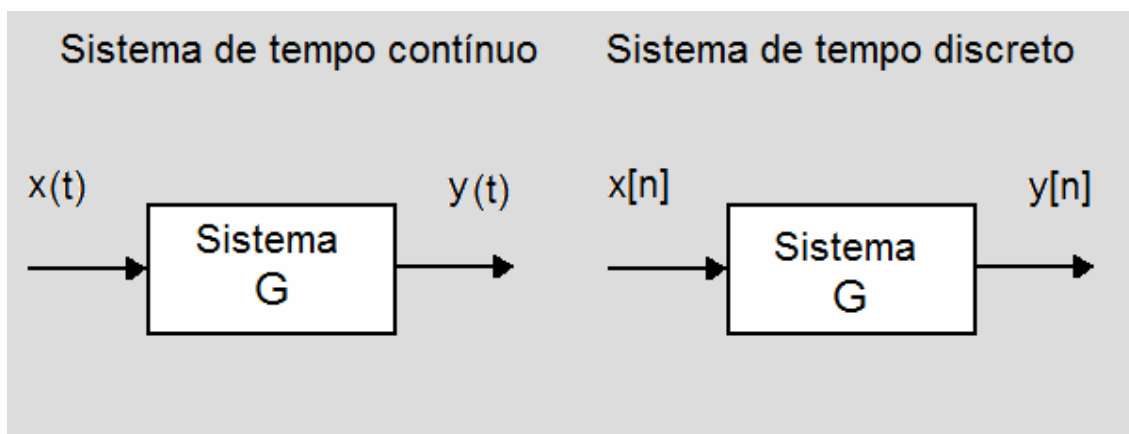


Fig. 2.4 Sistemas de tempo contínuo e tempo discreto.

#### 2.2.2.1.2 Sistemas lineares

Se um sistema  $G$  satisfizer as seguintes duas condições, aditividade e homogeneidade, respectivamente (2.2) e (2.3), então  $G$  é chamado de sistema linear:

$$G\{x_1\} = y_1 \quad e \quad G\{x_2\} = y_2 \quad \Leftrightarrow \quad G\{x_1 + x_2\} = y_1 + y_2 \quad \forall x_1 \text{ e } x_2 \quad (2.2)$$

$$G\{\alpha \cdot x_1\} = \alpha \cdot y_1 \quad \Leftrightarrow \quad \forall x \text{ e } \alpha \quad (2.3)$$

### 2.2.2.1.3 Sistemas invariante no tempo

Um sistema é invariante no tempo se um deslocamento no tempo (atraso ou avanço) no sinal de entrada, origina o mesmo deslocamento no sinal de saída. Temos então a definição de sistema invariante no tempo, discreto:

$$\begin{aligned}G\{x[n-k]\} &= y_1[n-k] \\G\{x[n+k]\} &= y_1[n+k]\end{aligned}\tag{2.4}$$

### 2.2.2.1.4 Sistemas Linear e Invariante no Tempo (SLIT)

Se um sistema é linear e invariante no tempo, então ele é chamado de sistema linear invariante no tempo.

### 2.2.2.1.5 Resposta de um SLIT

A saída de qualquer SLIT no tempo (contínuo ou discreto), obtém-se através da convolução da entrada,  $x(t)$  ou  $x[n]$ , com a resposta ao impulso  $h(t)$  ou  $h[n]$ , do sistema (ver figura 2.5):

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t - \tau) d\tau$$

ou

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k] \cdot h[n - k]\tag{2.5}$$

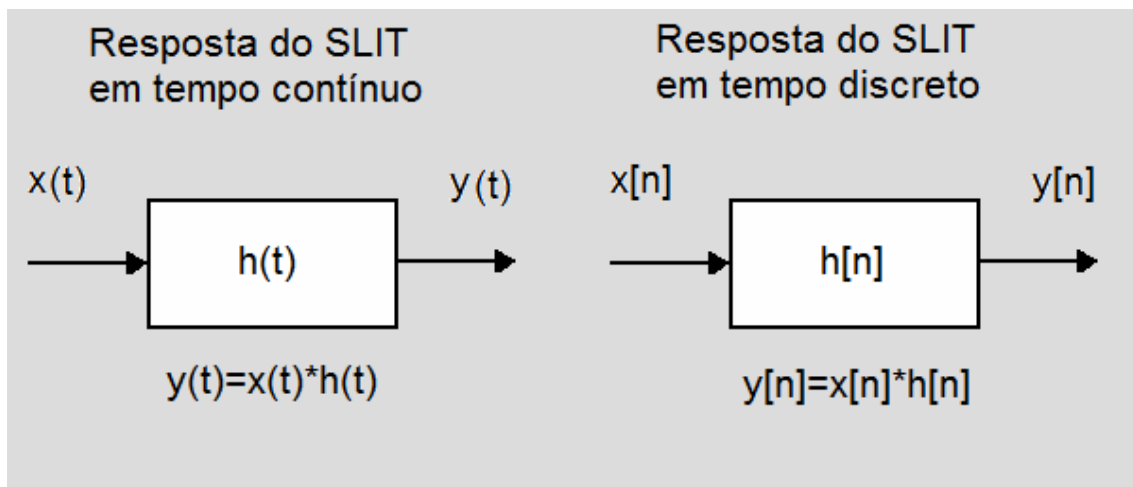


Fig. 2.5 Resposta do SLIT em tempo contínuo e discreto.

Se passarmos para campo da frequência, no caso contínuo (Transformada de Laplace), ou no caso discreto (Transformada em Z), a resposta do sistema SLIT será a seguinte, onde teremos a chamada função de transferência  $H(s)$  ou  $H[z]$  (com  $s=j.\omega$  e  $z=j.\Omega$ ), respectivamente para o sistema contínuo e para o sistema discreto (ver fig. 2.6), com a vantagem da convolução se “transformar” num produto:

$$y(t) = x(t) * h(t) \Leftrightarrow Y(s) = X(s) \cdot H(s)$$

ou (2.6)

$$y[n] = x[n] * h[n] \Leftrightarrow Y[z] = X[z] \cdot H[z]$$

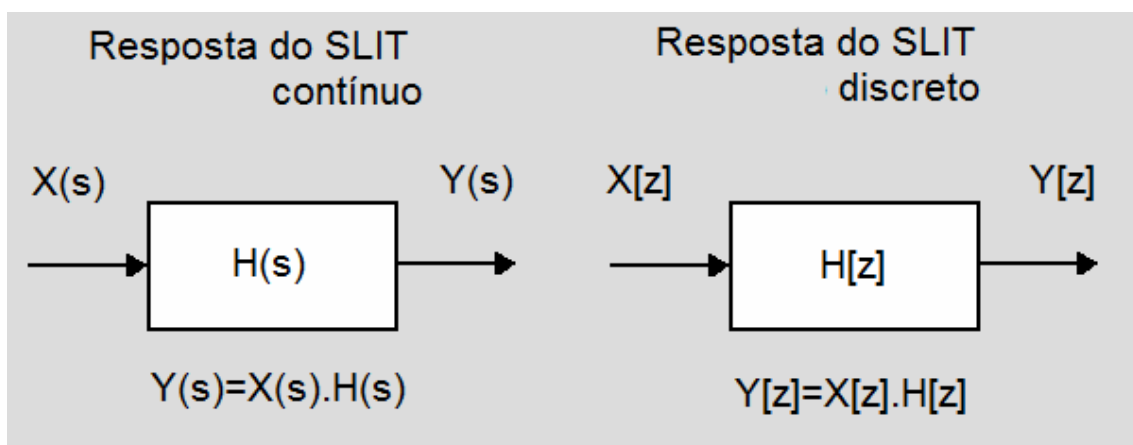


Fig. 2.6 Resposta do SLIT em frequência.

No processamento digital de imagem o sistema SLIT é caracterizado por  $h[n]$  que será um filtro ou uma transformação aplicada a imagem, a imagem inicial será então o sinal de entrada  $x[n]$  e a imagem resultante será o sinal de saída  $y[n]$ . A convulsão irá realizar-se no domínio discreto e espacial (2D) como se poderá ver no ponto 2.3.5 deste capítulo.

### 2.2.2.2 Digitalização

Através do processo de digitalização, uma imagem “real”, capturada por um sensor (exemplo: câmara), é discretizada, ou seja, digitalizada. A imagem digital obtida é resultado da aplicação de uma função que transforma a imagem numa ou varias matrizes, dentro do respectivo espaço de cor, ou seja, cada ponto da imagem (pixel) será o resultado de uma função de duas variáveis, que representam as coordenadas do ponto codificado:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N) \\ f(1,0) & f(1,1) & \dots & \dots \\ \dots & \dots & \dots & f(M-1,N) \\ f(M,0) & \dots & f(M,N-1) & f(M,N) \end{bmatrix} \quad (2.7)$$

onde  $M \times N$  são a dimensão da imagem em pixels

Pode considerar-se uma imagem digital monocromática, com 256 tons de cinzento, como uma matriz de  $n$  linhas por  $m$  colunas, em que cada pixel assume um valor entre 0 e 255 representando um tom de cinzento. O branco puro terá um valor de 255 e o preto puro terá um valor de 0. Para esta resolução, cada pixel será representado por um byte (8 bits) - ver Fig 2.7.

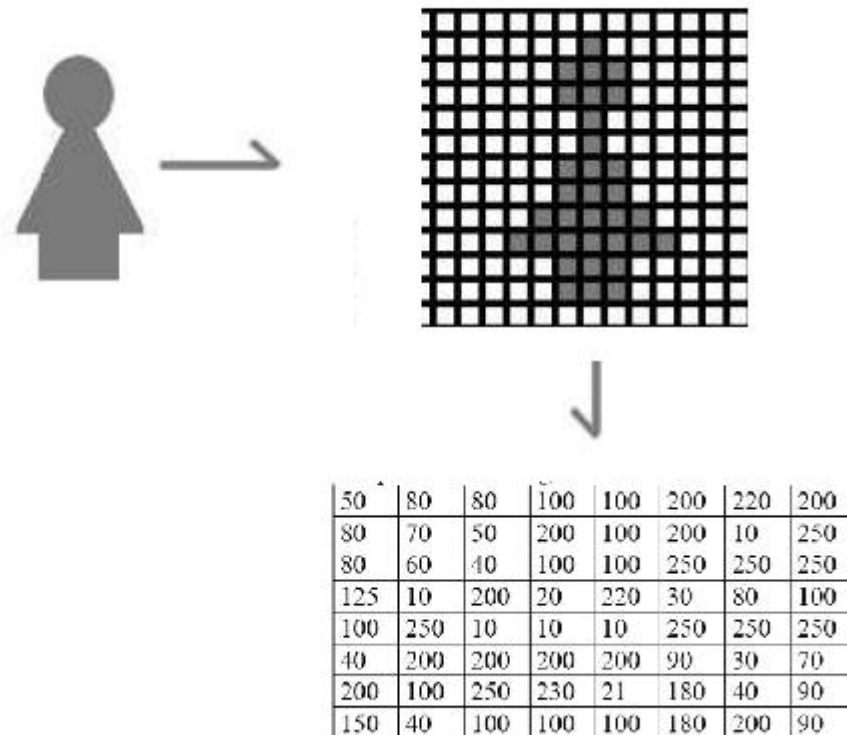


Fig. 2.7 Digitalização de uma imagem.

### 2.2.2.3 Segmentação

A segmentação de imagens é a técnica que subdivide uma imagem em partes, ou objectos, que a constituem. Quando o olho humano capta uma imagem e a analisa, o cérebro humano divide a cena em elementos, que poderão ser um único objecto ou um elemento mais complexo visualmente (ex.: uma árvore num bosque).

Se os pontos das áreas em que foi subdividida a imagem partilharem uma ou mais características comuns (ex.: cor), a utilização de algoritmos de reconhecimento de padrões poderão ser utilizados para as funções de análise, em processamento de imagem.

Na análise por segmentação, o número de objectos que resultam da mesma, depende do nível de detalhe (mais ou menos "profundo") ou da abstracção do processo. Por exemplo, na análise de uma imagem com blocos habitacionais, num primeiro nível de abstracção poderemos ter só as casas e as ruas, num nível mais profundo, poderemos identificar as portas e as janelas de cada casa. Logo, o algoritmo de segmentação não resulta numa só solução, dependendo sim do objectivo do processamento da imagem.

### 2.2.2.4 Imagens binárias

As imagens binárias caracterizam-se por serem constituídas por pontos pretos e brancos puros. Nestas imagens não há tons de cinzento intermédios. Esta é uma das formas utilizadas por muitos programas de computador para armazenarem grandes imagens, em ficheiros de reduzida dimensão, uma vez que para guardar a informação relativa a cada pixel basta 1 bit em vez dos 8 bits necessários para os 256 tons de cinzento (ver fig 2.8).

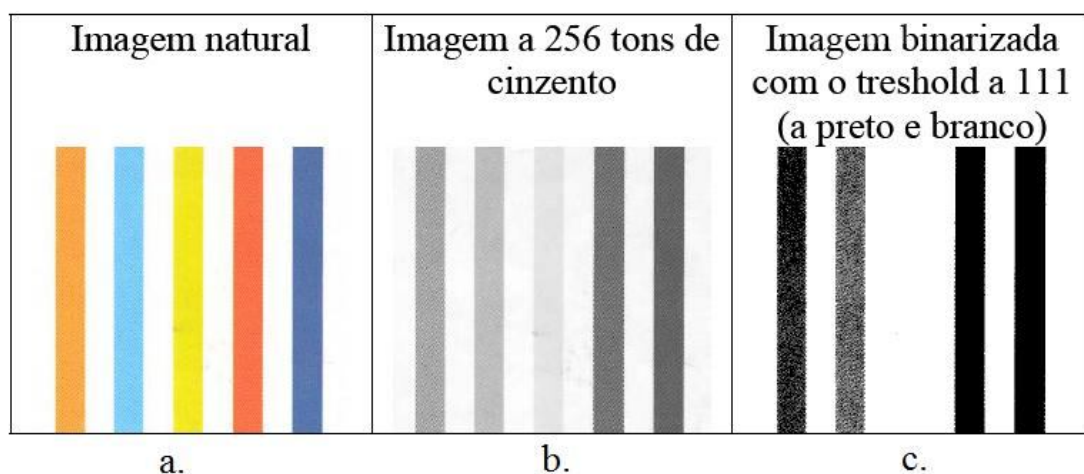


Fig. 2.8 Imagem a cores, imagens em tons de cinzento e imagem binarizada.

### 2.2.2.5 Espaços de Cor

A cor é uma importante fonte de informação no processamento digital de imagem, nomeadamente numa ampla gama de áreas de investigação como por exemplo segmentação, reconhecimento de padrões, análise espectral, etc. A forma como a cor de cada ponto (pixel) da imagem é codificado remete-nos para os espaços de cor. Adicionalmente, no processamento pode ser necessário “condicionar” as cores originais da imagem tendo em conta o tipo de análise pretendida. O “condicionamento” das cores pode ser realizado por transferência entre espaços de cor, preservando os detalhes e a aparência natural, ou não, da imagem original. A consideração dos espaços de cor é também de extrema importância na aquisição das imagens, no seu armazenamento e ainda na capacidade de computação necessária para a realização dos algoritmos no processamento digital. Além disso, alguns autores propõem que a selecção de um espaço de cor específico está mais relacionado com o gosto pessoal do autor do que com o seu fundamento em evidências experimentais. No entanto, a escolha do espaço de cor é muito importante tendo em conta os vários processos que se vão realizar no processamento de imagem, desde a aquisição, armazenamento, processamento e análise.



Os vários métodos de modelação reagem de formas diferentes com a mudança de espaço de cor. Seguidamente apresentam-se os vários tipos de espaços de cor.

### 2.2.2.5.1 RGB

Neste espaço de cor as cores são especificadas em termos das três cores primárias: vermelha (R-Red), verde (G-Green) e azul (B-Blue) (fig. 2.9). É um modelo aditivo, porque todas as cores são criados a partir das combinações das componentes primárias de cor (fig. 2.10). Se cada componente primário for codificado em 8 bits, teremos 24 bits para as três componentes primárias e com essa dimensão é possível representar até cerca de 16,7 milhões de cores ( $2^{24} = 16777216$ ). Por exemplo, para a cor vermelha temos o termo ordenado (255, 0, 0), para o verde (0,255, 0), e assim, combinado o R e G, em quantidades iguais, poderemos criar o amarelo, definido por (255, 255, 0). A quantificação de cada componente de cor primária dá a sua intensidade. Se os 3 componentes tiverem a intensidade máxima (255, 255, 255), teremos a cor branca. O RGB é um dos espaços de cor de maior utilização, sendo utilizado nas placas de vídeo dos computadores e na tecnologia de monitores. A sua utilização na detecção de pele humana é muito condicionada por alterações nas condições de luz (iluminação).

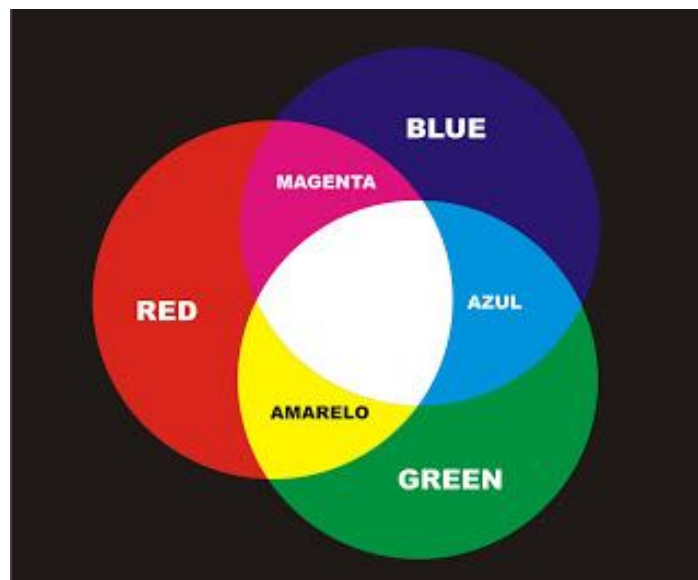


Fig. 2.9 Espaço de cor RGB ([www.graficacriasett.com.br/o-que-e-cromia-cmyk/rgb/](http://www.graficacriasett.com.br/o-que-e-cromia-cmyk/rgb/), acedido em Maio 2019).

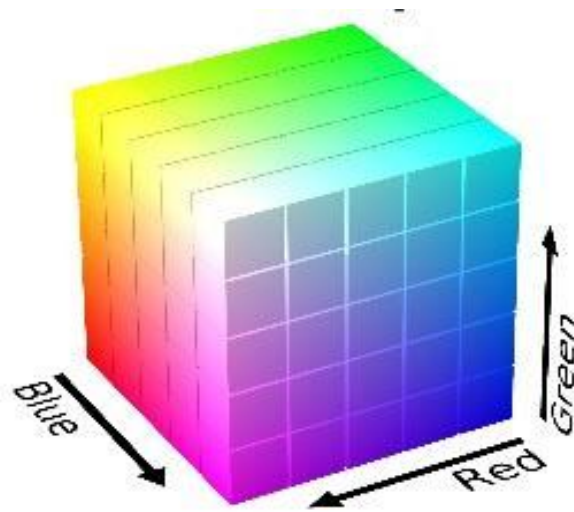


Fig. 2.10 Espaço de cor RGB ([https://www.researchgate.net/figure/RGB-left-and-HSV-right-color-spaces\\_fig1\\_310474598](https://www.researchgate.net/figure/RGB-left-and-HSV-right-color-spaces_fig1_310474598), acedido em Maio de 2019).

#### 2.2.2.5.2 RGB Normalizado

A fim de se reduzir a dependência do espaço RGB em relação às condições de iluminação, os valores RGB podem ser normalizados, por um simples procedimento de normalização conforme seguidamente se apresenta, dando origem ao RGB normalizado.

$$r = \frac{R}{R + G + B} \quad (2.8)$$

$$g = \frac{G}{R + G + B} \quad (2.9)$$

$$b = \frac{B}{R + G + B} \quad (2.10)$$

A soma dos componentes normalizados do RGB é igual à unidade ( $r + g + b = 1$ ), assim sendo,  $r$  e  $g$  são componentes suficientes para representar uma cor neste espaço de cor, pois a terceira componente,  $b$ , pode ser obtida directamente dos valores de “ $r$ ” e “ $g$ ”. O RGB e a sua versão normalizada estão entre os espaços de cor mais popularmente utilizados na detecção de pele humana.

### 2.2.2.5.3 YCbCr

O espaço de cor YCbCr é usualmente utilizada pelos canais de televisão Europeia. Tem como objectivo reduzir a redundância presente nos canais do espaço de cor RGB e representa a cor através de componentes estatisticamente independentes. “Y” representa a luminância, e Cr e Cb, as componentes de cromaticidade, sendo definidas respectivamente por:

$$Y = 0,299R + 0,587G + 0,114B \quad (2.11)$$

$$C_b = B - Y \quad (2.12)$$

$$C_r = R - Y \quad (2.13)$$

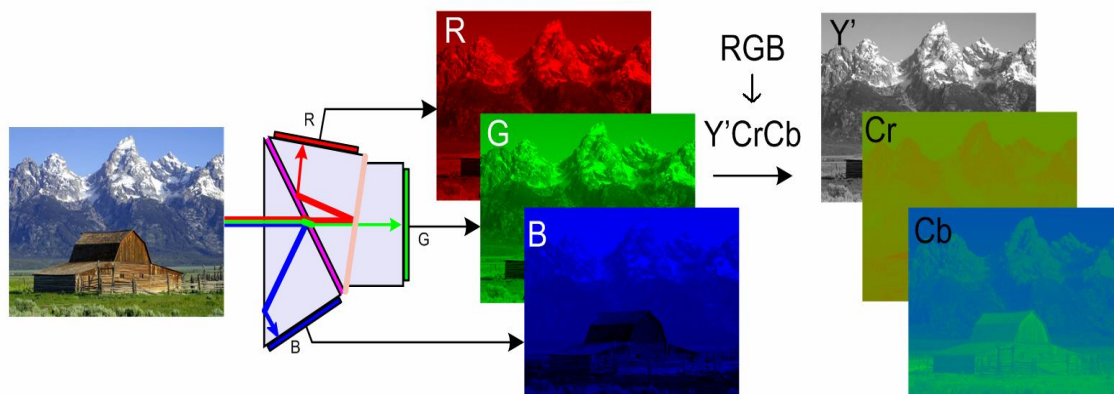


Fig. 2.11 Espaço de cor YCbCr (en.wikipedia.org/wiki/YCbCr, acedido em Maio de 2018).

Em comparação com o espaço de cor RGB, o YCbCr tem a separação explícita das componentes de luminância e cromaticidade (fig. 2.11), o que o torna muito utilizada para a detecção de pele por segmentação.

### 2.2.2.5.4 YIQ

O espaço de cor YIQ é semelhante ao espaço de cor YCbCr, pertencem à mesma categoria de espaços de cor ortogonais é o espaço de cores utilizado pelo sistema de TV a cores, NTSC, empregue principalmente na América do Norte, América Central e Japão. A luminância é representada por Y, e a

chrominância é representada por “I” e “Q”. O valor “I” descreve a mudança de laranja para azul, enquanto “Q” descreve a mudança de roxo para verde. A transformação do espaço de cor RGB para o espaço de cor YIQ permite separar a informação de luminância da de crominância (fig. 2.12). Esta separação eficaz da informação permite que o espaço de cor YIQ seja bastante útil em segmentação. O seguinte conjunto de equações é usado para transformar o espaço de cor RGB no espaço de cor YIQ:

$$Y = 0,299R + 0,587G + 0,114B \quad (2.14)$$

$$I = 0,596R - 0,275G - 0,321B \quad (2.15)$$

$$Q = 0,212R + 0,523G + 0,311B \quad (2.16)$$

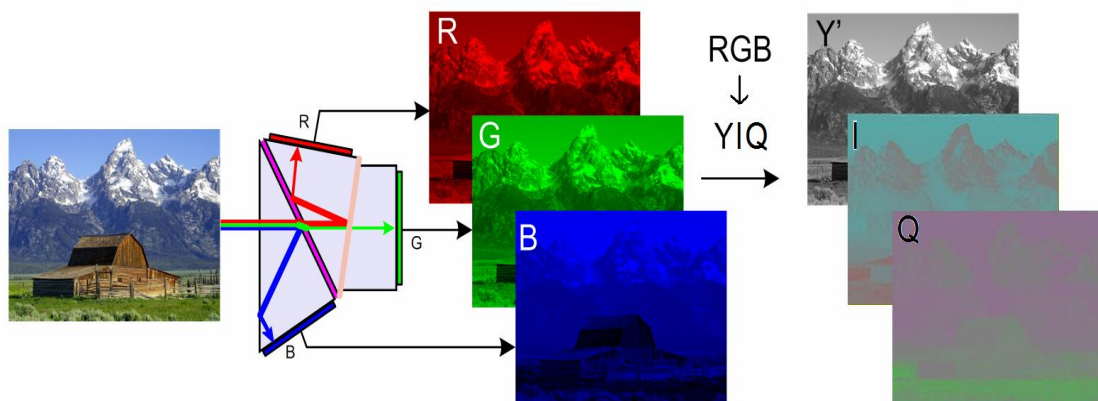


Fig. 2.12 Espaço de cor YIQ (en.wikipedia.org/wiki/YIQ, acedido em Maio de 2018).

### 2.2.2.5.5 HSV (HSL)

O espaço de cor Hue-Saturation-Value foi introduzido pela necessidade de especificar as propriedades de cor de forma quantitativa. As componentes descrevem a cores com valores intuitivos, baseados na ideia “artística” da coloração, saturação e tom. O Hue (matiz) define a cor dominante de uma área (vermelho, verde, roxo e amarelo), a Saturation (saturação) mede o colorido de uma área, em relação ao contraste, e por fim o Value (valor) é a componente que quantifica a luminância da cor (fig. 2.13). O seguinte conjunto de equações é usado para transformar o espaço RGB no espaço HSV:

$$H = \arccos \frac{\frac{1}{2}((R-G) + (R-B))}{\sqrt{((R-G)^2 + (R-B)(G-B))}} \quad (2.17)$$

$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B} \quad (2.18)$$

$$V(L) = \frac{1}{3}(R + G + B) \quad (2.19)$$

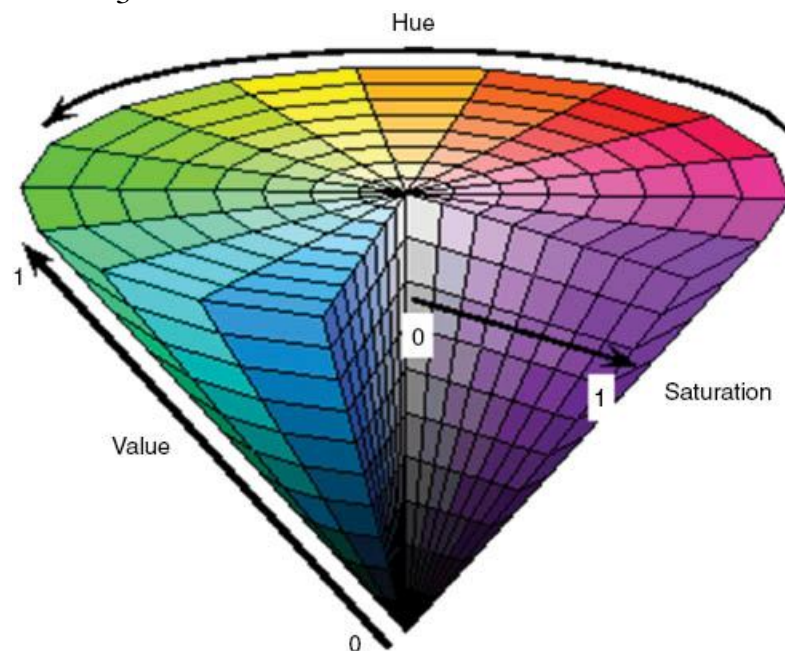


Fig. 2.13 Espaço de cor HSV ([www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html](http://www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html), acessado em Maio de 2019).

### 2.2.2.5.6 YUV

Este modelo também define o espaço de cor em três componentes. Um componente de luminância (canal Y) e dois componentes de crominância (canais U e V). O “U” representa a diferença entre o sinal azul e a luminância (B-Y) e “V” representa a diferença entre o vermelho e a luminância (R-Y). O sistema YUV é usado para transmissões analógicas televisivas, como o PAL ou o NTSC, permitindo manter a compatibilidade com os sistemas de recepção Preto-Branco (fig. 2.14). Sabendo que a visão humana é muito mais sensível a variações na luminância do que a variações na crominância, o sistema de codificação de cor YUV tem em conta esta característica, atribuindo assim uma maior largura de banda à luminância de modo a que o espaço de cor YUV se

aproxime da percepção humana, embora o espaço YUV não seja tão eficiente, neste aspecto, como o espaço HSV, anteriormente caracterizado).

Os sinais YUV são criados a partir dos componentes originais RGB. Os valores ponderados de R, G e B são adicionadas em conjunto para produzir um único sinal Y, representando o brilho geral ou luminância, desse ponto. O canal U é obtido subtraindo o canal Y à componente B do sinal RGB original e, em seguida multiplicado por um factor de escala. O canal V é obtido subtraindo ao canal Y à componente R do sinal RGB original multiplicando depois por um factor de escala de valor diferente. Matematicamente, a conversão RGB para YUV obtem-se:

$$Y = 0,299R + 0,587G + 0,114B \quad (2.20)$$

$$U = -0,14713R - 0,28886G + 0,436B \quad (2.21)$$

$$V = 0,615R - 0,51499G - 0,10001B \quad (2.22)$$

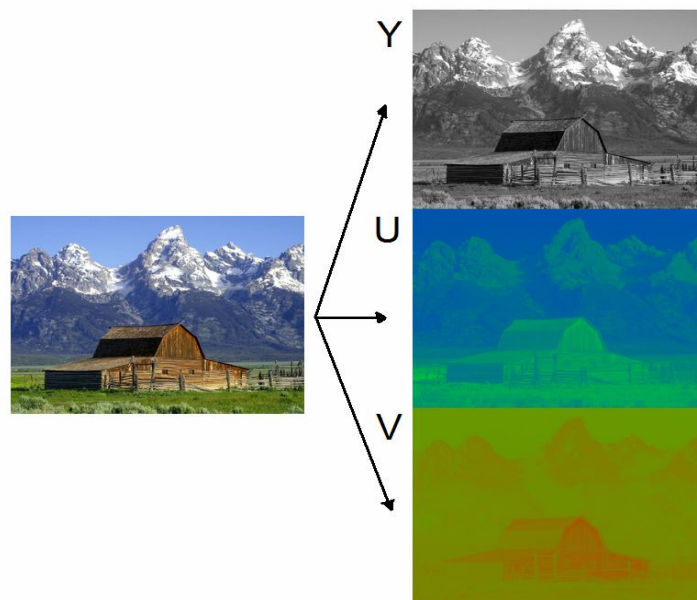


Fig. 2.14 Espaço de cor YUV (en.wikipedia.org/wiki/YUV, acedio em Maio de 2018).

### 2.2.2.5.7 YDbDr

Este espaço de cor é bastante semelhante aos anteriores, previamente caracterizados.

O sistema YDbDr é o espaço de cor utilizado no padrão de transmissão de televisão por satélite analógico, SECAM padrão, utilizado em França e em alguns países do antigo Bloco do Leste. É muito semelhante ao espaço YUV e está relacionado com os espaços de cor YIQ, YPbPr e YCbCr.

O YDbDr é composto por três componentes: Y; Db e Dr. A componente Y representa a luminância, As componentes Db e Dr representam as componentes de cromaticidade (sendo respectivamente as diferenças em relação ao azul, B e ao vermelho, R) (fig. 2.15). Tal como acontece com os anteriores modelos, também a transformação entre os espaços RGB e YDbDr é definido pelo conjunto de transformações lineares seguidamente apresentadas:

$$Y = 0,299R + 0,587G + 0,114B \quad (2.23)$$

$$Db = -0,450R - 0,883G + 1,333B \quad (2.24)$$

$$Dr = -1,333R + 1,116G + 0,217B \quad (2.25)$$

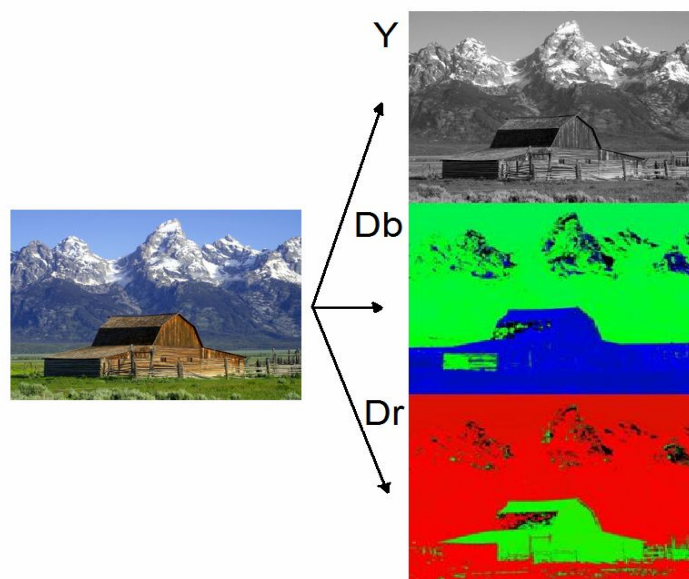


Fig. 2.15 Espaço de cor YDbDr (en.wikipedia.org/wiki/YDbDr, acedido em Maio de 2018).

### 2.2.2.5.8 CIE L\* a\* b

O CIE (Commission International de L'Éclairage) L\* a\* b é o espaço de cor onde as três coordenadas representam respectivamente: a luminosidade da cor (L\* = 0 indica o preto e L\* = 100 indica o branco difuso); a posição entre o vermelho e o verde (a\*, os valores negativos indicam verde, enquanto os valores positivos indicam vermelho) e por fim a posição entre o amarelo e o azul (b\*, os valores negativo indicam azul e valores positivos indicam amarelo), (fig. 2.16). A coordenada L\* varia de 0 a 100. A gama possível das coordenadas a\* e b\* é independente do espaço de cor que se está a transformar. Os valores X, Y e Z são obtidos a partir do espaço RGB, conforme as expressões:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0,17697} \cdot \begin{bmatrix} 0,49 & 0,31 & 0,20 \\ 0,17697 & 0,81240 & 0,01063 \\ 0 & 0,01 & 0,99 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.26)$$

$$L^* = 116 \times f\left(\frac{Y}{Y_n}\right) - 16 \quad (2.27)$$

$$a^* = 500 \times \left[ f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad (2.28)$$

$$b^* = 200 \times \left[ f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \quad (2.29)$$

$$f(t) = \begin{cases} \frac{1}{2} & : t > \left(\frac{6}{29}\right)^2 \\ \frac{1}{3} \left(\frac{29}{6}\right)^2 t + \frac{4}{29} & : \text{other } w. \end{cases} \quad (2.30)$$

onde X<sub>n</sub>, Y<sub>n</sub> e Z<sub>n</sub> são valores normalizados CIE XYZ do ponto branco de referência.



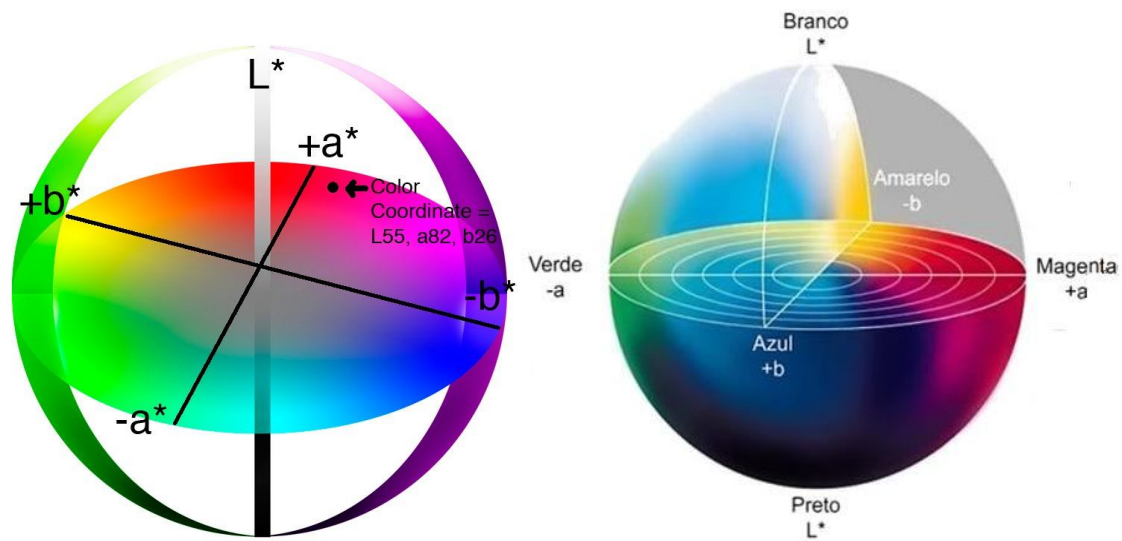


Fig. 2.16 Espaço de cor CIE  $L^* a^* b^*$  (www.researchgate.net/figure/Figura-26-Espaco-de-cor-CIELab\_fig4\_235438609, acessado em Maio de 2019).

### 2.2.2.6 Binarização (segmentação monocromática)

É frequente as imagens a processar estarem sobre um fundo de uma qualquer cor, o que dificulta o tratamento da imagem. A binarização é uma transformação de grande utilidade sempre que pretendemos localizar formas e contornos na imagem.

A binarização consiste em definir um valor como nível de separação (*threshold*), onde todos os pixels, com valor inferior a este, assumem o valor zero (preto), e todos os pixels, com valor acima deste, assumem o valor 255 (branco).

Definindo o nível de cinzento de separação (*threshold*) para um valor abaixo do fundo, todo o fundo ficará branco. Como exemplo, para um *threshold* de 111, a imagem utilizada anteriormente, na fig.2.7, após a sua binarização, ficará conforme é apresentado na fig.2.17.b

50	80	80	100	100	200	220	200
80	70	50	200	100	200	10	250
80	60	40	100	100	250	250	250
125	10	200	20	220	30	80	100
100	250	10	10	10	250	250	250
40	200	200	200	200	90	30	70
200	100	250	230	21	180	40	90
150	40	100	100	100	180	200	90

0	0	0	0	0	255	255	255
0	0	0	255	0	255	0	255
0	0	0	0	0	255	255	255
255	0	255	0	255	0	0	0
0	255	0	0	0	255	255	255
0	255	255	255	255	0	0	0
255	0	255	255	0	255	0	0
255	0	0	0	0	255	255	0

a.
b.

Fig. 2.17 a. Matriz original, b. Matriz depois da Binarização.

Caso se pretendesse uma compactação no ficheiro, a mesma poderia ser realizada em formato binário (apenas 1 bit por pixel), da seguinte forma (fig. 2.18):

0	0	0	0	0	1	1	1
0	0	0	1	0	1	0	1
0	0	0	0	0	1	1	1
1	0	1	0	1	0	0	0
0	1	0	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	1	1	0	1	0	0
1	0	0	0	0	1	1	0

Fig. 2.18 Matriz no formato binário (0,1).

Para efeitos de aplicação práticos observa-se o exemplo mostrados na fig.2.19.a. e 2.19.b :

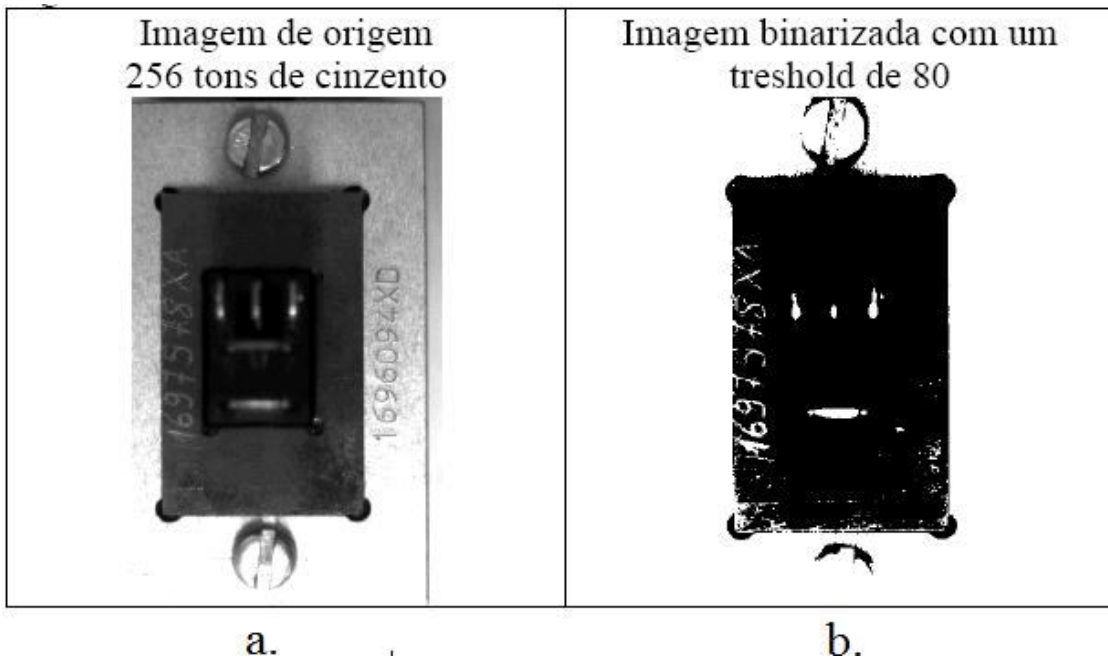


Fig. 2.19 Binarização.

### 2.2.2.7 Definição de pixel vizinho

Para a quantificação de certas áreas das regiões da imagem, assim como para a aplicação de alguns algoritmos de processamento de imagem, torna-se necessário definir o conceito de pixel vizinho.

- Diz-se que 2 pixels são 4-vizinhos se ao partilharem um limite, partilham também uma face (fig.2.20.a).
- Diz-se que 2 pixels são 8-vizinhos se ao partilharem um limite, partilham também pelo menos um canto (fig.2.20.b).

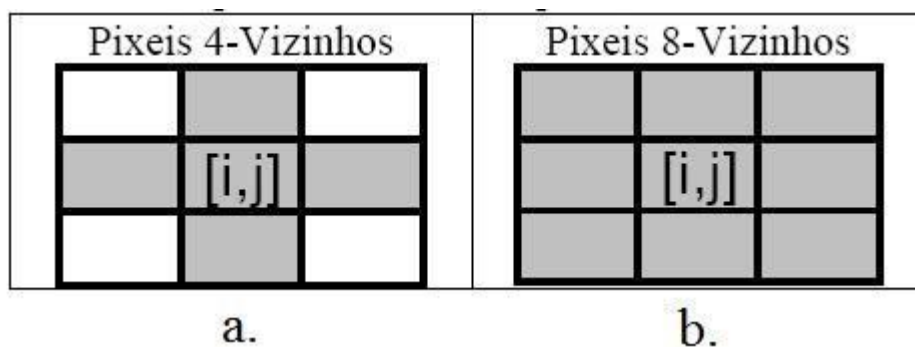


Fig. 2.20 Pixel vizinho.

### 2.2.2.8 Definição de Caminhos

Para a quantificação de arestas e perímetros de certas regiões da imagem, bem como para a aplicação de alguns algoritmos de processamento de imagem, torna-se necessário definir o conceito de caminho.

- Diz-se que um caminho é um 4-Caminho se as ligações entre os pixels que formam o caminho forem pixels 4-Vizinhos (fig.2.21.a).
- Diz-se que um caminho é um 8-Caminho se houver ligações entre os pixels que formam o caminho forem pixels 8-Vizinhos., fig.2.21.b.

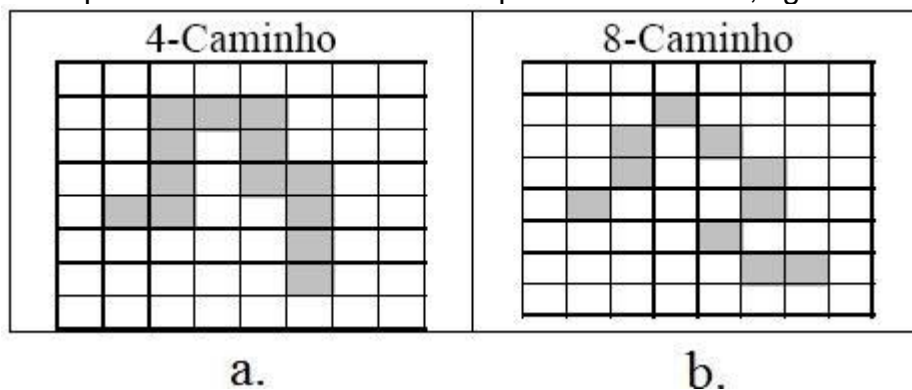


Fig. 2.21 Caminhos: 4-Caminho e 8-Caminhos.

### 2.2.2.9 Definição de Fronteira

Para a definição de certas regiões na imagem, bem como para a aplicação de alguns algoritmos de processamento de imagem (por ex. morfológicos), que serão referidos mais à frente neste trabalho, torna-se necessário definir o conceito de fronteira. Define-se como fronteira de um objecto, numa imagem, o conjunto de pontos que estão no limite do objecto e que têm 4-Vizinhos (fig.2.22.b).

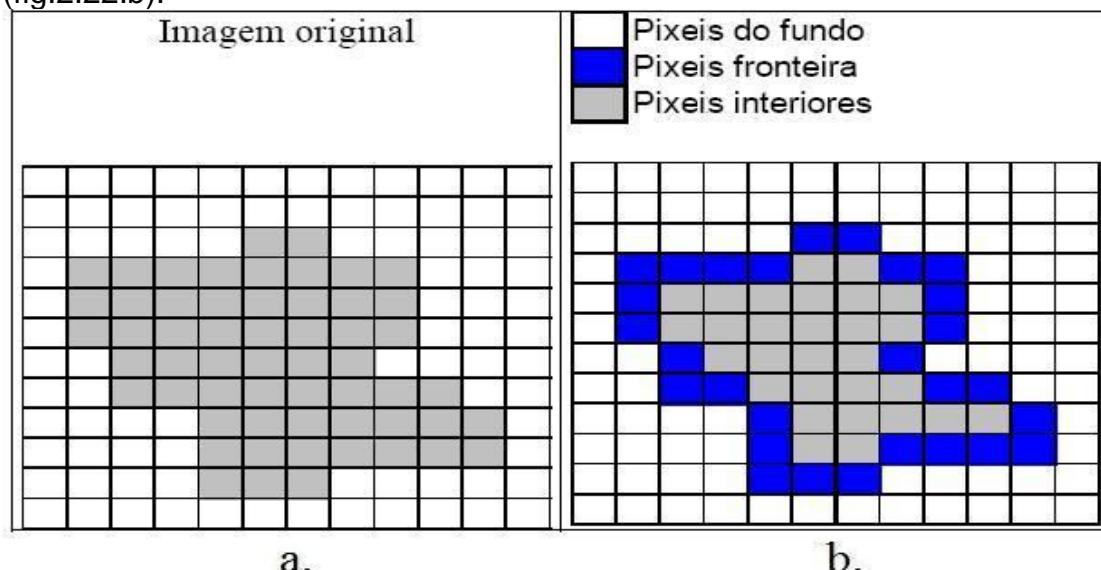


Fig. 2.22 Pixels Fronteira.

O tamanho de uma área da imagem, representado em notação matricial, será uma matriz com a dimensão  $n \times m$ , representada por:

$$A = \sum_{i=1}^n \sum_{j=1}^m I[i, j] \quad (2.31)$$

### 2.2.2.10 Definição de Distância

Para a quantificação da distância entre dois pontos de uma imagem torna-se necessário definir 3 tipos de distâncias.

- **Distância Euclidiana** (fig. 2.23.a)

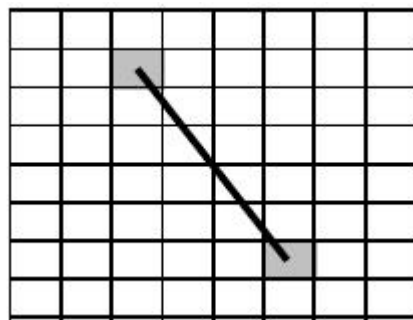


Fig. 2.23.a Distância euclidiana.

$$d_{Euclidiana} = ([i_1, j_1], [i_2, j_2]) = \sqrt{(i_2 - i_1)^2 + (j_2 - j_1)^2} \quad (2.32)$$

- **Distância quarteirão (city block)** (fig. 2.23.b)

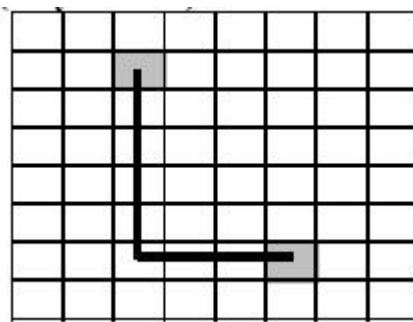


Fig. 2.23.b Distância quarteirão.

$$d_{\text{quarteirão}} = ([i_1, j_1], [i_2, j_2]) = |i_2 - i_1| + |j_2 - j_1| \quad (2.33)$$

- **Distância tabuleiro de xadrez (*Chessboard*) (fig2.23.c)**

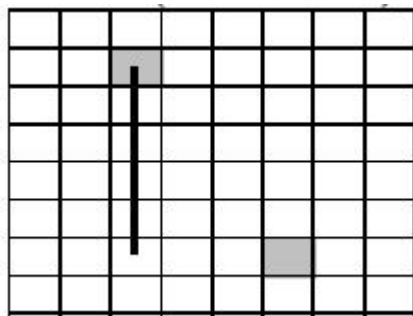


Fig. 2.23.c Distância tabuleiro de xadrez.

$$d_{\text{xadrez}} = ([i_1, j_1], [i_2, j_2]) = \max(|i_2 - i_1|, |j_2 - j_1|) \quad (2.34)$$

## 2.3 Algoritmos de processamento de imagem

### 2.3.1 Algoritmo de filtro de tamanho

Normalmente as imagens apresentarem-se com “ruído”. Este ruído surge por indefinição na obtenção do nível de intensidade a associar a determinado pixel, seja na aquisição directa da imagem, seja depois em pós processamento dessas imagens. Por exemplo, na definição do nível de *threshold* para uma binarização/segmentação, quando o nível escolhido é muito próximo do nível da intensidade do fundo, pode haver pontos pertencentes ao fundo que adquiram o valor zero (preto), ficando erradamente a ser identificados como objectos.

Exemplo de imagem com ruído, fig. 2.24.

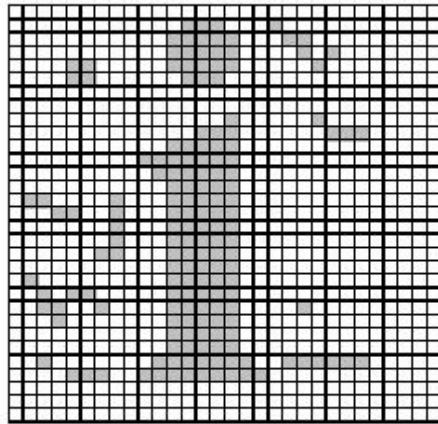


Fig. 2.24 Imagem original com ruído.

O filtro de tamanho é bastante simples e eficaz em imagens binárias. Vulgarmente o “ruído” da imagem resulta no aparecimento de pseudo-objects, normalmente de pequenas dimensões. O filtro de tamanho consiste em remover todos os objects que se apresentem abaixo de um determinado tamanho (área em pixels). Por exemplo, aplicando este filtro para a imagem anterior, bastaria definir  $T=10$  para obter a imagem isenta de ruído (ver fig. 2.25).

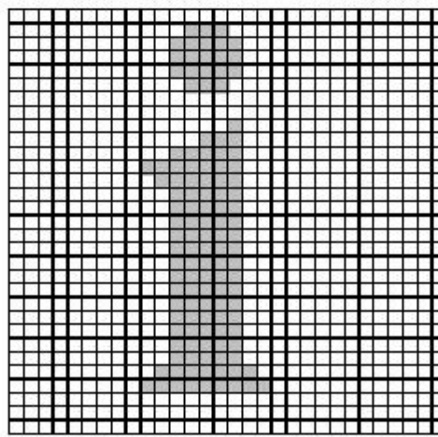


Fig. 2.25 Imagem sem ruído.

No caso analisado na fig. 2.25, teria que haver o cuidado de não utilizar  $T > 21$ , pois nesse caso o ponto do  $i$  também seria removido. Este filtro tem a sua aplicação mais adequada em imagens binárias.

### 2.3.2 Algoritmo de filtro morfológico Erosão e dilatação

Os filtros morfológicos pela sua definição alteram as dimensões das áreas onde são aplicadas, no caso da erosão e dilatação, são usualmente utilizados um em “sequência” do outro.

Os objectos ou áreas podem ser peças a analisar, mas também podem ser partículas de sujidade ou perturbações da imagem.

No algoritmo os objectos brancos sobre fundo negro reduzem-se por erosão e expandem-se por dilatação. Os objectos negros sobre fundo branco expandem-se por erosão e reduzem-se por dilatação, fig. 2.26.

Erosão – Muda o pixel de 1 para 0 se algum pixel vizinho for 0

Dilatação – Muda o pixel de 0 para 1 se algum pixel vizinho for 1.

Imagem original	Erosão	Dilatação
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 1 1 1 1 1 0
0 0 1 1 1 0 0	0 0 0 0 0 0 0	0 1 1 1 1 1 0
0 0 1 1 1 0 0	0 0 0 1 0 0 0	0 1 1 1 1 1 0
0 0 1 1 1 0 0	0 0 0 1 0 0 0	0 1 1 1 1 1 0
0 0 1 1 1 0 0	0 0 0 1 0 0 0	0 1 1 1 1 1 0
0 0 1 1 1 0 0	0 0 0 0 0 0 0	0 1 1 1 1 1 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 1 1 1 1 1 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 0 0 0 0 0 1	1 1 1 1 1 1 1
1 1 0 0 0 1 1	1 0 0 0 0 0 1	1 1 1 1 1 1 1
1 1 0 0 0 1 1	1 0 0 0 0 0 1	1 1 1 0 1 1 1
1 1 0 0 0 1 1	1 0 0 0 0 0 1	1 1 1 0 1 1 1
1 1 0 0 0 1 1	1 0 0 0 0 0 1	1 1 1 0 1 1 1
1 1 0 0 0 1 1	1 0 0 0 0 0 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 0 0 0 0 0 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1

Fig. 2.26 Filtro morfológico erosão – dilatação.

Como já referido anteriormente, obtêm-se bons resultados de redução de “ruído” em imagens adquiridas (fig. 2.27) aplicando consecutivamente: Erosão-Dilatação (fig. 2.27.a) ou Dilatação-Erosão (fig. 2.27.b).

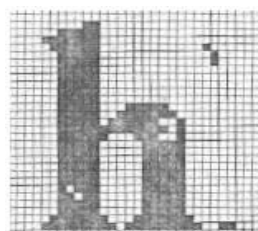


Imagem original

Fig. 2.27 Imagem original com ruído.



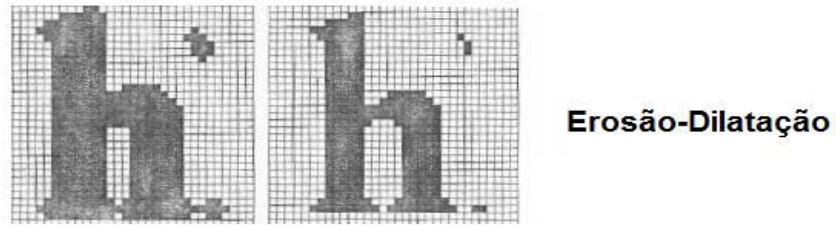


Fig. 2.27.a Aplicação da erosão-dilatação.

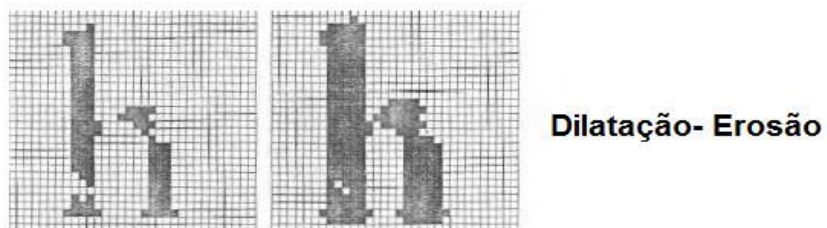


Fig. 2.27.b Aplicação da dilatação-erosão.

### 2.3.3 Algoritmo de Igualização do histograma

Por vezes as imagens a tratar apresentam-se com tons de cinzento, muito próximos, com pouco contraste, o que dificulta a sua análise. Pela observação do seu histograma, observa-se que determinados níveis de cinzento têm pouca probabilidade de ocorrer. A igualização do histograma permite redistribuir os níveis de cinzento pelos tons que não estão utilizados (fig. 2.28).

O algoritmo calcula os novos tons de cinzento da imagem utilizando a seguinte expressão:

$$l = \frac{255}{N} \cdot \sum_{i=0}^k n_i \quad (2.35)$$

onde:

$l$  – Novo tom de cinzento

$N$  – n<sup>o</sup> total de pixels

$n_i$  – pixel de índice  $i$  que pertence a todos os anteriores e ao nível antigo de cinzento entre 0 e  $k$  inclusive

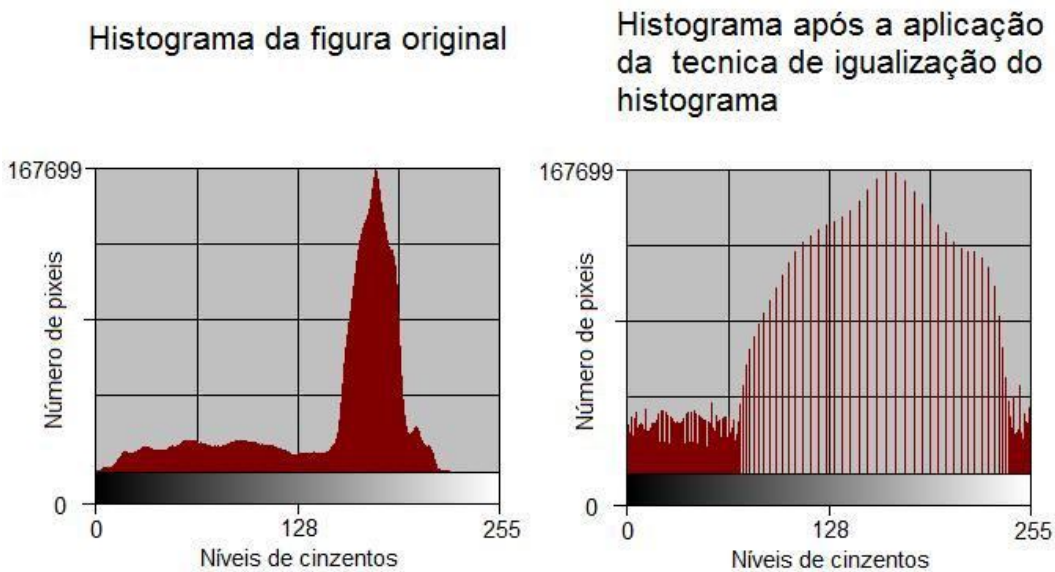


Fig. 2.28 Igualização do histograma.

### 2.3.4 Algoritmo segmentação dos níveis de cinzento

Este algoritmo de aumento de contraste faz com que o menor valor de nível de cinzento se altere para zero e que o maior valor de nível de cinzento se altere para 255, expandindo a distribuição pela maior gama possível de tons de cinzento:

$$l = (x - \min) \frac{255}{\max - \min}. \quad (2.36)$$

onde:

$l$  – Novo tom de cinzento

$x$  – Tom de cinzento original

$\min$  – Nível mínimo de tom de cinzento que ocorre na imagem

$\max$  – Nível máximo de tom de cinzento que ocorre na imagem

### 2.3.5 Algoritmos de filtro com elemento estruturante

Frequentemente as imagens apresentam-se com “ruído”. Este ruído surge por indefinição na obtenção do nível a associar a determinado pixel, seja na aquisição directa da imagem ou depois em pós-processamento dessas mesmas imagens. Para eliminar ou reduzir este ruído ou também para realizar transformadas na imagem utiliza-se a técnica de filtro no domínio espacial (2D).

Esta técnica de processamento digital de imagem consiste em calcular o novo valor de um pixel tendo em conta uma operação sobre os valores dos pixels de uma vizinhança na imagem digital. Para o efeito utiliza-se um elemento estruturante  $h[1..n,1..n]$  e realiza-se a convolução discreta no domínio espacial como é apresentada na expressão 2.37, no qual  $x[1..u,1..v]$  é a imagem inicial e  $y[1..u,1..v]$  a imagem resultante, ver a figura 2.29.

$$y[i,j] = x[i,j] * h[i,j] = \sum_{q=0}^{n-1} \sum_{k=0}^{n-1} x[i+q-1, i+k-1] \cdot h[n-q, n-k] \quad (2.37)$$

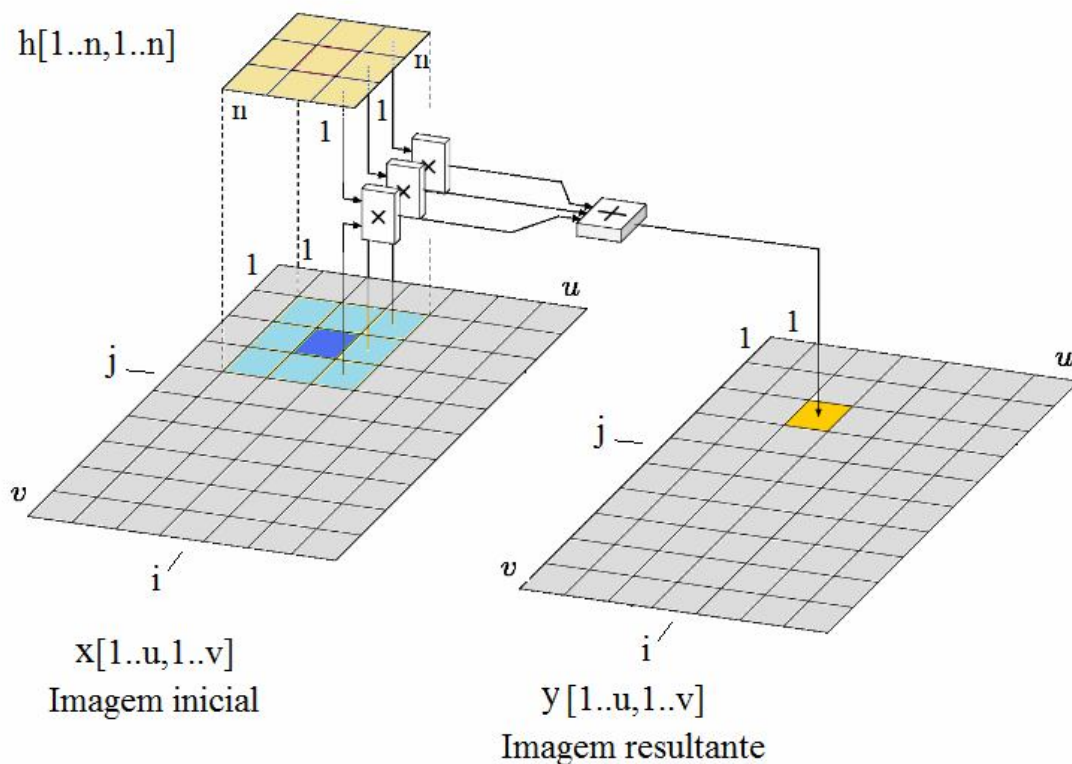


Fig. 2.29 Convolução discreta no domínio espacial.

### 2.3.5.1 Algoritmo de filtro de mediana

Na prática esta operação corresponde a fazer deslizar o elemento estruturante, a “janela” ( $n \times n$ ) ao longo da imagem original, substituindo o valor do pixel central, pelo valor da mediana dos pixels da “janela” ( $n \times n$ ).

### 2.3.5.2 Algoritmo de filtro de média

Este algoritmo tem uma aplicação muito semelhante ao filtro de mediana, apenas com a particularidade de vez de se substituir o valor do pixel central pela mediana, substitui-se pela média dos valores da janela.

### 2.3.5.3 Algoritmo de filtro de mínimo

Este algoritmo também tem uma aplicação muito semelhante ao filtro de mediana e de média, mas em vez de se substituir o valor do pixel central pela mediana ou pela média, substitui-se pelo mínimo dos valores da “janela” deslizante.

### 2.3.5.3 Algoritmo de filtro de Máximo

Como os anteriores este algoritmo também tem uma aplicação semelhante ao filtro de mediana e de média, mas em vez de se substituir o valor do pixel central pela mediana ou pela média, substitui-se pelo máximo dos valores da “janela” deslizante.

### 2.3.5.4 Algoritmo de filtro de Sobel

Sendo um algoritmo do tipo elemento estruturante, a sua aplicação é semelhante aos anteriores. Considerando também uma matriz de processamento 3x3 com o formato:

A	B	C
D	E	F
G	H	I

O operador Sobel incorpora a informação em 2 direcções, de acordo com:

$$M(S_x, S_y) = \sqrt{(A + 2.B + C - G - 2.H - I)^2 + (A + 2.D + G - C - 2.F - I)^2} \quad (2.38)$$
$$M(S_x, S_y) = \sqrt{S_x^2 + S_y^2}$$

Onde  $S_x$  e  $S_y$  também podem ser obtidos por convolução dos dois elementos estruturantes (matrizes 3x3):

$S_x$ :

$$\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

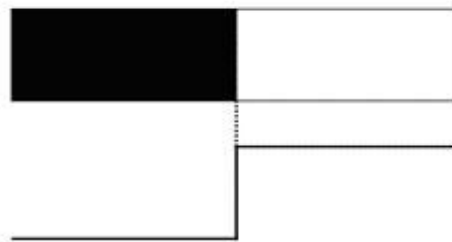
$S_y$ :

$$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

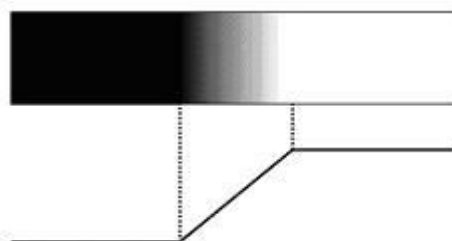
### 2.3.6 Algoritmo para detecção de arestas

Pode-se considerar uma aresta como sendo uma significativa alteração local na intensidade da imagem, habitualmente associada a uma descontinuidade na intensidade da imagem. As descontinuidades são normalmente classificadas de acordo com os tipos representados na figura 2.30.

Degrau



Rampa



Telhado

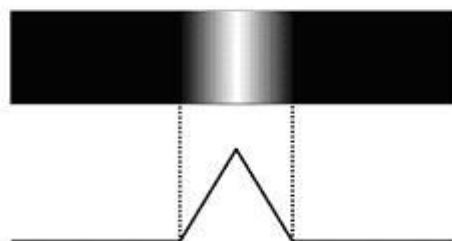


Fig. 2.30 Descontinuidade na intensidade da imagem.

As arestas “ideais” são as do tipo degrau ou linha, contudo são raras nas imagens reais, devido a componentes de baixa frequência, próprios da aquisição de imagem, ou devidos à utilização prévia de algum algoritmo de filtragem (*smoothing*) já abordado anteriormente. É então comum que as arestas se apresentem geralmente na forma de rampa ou “telhado”.

Também é frequente que as arestas reais se apresentem como um misto de degrau e linha. Este fenómeno acontece frequentemente em objectos com as arestas “boleadas” que, por reflexo da luz na zona boleada, se apresentam com um contorno de maior luminosidade.

Apresentam-se seguidamente algumas definições comuns na detecção e processamento de arestas:

- **Ponto de aresta** – É um ponto de uma imagem com coordenadas  $[i,j]$  numa localização com uma alteração significativa de intensidade na imagem;
- **Fragmento de aresta** – Corresponde às coordenadas  $i$  e  $j$  de uma aresta com orientação de aresta  $\theta$ , que pode ser o ângulo de gradiente;
- **Detector de arestas** – É um algoritmo que produz um conjunto de arestas (pontos de arestas ou fragmentos de arestas) de um objecto numa imagem;
- **Contorno** – É uma lista de arestas ou uma curva matemática que modela a lista de arestas;
- **Ligação de arestas** – É o processo de formação de uma lista de arestas ordenadas, a partir de uma lista desordenada. Por convenção, as arestas são ordenadas transversalmente e no sentido dos ponteiros do relógio;
- **Seguimento de aresta** – É um processo de procura para detectar contornos na imagem.

### 2.3.6.1 Algoritmo Gradiente

A detecção de arestas é essencialmente a detecção de alterações locais significativas numa imagem. Visto a uma dimensão, uma aresta do tipo degrau é associada com um pico local na primeira derivada (fig. 2.31). O gradiente é a medida da alteração da continuidade numa função, e uma imagem pode ser vista como uma matriz de amostras, de uma função contínua de intensidade dessa imagem. Por analogia, alterações significativas nos tons de cinzento, de uma imagem, podem ser detectados usando uma aproximação discreta do gradiente.

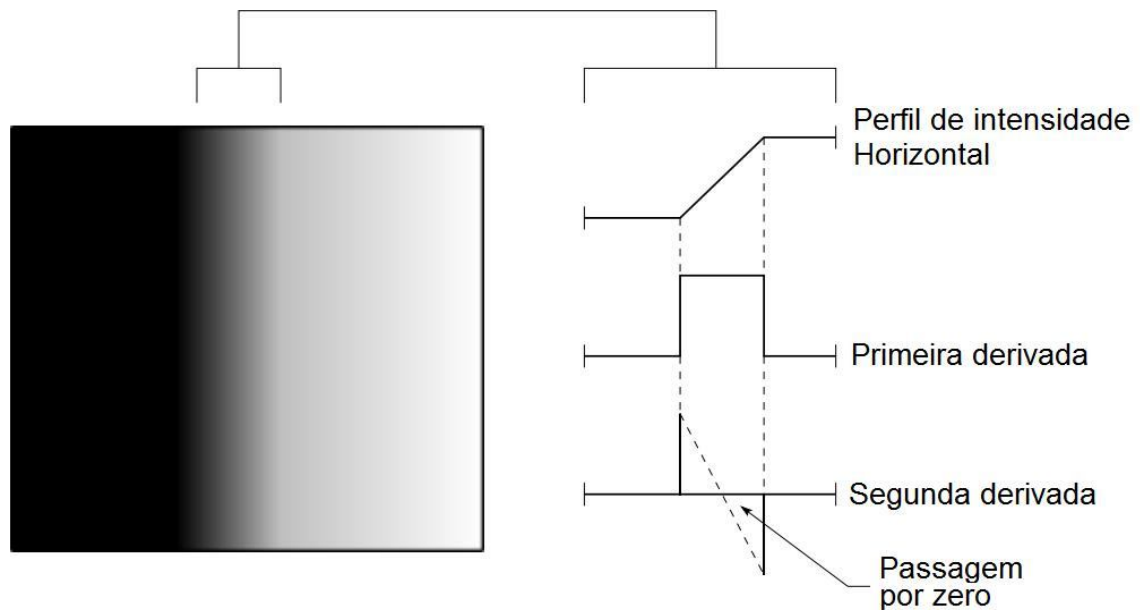


Fig. 2.31 Detecção de arestas.

O gradiente é o equivalente bidimensional da primeira derivada e pode ser definido pelo vector:

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.39)$$

Propriedades relevantes

- O vector  $G[f(x, y)]$  aponta na direcção do aumento máximo da função  $f(x, y)$
- A amplitude do gradiente é dada por

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2} \quad (2.40)$$

e representa a taxa de aumento máximo por distância unitária

- Da análise do vector gradiente, a sua direcção é definida por

$$\alpha(x, y) = \text{tg}^{-1} \left( \frac{G_x}{G_y} \right) \quad (2.41)$$

em que o ângulo  $\alpha$  é medido em relação ao eixo x

Para imagens digitais as derivadas são aproximadas por diferenças, em que a aproximação mais simples é:

$$\begin{aligned}G_x &\cong f[i, j+1] - f[i, j] \\G_y &\cong f[i, j] - f[i+1, j]\end{aligned}\tag{2.42}$$





# Capítulo 3

## Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células

*Neste capítulo é apresentado um novo método de detecção do movimento em imagens 2D, baseado na variação adaptativa da dimensão das células de busca rectangulares. A dimensão da célula é definida automaticamente com base num método de adaptação estatística, que tem em conta os resultados de pesquisa anteriores. A principal contribuição do presente método, em relação aos métodos usuais de pesquisa de movimento de imagem, é a metodologia que auto adapta e reduz significativamente o trabalho computacional em comparação com metodologias de células de imagem de dimensão fixa.*

### 3.1 Introdução

Nos métodos de detecção de movimento baseados no de fluxo óptico, utilizados em visão artificial, temos:

- Diferenciação entre imagens (*interframe differencing*) ;
- Diferenciação à imagem de referência (*reference frame differencing*).

Conforme já foi referido, o desempenho de ambos os métodos é drasticamente influenciado pela área dos pixels, onde é aplicado o método e a utilização subsequente dos resultados obtidos.

O método de processamento de imagem proposto neste capítulo, varia a área da célula de trabalho, tendo em conta a amplitude de movimento detectado (dado pelo número de pixels onde é detectado o movimento). A redução da área da célula de busca melhora a velocidade de processamento do método, sendo a redução da área da célula de trabalho feita de forma adaptável e criteriosa, pois tem em conta a avaliação estatística do processamento das imagens anteriores. Sendo assim a eficiência do método aumenta drasticamente.

Os algoritmos adaptativos são amplamente utilizados em aplicações de processamento digital de sinal, tais como: sistemas de cancelamento ruído; filtros digitais e transmissão dados.

Normalmente estes algoritmos adaptativos usam uma função adaptativa que varia de acordo com uma característica do sistema específico que descreve a evolução do próprio sistema, avaliando as características que são controladas. Essa capacidade é implementada através de pesos nominais que são definidos de acordo com uma avaliação estatística dos resultados anteriores processados. Referindo-se ao filtro digital FIR, com dimensão  $n$ , e seguindo o algoritmo LMS usual (Haykin, 1991) temos:

$$y(i) = k_{0i} \cdot x(i) + k_{1i} \cdot x(i-1) + \dots + k_{ni} \cdot x(i-n) \quad (3.3)$$

em que  $y(i)$  representa a resposta do filtro para a entrada  $x(i)$  no caso  $i$ , e  $k_i$  representa os factores de ponderação (pesos).

Considerando  $d(i)$  a resposta do filtro desejada no instante  $i$ , o erro pode ser calculado por:

$$e(i) = d(i) - y(i) \quad (3.4)$$

Os factores de ponderação modificados para ser utilizados na iteração seguinte ( $i+1$ ) são calculados de acordo com a regra adaptativa:

$$k_{n_{i+1}} = k_{n_i} + \mu \cdot e(i) \cdot x(i); \quad n \in N_0; \quad \mu \in [0,1] \quad (3.5)$$

onde que  $\mu$  é uma constante adaptativa de aprendizagem e  $X(i)$  é dada por (3.6).

$$X(i) = [x(i) \ x(i-1) \dots \ x(i-n)]^t \quad (3.6)$$

O método proposto é ligeiramente diferente do método descrito acima, na forma em que, na actualização dos factores de ponderação  $k_i$  o erro  $e(i)$  é substituído pela detecção de movimento, e a saída obtido é a nova dimensão da célula de trabalho da imagem, como será explicado na próxima secção.

## 3.2 Algoritmo de Processamento de Imagem

### 3.2.1 Descrição do algoritmo

O algoritmo de detecção de movimento proposto é iniciado com uma dimensão máxima de células de trabalho, igual à resolução da imagem  $P \times Q$ . A célula de trabalho no início é dividida em quatro secções de procura com dimensão  $P/2 \times Q/2$  como é mostrado na Fig. 3.1.



Fig. 3.1 Células de trabalho com a dimensão  $P/2 \times Q/2$ .

De acordo com a primeira divisão da célula original em quatro é aplicado o método de procura por diferenciação entre imagens (*interframe differencing*), sendo este aplicado ao longo das quatro linhas, como é mostrado na fig. 3.2.

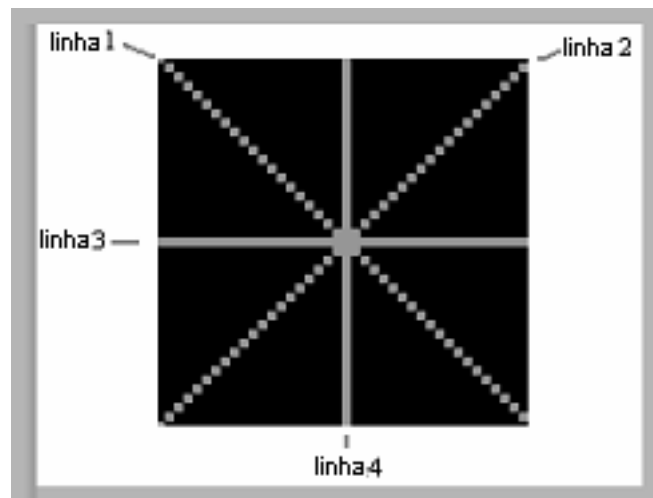


Fig. 3.2 Linhas de detecção 1 – 4.

Quando um movimento entre duas imagens (*frames*) consecutivas no tempo é detectado, a célula correspondente é dividida em:  $P/a_i \times Q/a_i$ . O parâmetro  $a_i$  é o coeficiente de divisão da célula e é calculado por (3.7).

Dependendo da função adaptativa de dimensão  $n$ , o cálculo dos factores  $a_i$  está relacionado com a amplitude do movimento detectado entre duas imagens (*frames*) consecutivos:

$$a_i = k_i \times a_{i-1} + k \times a_{i-2} + \dots + k_{i-n} \times a_{i-n-1} \quad (3.7)$$

Onde

$$k_i = D_i \times \alpha; \quad (3.8)$$

O parâmetro  $D_i$  representa o número pixels da célula na instância  $i$ , referente à célula onde o foi detectado o movimento, e  $\alpha$  é um factor adaptativo que define a evolução do algoritmo de aprendizagem. Quanto maior o valor  $\alpha$ , mais rápido se torna a progressão do algoritmo.

### 3.2.2 Redução do algoritmo

A fim de reduzir o esforço de cálculo computacional devido ao algoritmo adaptativo apresentado em (3.7) e portanto, melhorar a frequência de iteração do método, uma eventual simplificação pode ser obtida de acordo com a expressão (para a dimensão  $n = 1$ ) e com o limiar (*threshold*)  $\tau_p$ :

$$k_i = \begin{cases} 2, & D_i \geq \tau_p \\ 1/2, & otherwise \end{cases} \quad (3.9)$$

onde  $a_i$  (coeficiente de divisão da célula) é calculado por (3.10).

$$a_i = k_i \times a_{i-1} \quad (3.10)$$

Para melhor se compreender o algoritmo é apresentado um resumo na figura 3.3 que ilustra o seu funcionamento.

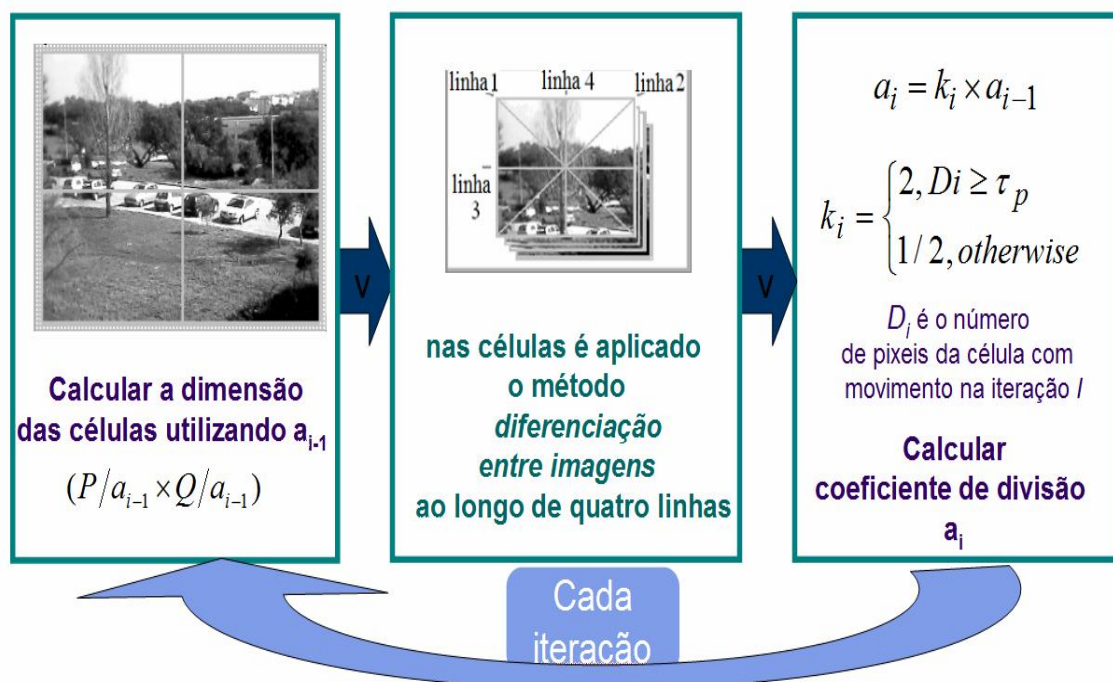


Fig. 3.3 Algoritmo de processamento de imagem.

Este algoritmo simplificado é muito eficaz em termos de esforço computacional, pois as operações em números inteiros de multiplicar ou de dividir por 2, são simples em cálculo binário: implicam apenas a rotação de um bit à direita ou à esquerda, respectivamente.

### 3.3 Implementação do sistema

Para validar a metodologia proposta, vários testes foram realizados no ambiente de desenvolvimento LabVIEW da National Instruments, com base nos dados adquiridos, através de uma *webcam* conectada a uma porta USB.

Duas aplicações de software foram desenvolvidas: um para realizar testes em imagens adquiridas pela *webcam*, em tempo real, e o outro para executar testes em formato de vídeo AVI.

Nas duas aplicações de software as imagens adquiridas foram convertidas para o formato IMAQ de LabVIEW. O formato IMAQ pode ser usado nas funções de visão da biblioteca LabVIEW, embora possa ser convertido para outro formato padrão de imagem. No trabalho descrito, os arquivos IMAQ foram convertidos em matriz RGB de 24 bits (8 bits para cada componente de cor).

Para alguns dos processos a matriz RGB é convertido numa matriz de 8 bits de níveis de cinzento. As funções do método de diferenciação entre imagens, (*interframe differencing*) para detecção de movimento, assim como as funções adaptativas, afectam directamente os elementos da matriz. No trabalho desenvolvido neste capítulo, para o processamento digital de imagem não foram utilizadas funções disponíveis nas bibliotecas do LabVIEW, foram desenvolvidas funções de raiz.



Fig. 3.4 Area 1.

Para realizar os testes de validação da metodologia proposta, as aplicações desenvolvidas no ambiente LabVIEW, apresentam uma interface do utilizador comum, a qual contém duas áreas, cujas funções são descritas. Área 1 (página anterior fig. 3.4) - Nesta área quatro imagens, janelas são apresentada. A primeira janela apresenta o vídeo a cores ou imagens, adquiridas em tempo real; a segunda janela mostra a mesma imagem em níveis de cinzento; a terceira janela identifica todos os pontos da imagem, onde algum movimento foi detectado, através do método de diferenciação entre imagens; e, finalmente, a quarta janela ilustra a imagem com as células, alcançados com o algoritmo adaptativo, que ilustra a evolução das dimensões da célula.

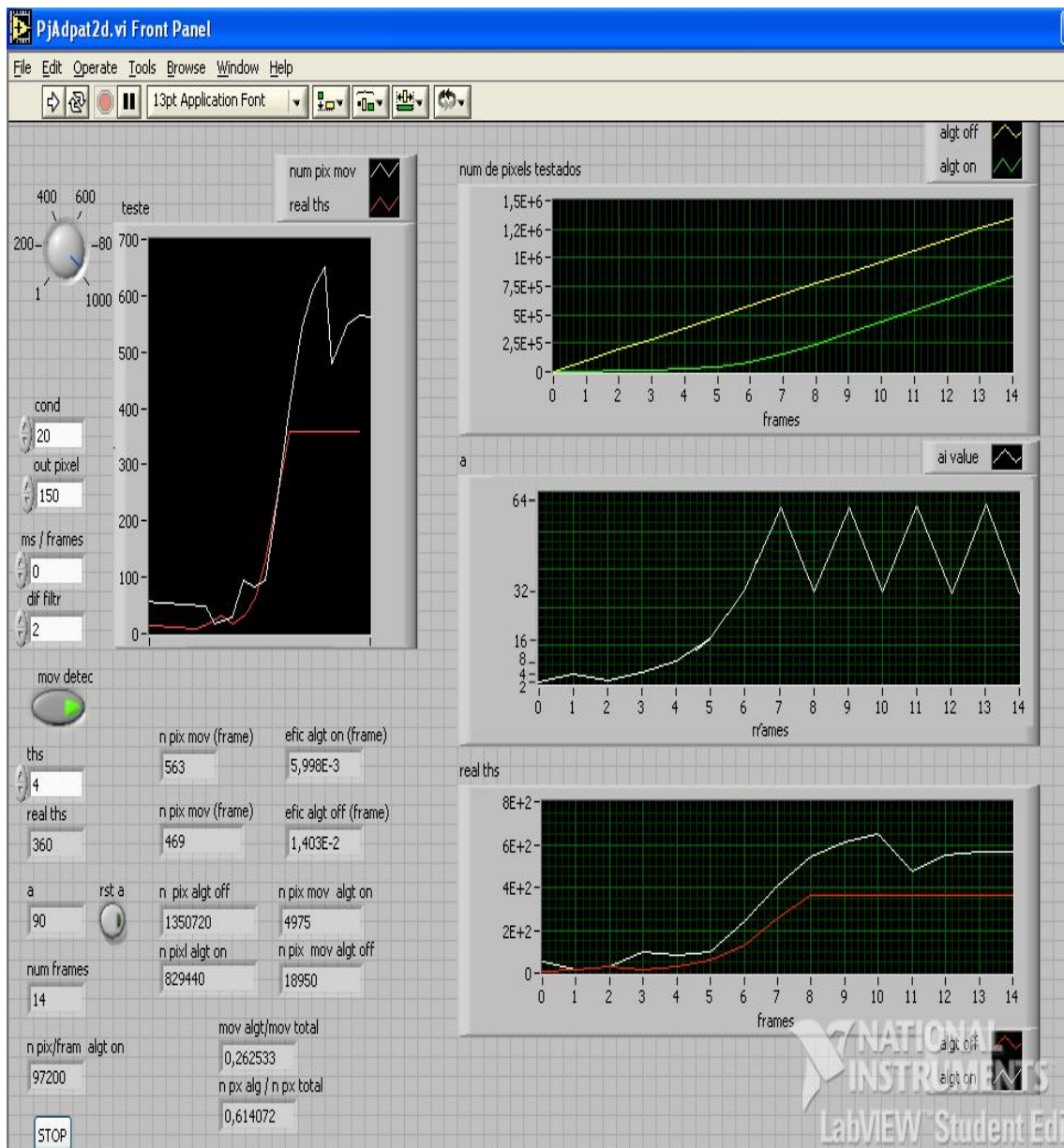


Fig. 3.5 Area 2.



A Área 2 (página anterior, Fig. 3.5) – Nesta área são apresentados os seguintes gráficos:

- I. o gráfico com o número total de pixels testados (com e sem o algoritmo);
- II. o gráfico com o  $a_i$  (coeficiente de divisão da célula) ;
- III. o gráfico com a variação do limiar dinâmico  $\tau_{pi}$ , utilizado pelo algoritmo, em cada *frame*.

Nesta área é igualmente apresentada a avaliação de desempenho do sistema. O significado de cada parâmetro utilizado para avaliar o desempenho do sistema é mostrado na secção 4.

O *software* desenvolvido para implementar a metodologia proposta, permite também gerar dados em formato ASCII que podem ser importados, como ficheiros Excel, como tabelas ou gráficos, que podem posteriormente ser incluídos em páginas HTML.

## 3.4 Análise e conclusões

### 3.4.1 Testes realizados

Nos testes realizados teve-se em conta o pior caso. Por conseguinte, foi opção usar um vídeo gravado em câmara lenta, no qual existe sempre movimento entre duas imagens (*frames*) consecutivas.

Percebeu-se que o limiar  $\tau_p$ , utilizado no algoritmo adaptativo (3.9) deveria ser dinâmico, porque com o aumento do número de células, resultante da partição da imagem, implica um aumento no número de pontos de movimento detectado entre duas imagens consecutivas. O novo limiar  $\tau_{pi}$  tem de ser adaptado ao aumento do número de células.

O limiar dinâmico é, por conseguinte, dado por:

$$\tau_{pi} = a_{i-1} \times \tau_p \quad (3.11)$$

Sendo assim, a expressão (3.9) pode ser escrita como:

$$k_i = \begin{cases} 2, & D_i \geq \tau_{pi} \\ 1/2, & otherwise \end{cases} \quad (3.12)$$

Os parâmetros seleccionados para avaliar o desempenho do sistema são os seguintes:

- Número total de pixels testados com o algoritmo desligado (n pix algth off -fig. 3.6);
- Número total de pixels testados com o algoritmo ligado (n pix algth on - fig. 3.6);
- Relação entre o número de pixels testados com e sem algoritmo ligado ( $n \text{ algoritmo algth pix on} / n \text{ pix algth off}$  - fig. 3.6);
- Número de pixels em que foi detectado movimento sem algoritmo ligado (n pix mov algth off -fig. 3.6);
- Número de pixels em que foi detectado movimento com o algoritmo ligado (n pix mov algth on - fig. 3.6);
- Relação entre o número de pixels em que foi detectado movimento, com e sem algoritmo ( $n \text{ pix algth mov on} / n \text{ pix mov algth off}$  - fig. 3.6).

Os valores dos parâmetros obtidos na realização destes testes são apresentados na Figura 3.6.

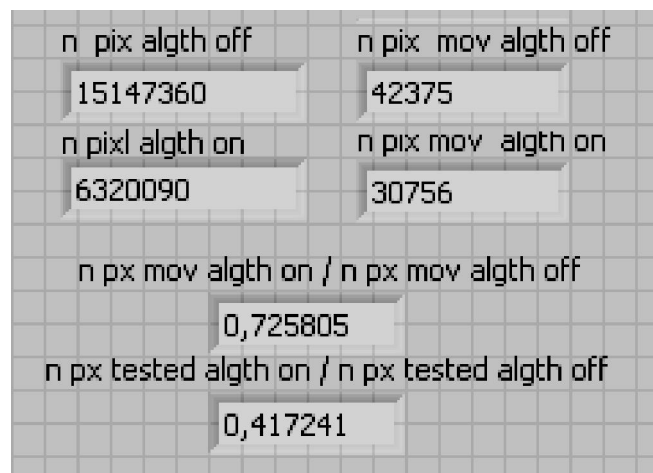


Fig. 3.6 Resultado dos testes.

Podemos ver, na figura 3.7, com a realização do teste, a progressão do valor do coeficiente de divisão da célula  $a_i$ , e a respectiva variação da dimensão e do número das células. Do *frame* 1 ao *frame* 6 assiste-se a uma mudança nas células, devido à acção do algoritmo de adaptação (3.12). Esta mudança é função da “quantidade” de movimento detectada nas células. A “quantidade” de movimento é medida pelo número de pixels que sofrem alteração, a partir de um *frame* para o seguinte.

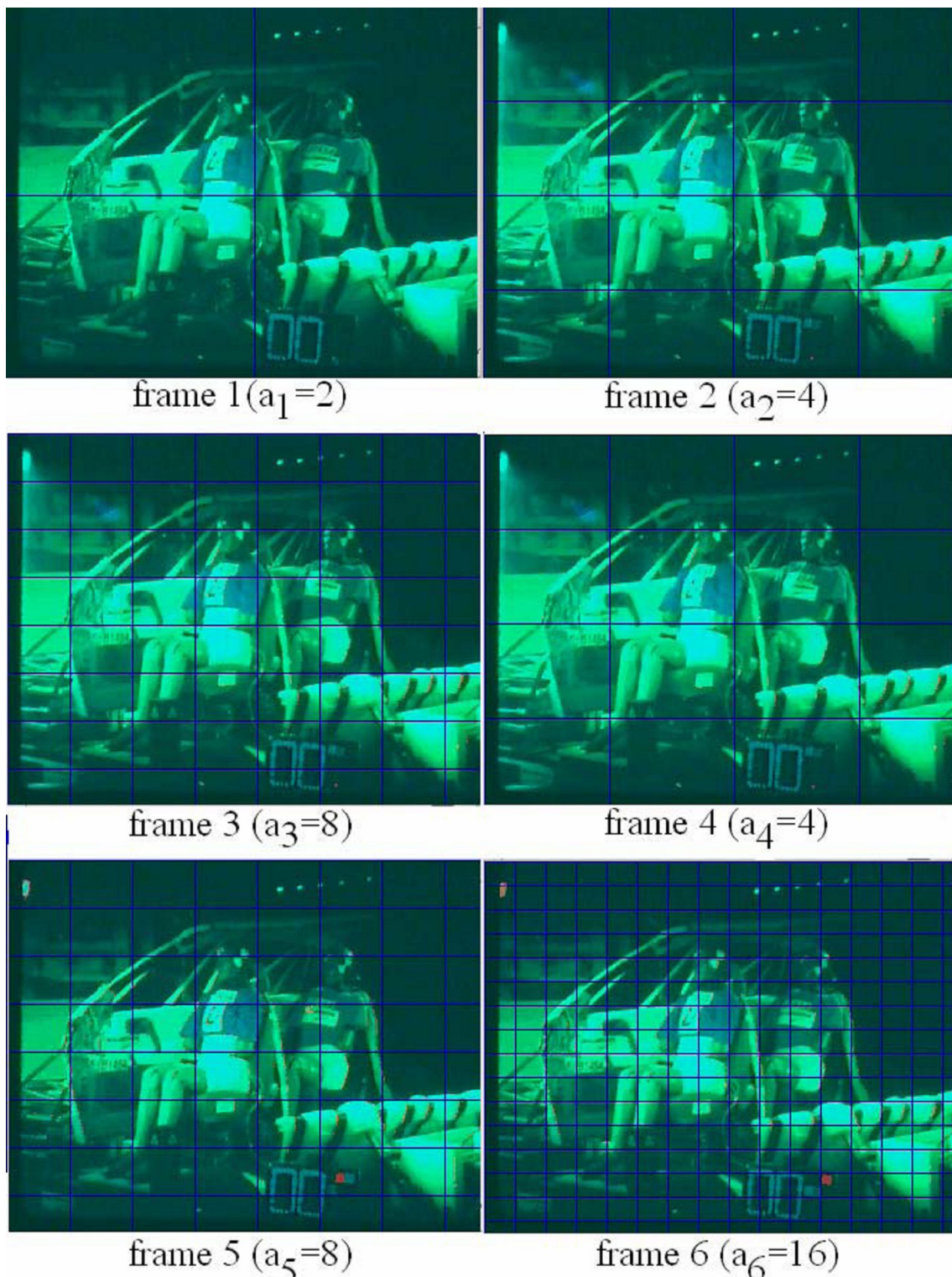


Fig. 3.7 Teste dos *frames* de 1 a 6.

Os testes realizados mostraram que o número total de pixels testados, com o algoritmo adaptativo, reduz a tendência crescente relativamente aos testes

realizados sem algoritmo adaptativo, como está ilustrado na fig. 3.8. De facto, a diferença obtida é de  $8,8 \times 10^6$  pixels quando comparados os testes realizados, com e sem algoritmo adaptativo (fig. 3.8). A relação entre o número de pixels testados com, e sem, o algoritmo adaptativo é 0,41721 (fig. 3.6).

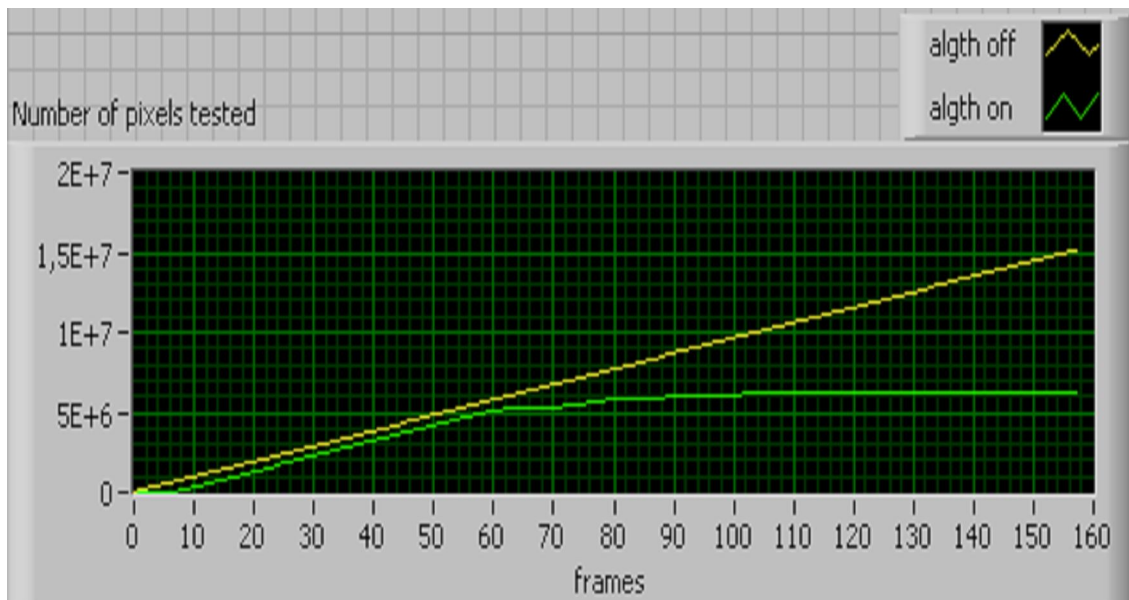


Fig. 3.8 Numero de pixels testados com e sem o algoritmo adaptativo.

Na figura 3.9, os resultados obtidos com o algoritmo adaptativo activo são apresentados. Esta figura apresenta o comportamento dinâmico do coeficiente de divisão da célula de  $a_i$ .

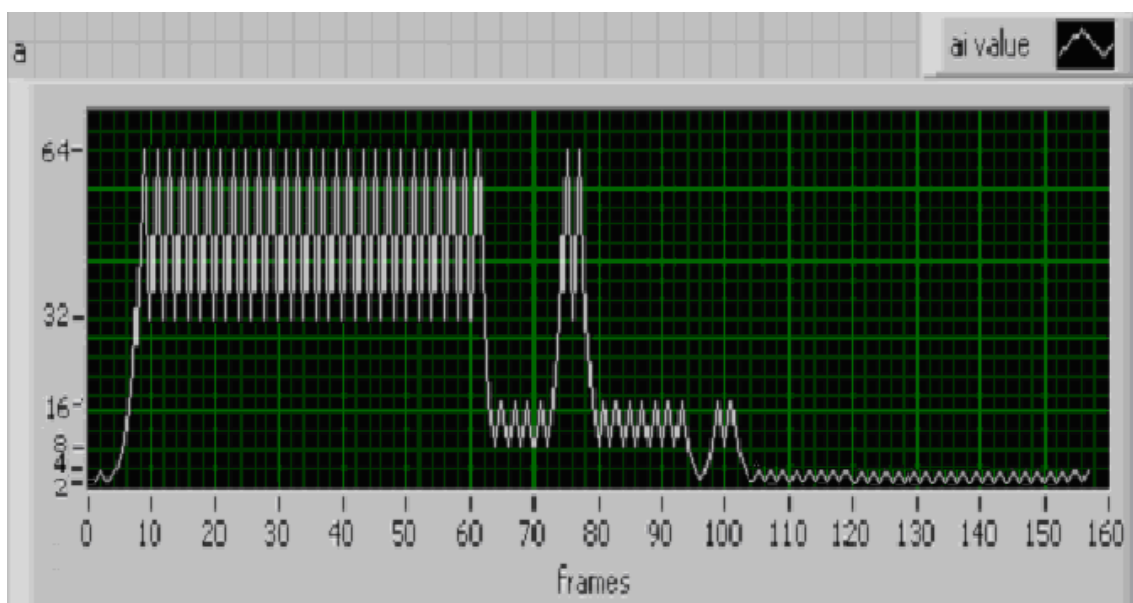


Fig. 3.9 Valor do factor  $a_i$ .

Analisando estes resultados pode-se concluir que:

- a) o valor  $a_i$ , a partir do *frame* 1 até ao *frame* 8, tem um crescimento de 2,4,2,4,8,16,32, até atingir o valor de 64;
- b) entre os *frames* 9 e 61 o valor oscila entre 32-64;
- c) entre os *frames* 62 e 74 o valor oscila entre 16-8;
- d) entre os *frames* 75 e 78 o valor volta a oscilar entre 32-64;
- e) entre os *frames* 79 e 94 varia volta a oscilar entre 8-16;
- f) o valor do coeficiente é 2 no *frames* 96;
- g) entre os *frames* 97 e 102, oscila entre 16-8;
- h) no *frame* 103 o valor é 4;
- e, finalmente,
- i) após o *frame* 107 o valor do coeficiente varia de 2 a 4 até ao fim.

Esta análise, com base na informação ilustrada nas figuras 3.8 e 3.9, caracteriza o comportamento dinâmico do algoritmo adaptativo proposto, ou seja, comprova-se que quando há um aumento de movimento na imagem, diminui a área da célula, mas aumenta o número de células de detecção e vice-versa, não comprometendo a capacidade de detecção e salvaguardando que não é atingido o número total de pixels da imagem a ser testado.

A Figura 3.10 mostra que o limiar dinâmico  $\tau_{pi}$  com capacidade de adaptação, evita a saturação do número de pixels testados. Desta forma, qualquer informação relativa às características de movimento nas imagens (*frames*) analisadas pode ser visualizada.

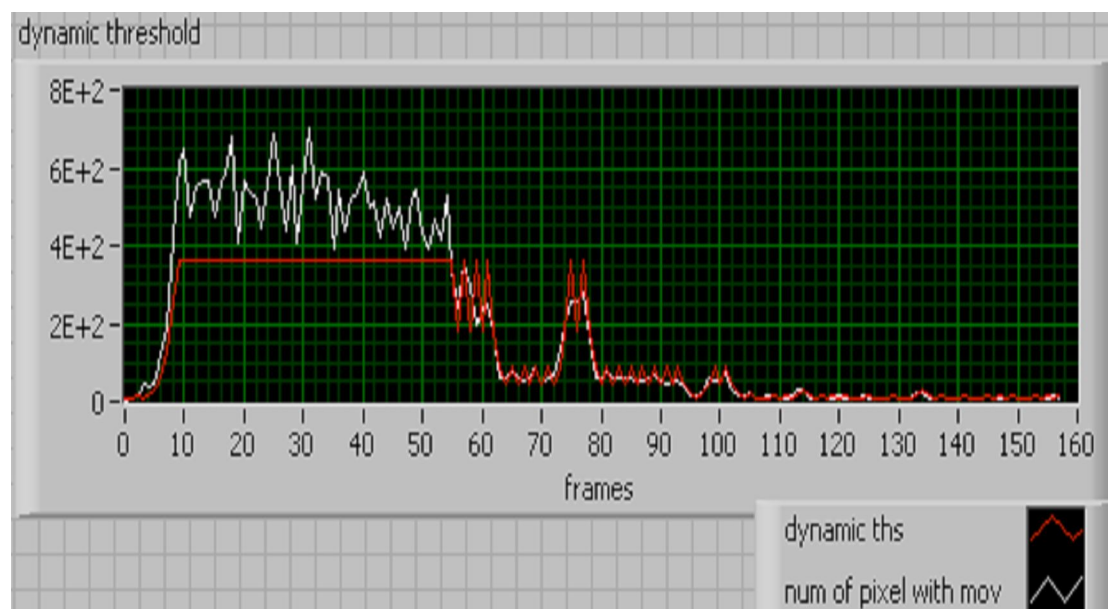


Fig. 3.10 Variação do limiar dinâmico (threshold)  $\tau_{pi}$ .

Para análise do algoritmo desenvolvido, teremos em conta os diversos tipos de resolução, em vídeo, apresentados na tabela 3.1.

Nome	Largura x Altura	Formato	Total de pixels
QQVGA	160 x 120	04:03	19.200
QVGA	320 x 240	04:03	76.800
VGA	640 x 480	04:03	307.200
VGA	720 x 480	04:03	345.600
SVGA	800 x 600	04:03	480.000
XGA	1024 x 768	04:03	786.432
WXGA-HD/720p	1280 x 720	16:09	921.600
WXGA	1366 x 768	16:09	1.049.088
WXGA+	1440 x 900	16:10 (8:5)	1.296.000
UXGA	1600 x 900	16:09	1.440.000
UXGA++	1680 x 1050	16:10	1.764.000
Full HD/1080p	1920 x 1080	16:09	2.073.600
WQHD	2560 x 1440	16:09	3.686.400
4K Ultra HD/2160p	3840 x 2160	16:09	8.294.400

Tabela 3.1: Resoluções de vídeo

### 3.4.2 Conclusões

Os testes realizados com o algoritmo de adaptação proposto neste capítulo, mostram que, relativamente aos resultados obtidos sem o algoritmo proposto, o número de pixels testados foi reduzida em cerca de 60 %, sem perda de informação relativa à detecção de movimento.

Para o estudo do caso apresentado neste capítulo (sequência de 157 *frames*), verifica-se uma redução de  $8,8 \times 10^6$  pixels testados. Na verdade, o algoritmo proposto apresenta uma maior eficiência em relação à detecção de movimento, o que é conseguido através da redução dos pixels testados e, ao mesmo tempo, ao aumento da probabilidade de identificar o movimento entre imagens (*frames*) consecutivas tendo em conta a capacidade adaptativa do mesmo.

Tendo em conta as várias resoluções de vídeo, conforme Tabela 3.1, é apresentada na Tabela 3.2 e na figura 3.11 a redução de dados obtida: Número total de pixels a processar vs. Número total de pixels, para cada resolução.

Nome (resolução)	Total de pixels	Redução de pixels	Total de pixels a processar
QQVGA	19.200	11.520	7.680
QVGA	76.800	46.080	30.720
VGA	307.200	184.320	122.880
VGA	345.600	207.360	138.240
SVGA	480.000	288.000	192.000
XGA	786.432	471.859	314.573
WXGA-HD/720p	921.600	552.960	368.640
WXGA	1.049.088	629.453	419.635
WXGA+	1.296.000	777.600	518.400
UXGA	1.440.000	864.000	576.000
UXGA++	1.764.000	1.058.400	705.600
Full HD/1080p	2.073.600	1.244.160	829.440
WQHD0	3.686.400	2.211.840	1.474.560
4K Ultra HD/2160p	8.294.400	4.976.640	3.317.760

Tabela 3.2 :Redução dos pixels nas várias resoluções de vídeo

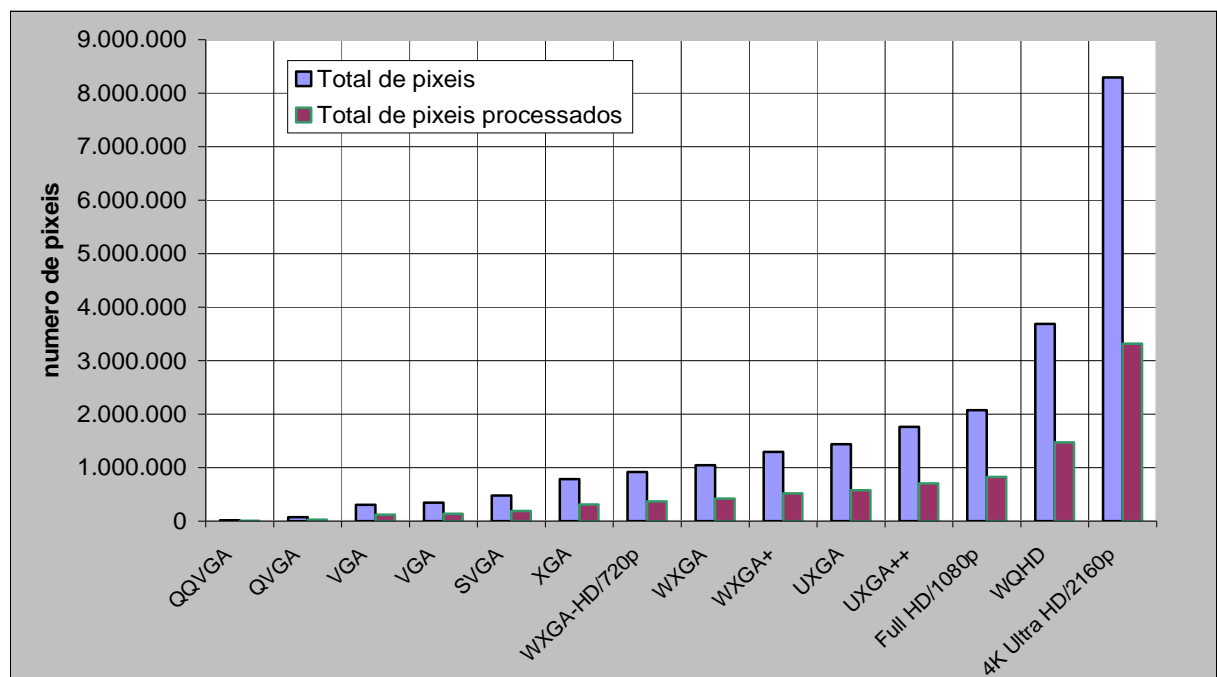


Fig. 3.11 Total de pixels a processar vs Total de pixels para cada resolução.

Trabalhos futuros poderão incidir sobre a aplicação do algoritmo actual, de forma independente, a cada célula, bem como com a ajuda de outras metodologias com capacidades de interpretação e previsão do movimento, no

contexto dos algoritmos genéticos e redes neurais. Metodologias que melhorariam possivelmente a capacidade de detecção de movimento, mas provavelmente não verificariam o critério de redução da carga computacional tendo em conta a complexidade do tipo de operações destes algoritmos.





# Capítulo 4

## Detecção de pele em Imagens a cores por Método adaptativo

*Neste capítulo é apresentada uma nova abordagem para detectar a pele em imagens a cores. O novo método utiliza uma metodologia adaptativa incorporada no algoritmo classificação/segmentação de pele que varia o brilho da imagem em cada um dos componentes do espaço de cor RGB para compensar as variações de iluminação e um novo método de detecção de face para determinar a localização da face na imagem, melhorando a detecção dos pixels de pele e reduzindo simultaneamente a carga computacional.*

## 4.1 Introdução

Os métodos que utilizam a relação entre os componentes do espaço de cor para construir um classificador de pele que compara em cada pixel o grau de semelhança entre os seus componentes de cor (Peer and Solina, 2003; KaKumanu et al, 2007), criou um algoritmo de decisão que identifica se o pixel corresponde a um pixel de pele: segmentação por cor. A sua principal vantagem é a simplicidade das regras de detecção da pele, que proporcionam o desenvolvimento de um rápido classificador (Vezhnevets et al, 2003; Peer and Solina, 2003; Albiol et al 2000).

Este tipo de abordagem tem bons resultados, se as condições do brilho da imagem não sofrem variações significativas. No entanto, se houver uma variação significativa da iluminação enquanto as imagens são adquiridas, erros significativos poderão acontecer na identificação dos pixels (Rein-lien and Abdel-Mottaleb, 2002). Esses erros ocorrem devido ao método quantificar a contribuição de cada uma das cores primárias (espaço de cor RGB) para cada pixel e é comum que o atributo de cor sofra variações em função do brilho da imagem

Neste capítulo um novo método é proposto para a identificação de pixels de pele. Este método funciona com base em dois algoritmos: um algoritmo adaptativo e um algoritmo classificador.

Esta nova metodologia permite tornar a identificação dos pixels de pele, menos sensível às mudanças de brilho da imagem. Na verdade, é capaz de detectar as variações dos componentes de cor no modelo RGB. Normalmente, o espaço de cor rgb (RGB normalizado) reduz as influências de imagem brilho (Vezhnevets et al, 2003). No entanto, o sistema não tem qualquer efeito positivo sobre a avaliação se a variação de brilho só afectar um componente de cor.

Os algoritmos adaptativos são tipicamente implementados em duas topologias: topologia de malha fechado e na topologia de malha aberta (Abrantes, 2000).

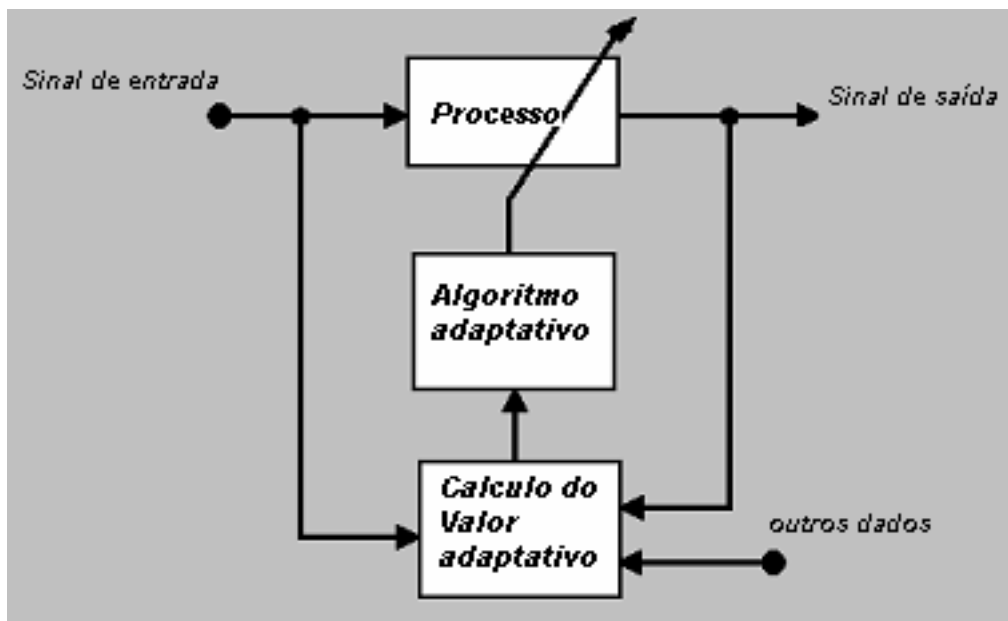


Fig. 4.1 Topologia de malha fechada.

Na malha fechada (Fig. 4.1) o processo de adaptação leva em conta o sinal de entrada, as alterações no sinal de saída e, por fim, outras fontes de dados relevantes (Grilo et al, 2007). Na malha aberta (Figura 4.2) a função de adaptação considera o sinal de entrada e se, eventualmente, as contribuições de outras fontes de dados relevantes, mas não considera o sinal de saída (variável controlada).

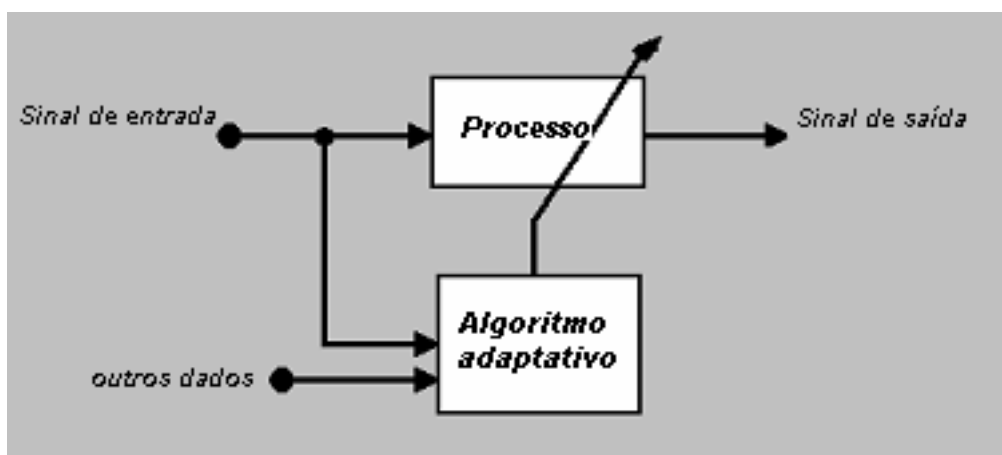


Fig. 4.2 Topologia de malha aberta.

A escolha de malha fechada vs. malha aberta de adaptação depende, principalmente, da disponibilidade de ambos os sinais, sinais de entrada e sinais indicadores de desempenho (Widrow and Stearns, 1985). Além disso, a carga computacional requerida é também de importância primordial pois esta

exigência define o período de amostragem mínimo para execução da aplicação.

Neste capítulo, a minimização da carga computacional desempenhava o principal papel na selecção da estratégia adaptativa porque esta aplicação destina-se a ser utilizado no campo de detecção e controlo de movimento. Assim, o método descrito usa a topologia de malha aberta tendo em conta a distribuição estatística de cada uma das componentes do espaço de cor (RGB) obtidas a partir do respectivo histograma.

## 4.2 Algoritmo de Processamento de Imagem

### 4.2.1 Algoritmo Base de detecção de pele

No algoritmo base de detecção de pixels de pele (classificador de pele), obtém-se as três matrizes correspondentes aos três componentes de cor do espaço RGB, para cada imagem. Em cada uma dessas matrizes, cada pixel da imagem é testado e a relação entre os três componentes de cor é encontrada.

Os parâmetros de  $C_R$ ,  $C_G$  e  $C_B$  são respectivamente os componentes vermelho, verde e azul do RGB. Os valores desses parâmetros determinam se o pixel é, ou não, um pixel de pele (segmentação). O algoritmo de pseudo - código é descrito como se segue (Gomez and Morales, 2002; Peer and Solina, 2003), ver Figura 4.3:

$$\begin{aligned} & \text{if } [ (C_R > 95) \& (C_G > 40) \& (C_B > 20) \& (C_R > C_G) \& \\ & (C_R > C_B) \& (|C_R - C_B| > 15) \& \\ & (Max(C_R, C_G, C_B) - Min(C_R, C_G, C_B) > 15) ] \\ & \text{then} \qquad \qquad \qquad \text{is a skin pixel} \\ & \text{else} \qquad \qquad \qquad \text{is not a skin pixel} \end{aligned} \tag{4.1}$$

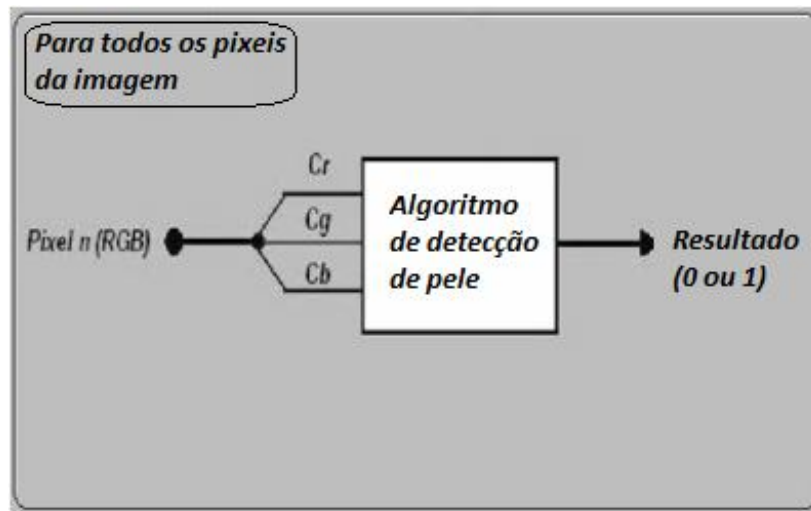


Fig. 4.3 Utilização do algoritmo Base de detecção de pixel de pele.

### 4.2.2 Algoritmo adaptativo

Para melhorar o desempenho do método anteriormente proposto para a identificação de pixels em imagens de cor de pele (Secção 4.2.1), foi desenvolvido um algoritmo adaptativo. Este algoritmo adaptativo foi desenvolvido para ser integrado no algoritmo base como uma melhoria no desempenho do método (ver Figura 4.4).

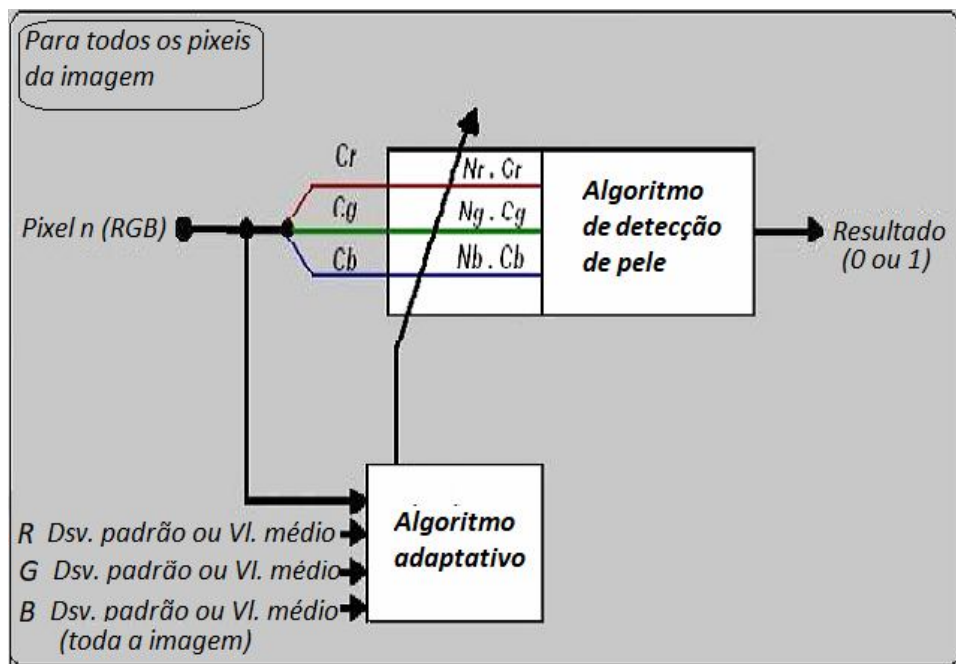


Fig. 4.4 Integração do algoritmo adaptativo no algoritmo base.

No algoritmo adaptativo, cada valor dos componentes do RGB base,  $C_r$ ,  $C_g$  e  $C_b$ , usados no algoritmo base são substituído pelo produto correspondente:  $N_r.C_r$ ,  $N_g.C_g$  e  $N_b.C_b$  onde os parâmetros  $N_r$ ,  $N_g$  e  $N_b$  dependem das variações de luminosidade. O método utilizado para calcular os factores de correcção  $N_r$ ,  $N_g$  e  $N_b$  baseia-se no sistema de RGB 24 bits. Para cada uma das três cores, o respectivo histograma é obtido na forma de que é mostrada na Figura 4.5. O histograma dá-nos o número de ocorrências em cada uma das três componentes de cor, para cada um dos 256 níveis (8 bits).

Para cada histograma, o valor médio,  $MnVI$  (*Mean Value*), é calculada pela equação (4.2).

$$MnVI = \frac{1}{n_t} \cdot \sum_{i=0}^{255} n_i \cdot i \quad (4.2)$$

Os parâmetros  $n_t$  e  $n_i$  são o número total de pixels e o número total de pixels que pertencem ao nível de  $i$  (ou classe), respectivamente.

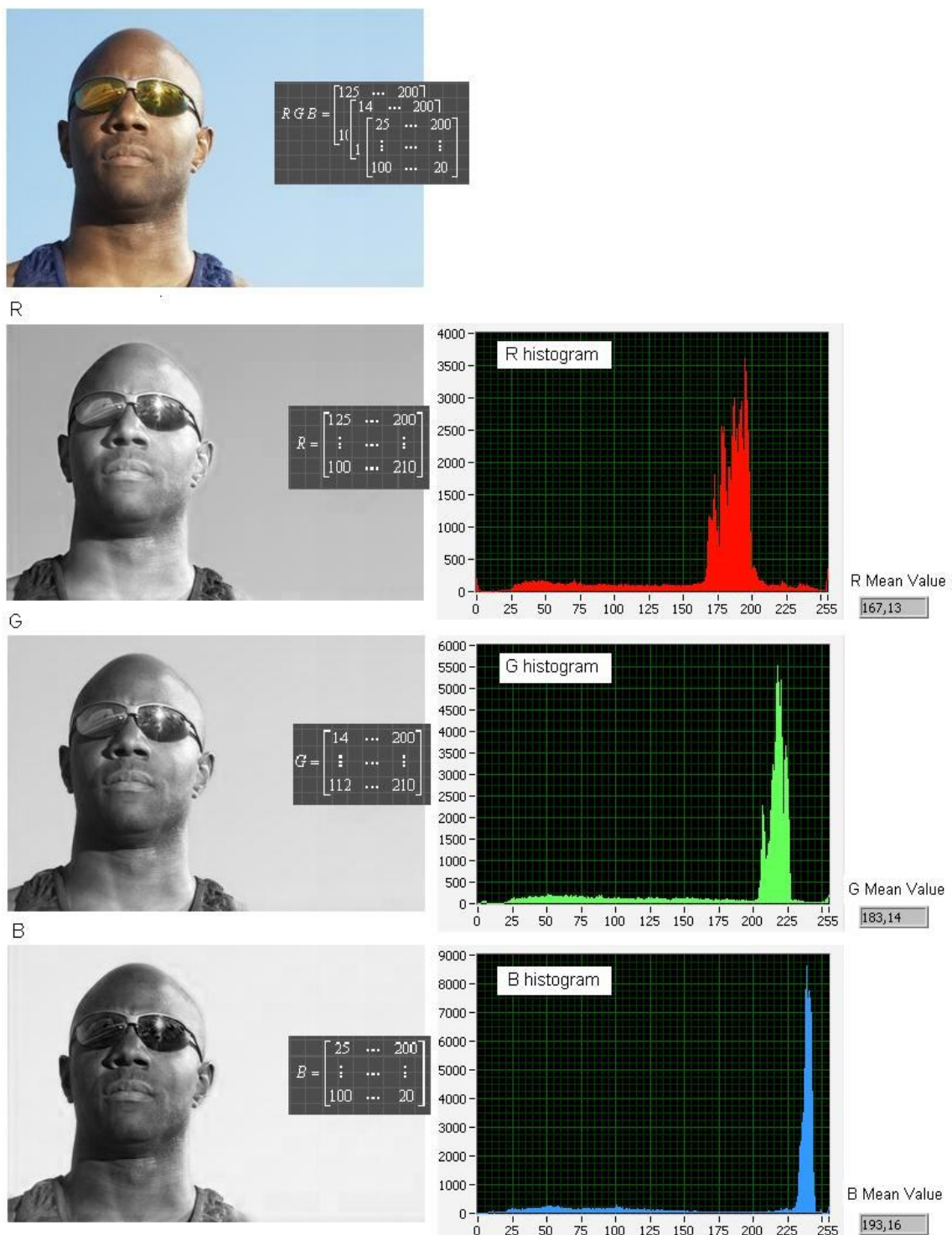


Fig. 4.5 Cálculo do valor médio em cada componente: de MnVIR, MnVIG e MnVIB.

O valor de cada um dos três factores (Nr, Ng e Nb) é calculado para o respectivo componente de cor considerando o valor médio correspondente (MnVI):



$$\begin{aligned} \text{if } (MnVI_R < 140) \quad & \text{then} \\ & N_R = 2 - 0,1 \cdot \alpha \cdot MnVI_R \\ \\ \text{if } (MnVI_G < 140) \quad & \text{then} \\ & N_G = 2 - 0,1 \cdot \alpha \cdot MnVI_G \\ \\ \text{if } (MnVI_B < 140) \quad & \text{then} \\ & N_B = 2 - 0,1 \cdot \alpha \cdot MnVI_B \\ \\ \text{if } (140 < MnVI_R < 200) \quad & \text{then} \\ & N_R = 1,5 - 0,05 \cdot \alpha \cdot MnVI_R \\ \\ \text{if } (140 < MnVI_G < 200) \quad & \text{then} \\ & N_G = 1,5 - 0,05 \cdot \alpha \cdot MnVI_G \\ \\ \text{if } (140 < MnVI_B < 200) \quad & \text{then} \\ & N_B = 1,5 - 0,05 \cdot \alpha \cdot MnVI_B \\ \\ \text{if } (MnVI_R > 200) \quad & \text{then} \\ & N_R = 1,25 - 0,025 \cdot \alpha \cdot MnVI_R \\ \\ \text{if } (MnVI_G > 200) \quad & \text{then} \\ & N_G = 1,25 - 0,025 \cdot \alpha \cdot MnVI_G \\ \\ \text{if } (MnVI_B > 200) \quad & \text{then} \\ & N_B = 1,25 - 0,025 \cdot \alpha \cdot MnVI_B \end{aligned} \tag{4.3}$$

onde  $\alpha$  é o factor de escala adaptativa. O valor de  $\alpha$  foi obtido experimentalmente, foi ajustado para o conjunto de imagens usado no desenvolvimento do algoritmo base de detecção da pele (ver Secção 4.2.1). Este valor foi dimensionado para evitar variações extremas que poderiam mudar significativamente a relação dos três componentes RGB e, por conseguinte, comprometer o algoritmo base de detecção de pele. O algoritmo desenvolvido implementa os três níveis adaptativos definidos, um para cada um dos três componentes RGB, dependendo da amplitude dos valores médios dos três componentes RGB (MnVl<sub>r</sub>, MnVl<sub>g</sub> e MnVl<sub>b</sub>) (Ver Figura 4.5).

Três intervalos diferentes para valores médios foram definidos: MnVI <140; 140 <MnVI <200 e MnVI > 200.

Dependendo do intervalo de trabalho, o que corresponde cálculo dos factores de adaptação é realizada ( $N_r$ ,  $N_g$ ,  $N_b$ ).

O algoritmo foi desenvolvido para os valores do factor  $\alpha$  no intervalo de 0,055 a 0,085.

Nos testes realizados, o valor  $\alpha = 0,060$  reproduziu os melhores resultados experimentais (Ver Figura 4.6).

A estratégia desenvolvida demonstrou a sua capacidade de compensar as alterações de brilho da imagem para cada um dos componentes de cor, tal como será demonstrado em Secção 4.4.

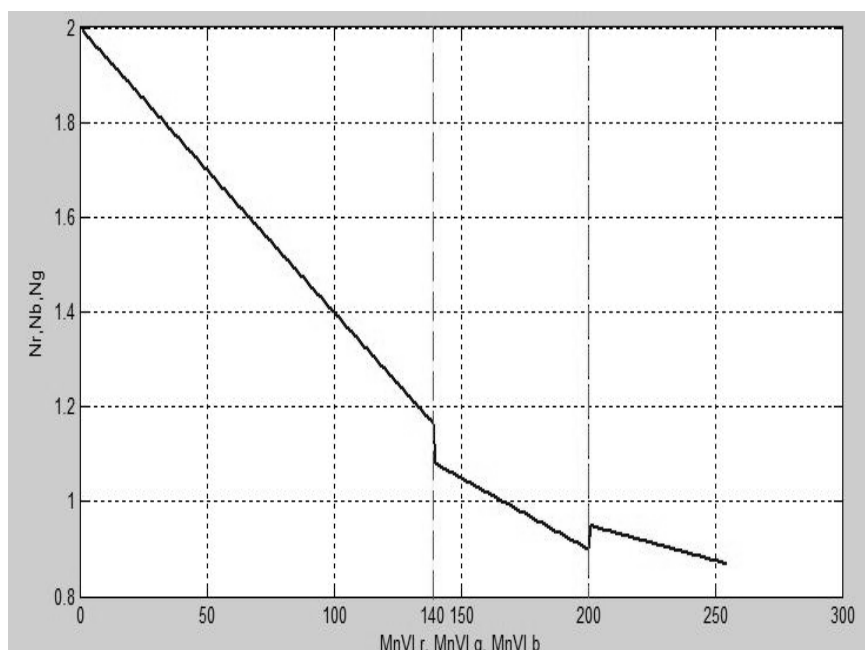


Fig. 4.6 Os 3 níveis dos factores adaptativos  $N_r$ ,  $N_g$ ,  $N_b$  com  $\alpha=0,06$ .

### 4.3 Implementação do sistema

Para validar a metodologia proposta desenvolveu-se um sistema que foi implementado com a ferramenta de desenvolvimento LabVIEW (National Instruments) e é executado no sistema operativo Windows 7. A aplicação de *software* de processamento permite receber os dados de três fontes diferentes:

- as imagens importadas (BMP ou JPEG);
- vídeos importados (formato AVI);
- imagens (vídeo) recebidas de uma *webcam* comum.

Na aplicação desenvolvida, as imagens obtidas são convertidas para o formato IMAQ a partir do LabVIEW. O formato IMAQ também pode ser usado com as funções de visão da biblioteca LabVIEW. No processamento de imagem

desenvolvido no sistema, o formato IMAQ foi convertido numa matriz de 24 bits RGB (8 bits para cada componente de cor).

Para validar a metodologia proposta, foi desenvolvida a aplicação cuja interface de utilizador contém duas áreas principais tal como é ilustrado na Figura 4.7 (áreas 1 e 2).

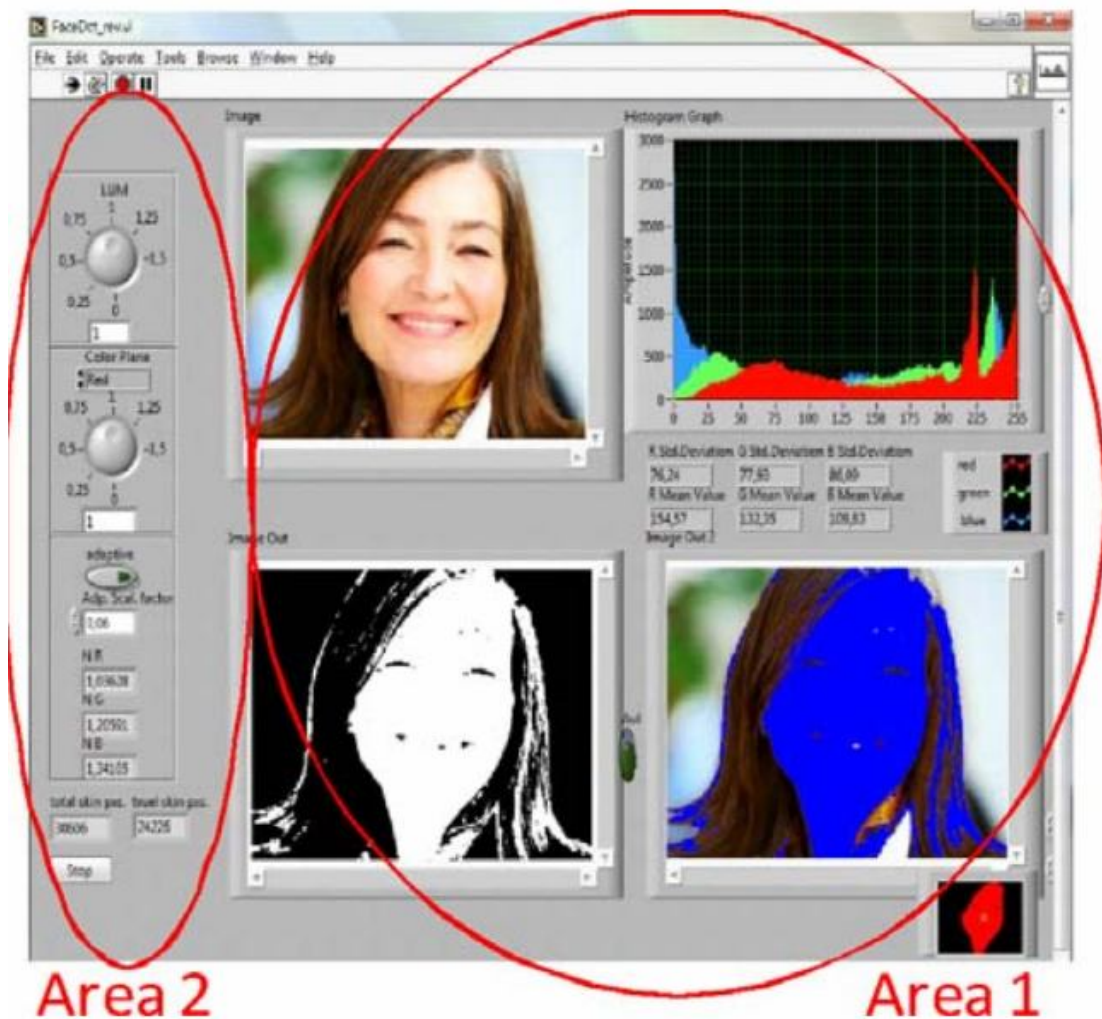


Fig. 4.7 Interface do utilizador.

Na área do lado direito da interface do utilizador (Área 1), existem cinco imagens, janelas (Figura 4.8). A primeira janela apresenta o vídeo original a cores ou a imagem recolhidas em tempo real ou carregado a partir de um arquivo. A segunda janela identifica todos os pontos (pixeis) na imagem com classificação positiva de pele, de acordo com o método desenvolvido, em preto / branco. A terceira janela ilustra a imagem com os pixeis de pele identificados

em cor azul sobre a imagem original. Este método permite avaliar a robustez do algoritmo desenvolvido com, ou sem, o método adaptativo.

A quarta janela mostra os três histogramas, bem como os três valores médios e os valores do desvio padrão correspondente de cada um dos componentes RGB relativamente à imagem capturada inicial.

Finalmente, a quinta, a pequena janela no canto inferior direito, mostra a área calculada de pixels de pele verdadeiros após a aplicação de ambos, a validação pele e a função de detecção de rosto, com a identificação do Centróide ou centro geométrico correspondente (pequeno círculo). A identificação do Centróide é de grande importância para aplicações relacionadas com controlo de movimento.



Fig. 4.8 Interface do utilizador (Área 1).

Na área do lado esquerdo da interface do utilizador (Área 2) -Figura 4.9 - existem dois botões de controlo: LUM e COLOUR PLANE. O botão LUM permite que o utilizador varie o brilho da imagem inicial (para simular a iluminação / variação de iluminação) e o botão COLOUR PLANE permite alterar o valor de cada um dos três componentes RGB, pré-seleccionado pelo utilizador. Estas funcionalidades foram implementadas para melhor testar o desempenho do algoritmo adaptativo.

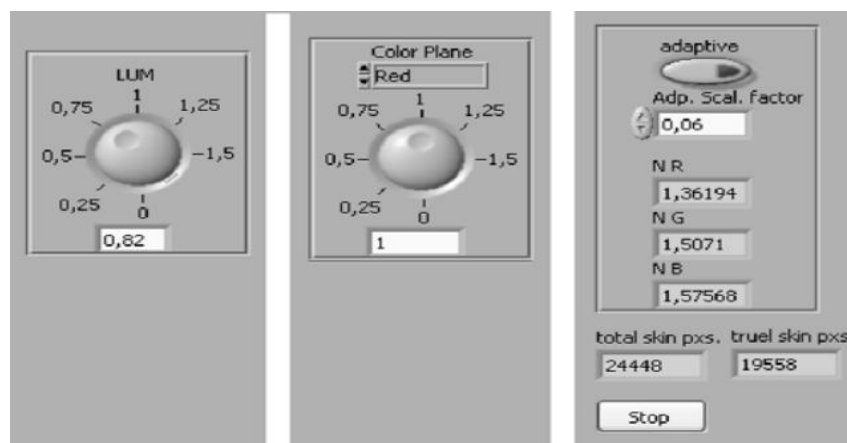


Fig. 4.9 Interface do utilizador (Área 2).

A funcionalidade implementada para modificar o brilho da imagem é ilustrada na Figura 4.10.

### 4.3.1 Funcionalidade de variação do brilho da imagem

O processo de variação de brilho é iniciado com a conversão da imagem a partir do espaço de cor de RGB para o espaço HSL (Jones e Rehg , 2002). Obtendo o componente L do sistema de cor, a matriz correspondente é isolada e multiplicada por um valor definido pelo utilizador (nível de brilho), seleccionado através do botão LUM na aplicação (Figura 4.9). A nova matriz calculada para a componente G é então reconvertida a partir do espaço de cor de HSL para o espaço RGB original.

A implementação correspondente na plataforma LabVIEW (National Instruments, 2011) é mostrada na Figura 4.11.

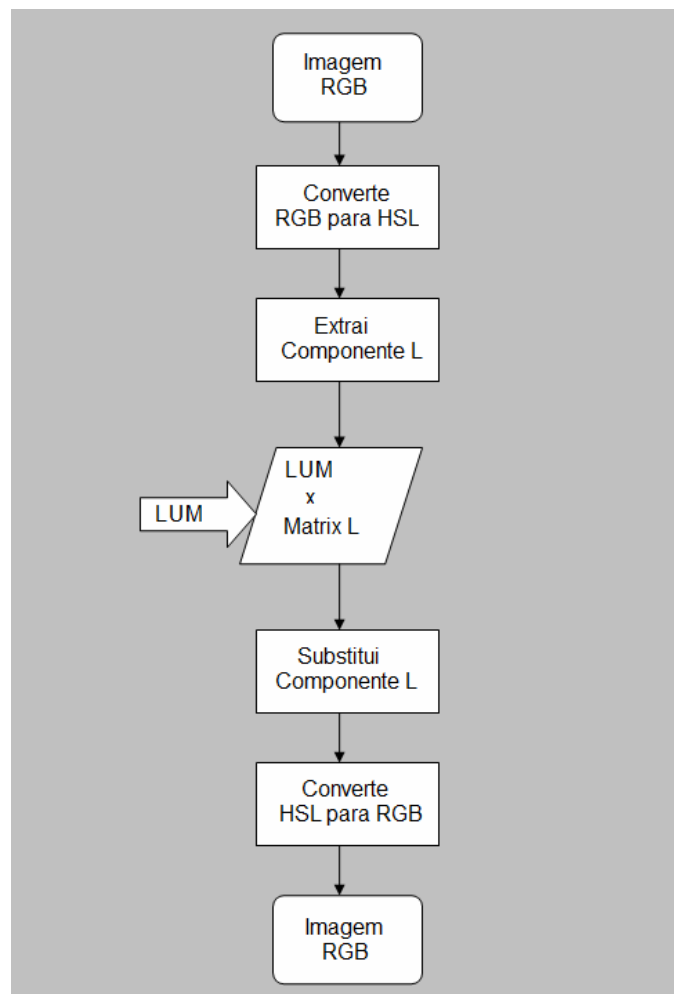


Fig .4.10 O fluxograma da funcionalidade de variação do brilho da imagem.

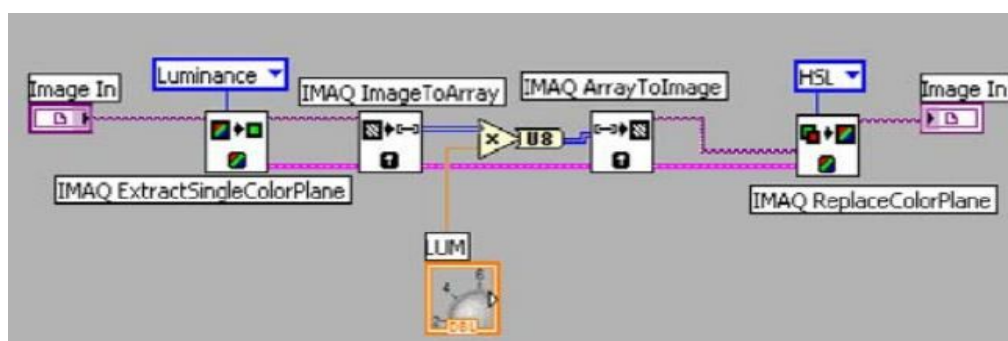


Fig. 4.11 A implementação da funcionalidade de variação do brilho da imagem (desenvl. em LabVIEW).

Na interface do utilizador - Área 2 - (Figura 4.9), também existem os botões : ADAPTIVE, que permite ao utilizador activar /desactivar o algoritmo adaptativo e o botão ADP. SCL. FACTOR que permite ao utilizador alterar o factor adaptativo  $\alpha$  (de escala). A interface desenvolvida também mostra os valores calculados dos parâmetros NR, NG e NB (expressões 4.3), bem como o

número total de pixels de pele identificados e o número total de pixels (validados) verdadeiros-positivos de pele.

### 4.3.2 Algoritmo da funcionalidade de validação de pixel de pele para detecção de face

A funcionalidade de validação de pixel de pele e detecção de face utiliza um conjunto de funções básicas de processamento de imagem, funções que são utilizadas para executar o algoritmo desenvolvido. O algoritmo desenvolvido toma a imagem composta por todos os pixels inicialmente identificados como pixels de pele e processa os dados através de várias funções para validar os pixels, inicialmente identificados como pixels de pele. O processo de validação de pixels verdadeiros de pele implementa o algoritmo mostrado na Figura 4.12. A explicação detalhada da metodologia desenvolvida segue, juntamente com a visualização dos resultados produzidos para uma determinada imagem (Figura 4.13).

1. Em primeiro lugar, é executada uma transformação morfológica primária com o auxílio da função de erosão, que, basicamente, recebe a informação a partir do algoritmo de detecção de pele e elimina os pixels isolados de não-pele (ver resultados na Figura 4.13 - Passo 1).
2. Em segundo lugar, a imagem obtida é processada com a função dilatação para preencher os buracos dentro das áreas já definidas como pele. Ambas as funções erosão e dilatação utilizam como elemento estruturante a matriz  $5 \times 5$  (ver os resultados em Figura 4.13 - Passo 2).
3. Em terceiro, a função remover partículas abaixo do valor de referência (*Remove Particle*) elimina as regiões mais pequenas que não estão contidos dentro da região principal definida como pele (ver resultados na Figura 4.13 - passo 3).
4. Em seguida, a função preencher área convexa (*Convex Hul*) elimina os buracos existentes no interior da área da pele considerada (ver os resultados em Figura 4.13 - Passo 4).
5. Em seguida, a função de análise de partículas (*Particle Analysis Report*) determina a maior área independente (partículas) na imagem. A função de remover partículas abaixo do valor de referência (*Remove Filter*) elimina todas as outras áreas pequenas (partículas) abaixo do valor maior área independente obtido anteriormente (ver os resultados na Figura 4.13 - passo 5).
6. Em seguida, a função lógica E (*and*) compara cada um dos pixels da imagem inicial com os pixels de a etapa anterior 5 para validar os verdadeiros pixels de pele (Ver Figura 4.13 - Passo 6). Esta operação cria uma imagem constituída apenas pelos verdadeiros pixels de pele, o qual pode ser visualizado na interface do utilizador (Ver Figura 4.13 - a etapa 7).
7. Finalmente, a função Centróide (NI IMAQ lib.) calcula o centro de a área definida a partir da etapa anterior. Esta função é de grande importância para aplicações relacionadas com detecção de movimento / controlo.

A implementação do algoritmo explicado anteriormente foi realizada na plataforma LabVIEW (National Instruments, 2011). A aplicação desenvolvida é mostrada na Figura 4.14.

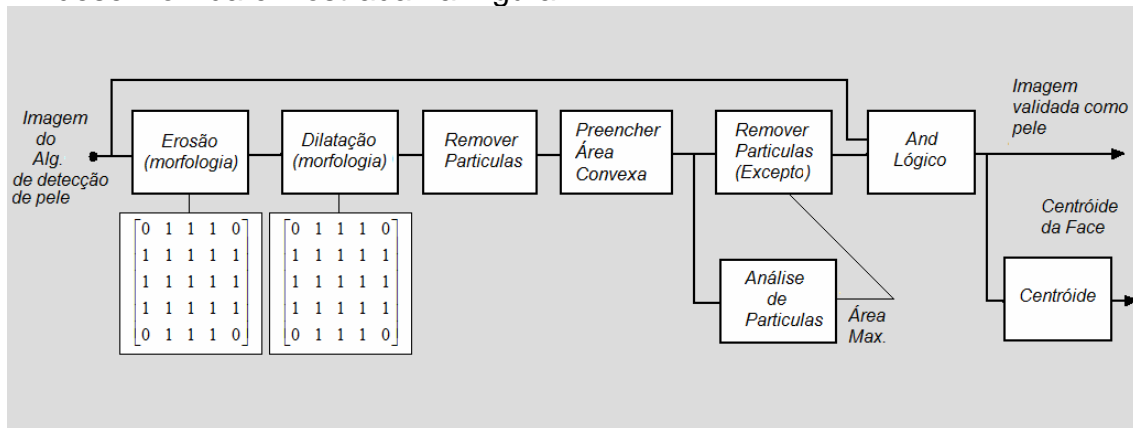


Fig. 4.12 Funcionalidade de validação de pixel de pele e detecção de face.

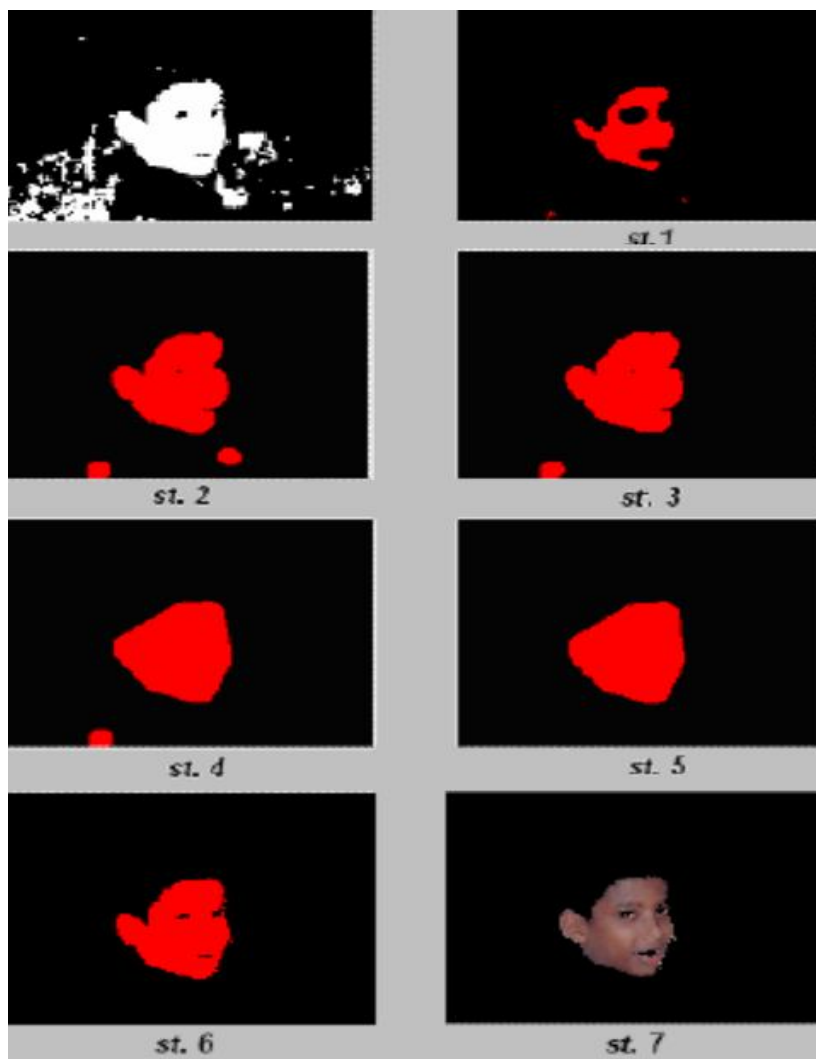


Fig. 4.13 Validação de pixel verdadeiro de pele (passos 1 a 7).



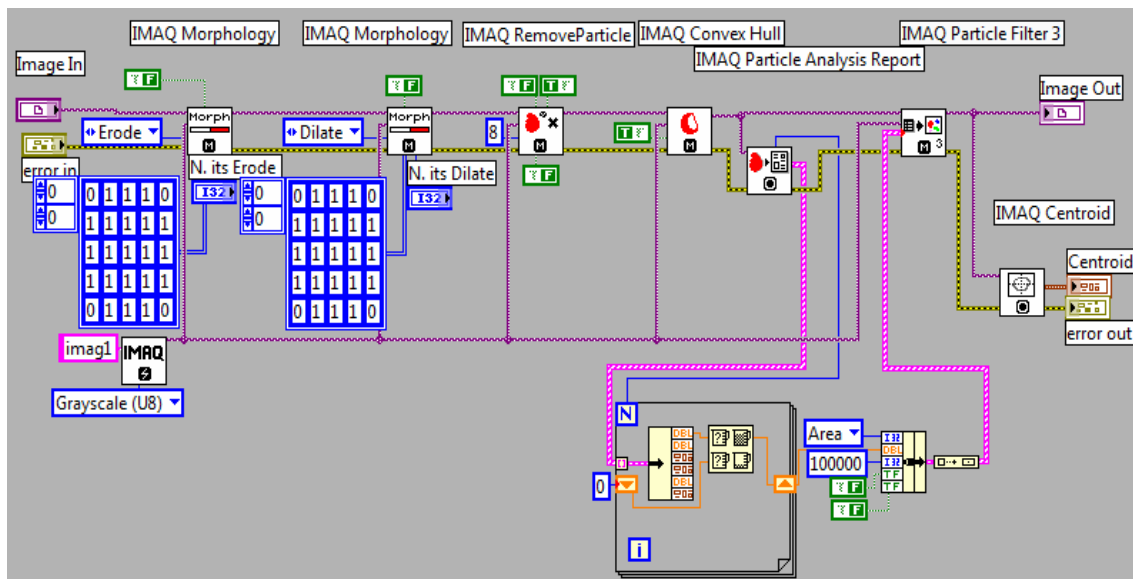


Fig. 4.14 Funcionalidade de validação de pixel de pele e detecção de face (implementado em LabVIEW).

#### 4.4. Testes realizados

Nos testes realizados utilizaram-se dois conjuntos diferentes de imagens:

- imagens capturadas em condições laboratoriais;
- imagens recolhidas a partir de bases de dados na Internet.

O conjunto de imagens é constituído por faces de homens, mulheres e crianças, de vários grupos étnicos, com diferentes origens e diferentes condições de iluminação (LFW Database, <http://vis-www.cs.umass.edu/lfw/>). O principal objectivo dos testes realizados foi o de avaliar o desempenho do novo algoritmo adaptativo desenvolvido. Portanto, em todos os seguintes exemplos ilustrativos, os resultados comparativos com, e sem, o algoritmo adaptativo (Adp on e Adp off) são mostrados. Os testes experimentais permitem avaliar os resultados em termos de dois parâmetros: a qualidade obtida (com base na detecção da face) e a eficiência do peso computacional (com base na avaliação do pixel da pele, com e sem algoritmo, Adp on e Adp off).

É visto que em todos os testes realizados, o algoritmo adaptativo melhora significativamente o número de pixels de pele identificados. Este número é geralmente maior do que o número de pixels de pele identificados sem algoritmo adaptativo com a razão.

O desempenho do algoritmo adaptativo em termos da qualidade dos resultados produzidos é ilustrado nas Figuras 4.15-4.26. Este conjunto de resultados

permite a visualização os diferentes números pixeis de identificado de pele com e sem o algoritmo adaptativo, em termos da área e de face detectada.

Nos testes realizados, foi tido em conta, diferentes condições de iluminação para testar a robustez do algoritmo adaptativo desenvolvido.

Na Figura 4.15, é mostrada uma análise comparativa típica do desempenho do algoritmo adaptativo. Os resultados obtidos sem o algoritmo adaptativo são mostrados na coluna da esquerda e os resultados correspondentes obtidos com o algoritmo adaptativo são mostrados na coluna da direita.

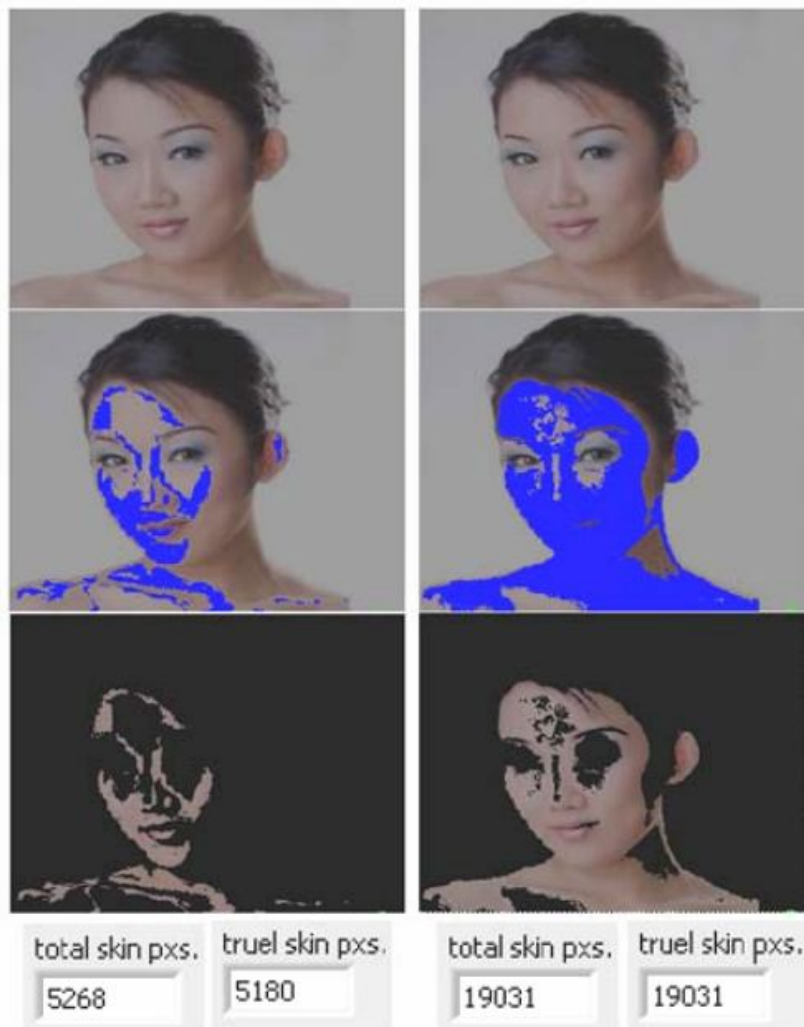


Fig. 4.15 Detecção de pele sem (esquerda) e com (direita) algoritmo adaptativo, com Lum = 0.45, Pixeis de pele-detectados (*total skin pxs*) e **Pixeis de pele-verdadeiros-positivos-da-face** (*true skin pixel*), para uma imagem com o numero de **Pixeis total-de pele-da-face** de 19.800.

A análise de ambas as colunas da figura 4.15 (direita e esquerda), mostra forma clara que o algoritmo adaptativo traz melhorias para o resultados obtidos. De facto, em ambos os testes realizados com Lum = 0,45, podemos verificar

que o número de pixels de pele detectados e validados verdadeiros-positivos-da-face tem um aumento significativo quando o algoritmo adaptativo é activado (Figura 4.15 - direita). Neste caso, tendo em conta que o numero de pixels total-de-pele-da-face da figura são 19.800, a razão de pixels de pele-verdadeiros-positivos-da-face pelo numero de pixels total-de-pele-da-face, sem e com algoritmo será respectivamente: *Adp off*,  $5180 / 19800 = 0,26$  (26 %); *Adp on*,  $19031 / 19800 = 0,96$  (96 %).

A Figura 4.16 mostra os dados relativos às imagens testadas e mostradas na Figura 4.15. A Figura 4.16 apresenta os parâmetros numéricos que caracterizam o algoritmo adaptativo: os valores médios e os factores adaptativos NR, NG e NB.

NR	R Mean Value
1,46693	88,85
NG	G Mean Value
1,50637	82,27
NB	B Mean Value
1,50982	81,70

Fig. 4.16 Valores médios de R, G e B; valores NR, NG e NB.

O seguinte conjunto de testes (Figuras 4.17-4.19) está relacionado com a qualidade dos resultados produzidos pelo algoritmo adaptativo, mas agora num contexto laboratorial e com as variações de iluminação associadas à captação da imagem. Os seguintes resultados referem-se ao processamento de imagens captadas por uma *webcam* comum, com fundos complexos e três diferentes condições de iluminação. Na Figura 4.17, a principal fonte de iluminação é uma lâmpada fluorescente, no tecto da sala. Na Figura 4.18, a fonte de iluminação principal é uma lâmpada de halogéneo, de fundo, e finalmente, na Figura 4.19, a única iluminação presente resulta da luminosidade do próprio monitor do computador.

A linha superior das figuras diz respeito ao processamento sem o algoritmo adaptativo e as figuras na linha em baixo mostram os resultados comparativos com o algoritmo adaptativo ligado.

A análise das Figuras 4.17-4.19 mostra a robustez do novo algoritmo adaptativo desenvolvido em relação às condições de iluminação extremas. O desempenho do algoritmo adaptativo mostra sempre melhores resultados na detecção do rosto quando comparada com os resultados fornecidos sem o algoritmo adaptativo. O melhor desempenho é particularmente evidente quando as condições de iluminação são mais fracas.

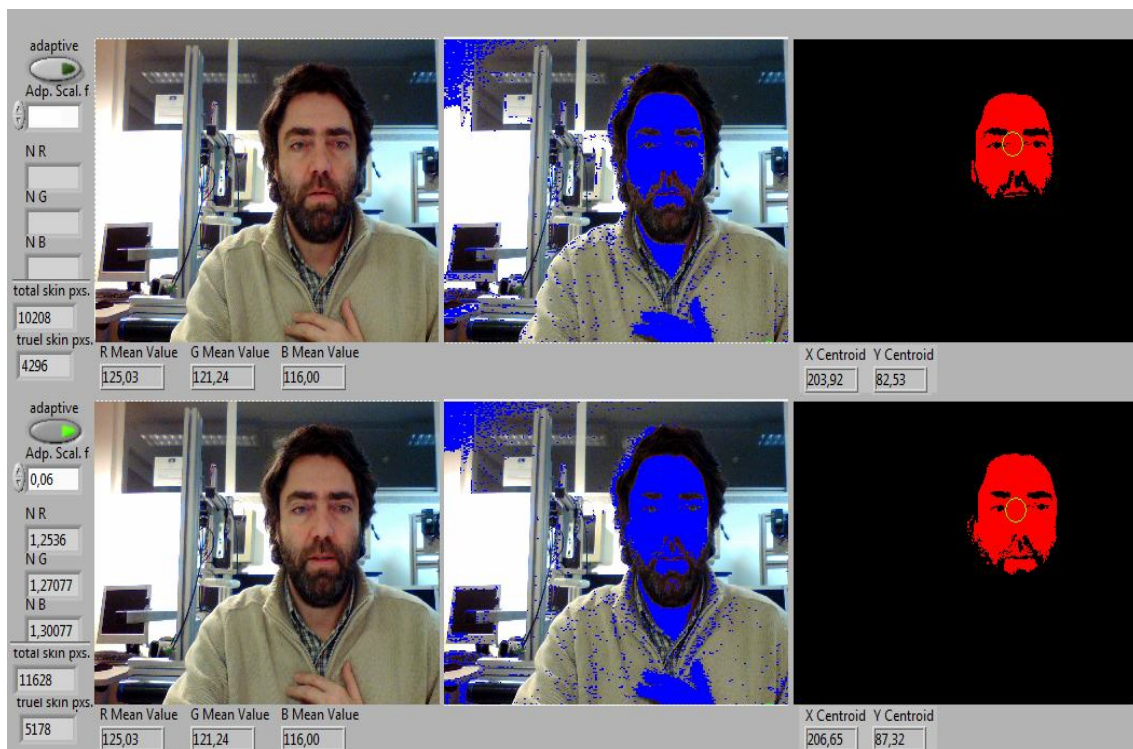


Fig. 4.17 Detecção de pele e detecção de face sem (cima) e com (baixo) algoritmo adaptativo, Total Pxs pele (Adp on/Adp off) = 114% e Total Pxs pele-verdadeiros-post-face (Adp on/Adp off) = 120%. Fonte de iluminação: lâmpada fluorescente no tecto.

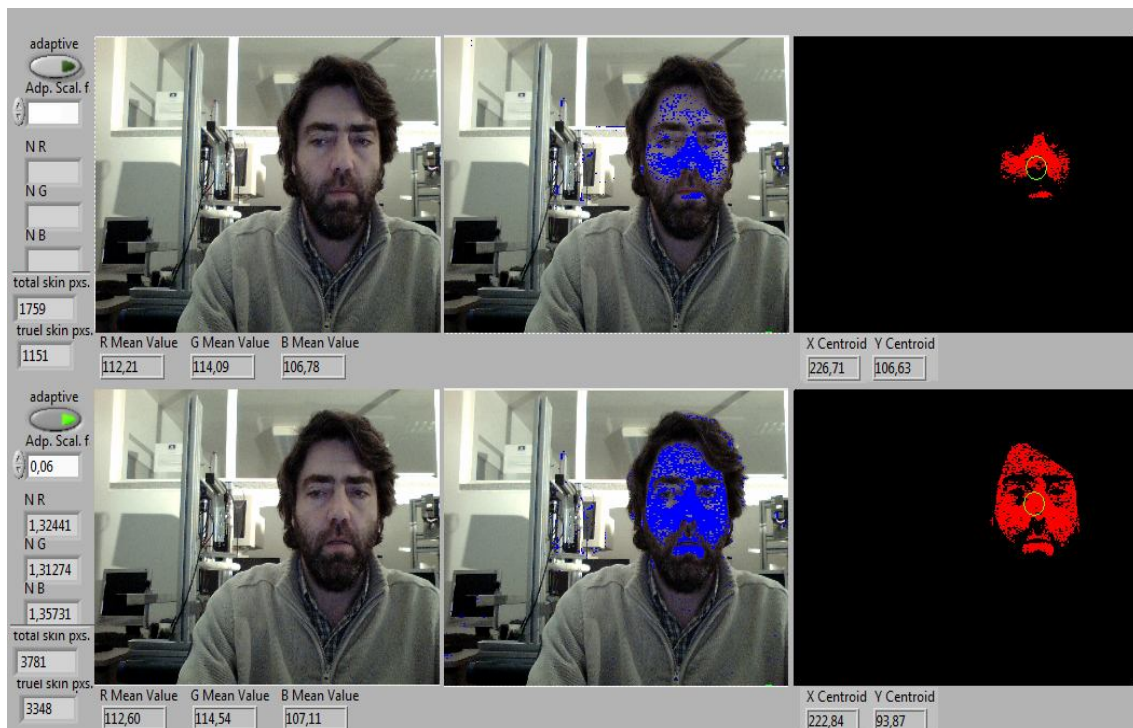


Fig. 4.18 Detecção de pele e detecção de face sem (cima) e com (baixo) algoritmo adaptativo, Total Pxs pele (Adp on/Adp off) = 215% e Total Pxs pele-verdadeiros-post-face (Adp on/Adp off) = 291%. Fonte de iluminação: lâmpada de halogéneo de trás.

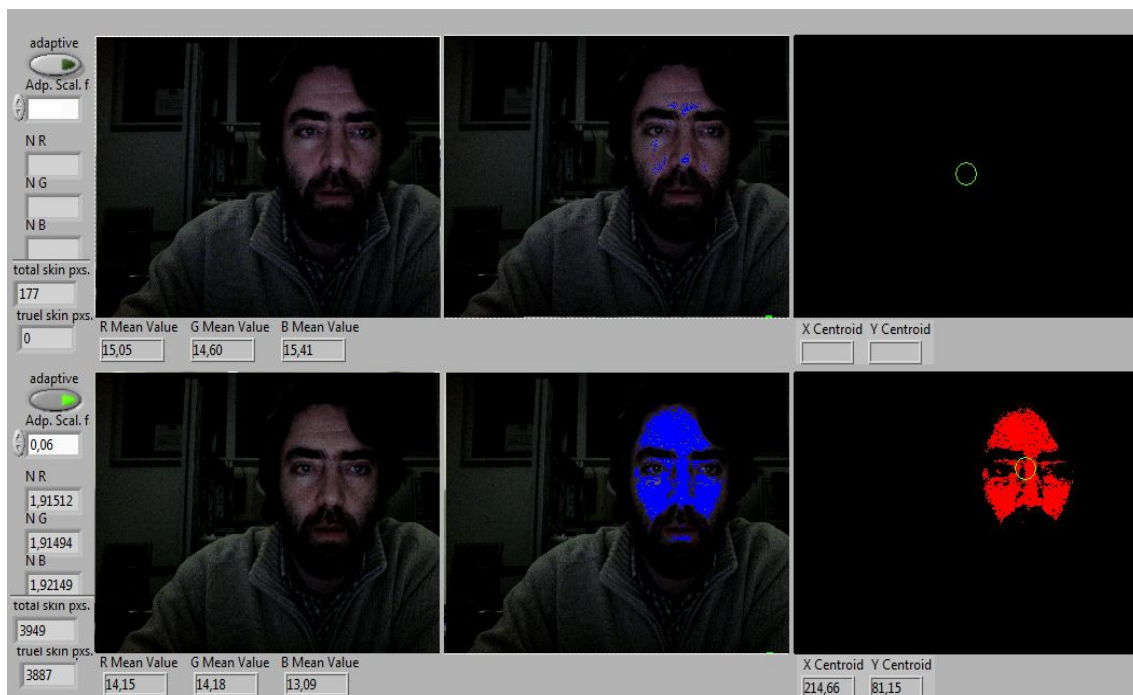


Fig. 4.19 Detecção de pele e detecção de face sem (cima) e com (baixo) algoritmo adaptativo, Total Pxs pele (Adp on/Adp off) = 2230% e Total Pxs pele verdadeiros-post-face (Adp on/Adp off) = n.c.. Fonte de iluminação: iluminação directa da luz do monitor do computador.

Nas Figuras 4.20-4.25, são mostrados os resultados dos testes realizados no conjunto do banco de imagens numa análise quantitativa da robustez do algoritmo adaptativo em relação à variação de brilho para a mesma imagem.

Para cada imagem, três níveis de brilho são utilizados no teste: 1,1, 0,75 e 0,35 (Figuras 4.20-4.22). E também uma variação independente em cada um dos componentes RGB, de 1,1 (figuras 4.23) e por fim a variação 0,5 só na componente R (figuras 4.24-4.25). Foi processado um conjunto de 10 imagens diferentes, o que dá um total de 80 testes, de acordo com as combinações acima descrito ( $3 \times 10 + 3 \times 10 + 2 \times 10$ ).

Nos teste em imagens de bases de dados, os resultados são apresentados pela razão de pixeis de **pele-verdadeiros-positivos-da-face** por pixeis **total-pele-da-face**, sem e com algoritmo, respectivamente Adp off e Adp on. Avaliando assim nos dois caso o nível de sucesso na detecção de todos os pixeis da face que são pele na imagem

O numero total-pele-da-face para cada imagem de teste foi obtido utilizando uma aplicação desenvolvida no ambiente LabView, na qual em cada imagem são definidas manualmente as várias áreas da face que são pele, retirado as várias áreas que não o são, como por exemplo os olhos. Calculando a aplicação, posteriormente, o número total de pixeis.

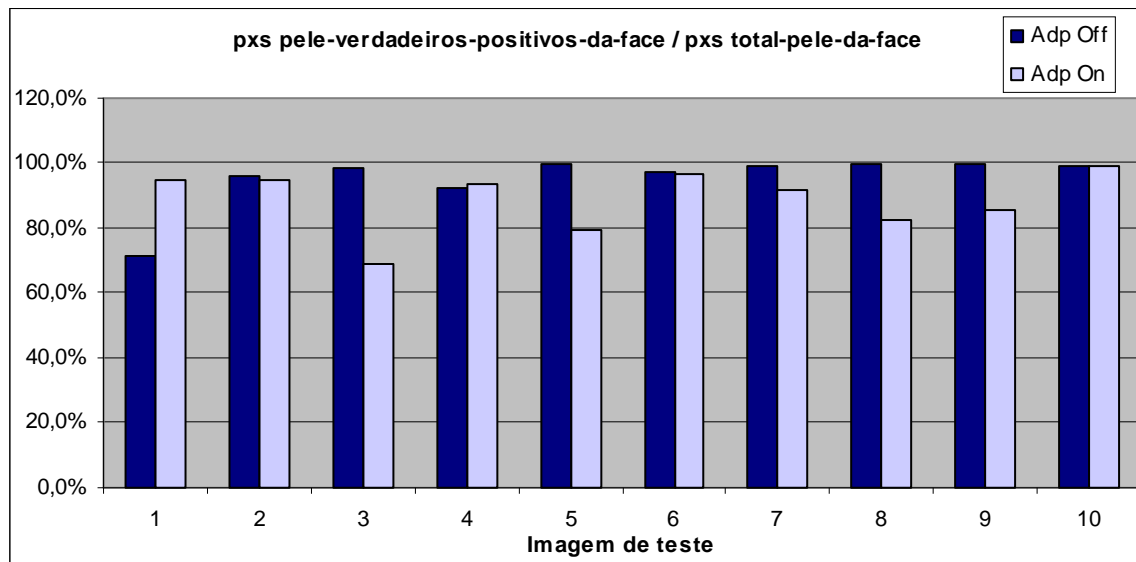


Fig. 4.20 Resultado dos testes com Lum = 1,1.

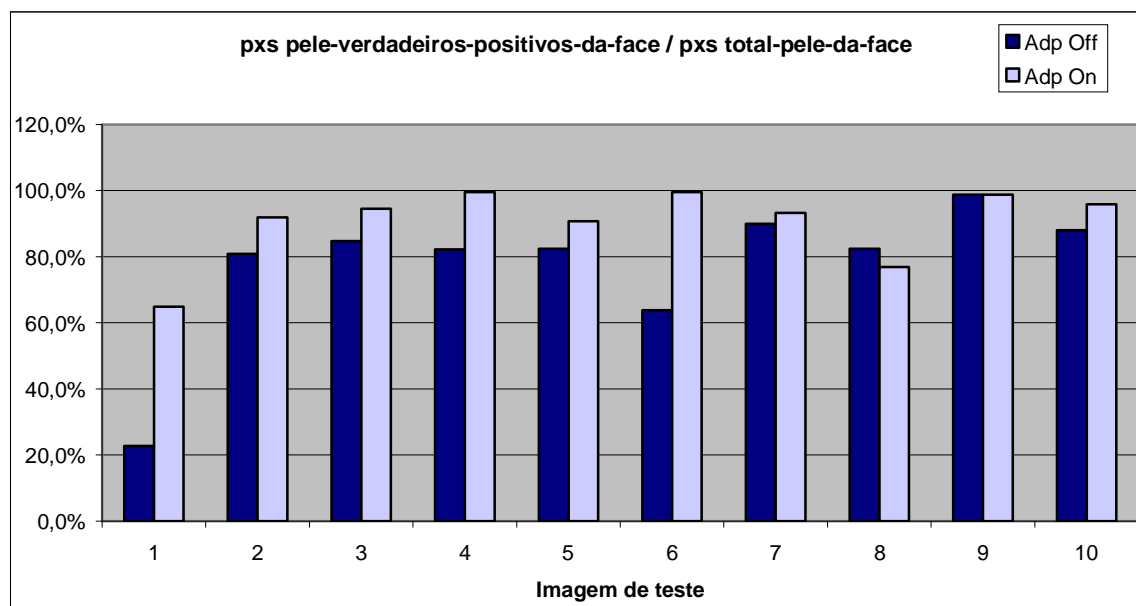


Fig. 4.21 Resultado dos teste com Lum = 0,75.

De referir que nos testes realizados com Lum = 0,75, com o algoritmo adaptativo activo (Adp on ), a maioria dos teste, apresentam resultados entre 90% e 100% o que representa uma melhoria entre 10% a 40% em relação os resultados sem algoritmo.

Nos teste com o brilho Lum = 0,35, os resultados mostram que com o algoritmo adaptativo (Adp on ), é apresentada uma significativa melhoria em relação aos testes sem o algoritmo adaptativo (Adp off ), uma melhoria com o valor médio de 60% e de valor máximo de 90%.

Na verdade, com este brilho, a totalidade dos pixeis é quase validada como verdadeira; mas sem algoritmo adaptativo o número de pixeis de pele verdadeira está perto de zero. Esta melhoria significativa obtida com o algoritmo de adaptação é devido ao facto do algoritmo base ter um desempenho mais fraco para imagens com brilho reduzido.

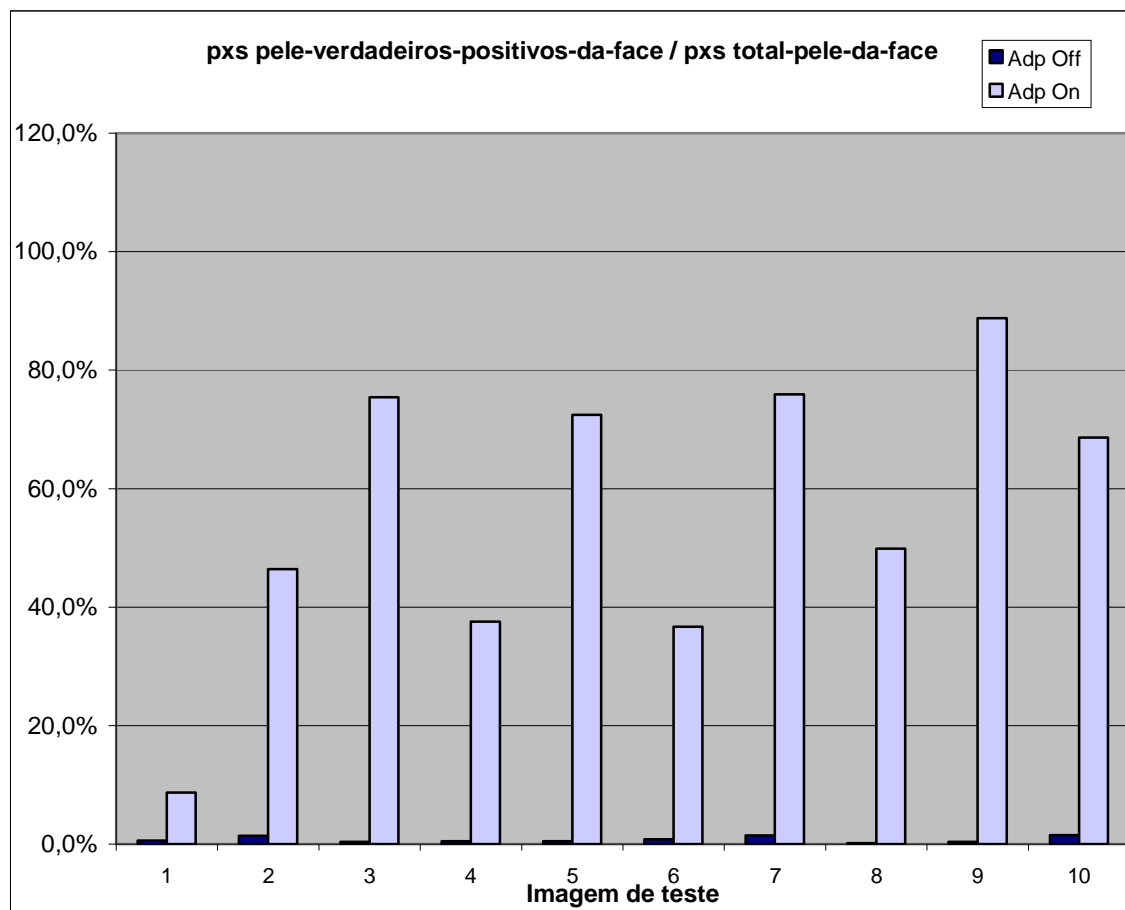


Fig. 4.22 Resultado dos testes com Lum = 0,35.

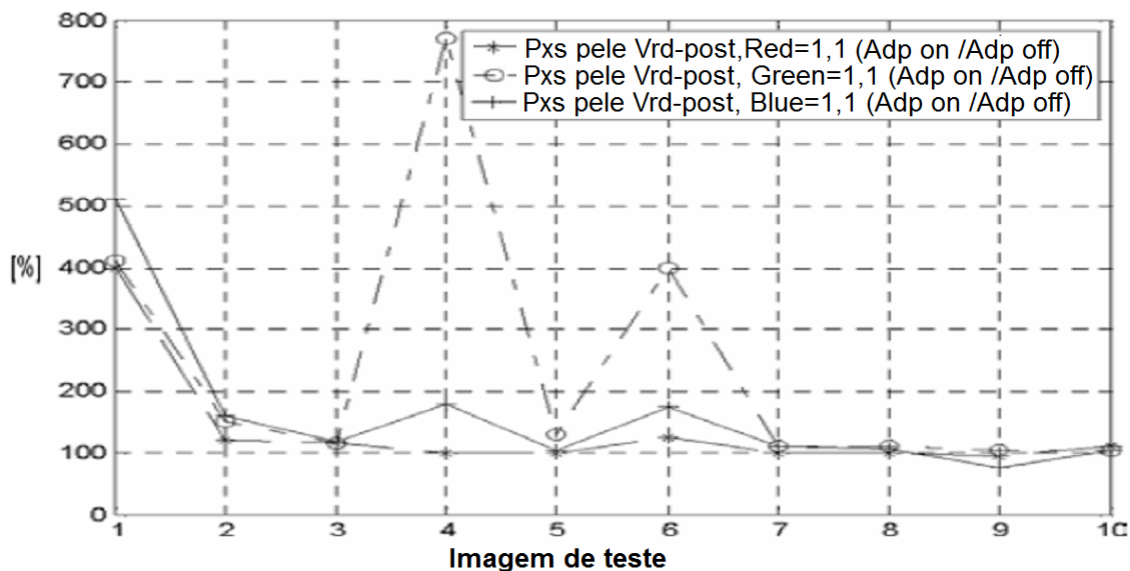


Fig. 4.23 Resultado dos testes com variação  $R = G = B = 1,1$  de forma independente, razão dos pixels pele-verdadeiros-positivos-da-face com e sem algoritmo (Adp On /Adp Off).

Finalmente, as Figuras 4.24 e 4.25 mostram os valores dos factores adaptativos ( $N_r$ ,  $N_g$  e  $N_b$ ) para os diferentes testes realizados nas imagens dos bancos de imagem, no caso 10 imagens diferentes.

Na Figura 4.25 são mostrados os valores de  $N_r$ ,  $N_g$  e  $N_b$  para  $Lum = 1,0$  aplicada a todos os três componentes de cor ( $R$ ,  $G$  e  $B$ ) e na Figura 4.24 são mostrados os mesmos parâmetros, mas apenas com  $Lum = 0,5$  aplicado à componente  $R$ . Estes dois exemplos mostram a capacidade do algoritmo desenvolvido para compensar a mudança na luminosidade provocada por uma mudança na iluminação não branca.

## 4.5 Conclusões

Neste capítulo, foi proposta uma nova abordagem para detectar pixels de pele em imagens a cores. A abordagem proposta combina um algoritmo-base para identificação de pele com um algoritmo adaptativo que ajuda a minimizar o efeito das variações do brilho sobre a imagem na detecção de pixels da pele. O método desenvolvido pode processar imagens, vídeos (importado de bases de dados) e imagens adquiridas por *webcams*. Os resultados experimentais obtidos mostram que a abordagem proposta é uma boa solução para compensar a variação do brilho da imagem causada pelas diferentes condições de iluminação. Os resultados do teste mostram que o algoritmo adaptativo tem, significativamente, melhores resultados comparativos quando o brilho diminui.



Quando o brilho da imagem aumenta, o algoritmo adaptativo reduz o erro (diferença) entre os pixels detectados e os pixels da pele validados.

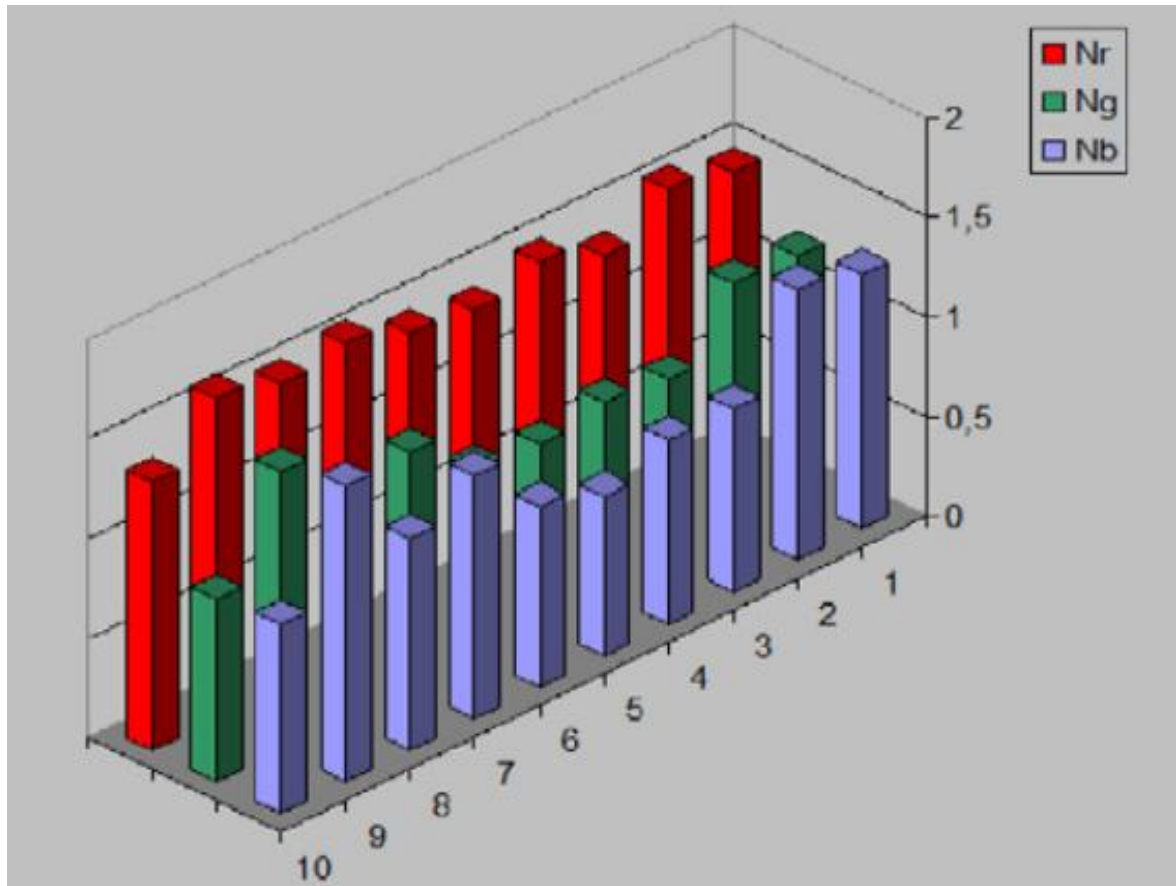


Fig. 4.24 Factores Nr, Ng e Nb com R = 0,50, G = 1,0 e B = 1,0.

Nos testes realizados para avaliar as variações independentes em cada um dos três componentes do espaço de cor (R,G e B), o algoritmo apresenta melhores resultados em alguns casos do que em outros. Este comportamento está relacionado com a distribuição estatística da base de cor da imagem (R, G, B).

Comparando-se a evolução dos factores de adaptação (NR, Ng e Nb) para as diferentes condições de iluminação, os testes mostram bom desempenho, nos dois casos, quando não há luminosidade alterada (Figura 4.25) e quando as mudanças de luminosidade ocorrem em apenas um componente de cor (Figura 4.24). Os testes provaram a capacidade do algoritmo adaptativo para reagir eficientemente às variações na luminosidade imagem, tanto nas variações na cor branca como nos outros componentes de cor.

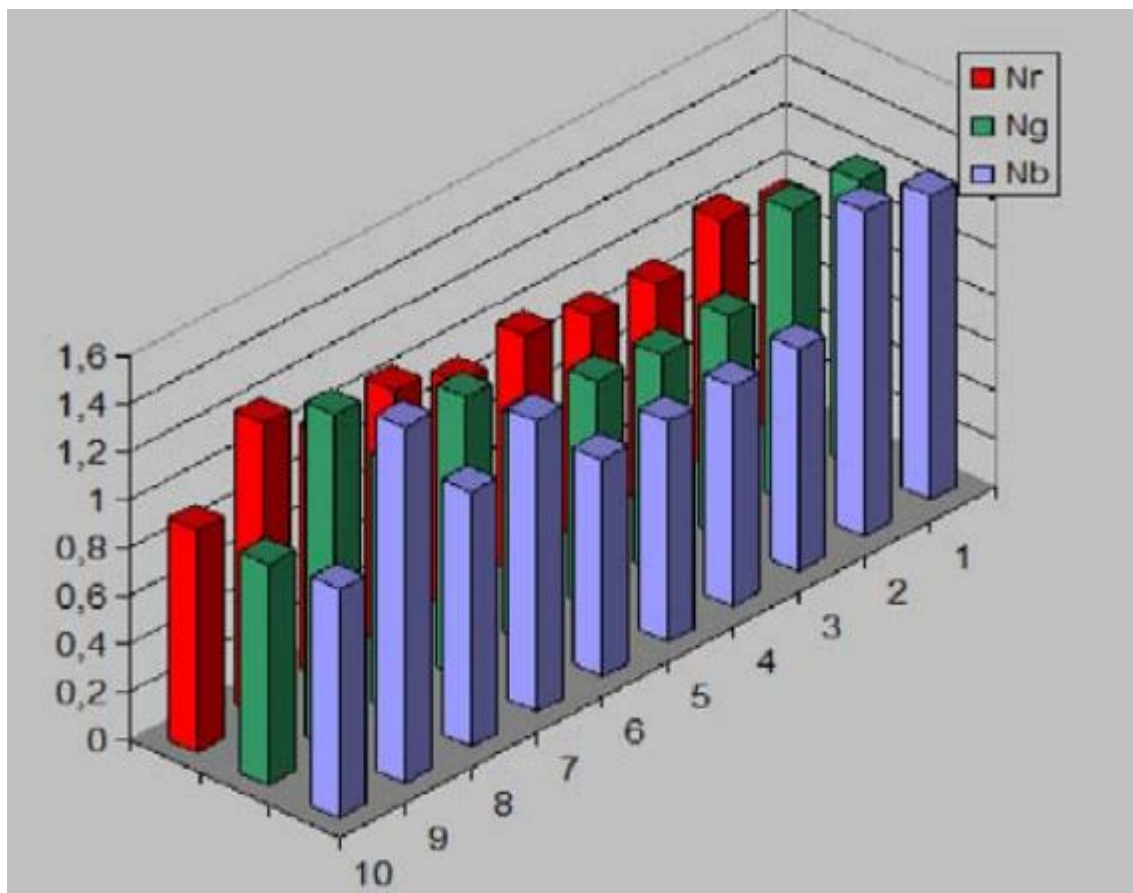


Fig. 4.25 Factores Nr, Ng e Nb com R = G = B = 1,0.

Os resultados fornecidos pelo processamento de imagens capturadas por uma *webcam* comum, com fundos complexos e diferentes condições de iluminação demonstraram a robustez do novo algoritmo adaptativo desenvolvido em relação às condições de iluminação extremas.

Tendo em conta a figura 4.26 onde é apresentado o desempenho do algoritmo adaptativo para três condições de iluminação:

- Fonte de iluminação: lâmpada fluorescente no tecto;
- Fonte de iluminação: lâmpada de halogéneo por trás;
- Fonte de iluminação: iluminação directa pela luz do monitor do computador.

Os testes mostram sempre melhores resultados na detecção do rosto, quando comparados com os resultados fornecidos sem o algoritmo adaptativo. O melhor desempenho é particularmente evidente quando as condições de iluminação são mais fracas. Com os resultados entre 100% e 2300%.

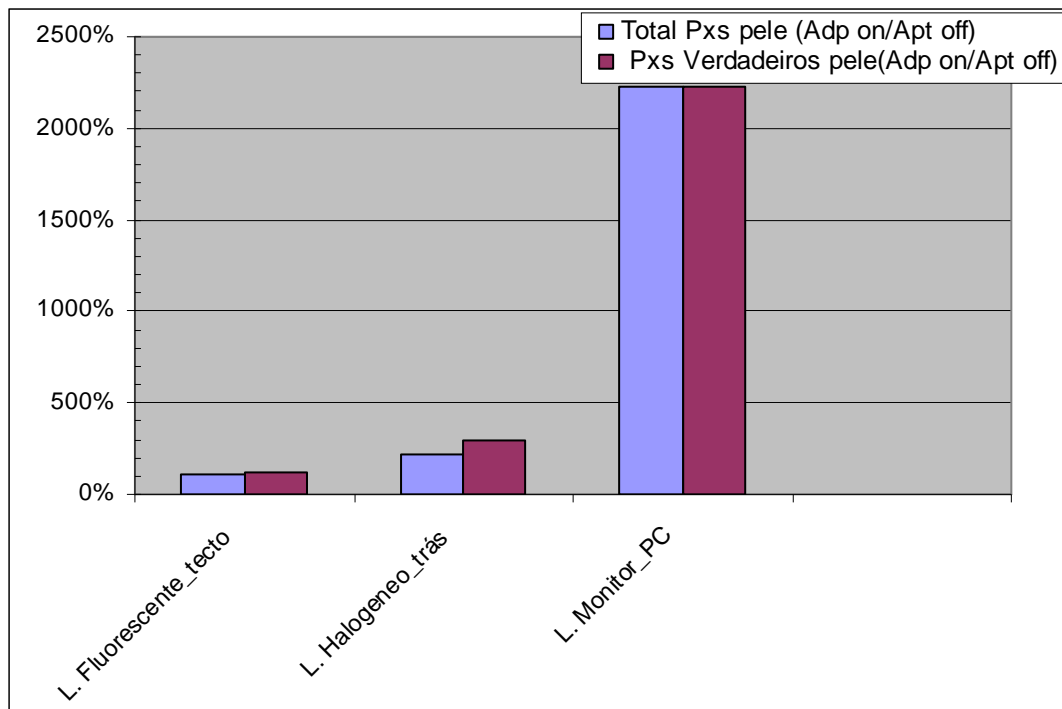


Fig. 4.26 Desempenho do algoritmo adaptativo nas três condições de iluminação.

Finalmente, a utilização da função centróide mostra um grande potencial em aplicações relacionado com a detecção de movimento / controlo em tempo real.

# Capítulo 5

## Detecção da variação de posição da face humana por enquadramento de rectângulo áureo com Método adaptativo

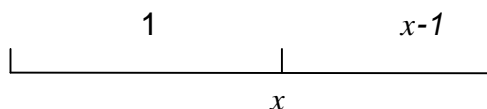
*Neste capítulo são apresentadas novas abordagens utilizando a relação do rectângulo áureo para enquadramento da face humana como validação e detecção da face humana num sistema de classificação de pele em imagens a cores para posterior avaliação da sua posição em imagens consecutivas, com o objectivo de controlo de um sistema por movimentação da face. O novo método utiliza um algoritmo adaptativo no algoritmo de detecção de base, para cada um dos componentes de cores RGB, a fim de reduzir a influência de diferentes ambientes de iluminação, com a capacidade de validação da face por enquadramento áureo, além de uma metodologia adaptativa na área de detecção de movimento para avaliação da variação da posição do rosto.*

## 5.1 Introdução

O uso do rosto humano como elemento de comando e controlo teve um tremendo crescimento em muitos campos - domésticos, industriais, militares - especialmente em ligação com questões de segurança. Em alguns sistemas interactivos, uma abordagem ao controlo de movimento é realizada através da detecção de rosto do operador. Outro campo importante na detecção de rosto humano está relacionado com a localização de pessoas em áreas específicas

### 5.1.1 O número de ouro ou razão divina

O número de ouro ou razão divina tem como símbolo a letra grega  $\Phi$  (phi) como homenagem ao construtor do Pártenon em Atenas, Fídias, sendo o próprio Pártenon um monumento que apresenta na sua forma vários exemplos de rectângulos áureos. Ao longo dos tempos muito se estudou a presença deste número tanto na natureza como nas obras de construção humana. Na própria morfologia humana existem muitos exemplos, os quais foram representados, de forma artística, por uma das principais figuras do renascimento, Leonardo da Vinci no famoso Homem Vitruviano. O número de ouro pode ser calculado tendo em conta a seguinte relação:



Dividindo-se um segmento de recta, com dimensão  $x$ , em dois segmentos com dimensão  $1$  e  $x-1$ , obtém-se uma partição áurea, se for válida a relação apresentada em (5.1)

$$\frac{x}{1} = \frac{1}{x-1} \quad (5.1)$$

Obtém-se então:

$$x(x-1) = 1 \quad (5.2)$$

$$x^2 - x - 1 = 0 \quad (5.3)$$

que apresenta duas soluções para  $x$

$$x = \frac{1 + \sqrt{5}}{2}, x = \frac{1 - \sqrt{5}}{2} \quad (5.4)$$

a solução positiva será o número  $\Phi$

$$x = \Phi \approx 1,618 \quad (5.5)$$

O número  $\Phi$  é um número irracional cuja componente decimal tem o número de casas infinito e não periódico.

Ao construir-se um retângulo em que a relação entre a altura e o largura é o número  $\Phi$ , o número de ouro, teremos o que se chama um retângulo áureo.

Os retângulos áureos devido à característica do próprio número áureo apresentam uma propriedade muito importante, que é a seguinte, se ao retângulo áureo se retirar um quadrado de lado igual à altura, o retângulo resultante continuam a ser um retângulo áureo, ou seja, a relação entre a altura e a largura continua a ser o número de ouro. Se voltarmos a ter em conta as relações do número  $\Phi$  na morfologia humana ressalta uma característica importante para o desenvolvimento do presente capítulo, que é a seguinte: a face humana, vista de frente, tem uma relação na suas dimensões que a leva a ser inscrita num retângulo áureo, ou seja, a distância do queixo ao topo a testa tem uma relação de  $\Phi$  com a largura da face, figura 5.1.

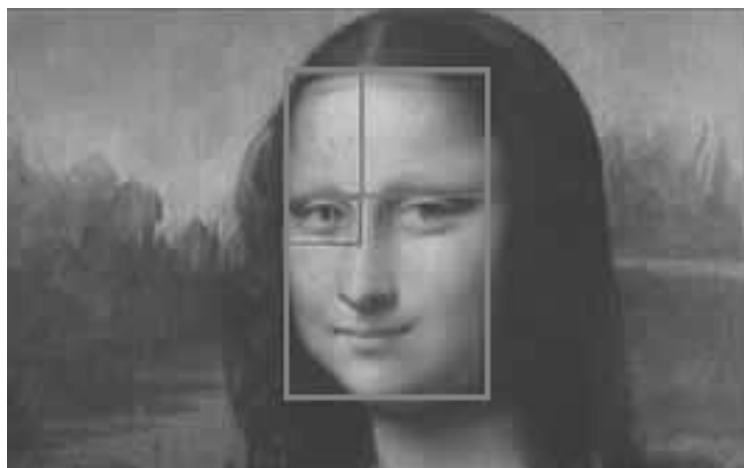


Fig. 5.1 Rectângulo áureo na face humana  
(webeduc.mec.gov.br/portaldoprofessor/matematica/condigital2/guias/Guia\_audiovisual\_i.pdf, acedido em Maio de 2019).

## **5.2 Algoritmo de Processamento de Imagem**

A explicação da metodologia desenvolvida vai ser feita em duas partes: Em 5.2.1 será apresentada a detecção de pele para a identificação da face; Em 5.2.2 será apresentada a detecção da variação da posição da face pela utilização do enquadramento da face no respectivo rectângulo áureo.

### **5.2.1 Detecção de pele**

#### **5.2.1.1 Algoritmo de detecção de pele com Método adaptativo**

O algoritmo utilizado na detecção de pele foi o algoritmo apresentado no capítulo 4. Este algoritmo utiliza segmentação por classificação nos três componentes de cor do sistema RGB (vermelho, verde e azul), para detecção de pele. O método proposto utiliza a metodologia adaptativa incorporada no algoritmo classificador da pele para a classificação do pixel (Grilo and Figueiredo, 2014). O algoritmo adaptativo varia o brilho da imagem em cada um dos componentes do espaço de cor RGB, com uma nova abordagem para a segmentação de pele e detecção de rosto.

O algoritmo desenvolvido implementa os três níveis adaptativos definidos para cada um dos três componentes de cor RGB, dependendo da amplitude do valor médio de cada componente ( $MnVI$ ),

A estratégia desenvolvida demonstrou a sua capacidade para compensar as alterações de brilho da imagem para cada um dos componentes de cor.

### **5.2.2 Detecção da variação da posição da face**

#### **5.2.2.1 Detecção da posição da face**

Depois da detecção das áreas classificadas como pele, a área com maior área que tenha uma dimensão na horizontal menor que a vertical e que, devido às dimensões que apresenta, seja enquadrada num rectângulo áureo, é classificada como face, fig. 5.2. Tendo em conta a dimensão horizontal, é calculada a altura (dimensão vertical) do rectângulo com a utilização do número áureo e se a área classificada como pele, ficar no interior do rectângulo, é considerado face. Será a posição do rectângulo que será posteriormente utilizada na detecção da variação da posição da face.

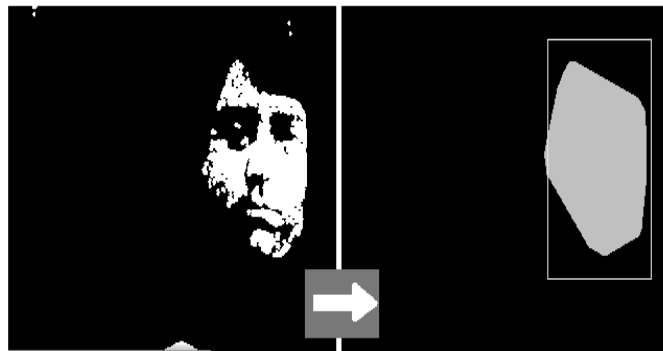


Fig. 5.2 Detecção da posição da face com o rectângulo áureo.

### 5.2.2.2 Detecção da variação da posição da face

A detecção da variação da posição da face é feita com a detecção da variação do centro do rectângulo áureo que enquadra a face, ou seja, com a aquisição de imagem em tempo real e com a detecção da face activa, o processo vai determinar a variação nos eixos x e y, das coordenadas do centro do rectângulo que enquadram a face entre duas imagens consecutivas, adquiridas no tempo ou entre um ponto previamente determinado, por exemplo centro da imagem (Fig. 5.3).

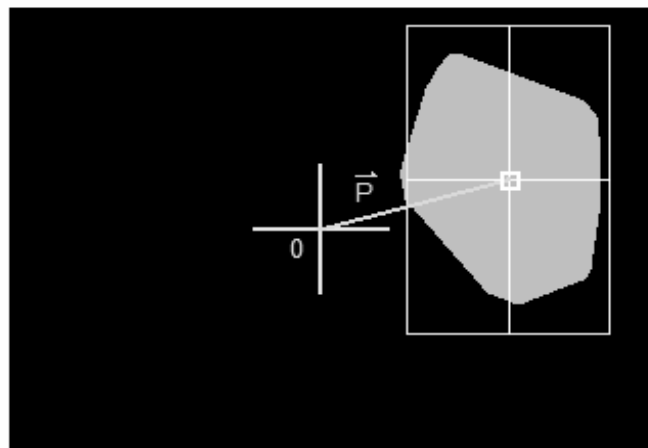


Fig. 5.3 Detecção variação da posição da face em relação à origem de um referencial do centro da imagem,  $P = P_x + P_y$ .

### 5.2.2.3 Área de detecção adaptativa da pele (Face)

A área da imagem de teste para a detecção de pele (face) será função da variação de posição da área da face, detectada nas iterações anteriores, aumentado assim a eficiência do processo nomeadamente no volume de dados



tratados e na velocidade de processamento. Ao realizar-se a primeira detecção da face, o respectivo rectângulo áureo será utilizado como base  $R_b$ , no processo, calculando-se depois a variação da posição no eixo  $x$  dos centros dos rectângulos, entre duas imagens adquiridas em iterações consecutivas,  $dx$ . O valor de  $dx$  é calculado no processo adaptativo que em conjunto com o rectângulo base  $R_b$  determinam a próxima dimensão da área de imagem a ser testada no sistema de detecção de pele  $Ad$ , em vez de ser testada toda a área a imagem adquirida (fig 5.4). O processo adaptativo reage (adapta-se), ou seja calcula a dimensão da area de procura  $Ad$  tendo em conta a variação da posição do rectângulo que enquadra a face entre duas imagens consecutivas, ao que se pode chamar a "velocidade" de movimento  $V_x$ , expressões 5.6 e 5.7 e figs. 5.4 e 5.5.

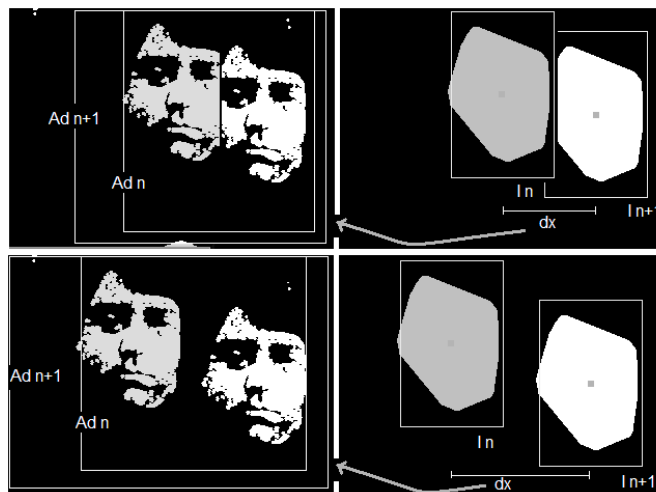


Fig. 5.4 Variação da área de detecção de pixels de pele  $Ad$  em função da variação de posição da área da face detectada nas iterações anteriores  $dx$ .

$$dx = |x(i+1) - x(i)|; \quad V_x = \frac{dx}{2} \quad (5.6)$$

$$Ad = f(R_b, V_x) \quad (5.7)$$

Onde:

$x(i)$  e  $x(i+1)$  são as coordenadas em  $x$  dos rectângulos que enquadram os pixels de pele detectados em imagens consecutivas;

$R_b$  é o rectângulo base;

$Ad$  é a dimensão da área de imagem a ser testada no sistema de detecção de pele.

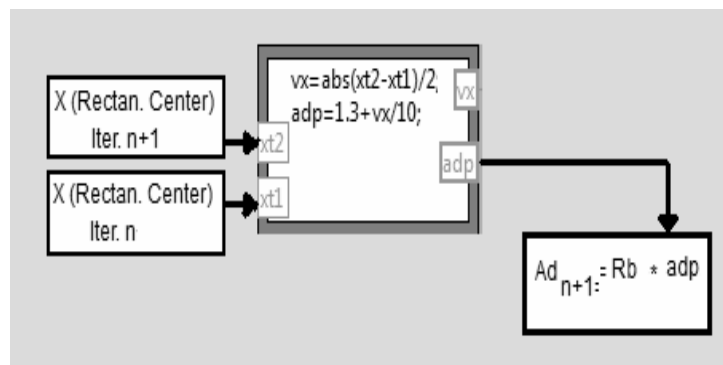


Fig. 5.5 Processo adaptativo da variação da área de detecção de pixels de pele Ad.

### 5.3 Implementação do sistema

Na implementação, desenvolveu-se um sistema dividido em dois algoritmos que se complementam: algoritmo A - *Algoritmo Adaptativo para Detecção de Pele em Imagens As cores*; e algoritmo B – *Algoritmo de detecção da variação de posição da face humana por enquadramento de rectângulo áureo com Método adaptativo* (fig. 5.6).

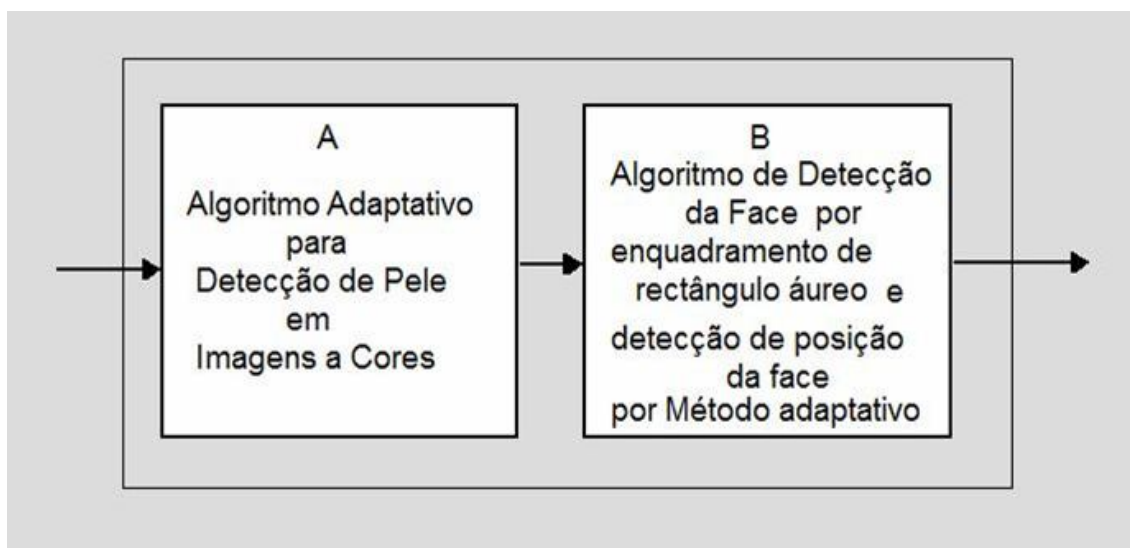


Fig. 5.6 Implementação do Sistema.

O sistema foi implementado com a ferramenta de desenvolvimento de software LabVIEW, a correr sobre o sistema operativo Windows.

A aplicação de software desenvolvida teve como principais objectivos a implementação e monitorização dos algoritmos e a geração de informação em

ficheiro formato ASCII (.txt), para posterior utilização em qualquer ferramenta de análise de resultados.

Também foi desenvolvida a funcionalidade que possibilita a sua *interface* com outras aplicações, na utilização da detecção da posição de face, para o potencial controlo de posição num sistema 2D (2-Dimensões, x e y), através do protocolo *DataSocket*, na mesma máquina, através do sistema operativo, ou em rede, via *ethernet*. Ou através do protocolo Xbee.

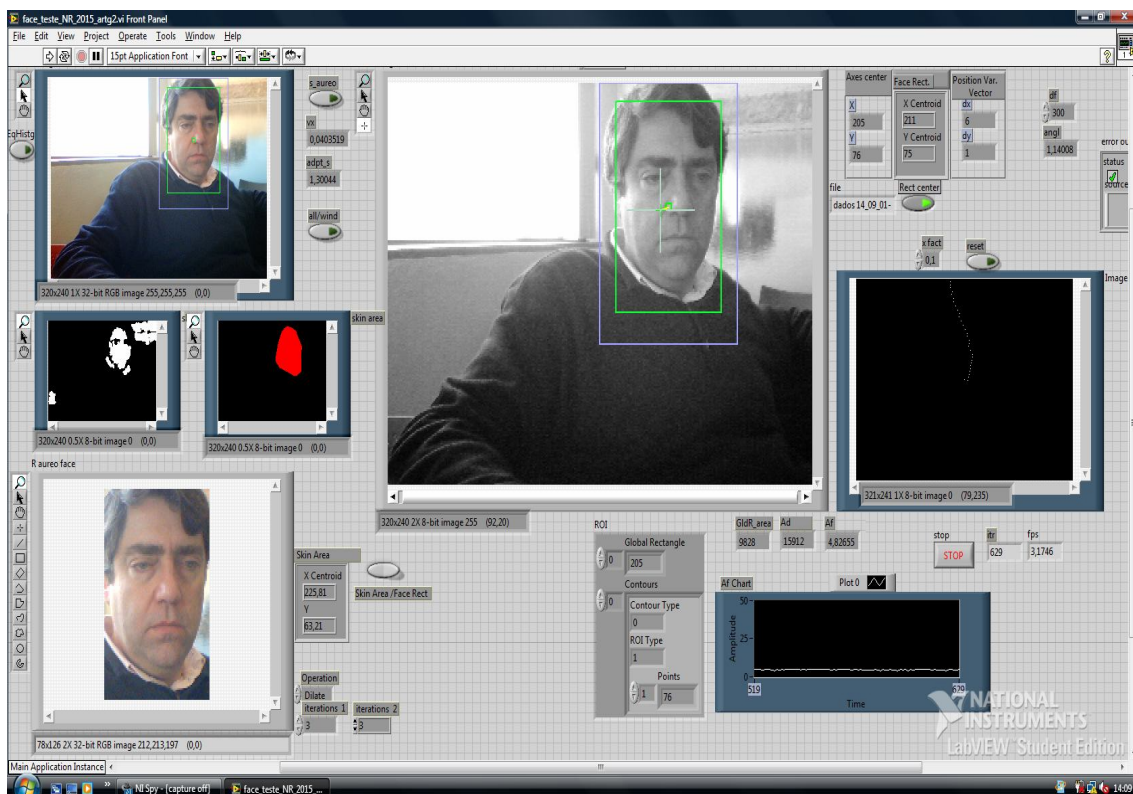


Fig. 5.7 Painel do utilizador da aplicação do sistema

A aplicação de *software* desenvolvida processa a informação recebida de uma qualquer câmara, via USB, no caso presentemente apresentado foi utilizada a *webcam* do computador, mas para outro tipo de teste, poderá processar informação a partir de um ficheiro que contenha vídeo.

As imagens adquiridas pela aplicação são convertidas para o formato IMAQ do LabVIEW. O formato IMAQ possibilita a utilização das funções de visão a partir da biblioteca LabVIEW. No trabalho desenvolvido o formato IMAQ foi convertido numa matriz RGB 24 bits (8 bits para cada componente de cor).

A aplicação desenvolvida tem a interface de utilizador apresentada na figura 5.7, a qual é constituído por três áreas distintas: i) Área das imagens; ii) Área dos comandos; e iii) Área de apresentação dos resultados, em tempo real.

### 5.3.1 Algoritmo A

#### 5.3.1.1 Área de imagens do Sistema A

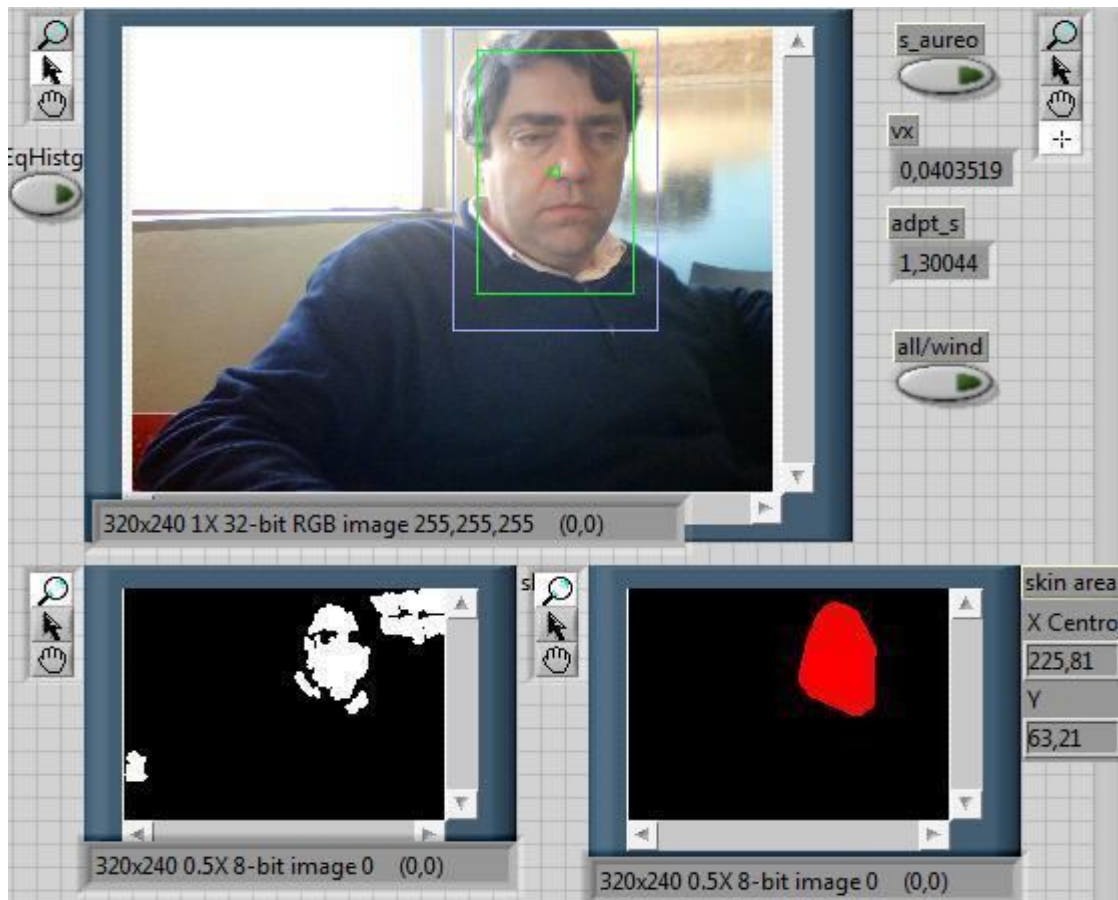


Fig. 5.8 A Área das imagens dos sistemas 1, 2A e 3A.

Na Área da Imagem existem cinco imagens, as três primeiras (1, 2A e 3A) são apresentadas na figura 5.8. A primeira imagem (1) apresenta o vídeo original a cores, sendo a imagem recolhida em tempo real, ou carregada a partir de um ficheiro. A segunda imagem (2A) identifica todos os pontos na imagem, com uma classificação de pele positiva, de acordo com o método desenvolvido, em preto / branco. A terceira imagem (3A) ilustra o resultado da análise para identificação de face na imagem (cor vermelha) explicado seguidamente, no ponto 5.3.1.2. Esta imagem, 3A, permite visualizar o resultado do algoritmo de detecção da pele, para posterior detecção da face, assim como as coordenadas do centro (x,y) na imagem original. Na imagem 1 são apresentados dois rectângulos: o primeiro rectângulo (verde), que enquadra a

face com o rectângulo áureo, o qual valida assim, com o algoritmo B, que é uma face humana, o código em LabVIEW desenvolvido que implementa a função é apresentada na figura 5.9; o segundo rectângulo (azul) que também é um rectângulo áureo, equivalente ao verde, mostra a área de busca de detecção de movimento, a qual irá variar, tendo em conta a “dinâmica” da variação da posição da face, enquadrada pelo primeiro rectângulo (verde), segundo a funcionalidade do algoritmo B, o código LabVIEW desenvolvido que implementa a função é apresentada na figura 5.10.

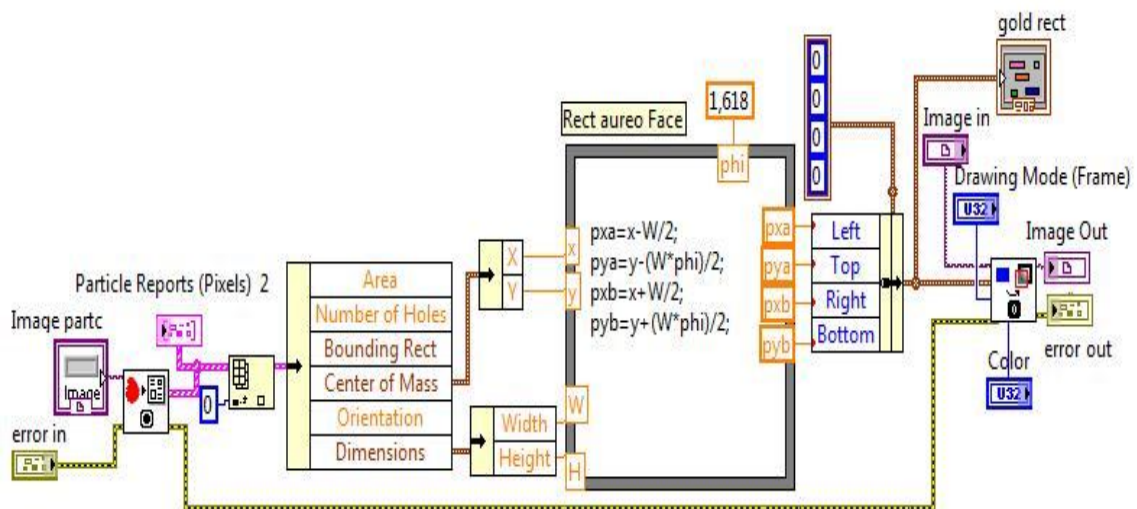


Fig. 5.9 O código LabVIEW desenvolvido que implementa a função que enquadra a face com o rectângulo áureo para detecção da face.

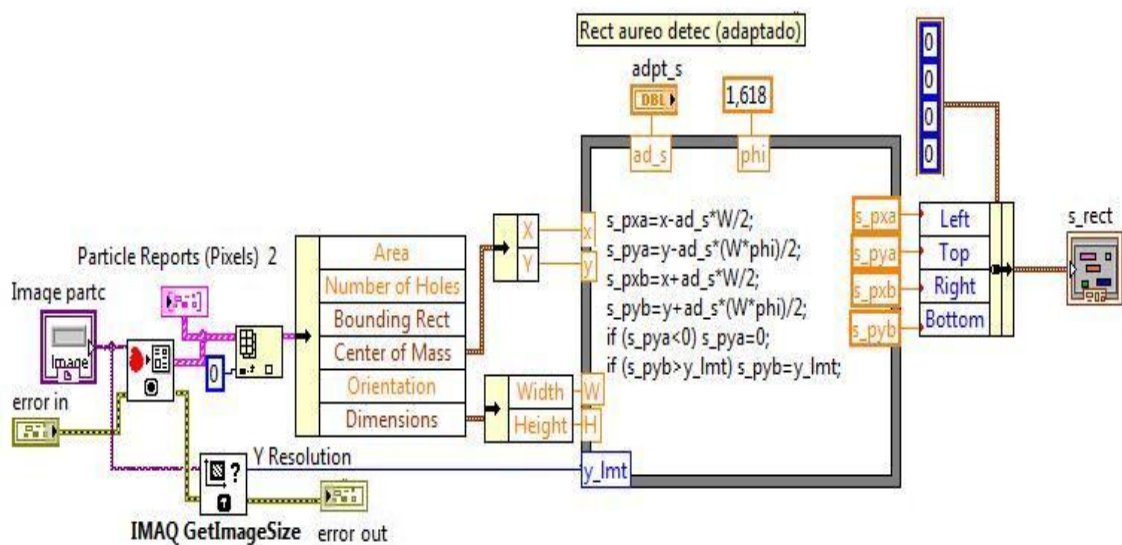


Fig. 5.10 O código LabVIEW desenvolvido que implementa o rectângulo áureo da área de busca de detecção de movimento a qual irá variar tendo em conta a “dinâmica” da variação da posição da face enquadrada.

### 5.3.1.2 Detecção e validação dos pixels de pele e sua detecção no bloco de face

Para a validação de pixels de pele verdadeiros a aplicação inclui uma funcionalidade que implementa o algoritmo que toma a imagem composta por todos os pixels inicialmente identificados como pixels de pele e processa esses dados através de várias funções (no ambiente LabVIEW) para validar os pixels. O algoritmo de validação de pixels de pele verdadeira foi apresentado e desenvolvido no capítulo 4.

Por fim com o Algoritmo B se a área detectada for possível esquadrar num retângulo de proporções áureas é validado como Face humana, com explicado no ponto 5.2.2.1.

## 5.3.2 Algoritmo B

### 5.3.2.1 Área de imagens do Sistema B

A imagem central 4B (fig. 5.11) é semelhante à imagem 1 (fig. 5.8), embora com a imagem em tons de cinzento, onde é apresentado o centro do retângulo que enquadra a face, e o centro de posicionamento em 2D. Com a utilização do rato, colocando o segundo cursor (cruz verde claro) em cima do primeiro cursor (pequeno quadrado verde), calibra-se o início do sistema de posicionamento por movimento da face. O resultado do posicionamento 2D é apresentado no gráfico à direita do painel (fig. 5.12). Também são apresentados os valores do centro do retângulo que enquadra a face (*Face Rect X, Y centroid*) e os valores da variação do mesmo, entre *frames* (*Position Var, Vector dx, dy*), utilizados no algoritmo para adaptar a área de detecção de movimento, conforme foi explicado em 5.2.2.3

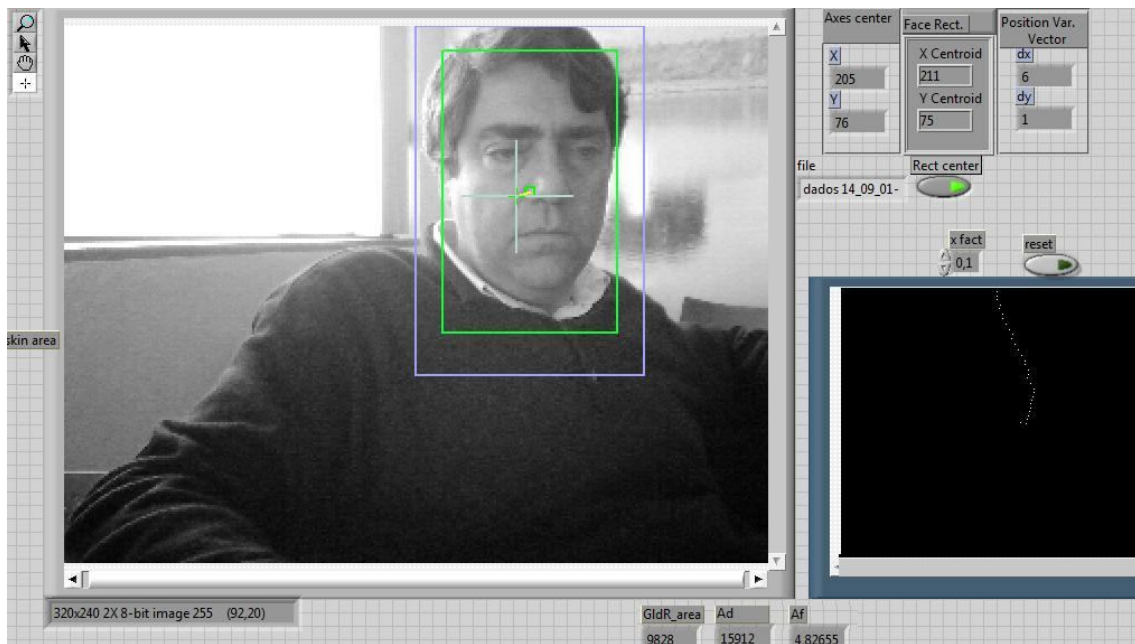


Fig. 5.11 A imagem 4B.

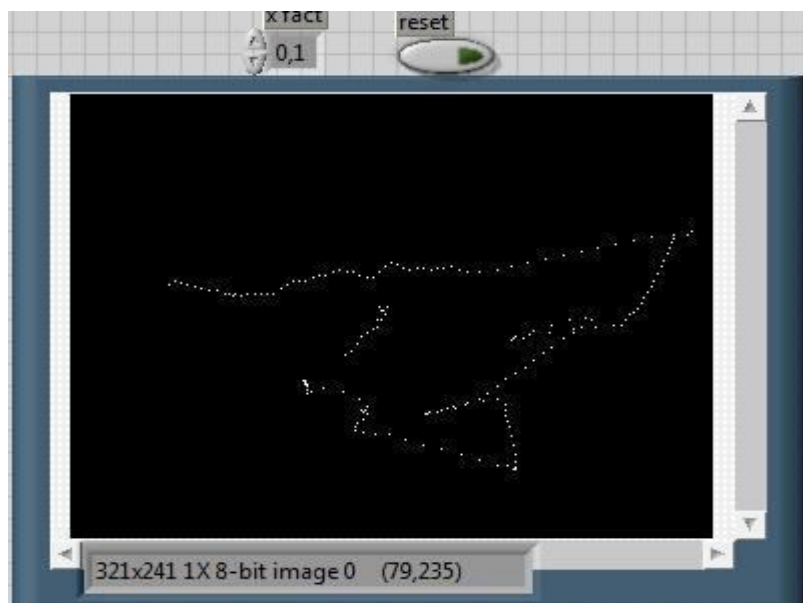


Fig. 5.12 Posicionamento 2D.

A quinta imagem 5B (fig. 5.13) apresenta a imagem da face isolada, enquadrada no rectângulo áureo (verde), da imagem 1 (fig. 5.7).



Fig. 5.13 A imagem 5b.

Por fim é apresentado um gráfico que mostra os valores ao longo do teste do factor  $A_f$  (fig. 5.14). Este teste irá avaliar a eficiência do algoritmo, pois como mostra a expressão (5.8) o factor  $A_f$  relaciona o  $N_t$  (numero total de pontos da imagem) com o  $A_d$  (numero de pontos da área onde feita a detecção do movimento).

$$A_f = \frac{N_t}{A_d} \quad (5.8)$$

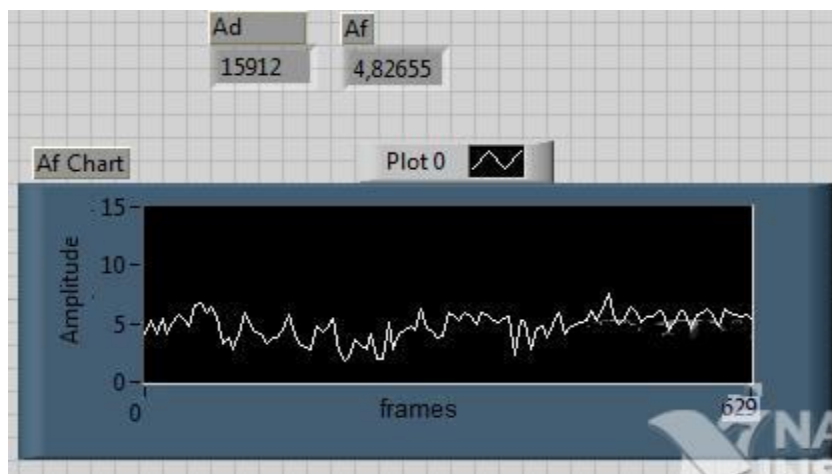


Fig. 5.14 O gráfico do factor  $A_f$ .

Os dados deste gráfico, bem como outros valores de teste, são guardados num ficheiro em formato ASCII (.txt) que será usado na análise de resultados.



### 5.3.3 Sistema de utilização prática de posicionamento 2D

Para testar de forma prática a utilização da detecção da posição do rosto como um elemento de comando para sistema de posicionamento 2D, desenvolveu-se a funcionalidade que pode controlar um robô móvel

O robô móvel utiliza uma placa Arduino Uno que controla os motores com uma placa dedicada. O Robô móvel recebe os comandos do PC, onde corre a aplicação desenvolvida, através do protocolo de comunicação implementado sobre o protocolo Xbee via rádio frequência.

Os comandos desenvolvidos para os testes são: Frente; Trás, Direita e Esquerda.

## 5.4 Análise e conclusões

### 5.4.1 Testes realizados

Os testes realizados tiveram como objectivo testar a eficiência do sistema na capacidade de detecção do movimento da face humana e na redução da respectiva área de teste de detecção.

Os testes realizados tiveram em conta vários tipos de situação:

- i) imagem com tempo de actividade, ou seja movimento da face , inferior, ou igual, a 30% da duração do teste;
- ii) imagem com tempo de actividade entre o 30% e 60% da duração do teste;
- iii) imagem com tempo de actividade entre 60% e 100% da duração do teste.

A realização dos três tipos de teste teve como objectivo medir a eficiência do algoritmo em diferentes situações de dinâmica da face detectada.

Para cada teste são apresentados dois gráficos:

- i) onde se mostra Rect 1, o número de pixels do rectângulo que enquadra a face e Rect 2 a área de detecção de movimento;
- ii) onde se mostra a variação do factor  $A_f$ .

O factor  $A_f = \frac{N_t}{A_d}$  relaciona o número total de pixels da imagem com a área de detecção de movimento, dando assim uma medida da eficiência do método. Nos testes aqui apresentados o número máximo de pixels foi:  $320 \times 240 = 76.800$ .

Nas figuras 5.15 e 5.16, são apresentados os resultados referentes ao primeiro tipo de testes (imagem com tempo de actividade inferior, ou igual, a 30% da duração do teste).

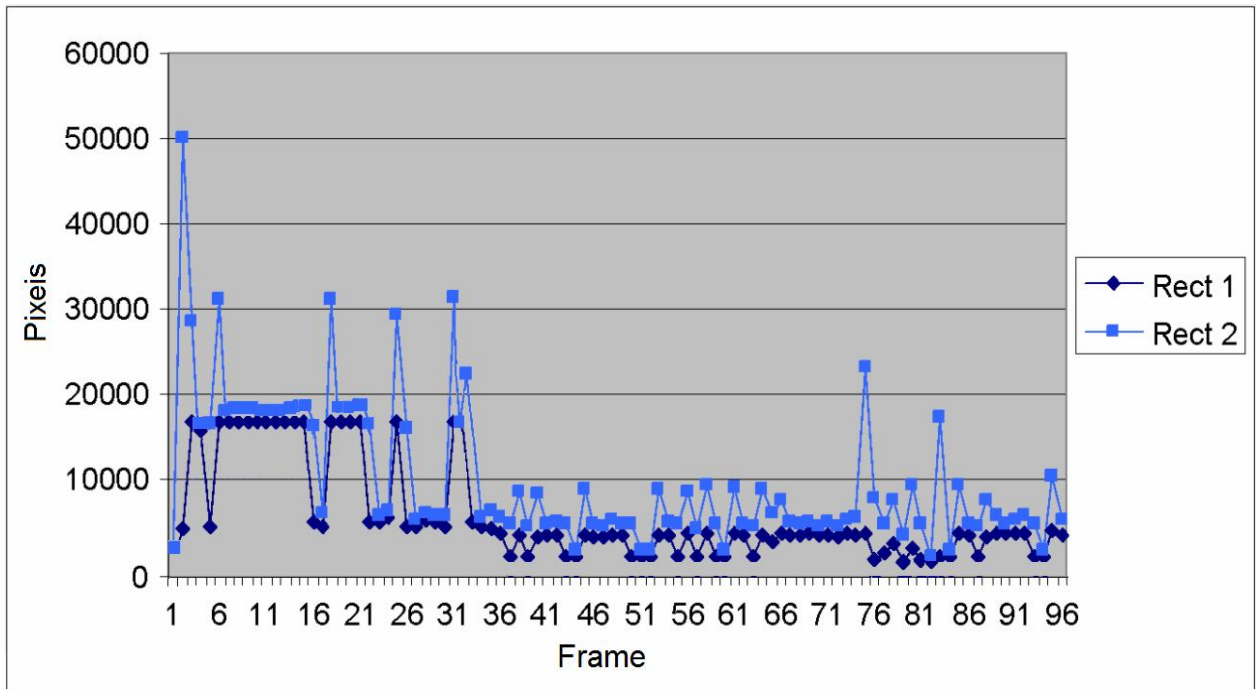


Fig. 5.15 Valor de Rect 1 e Rect 2 (area de busca).

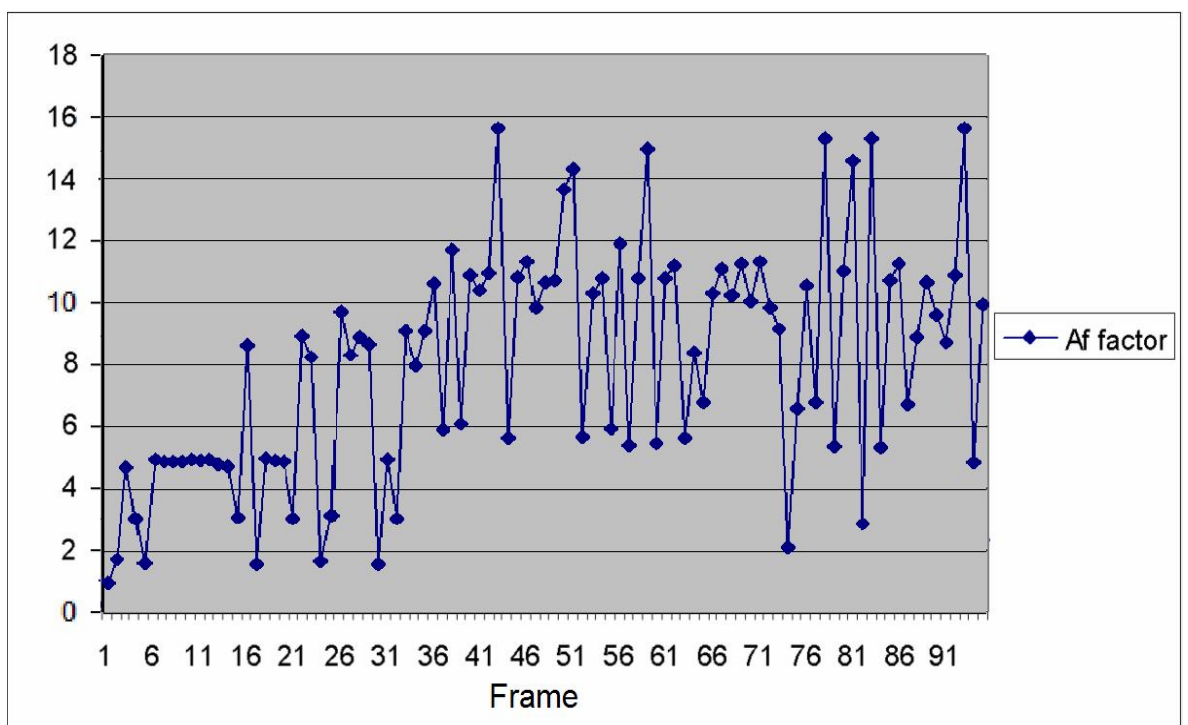


Fig. 5.16 Valor do factor Af.

Nas figuras 5.17 e 5.18, são apresentados os resultados referentes ao segundo tipo de testes (imagem com tempo de actividade entre o 30% e 60% da duração do teste).

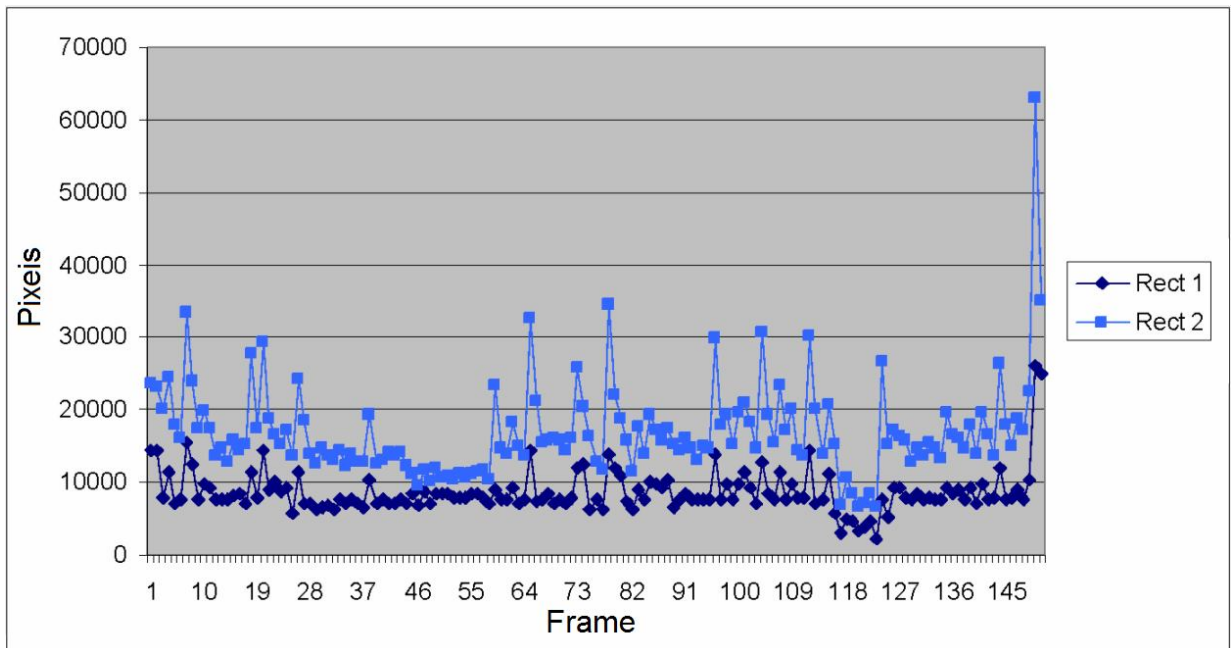


Fig. 5.17 Valor de Rect 1 e Rect 2 (area de busca).

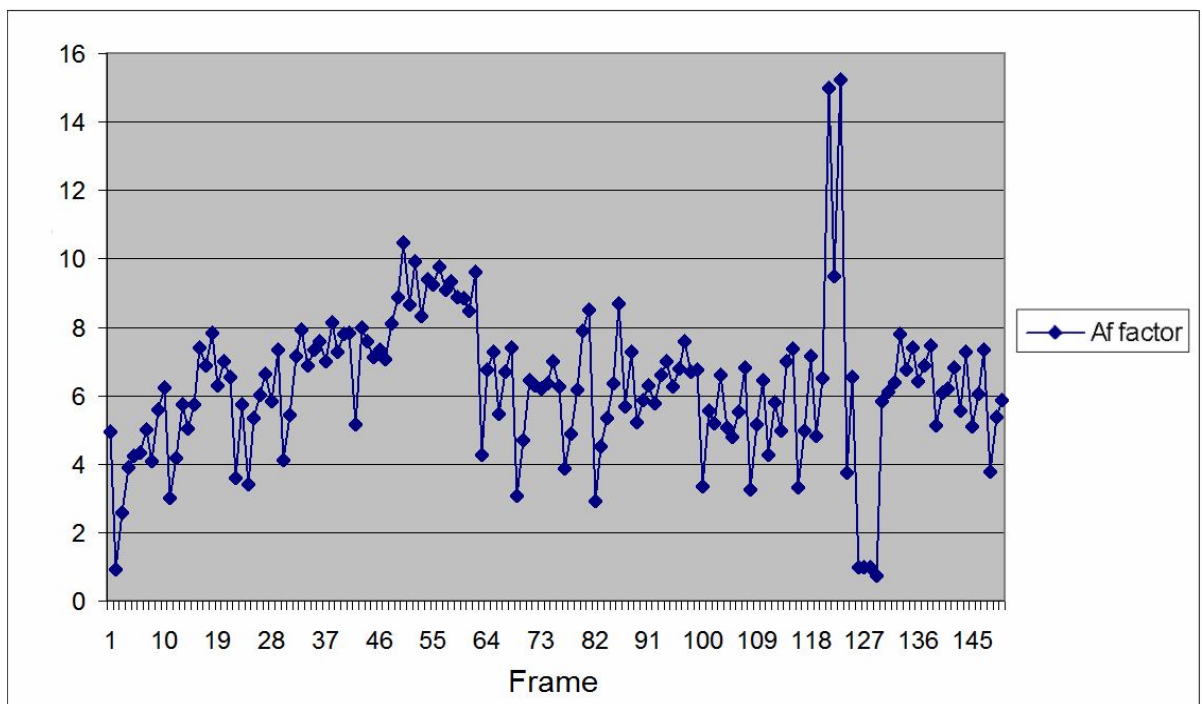


Fig. 5.18 Valor do factor Af.

Nas figuras 5.19 e 5.20 são apresentados os resultados referentes ao terceiro tipo de testes (imagem com tempo de actividade entre 60% e 100% da duração do teste).

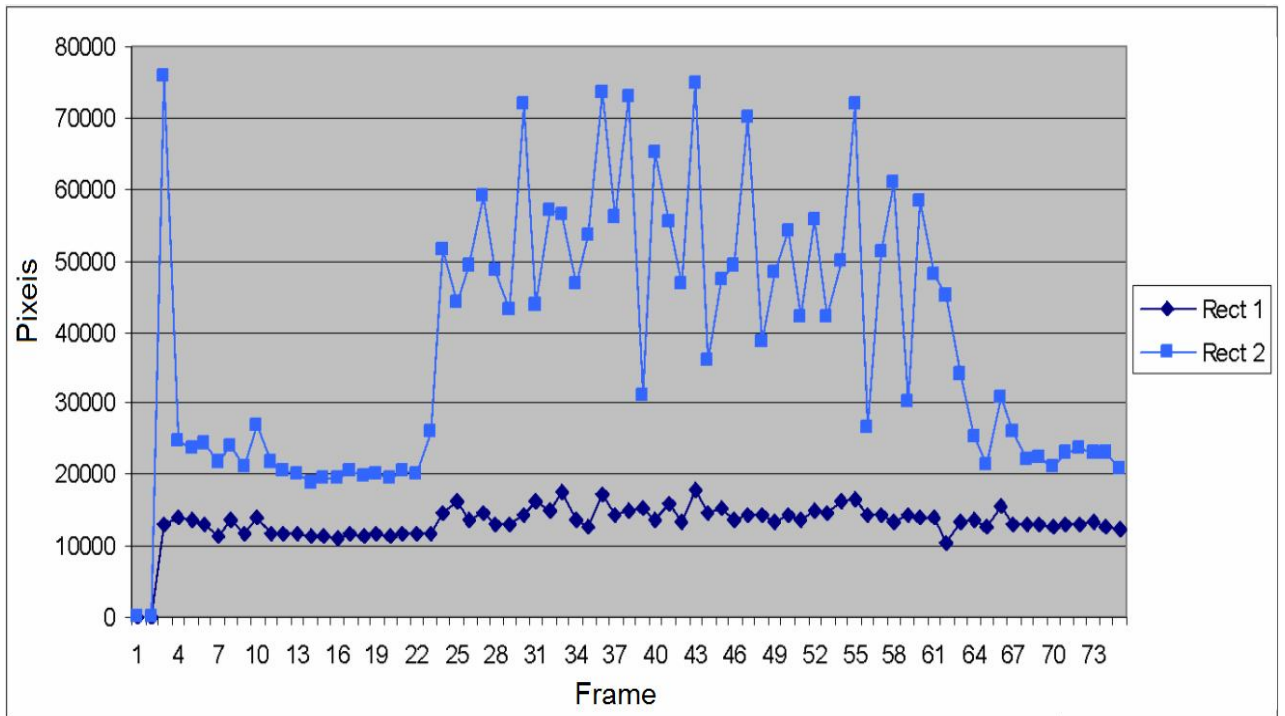


Fig. 5.19 Valor de Rect 1 e Rect 2 (area de busca).

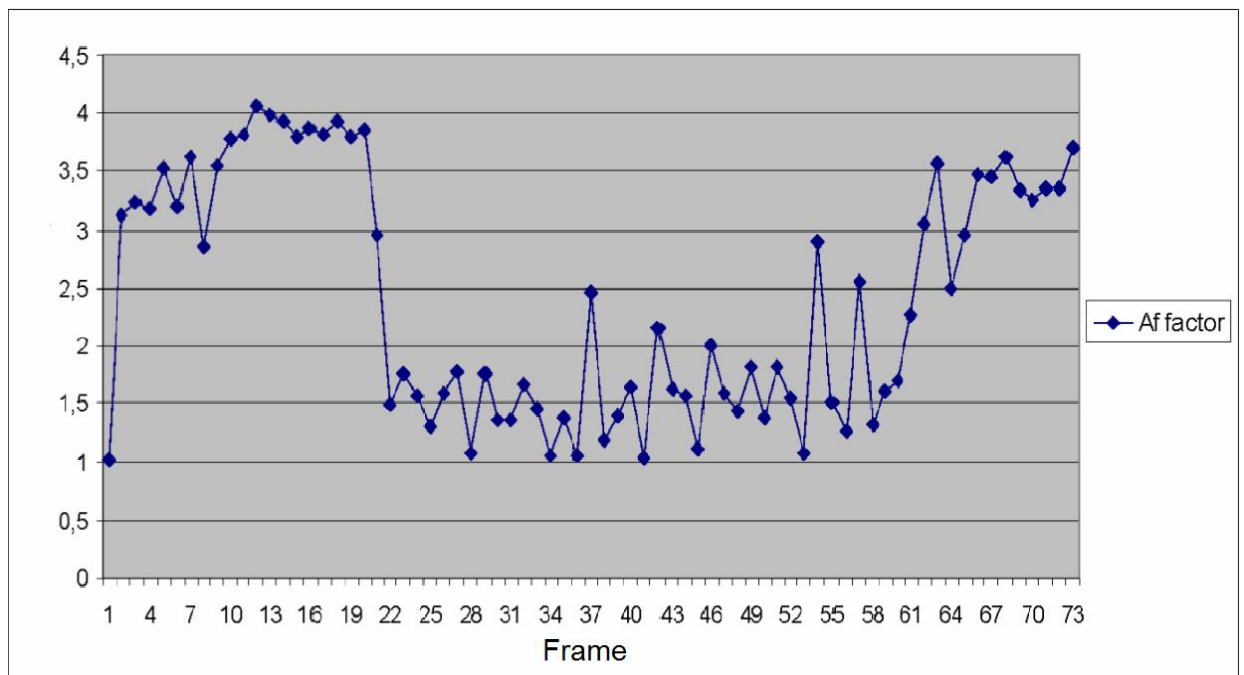


Fig. 5.20 Valor do factor Af.

### 5.4.2 Análise com o factor Af

Na figura 5.21 apresentam-se os resultados para os testes realizados com o algoritmo de adaptação proposto, utilizando o factor Af (rácio do numero total de pixeis e o número de pixeis testados na detecção de movimento), para avaliar o desempenho do algoritmo nos 3 cenários de teste estudados:

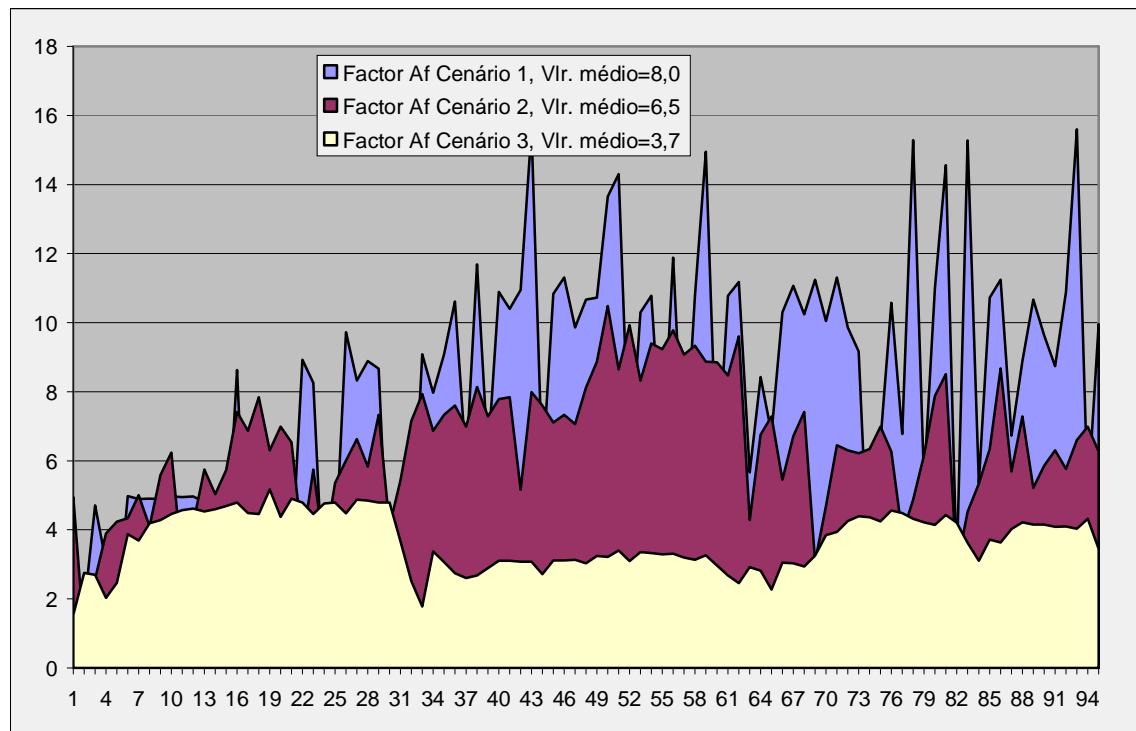


Fig. 5.21 A variação de Af nos três cenários de teste e o seu Valor médio.

Verifica-se no três cenários de teste que o valor médio do factor Af é no pior caso superior a 3,7, o que mostra que em média há sempre redução do número de pixeis a serem testados em relação ao número total de pixeis da imagem.

### 5.4.3 Conclusões

Os testes realizados com o algoritmo de adaptação proposto neste capítulo nos três cenários de teste propostos, mostram que tendo em conta a análise dos valores do factor Af (razão entre número total de pixeis e o número de pixeis testados na detecção de movimento) os casos mais desfavoráveis verificaram-se, quando o movimento se torna constante, ao longo do teste, apresentando valores entre 1,5 e 2. Quando o movimento diminui, verificaram-se valores entre 4 e 6 chegando, por vezes a valores máximos de 10 e 11.

Verifica-se também que, no caso mais desfavorável, o melhoramento para uma imagem com 80.000 pixels, traduz-se numa redução de 58.400 pixels (-73%) a processar, com valores médios de redução de cerca de 68.000 pixels (-85%) com máximos de 70.400 pixels (-88%).

Nos 3 cenários testados obtiveram-se 3 casos diferentes, tendo em conta o valor médio do factor Af. Assim, verificou-se, para o caso mais desfavorável, (Vlr. Médio de Af =1,5), um melhoramento traduzido numa redução do volume de dados de 73%.

Para o caso correspondente a Vlr. Médio de Af =6,5, a redução do volume de dados correspondente foi de 85%. Finalmente, e para o caso mais favorável (Vlr. Médio de Af =8,0), verificou-se uma redução do volume de dados de 88%. Se tivermos em conta 4 tipos de resolução (ver tabela 3.1, cap. 3):

- QVGA (*Quarter Video Graphics Array*),
- VGA (*Video Graphics Array*),
- SVGA (*Super Video Graphics Array*)
- XGA (*eXtended Video Graphics Array*) ;

respectivamente

- 360 x 640 (76.800 pixels),
- 640 x 480 (307.200 pixels),
- 800 x 600 (480.000 pixels)
- 1024 x 768 (786.432 pixels)

A figura 5.22 apresenta o desempenho do algoritmo desenvolvido, em termos de redução de pixels em processamento.

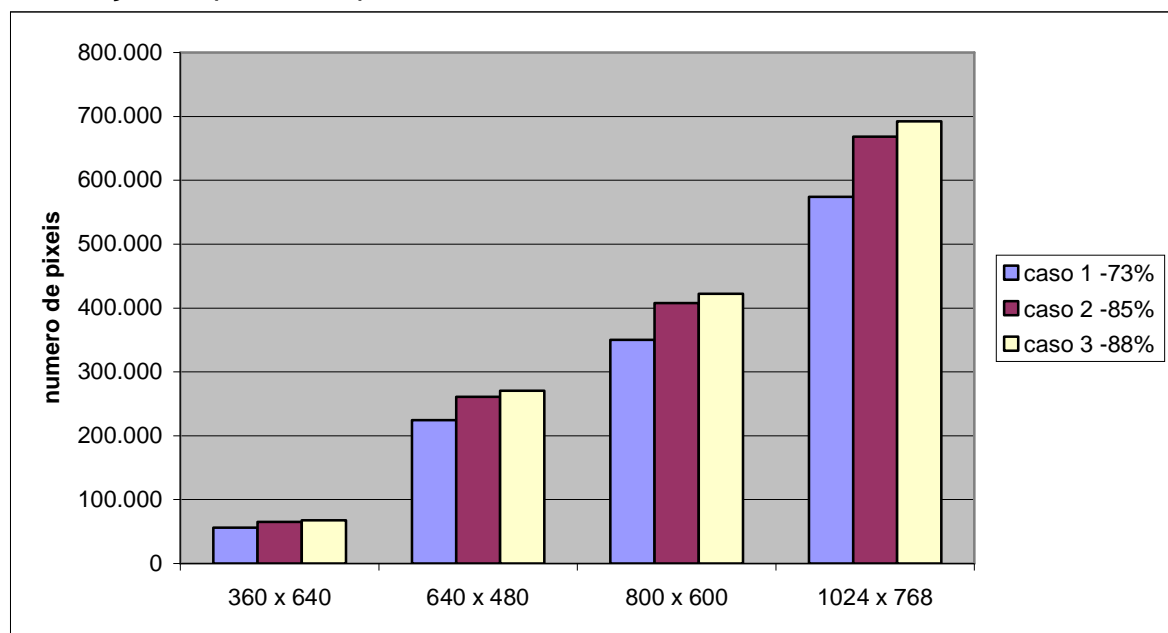


Fig. 5.22 A redução de pixels a ser processados nos 4 tipos de resolução.

Analisando a fig. 5.22, verifica-se que para a resolução mais “usual” (VGA) temos no melhor caso uma redução de cerca de 270.000 pixels e no pior caso uma redução de cerca 225.000 pixels.

Considerando a plataforma de maior resolução (XGA) temos no melhor caso uma redução de cerca de 690.000 pixels e no pior caso uma redução de cerca de 570.000 pixels.

Analisando apenas estes dois casos significativos, constata-se do enorme potencial do algoritmo desenvolvido na significativa redução de dados a processar, assim como da consequente redução substancial da carga computacional de processamento digital de imagem, em qualquer sistema onde este algoritmo seja implementado.

As significativas reduções na carga de processamento, revelam o importante contributo da aplicação do algoritmo desenvolvido para o processo de detecção do movimento da face nas variadas aplicações estudadas, bem como na potencial utilização futura de diferentes plataformas de processamento, com maior ou menor capacidade de processamento, com possíveis aplicações em sistemas dedicados embebidos, com microcontroladores dedicados, ou com o recurso a sistemas de lógica programável, onde é mais sensível a capacidade de processamento, com a redução do volume de dados a processar, e o consequente melhoramento nas velocidades de processamento.

A significativa redução do volume de dados a processar, obtida com as metodologias aqui desenvolvidas, permitem alargar a capacidade de utilização destes algoritmos a equipamentos que não fiquem dependentes de sistemas de arquiteturas de plataformas “monolíticas” (tipo PC), mas integradas em estratégias modulares de desenvolvimento de diferentes blocos, em software ou hardware, integrados noutros sistemas ou em funcionamento isolado (*standalone*), em arquitetura modular (tipo “*Lego*”), ou seja com a possibilidade de ser ou não, integradas em topologias variáveis com configurações desenhadas à medida como solução para funções específicas, em aplicações variadas desde a robótica móvel à domótica.

# Capítulo 6

## Visão por computador e robôs móveis

*Neste capítulo é apresentado o potencial da visão por computador aplicada à navegação de robôs móveis através da avaliação de movimento da face humana.*



## 6.1 Introdução

Como base para a implementação experimental do trabalho desenvolvimento utilizou-se um robô móvel com capacidade de implementação de condução automática, AGV – *Automated Guided Vehicle*.

Deste modo, será apresentada seguidamente uma breve introdução aos veículos de condução automática - AGVs.

### 6.1.1 Introdução histórica dos AGVs

O primeiro *Automated Guided Vehicle* (AGV) foi inventado em 1953. A ideia surgiu a um inventor americano o qual pensou na possibilidade de poder substituir o motorista dum tractor. A *Savant Automation*<sup>1</sup> implementou a invenção e o primeiro Sistema Automatizado de condução de Veículos Guiados (AGVs) o qual foi instalado como um tractor-reboque em 1954 na empresa *Mercury Motor Freight Company* na Columbia, Carolina do Sul, para operações de transporte de longa distância, na sua fábrica (Ullrich, 2015).

Os primeiros AGVs tinham pára-choques mecânicos de segurança e eram dirigidos por um fio simples colocado no solo (orientação indutiva activa) ou com um sensor óptico (Ullrich, 2015). Durante os primeiros anos, a grande procura desta tecnologia, principalmente pela indústria automóvel, promoveu o seu desenvolvimento a um ritmo acelerado. Os AGVs apresentavam várias vantagens significativas, nomeadamente na economia de custos, na maior eficiência e na redução de acidentes com os materiais transportados (Martínez-Barberá and Herrero-Pérez, 2010b) adicionalmente, também apresentavam maior flexibilidade, melhor utilização do espaço e um maior nível de segurança (Ganesharajah and Sriskandarajah, 1998).

No final da década de 80, do século XX, a indústria foi atingida por uma grave recessão, sensivelmente na mesma época era introduzida a Metodologia Lean aplicada à produção industrial. Esta filosofia, com origem na indústria automóvel japonesa, tinha como objectivos, aumentar a qualidade e reduzir os custos de fabricação. Na visão desta metodologia os AGVs eram inflexíveis e por outro lado a redução de custos não era compatível com os gastos em desenvolvimento de novas tecnologias dispendiosas. No final dos anos 90, porém, os avanços tecnológicos permitiram uma nova era dos AGVs, mais flexíveis e fiáveis. A capacidade de orientação electrónica dos AGVs, torna-se possível com o desenvolvimento de novos sensores sem contacto. A orientação indutiva activa já não era a única solução.

Novas tecnologias surgiram, como a navegação por laser. Também se tornou possível o controlo de AGVs através de um PC padrão (Ullrich, 2015). Esta nova era de desenvolvimento tecnológico dos AGVs vem-se mantendo a um ritmo acelerado até aos dias de hoje tornando actualmente estes veículos muito populares em ambientes industriais e com uma crescente penetração na domótica.

Os AGVs são hoje parte integrante da Indústria 4.0 e muitos fabricantes da indústria automóvel estão a aderir a esta nova tecnologia. Esta tecnologia também se expandiu para novos segmentos de mercado, como por exemplo o retalho. Em 2012 a *Amazon*, maior retalhista da Internet, nos EUA, adquiriu o produtor de AGVs, *Kiva Systems* e implantou 15000 AGVs em 10 dos seus armazéns, com o objectivo de reduzir o tempo de entrega e aumentar os níveis de serviço ao cliente (D'Andrea, 2012). É previsível no futuro a expansão desta tecnologia a novos domínios e o seu continuado desenvolvimento tecnológico.

### 6.1.2 Métodos de navegação dos AGVs

O AGV tem como função navegar entre diferentes locais, na sua área de operação, para poder cumprir as suas tarefas. Para realizar a sua função deve conhecer a sua própria posição no seu sistema de coordenadas fixo, o que é realizado através de diferentes métodos de navegação (Ullrich, 2015). É importante notar que existe uma diferença entre os termos posição e localização. A posição é um conjunto exacto de coordenadas que correspondem ao sistema de coordenadas global da área de operação. A localização contém mais informação. Por exemplo, informação sobre o que o AGV tem que fazer nessa posição, como parar, comunicar com o sistema de controlo de orientação ou realizar uma determinada tarefa (Ullrich, 2015). Os métodos de navegação para os AGVs podem ser classificados em dois grupos distintos: navegação por caminho fixo ou navegação por caminho aberto (Martínez-Barberá and Herrero-Pérez, 2010a).

Os dois grupos de métodos de navegação têm em comum a necessidade da utilização de um processo para determinar a sua posição. O cálculo da posição e da orientação actual do veículo é baseado na informação fornecida pelos sensores e determinado mediante a actualização das suas posição e orientação, medidas anteriormente. A selecção dos processos a utilizar no cálculo da posição é opção do método de navegação (Ullrich, 2015). A odometria, sendo um desses processos, é usualmente utilizado pela generalidade dos métodos de navegação. Esta técnica consiste na integração da informação incremental do movimento linear das rodas motrizes ao longo do tempo, ou seja, mede a rotação das rodas motrizes cujo raio é conhecido e determina a distância percorrida pelo veículo, a sua velocidade e a direcção do movimento. Assim, com estas informações, o AGV determina a sua posição actual relativa à sua posição inicial (Lee and Yang, 2012). Devido ao deslizamento das rodas e a alterações na dimensão do raio das rodas, provocado por exemplo, pela variação do peso de carga, este cálculo é impreciso (Ullrich, 2015).

A utilização de sensores do tipo *encoders* é necessário para a odometria. Ao realizar um movimento, o AGV compara a sua posição calculada com a posição pretendida, que foi previamente programada durante a configuração do

sistema e, deste modo, pode corrigir a posição calculada e recuperar a sua posição absoluta. Dependendo do método de navegação que é usado, os sensores utilizados serão diferentes, conforme seguidamente se descreve.

### **6.1.2.1 Navegação fixa**

Na navegação fixa, o AGV é guiado com a ajuda de orientações físicas colocadas sobre, ou sob, o plano de deslocamento. Assim poderemos ter orientação indutiva ou orientação óptica (Ullrich, 2015).

Para a navegação fixa, os caminhos são pré-definidos e a navegação torna-se, à partida, mais fácil, pois só um sensor é necessário para detectar o caminho (a directriz). As possíveis modificações ao caminho requerem mudanças físicas e a paragem do sistema (Martínez-Barberá and Herrero-Pérez, 2010a). Os locais de referência, no sistema, são implementações físicas, e exigem marcações especiais que contenham informações adicionais para o próprio sistema. As marcações podem ser tiras de metal, ímanes, fitas ou marcas coloridas para as quais o AGV está programado para identificar o que fazer quando a marcação é reconhecida (Ullrich, 2015). Pode também utilizar-se tecnologia mais avançada tal como etiquetas de identificação por radiofrequência (RFID *Rádio Frequency IDentification*) ou marcas visuais (ex.:código de barras) as quais podem referenciar diferentes instruções dependendo da tarefa ou da situação do tráfego (Shneier and Bostelman, 2015). Estas marcações funcionam como pontos de apoio para os sistemas com caminhos-guia físicos e são adicionados ao respectivo trajecto. Por exemplo, etiquetas RFID que fornecem informações ao AGV sobre como agir, num cruzamento, e que caminho seguir.

#### **6.1.2.1.1 Orientação indutiva**

A orientação indutiva funciona através da criação de um campo magnético ao longo do caminho guia. O AGV mede o fluxo magnético e verificando que se desvia do caminho-guia, corrige o movimento para voltar ao caminho (Lee and Yang, 2012). A orientação indutiva pode ser activa ou passiva, correspondendo a um caminho-guia com um campo magnético activo ou não. Com a orientação indutiva activa, um condutor de corrente eléctrica é colocado sob o piso (a alguns centímetros de profundidade) criando um campo magnético. Duas bobinas são montadas sob o AGV, as quais medem o desvio relativamente ao caminho-guia (Ullrich, 2015).

Na orientação indutiva passiva é utilizada uma fita magnética fixada ao chão. O AGV é equipado com dois ou três sensores, de campo magnético, que detectam a fita (fig. 6.1). O AGV pode, se necessário, navegar sem o caminho-guia com recurso ao método de cálculo da odometria, o que torna possível viajar entre diferentes caminhos e torna o AGV menos dependente da condição

da fita (Shneier and Bostelman, 2015). As duas técnicas têm diferentes vantagens. O condutor colocado sob o piso é mais robusto, já que o caminho-guia é protegido contra rasgos causados pelo tráfego e não é afectado por sujidade. No entanto, a alternativa passiva é mais flexível, pois é mais fácil modificar a trajectória (Sankari and Imtiaz, 2016). Tanto a navegação indutiva passiva como a activa não são adequadas na presença de piso metálico, pois isso afectará todo o campo magnético (Ullrich, 2015).

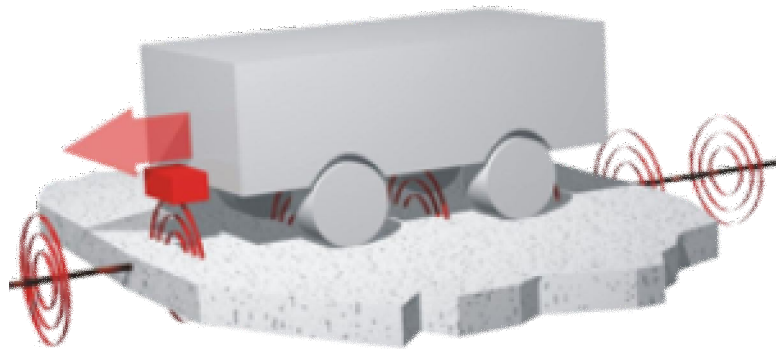


Fig. 6.1 - AGV com orientação indutiva (www.Goetting-agv.com, acedido em Maio de 2019)

#### 6.1.2.1.2 Orientação óptica

A orientação óptica funciona de forma semelhante à orientação indutiva passiva. Usualmente com uma câmara de vídeo apontada para o solo, o AGV segue o caminho-guia que poderá ser uma fita colorida ou uma linha pintada. A cor tem de contrastar com o chão de modo a ser facilmente detectada a aresta da linha, conforme ilustrado na figura 6.2. Como na navegação de indução passiva, a navegação óptica é flexível e é capaz de operar sem o caminho-guia. Também tem a vantagem de não ser afectada pela presença de metais no piso que perturbariam o campo magnético (Ullrich, 2015).

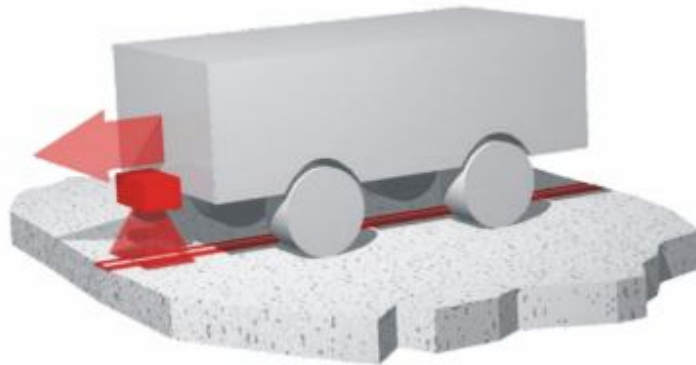


Fig. 6.2 - AGV com orientação optica (www.Goetting-agv.com, acedido em Maio de 2019)

### 6.1.2.2 Navegação aberta

A navegação aberta, também denominada de navegação livre (Ullrich, 2015), é um método de navegação sem caminhos-guia físicos. Os caminhos-guia são virtuais e quando são necessárias mudanças no caminho, é necessário reprogramar o AGV. Na realidade os AGVs de navegação livre podem tomar qualquer caminho desde que previamente planeado. A limitação de caminhos possíveis torna o sistema menos flexível, porém reduzem enormemente a sua complexidade, em termos de projecto e implementação (Digani et al., 2014).

#### 6.1.2.2.1 Navegação por marcas ou pontos de ancoragem

Atribui-se a designação de navegação por pontos de ancoragem a um grupo de métodos que usa a odometria e a orientação por marcas artificiais no chão. Os métodos de navegação por pontos de ancoragem são altamente dependentes da qualidade da unidade de odometria. Existem diferentes exemplos de navegação por pontos de ancoragem dependendo dos diversos tipos de marcas de orientação artificial. O de maior utilização é a navegação por marcas magnéticas (Ullrich, 2015).

##### 6.1.2.2.1.1 Navegação por marcas magnéticas

Uma alternativa aos ímãs permanentes passivos é a utilização de *transponders* quase-activos. Em alternativa aos ímãs passivos que geram um campo magnético, as marcas, ou pontos de ancoragem, são *transponders* quase-activos, os quais são activados por indução da unidade de leitura, localizada na

parte inferior do veículo. Isto permite que a precisão seja maior e capacita o AGV a ler informações adicionais de *layout*, para posicionamento, conferindo-lhe maior liberdade de movimentação (Ullrich, 2015).

#### **6.1.2.2.1.2 Navegação por marcas de radiofrequência**

A navegação por RFID é semelhante à navegação por marcas magnéticas, com a diferença de que estas são substituídas por etiquetas RFID. As etiquetas são lidas por um leitor RFID montado no AGV. Através da medição da intensidade do sinal recebido (*RSS Received Signal Strength*) e do conhecimento da localização das marcas na trajectória, pode ser determinada a posição exacta do AGV (Luimula et al., 2009). O sistema tem poucos custos e é simples de implementar, por outro lado é sensível às condições externas do ambiente (Lee et al., 2016).

#### **6.1.2.2.1.2 Navegação por marcas visuais**

Na navegação por marcas visuais, a marca pode ser de vários tipos, com maior ou menor complexidade, para que possa ser identificada e reconhecida. Existem inúmeros tipos de marcas visuais. Estas podem ir desde simples formas geométricas, até padrões mais complexos, a preto e branco, a cores ou até mesmo tridimensionais. Por exemplo o código de barras unidimensional ou os códigos bidimensionais tais como os de uso livre nomeadamente: o *Aztec Code*, o *DataMatrix*, o *MaxiCode*, o *PDF-417* e o *QR Code (Quick Response)*. Correntemente, as marcas com maior utilização industrial são o *DataMatrix* e o *QR Code*.

#### **6.1.2.2.2 Navegação por laser**

Na navegação por laser, o AGV toma referências de maneira semelhante a um barco. Na navegação são usadas marcas artificiais no terreno, que de forma semelhante à navegação marítima funcionam como faróis, ou marcas naturais que orientam a navegação do AGV pelo cálculo da distância às mesmas. As marcas artificiais devem portanto ser instaladas na área de operação do AGV (Ullrich, 2015).

#### 6.1.2.2.1 Navegação por laser artificial (laser com marcas artificiais)

Na navegação por laser artificial, um laser rotativo é colocado no topo do AGV, emitindo radiação laser que é reflectida nas superfícies rectro-reflectivas que são montadas, com visibilidade, no espaço de trabalho. Os feixes reflectidos para o AGV são então detectados. As coordenadas das marcas são adicionadas ao mapa da área de operação na fase de configuração do sistema. Quando a radiação laser é reflectida, o AGV é capaz de triangular a sua posição absoluta com base no conhecimento das coordenadas de cada reflector (ver figura 6.3). Pelo menos dois ou três marcas necessitam de ser visíveis para que o AGV se posicione (Ullrich, 2015). A navegação por laser é o método mais utilizado na navegação livre, uma vez que tem alta precisão e não apresenta restrições relativamente à qualidade do piso (Ullrich, 2015). As limitações deste método são os custos de equipamento (Lee e Yang, 2012) e o projecto do AGV, uma vez que a rotação do laser deve ter uma linha de visão, no espaço de trabalho, completamente desimpedida (Sobreira et al., 2016).

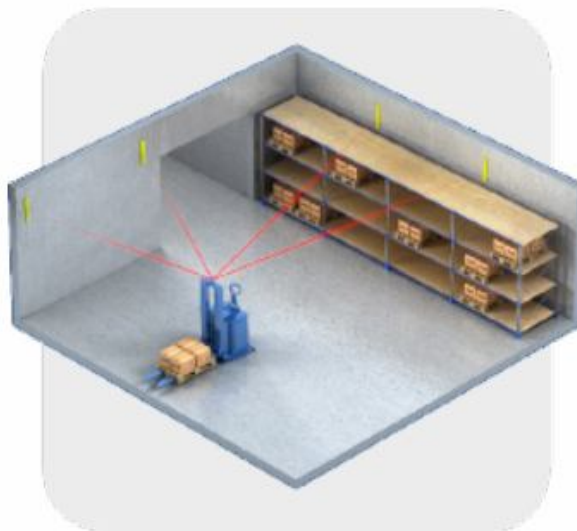


Fig. 6.3 - Navegação por laser artificial (NDC Solutions)

#### 6.1.2.2.2 Navegação por contorno (navegação a laser com marcas naturais)

Na navegação por contorno não são usados reflectores para determinar a posição do AGV. Em vez disso o AGV usa o ambiente já existente, no espaço navegável, tomando assim o nome de navegação por contorno. A vantagem desta técnica é não ser limitada pela necessidade da utilização de reflectores externos que necessitam ser calibrados e inseridos no sistema de controlo de navegação. Isto torna o sistema mais flexível do que a navegação por laser artificial. O tipo mais comum de navegação por contorno é a que utiliza um

laser de varrimento, que varre o ambiente e reconhece elementos estruturais (paredes, pilares, etc.) e outros itens fixos - figura 6.4 (Ullrich, 2015). A desvantagem desta metodologia é que a precisão e a robustez do sistema são inferiores às obtidas na navegação artificial por laser (Ullrich, 2015).



Fig. 6.4 - Navegação por contorno - navegação por laser com marcas naturais (NDC Solutions)

A navegação por contorno pode ser utilizada de diferentes formas. Uma alternativa é utilizar a medição de distância laser, a fim de conduzir o veículo ao longo das paredes, funcionando como uma ajuda a integrar com outros métodos de navegação. Por outro lado, o laser de varrimento ao poder mover-se ao longo do terceiro eixo, torna possível criar uma imagem 3D do ambiente, em comparação com a normal imagem a 2D (Shneier and Bostelman, 2015).

A navegação por contorno implica que o AGV tenha um mapa interno do ambiente, como referência. Este pode ser obtido de duas maneiras: pode ser pela digitalização manual feita por uma pessoa ou por uma técnica chamada SLAM (*Simultaneous Localization and Mapping*). Isto significa que o laser usado para navegação, também mapeia o ambiente desconhecido, medindo distâncias a diferentes objectos (Riisgaard e Blas, 2005). Ao usar o SLAM, o sistema torna-se mais flexível e mais fácil de utilizar em ambientes novos, ou dinâmicos (Klančar et al., 2017).

A tecnologia SLAM combina odometria, *scanners* a laser e filtros de Kalman estendidos (EKF *Extended Kalman Filter*). O AGV viaja pelo ambiente e detecta novos objectos com a ajuda do varrimento a laser. Pela odometria determina a sua posição. Tendo em conta que a odometria irá introduzir erros, surge a necessidade de utilizar o filtro EKF. O filtro EKF compara a posição



determinada pela odometria, com a forma como os objectos detectados se movem quando o robô está em movimento (Riisgaard and Blas, 2005) e, desta forma, é capaz de mapear a sua área operacional e calcular os caminhos possíveis, livres de colisão.

Refere-se finalmente que, nesta tecnologia, os materiais transparentes, como o vidro, não são detectados pelo laser (Zunino, 2002).

### **6.1.2.2.3 Navegação por sonar**

Existem métodos alternativos à utilização de sistemas laser na navegação. O sonar (ultrasons) tem menores custos, relativamente ao laser, mas a sua precisão é inferior. Para além disso é susceptível a perturbações induzidas por factores externos ambientais. A sua utilização é, por vezes, integrada com outros métodos, funcionando como sistema primário reactivo, na segurança a colisões (Zunino, 2002).

### **6.1.2.2.4 Navegação por visão com câmara**

A visão por câmara tem vantagens claras, relativamente a outras tecnologias, pois permite ao AGV o acesso a uma quantidade maior de informação. Com uma imagem colorida, ou em tons de cinzento, é possível identificar objectos e outros recursos no ambiente, que são vedados ao laser ou ao sonar (Zunino, 2002). A visão, com processamento de imagem, permite “interpretar” o ambiente de navegação e assim utilizar sistemas, com maior ou menor “inteligência” para uma navegação aberta, em ambientes complexos. A visão tem, no entanto, uma considerável limitação que resulta da enorme influência das condições de iluminação nos resultados do processamento da imagem, o que não acontece no laser ou no sonar (Zunino, 2002).

### **6.1.2.2.5 Navegação GPS**

A maioria dos métodos de navegação abertos usam marcas passivas (reflectores, imanes). A tecnologia activa, mais comumente usada em alternativa à odometria, é o sistema GPS (*Global Positioning System*) (Ullrich, 2015), onde satélites em órbita são usados para determinar a posição do receptor GPS. A limitação deste sistema de navegação é a necessidade do AGV estar em linha de vista com o céu, o que representa uma enorme restrição para a sua utilização em ambiente industrial (Ullrich, 2015). Uma maneira de ultrapassar este constrangimento é através da instalação de um radar de

posicionamento local (LPR *Local Positioning Radar*), que é utilizado em alternativa aos satélites, para determinar a posição do AGV (Ullrich, 2015). O problema do uso do LPR é que a precisão de posicionamento do AGV é diminuída, verificando-se valores comuns de precisão na ordem dos +/-10 cm. A navegação GPS é vantajosa, quando utilizada em espaços livres, onde a navegação a laser ou navegação de caminho fixo não é economicamente viável (Carlo et al., 2014).

## **6.2 Implementação do sistema**

Na implementação laboratorial das metodologias desenvolvidas nos capítulos anteriores recorreu-se à implementação de um robô móvel no qual foram desenvolvidas capacidades de condução AGV. O veículo AGV terá a capacidade de condução autónoma e condução remota com a utilização da face humana como elemento de comando.

### **6.2.1 Caracterização do AGV**

O AGV apresentado neste capítulo é composto por dois sistemas principais: a) o sistema motor e b) o sistema de processamento.

O sistema motor é a plataforma motorizada que transporta um computador com uma câmara de vídeo (PC-Slave) que processa a imagem e controla os motores.

O sistema de processamento é composto por dois computadores (PC-Master e PC-Slave) que estão conectados através de uma rede sem fios – Xbee (<https://www.digi.com/>).

O sistema motor é composto fisicamente por 5 componentes principais, conforme ilustrado na figura 6.5. Um desses componentes, a placa controladora do motor (<http://www.robot-electronics.co.uk/html/md49tech.htm>), é tecnicamente caracterizada nas figs. 6.6 e 6.7.

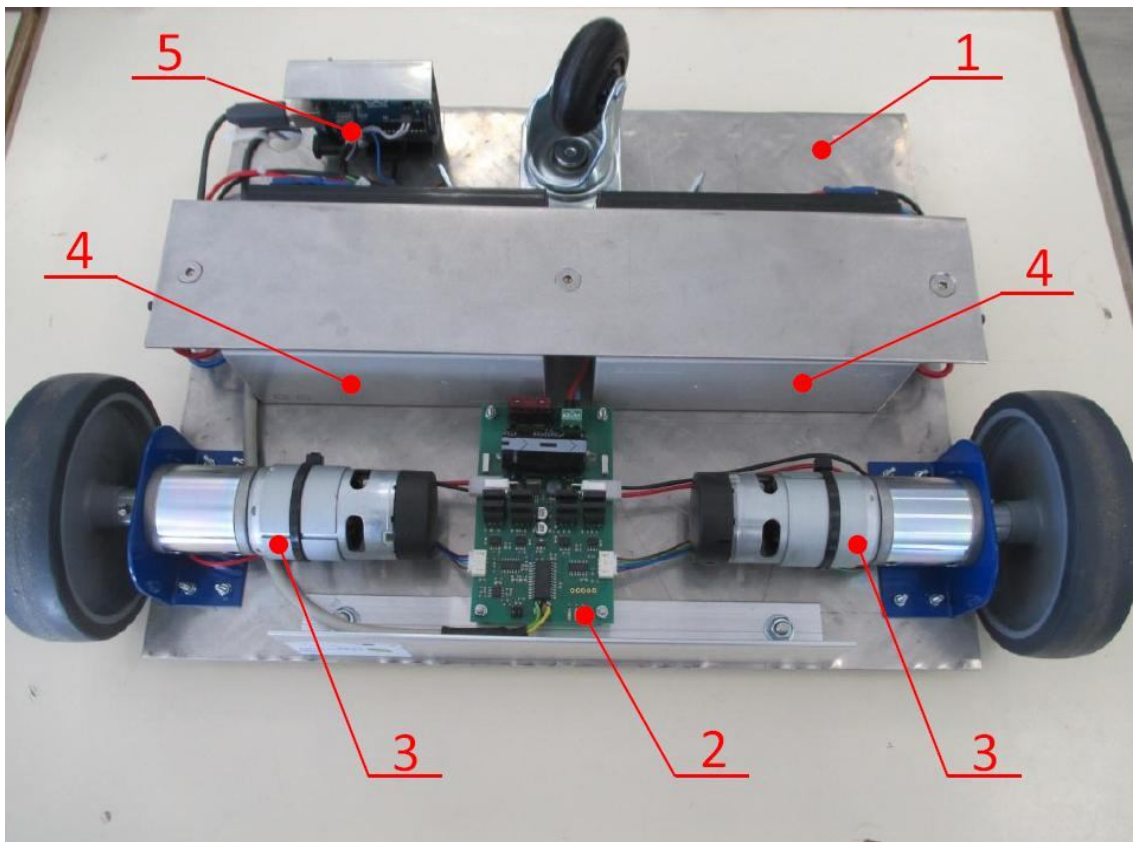


Fig. 6.5 Caracterização do Sistema Motor: 1-Estrutura metálica; 2-Placa controladora dos motores; 3-Motores DC; 4-Bateria 24V DC; 5-Controlador Arduino.

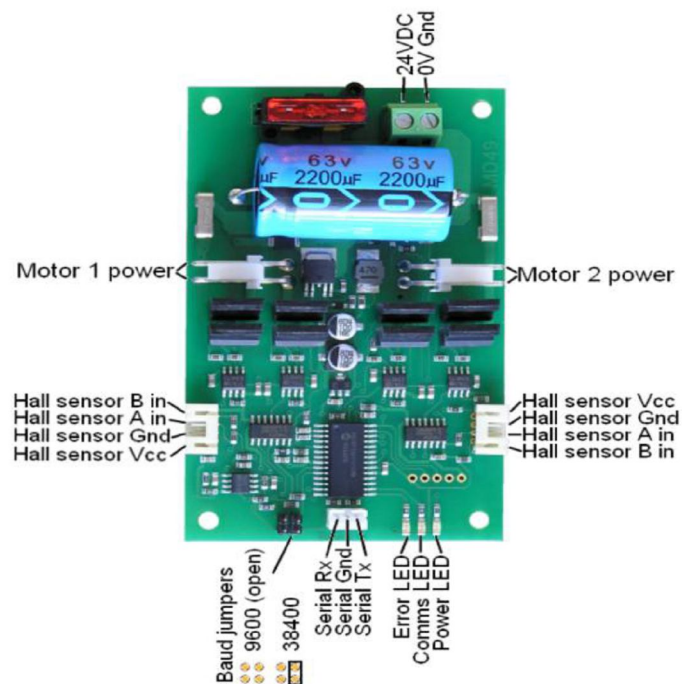


Fig. 6.6 Placa controladora dos motores  
([www.robot-electronics.co.uk/html/md49tech.htm](http://www.robot-electronics.co.uk/html/md49tech.htm), acedido em Maio de 2019).

command	Name	Bytes sent to MD49	Bytes returned by MD49	Description
0x21	<a href="#">GET SPEED 1</a>	2	1	returns the current requested speed of motor 1
0x22	<a href="#">GET SPEED 2</a>	2	1	returns the current requested speed of motor 2
0x23	<a href="#">GET ENCODER 1</a>	2	4	motor 1 encoder count, 4 bytes returned high byte first (signed)
0x24	<a href="#">GET ENCODER 2</a>	2	4	motor 2 encoder count, 4 bytes returned high byte first (signed)
0x25	<a href="#">GET ENCODERS</a>	2	8	returns 8 bytes - encoder1 count, encoder2 count
0x26	<a href="#">GET VOLTS</a>	2	1	returns the input battery voltage level
0x27	<a href="#">GET CURRENT 1</a>	2	1	returns the current drawn by motor 1
0x28	<a href="#">GET CURRENT 2</a>	2	1	returns the current drawn by motor 1
0x29	<a href="#">GET VERSION</a>	2	1	returns the MD49 software version
0x2A	<a href="#">GET ACCELERATION</a>	2	1	returns the current acceleration level
0x2B	<a href="#">GET MODE</a>	2	1	returns the currently selected mode
0x2C	<a href="#">GET VI</a>	2	3	returns battery volts, motor1 current and then motor2 current
0x2D	<a href="#">GET ERROR</a>	2	1	returns a byte within which the bits indicate errors on the MD49
0x31	<a href="#">SET SPEED 1</a>	3	0	set new speed1
0x32	<a href="#">SET SPEED 2 / TURN</a>	3	0	set new speed2 or turn
0x33	<a href="#">SET ACCELERATION</a>	3	0	set new acceleration
0x34	<a href="#">SET MODE</a>	3	0	set the mode
0x35	RESET ENCODERS	2	0	zero both of the encoder counts
0x36	DISABLE REGULATOR	2	0	power output not changed by encoder feedback
0x37	ENABLE REGULATOR	2	0	power output is regulated by encoder feedback
0x38	DISABLE TIMEOUT	2	0	MD49 will continuously output with no regular commands
0x39	ENABLE TIMEOUT	2	0	MD49 output will stop after 2 seconds without communication

Fig. 6.7 Conjunto de comandos para a placa controladora dos motores ([www.robot-electronics.co.uk/html/md49tech.htm](http://www.robot-electronics.co.uk/html/md49tech.htm), acessado em Maio 2019).

O sistema de processamento é composto fisicamente por 3 partes principais, conforme ilustrado na fig. 6.8:

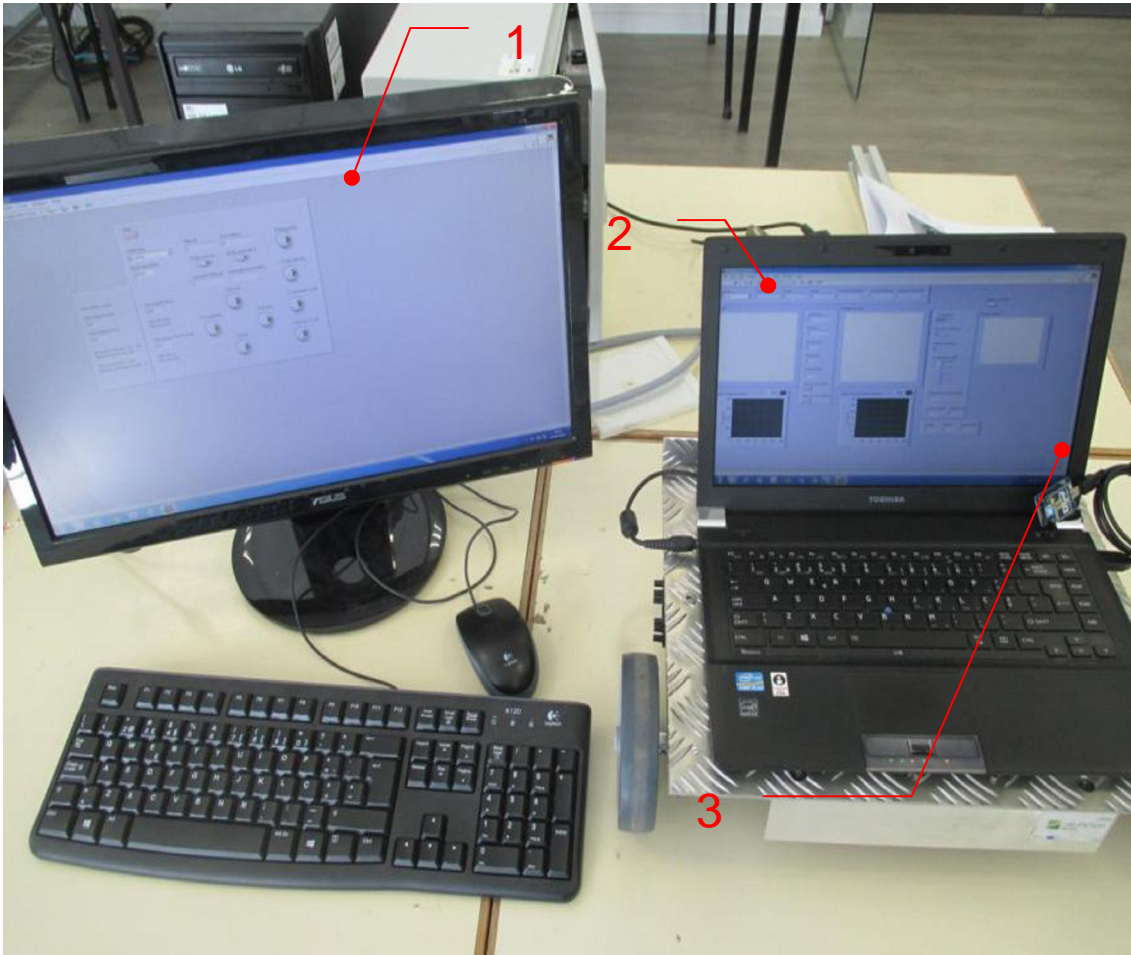


Fig. 6.8 Caracterização do sistema processamento  
1-PC-Master; 2-PC-Slave com câmara de vídeo; 3-Rede sem fios Xbee.

A estratégia de controlo do AGV é desenvolvida num conjunto de software industrial, hospedado no sistema de processamento. O *software* usado é o seguinte:

- NI LabVIEW (<http://www.ni.com>);
- NI VISA (<http://www.ni.com>);
- NI Vision Acquisition Software (<http://www.ni.com>);
- Drivers de Dispositivos da NI (<http://www.ni.com>);
- NI Vision Development Module (<http://www.ni.com>);
- Arduino Uno (<https://www.arduino.cc>).

## 6.2.2 A Estratégia de Controlo do AGV - *Master/ Slave*

A estratégia desenvolvida para o controlo de movimento AGV é baseada em três modos principais:

- I. **Modo manual** - que permite ao utilizador controlar o AGV, através do *PC-Master*, usando 4 comandos principais (mover para frente - mover para trás - virar à esquerda - virar à direita);
- II. **Modo automático** - que passa o controlo de movimento completo para o *PC-Slave*, e o AGV realiza automaticamente o seguimento de uma faixa preta marcada no chão, por processamento de imagem.
- III. **Modo de controlo remoto por face humana** - que permite ao utilizador controlar o AGV, através do movimento da sua face, em frente do *PC-Master*, ou de qualquer outra unidade que funcione como unidade remota de controlo. Também neste modo de controlo são utilizados os 4 comandos principais (mover para frente - mover para trás - virar à esquerda - virar à direita);

Na figura 6.9 ilustra-se a estratégia de controlo do AGV. Na figura 6.10 caracteriza-se o fluxo de comunicação entre o *PC-Master* e o *PC-Slave*.

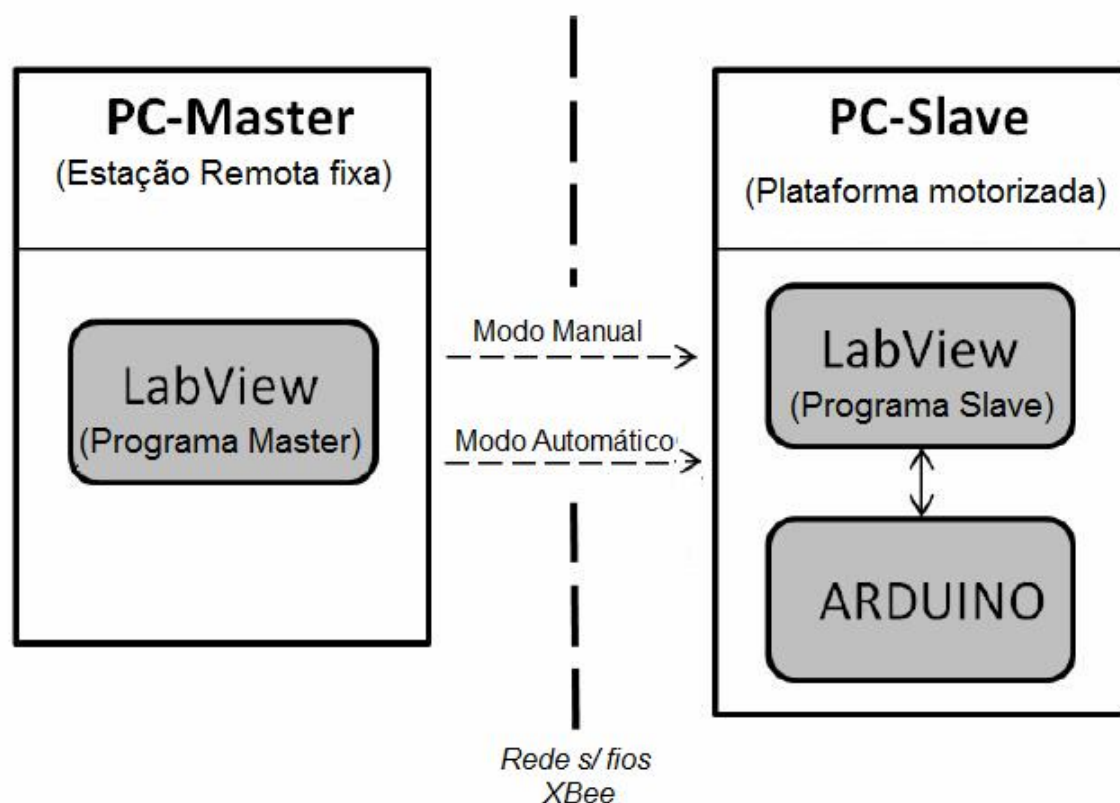


Fig. 6.9 Estratégia de controlo do AGV.

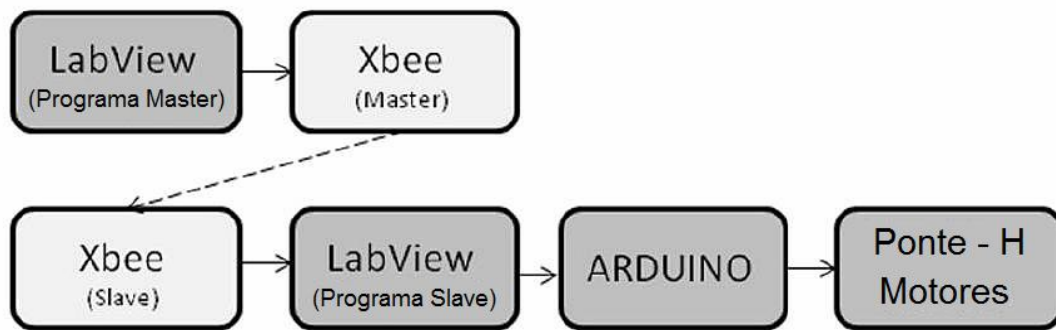


Fig. 6.10 Fluxo de comunicação entre o *PC-Master* e o *PC-Slave*.

### 6.2.2.1. Programa LabView do *PC-Master*

Este programa é executado no *PC-Master*, que é uma unidade remota, e é responsável pelo controlo principal do AGV. Basicamente tem três modos de comando, conforme já atrás referido em 6.2.2: i) Manual; ii) Automático e iii) Modo de controlo remoto por face humana.

A Interface gráfica de utilizador (GUI) desenvolvida para o *PC-Master* é ilustrada na fig. 6.11.

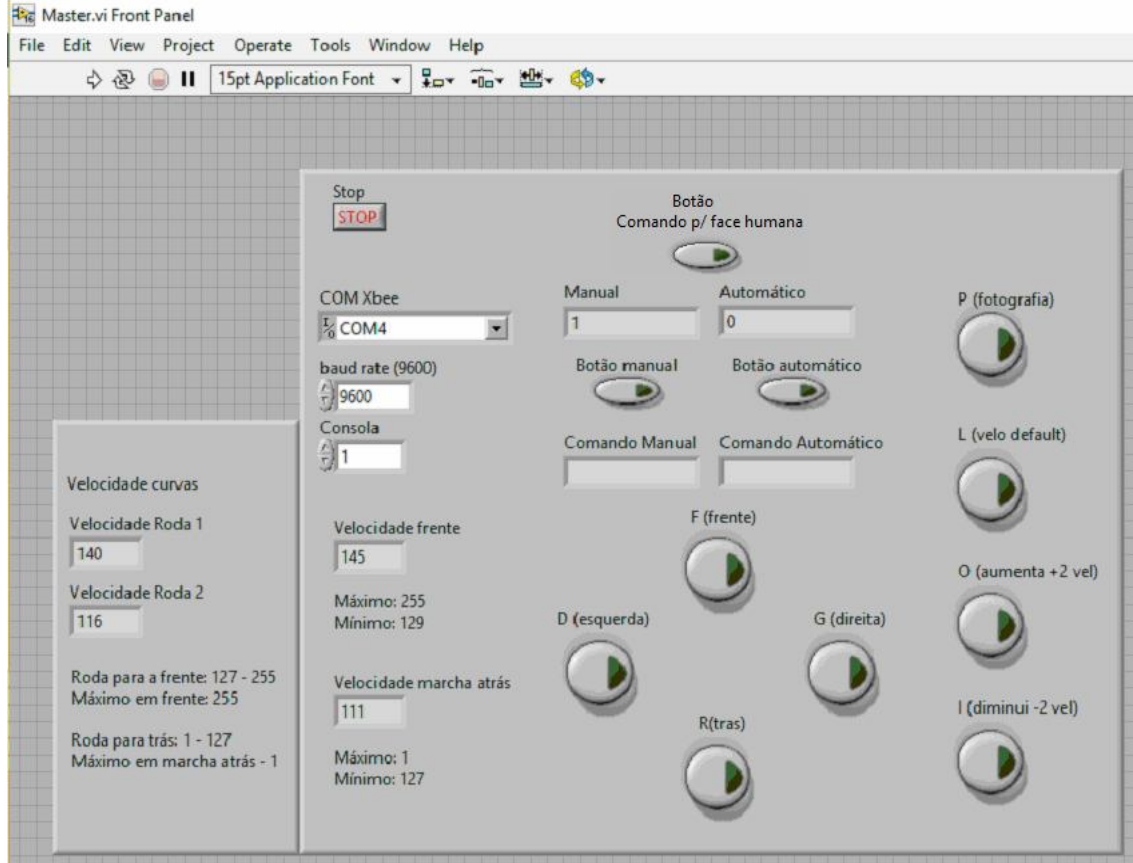


Fig. 6.11 Interface gráfica do utilizador do *PC-Master* (implementação em LabVIEW).

A estrutura base do programa desenvolvido é uma “Estrutura de Sequência Plana” composta por duas subestruturas. Este programa inicia a comunicação sem fios, Xbee, define as principais variáveis - velocidade padrão, movimento para frente, movimento para trás -, selecciona o Modo Manual (0/1), o Modo Automático (0/1) e ainda o modo de controlo remoto por face humana, que lança outra aplicação.

A Fig. 6.12 ilustra uma pequena parte do código LabView desenvolvido para a selecção do modo: Manual/ Automático.

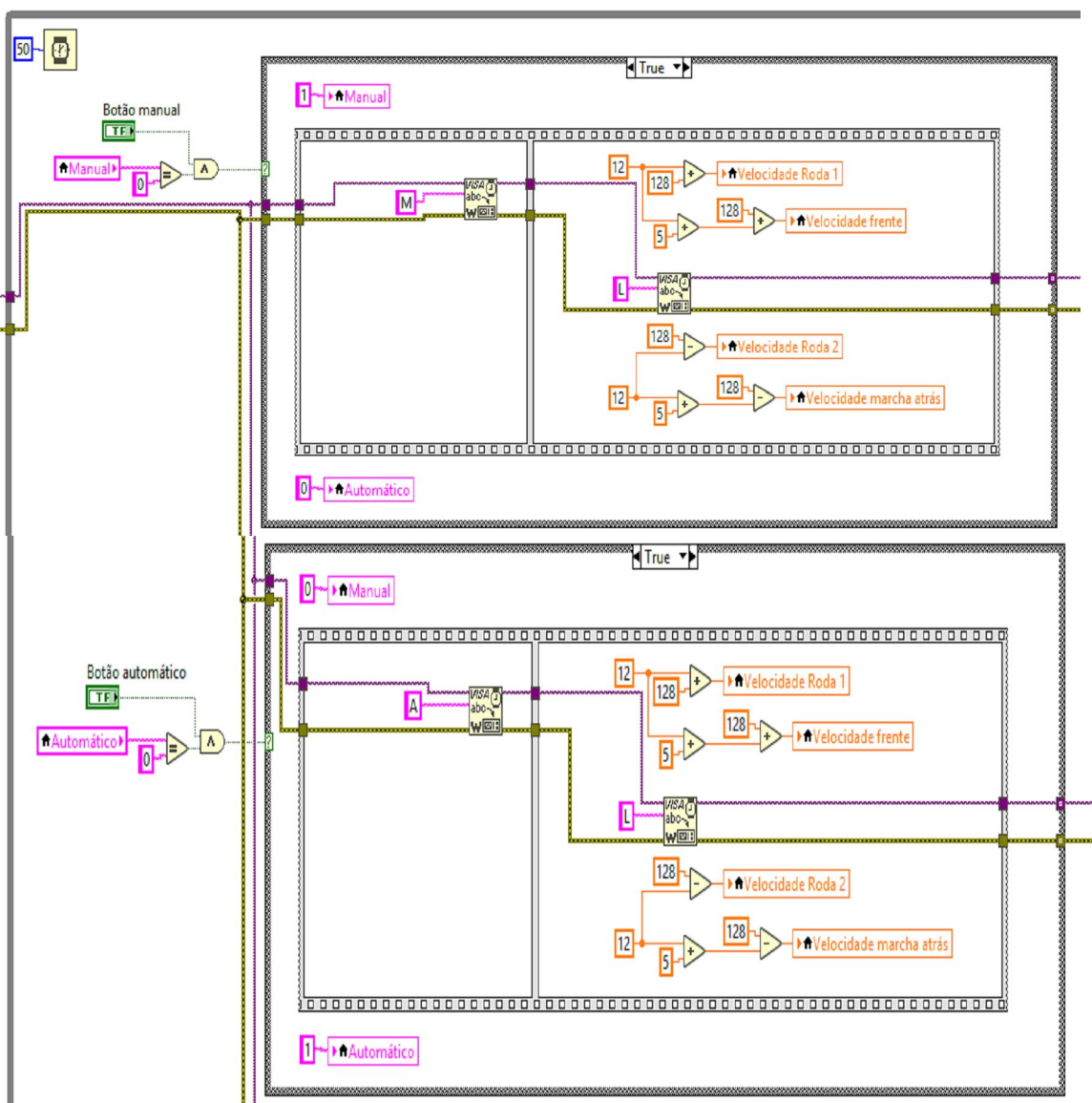


Fig. 6.12 LabView Master: selecção de modo manual / automático.

Tanto a aplicação Master, em modo Manual, como a aplicação do Modo de controlo remoto por face humana, ambas necessitam de comunicar entre o PC-



Master e o PC-Slave os comandos de movimento (mover para frente - mover para trás - virar à esquerda - virar à direita). Esta comunicação, conforme já atrás referido, utiliza a rede sem fios Xbee. Para a implementação desta capacidade de comunicação, foi desenvolvida a função apresentada na figura 6.13.

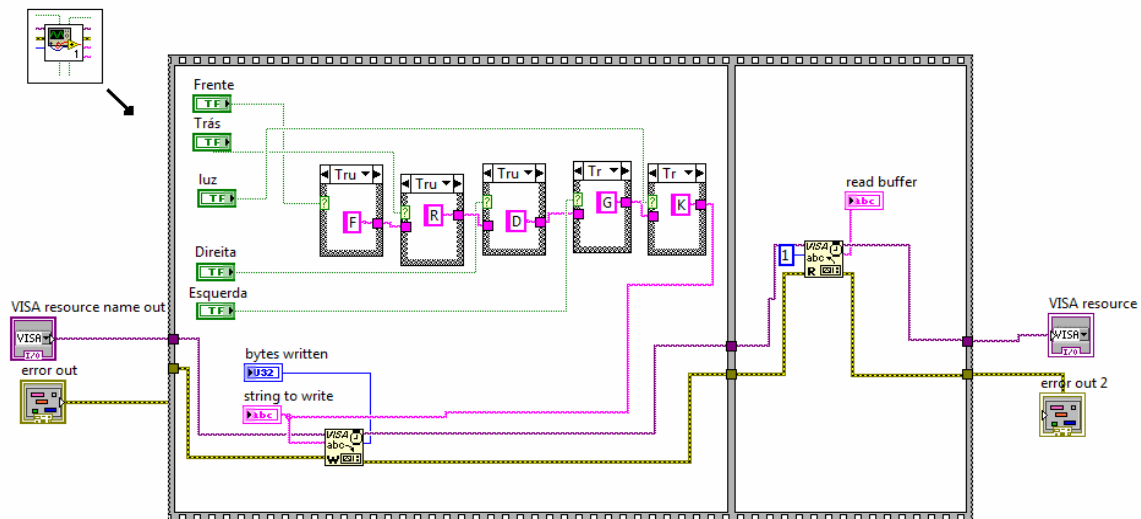


Fig. 6.13 LabView *Master*: função xbee para envio dos comandos de movimento ao AGV.

### 6.2.2.2. Programa LabView do *PC-Slave*

Este programa é executado no *PC-Slave*, que está localizado na plataforma do AGV, e que portanto se move solidariamente com o veículo. Este programa é directamente responsável pelo comando do movimento do AGV. Basicamente, comanda os motores do veículo, através da comunicação com o processador da placa Arduino, que por sua vez interage directamente com a placa controladora dos motores H-Bridge. Paralelamente, o programa *PC-Slave*, processa a imagem capturada pela câmara de vídeo-PC de modo a seguir automaticamente a faixa preta marcada no chão (Modo Automático). A Interface gráfica de utilizador (GUI) desenvolvida para o *PC-Slave* é ilustrada na fig. 6.14.

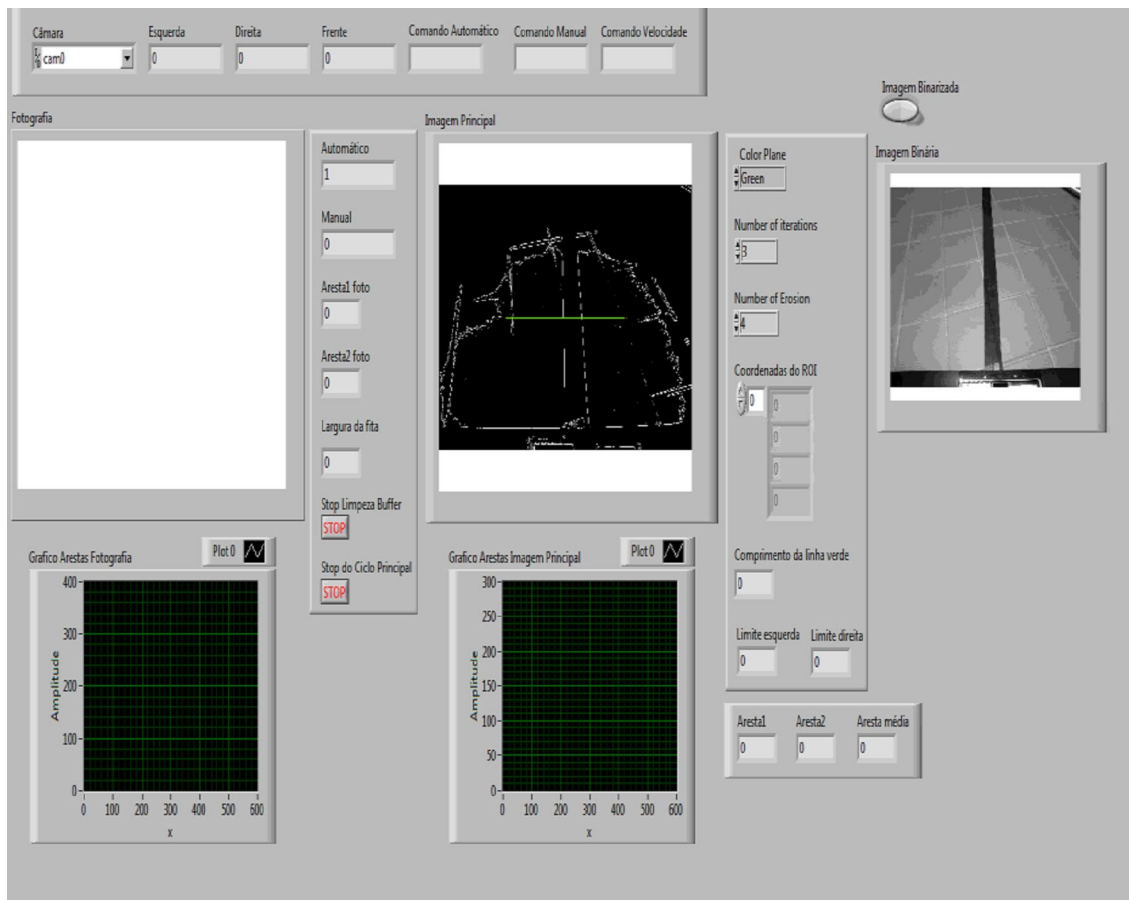


Fig. 6.14 Interface gráfica do utilizador do *PC-Slave* (implementado em LabVIEW).

A estrutura base do programa desenvolvido é uma “Estrutura de Sequência Plana” composta por três subestruturas. Este programa inicia a comunicação *Wireless Xbee*, a comunicação do Arduino, define as principais variáveis e processa as imagens capturadas.

A Fig. 6.15 ilustra uma pequena parte do código LabView desenvolvido para a inicialização das comunicações.

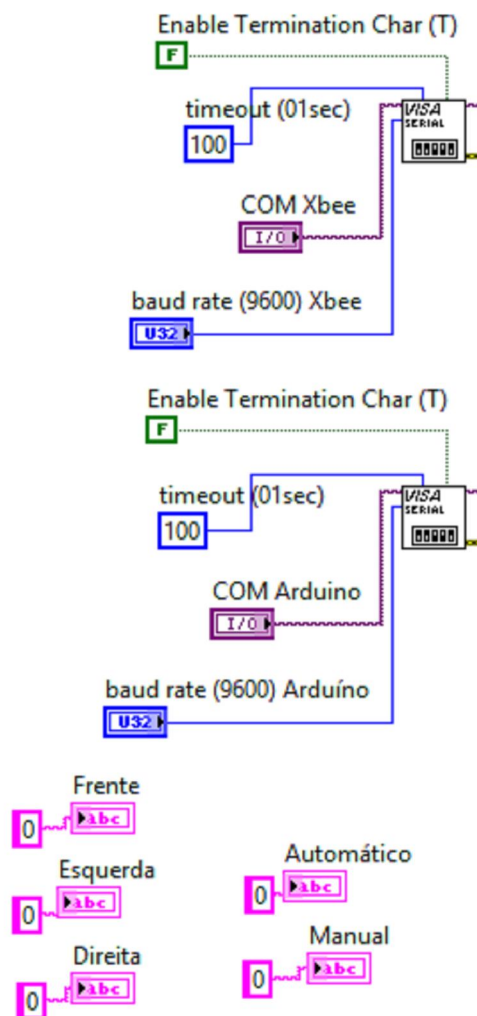


Fig. 6.15 Programa LabView - inicialização de comunicações *PC-Slave*.

### 6.2.2.3. Programa do Arduino

O processador Arduino é responsável pela comunicação série entre o programa LabView do *PC-Slave* e a placa H-Bridge que controla os dois motores DC que movimentam o AGV.

De acordo com as especificações da placa H-Bridge, controladora dos motores, (ver fig. 6.7) foi desenvolvido um programa para comandar os dois motores DC, de forma a permitir ao AGV executar os seus 4 movimentos principais: mover para frente, mover para trás, virar à esquerda, virar à direita.

A parte inicial deste programa, onde as variáveis são definidas de acordo com as especificações da H-Bridge, (ver fig. 6.7) é ilustrada na fig. 6.16.

A Fig. 6.17 exemplifica uma parte do programa em que é implementado o movimento para a frente.

```
#define CMD (byte)0x00
#define GET_SPEED1 0x21
#define GET_SPEED2 0x22
#define GET_ENC1 0x23
#define GET_ENC2 0x24
#define GET_ENCS 0x25
#define GET_VOLTS 0x26
#define GET_I1 0x27
#define GET_I2 0x28
#define GET_VER 0x29
#define GET_ACCEL 0x2A
#define GET_MODE 0x2B
#define GET_VI 0x2C
#define GET_ERROR 0x2D
#define SET_SPEED1 0x31
#define SET_SPEED2 0x32
#define SET_ACCEL 0x33
#define SET_MODE 0x34
#define RESET_ENCS 0x35
#define DISABLE_REG 0x36
#define ENABLE_REG 0x37
#define DISABLE_TIME 0x38
#define ENABLE_TIME 0x39
```

Fig. 6.16 Programa do Arduino: Definição de variáveis.

```
// Controlo e condução
else if (letra=='F' )
{
    Serial.println("Em frente");
    mySerial.write(CMD);
    mySerial.write(SET_ACCEL);
    mySerial.write(10);
    mySerial.write(CMD);
    mySerial.write(SET_SPEED1);
    mySerial.write(128+velocidade+5);
    mySerial.write(CMD);
    mySerial.write(SET_SPEED2);
    mySerial.write(128+velocidade+5);
}
}
```

Fig. 6.17 Programa do Arduino: Movimento para frente.

### 6.2.3 Processamento de imagem para seguimento automático de trajectória

Nesta secção descreve-se o processamento de imagem responsável pelo seguimento automático de trajectórias.

Sempre que é seleccionado o Modo Automático, na aplicação *Master*, o controlo de movimento do AGV é transferido para o programa LabView do *PC-Slave*. O objectivo deste processamento de imagem é gerar as acções necessárias para comandar os dois motores DC do AGV de tal forma que o veículo siga autonomamente uma faixa preta marcada no chão (ver fig. 6.18).



Fig. 6.18 Trajectória marcada no chão com uma faixa preta.

Para o desenvolvimento da aplicação de processamento de imagem foi utilizado um conjunto de funções que estão disponíveis no NI Vision Development Module [16].

Basicamente, em cada imagem capturada, são executadas sequencialmente as seguintes operações principais: i) captura de imagem; ii) conversão para imagem de 8 bits; iii) processamento de imagem – binarização; iv) erosão de imagem; v) detecção de contorno de imagem.

Estas transformações principais são usadas para identificar a linha preta marcada no chão. O critério adoptado para guiar o veículo ao longo dessa linha preta foi manter a localização relativa dessa linha entre duas linhas de fronteira virtuais - linhas esquerda e direita. É ao utilizador que cabe a definição da distância das linhas de fronteira virtuais à faixa preta central (ver fig. 6.19).

A fig. 6.20 ilustra parte do programa desenvolvido onde, na imagem já binarizada, são utilizadas funções de modificação de morfologia. Finalmente, a

figura 6.21 ilustra o AGV executando o seguimento de trajetória (faixa negra marcada no pavimento)

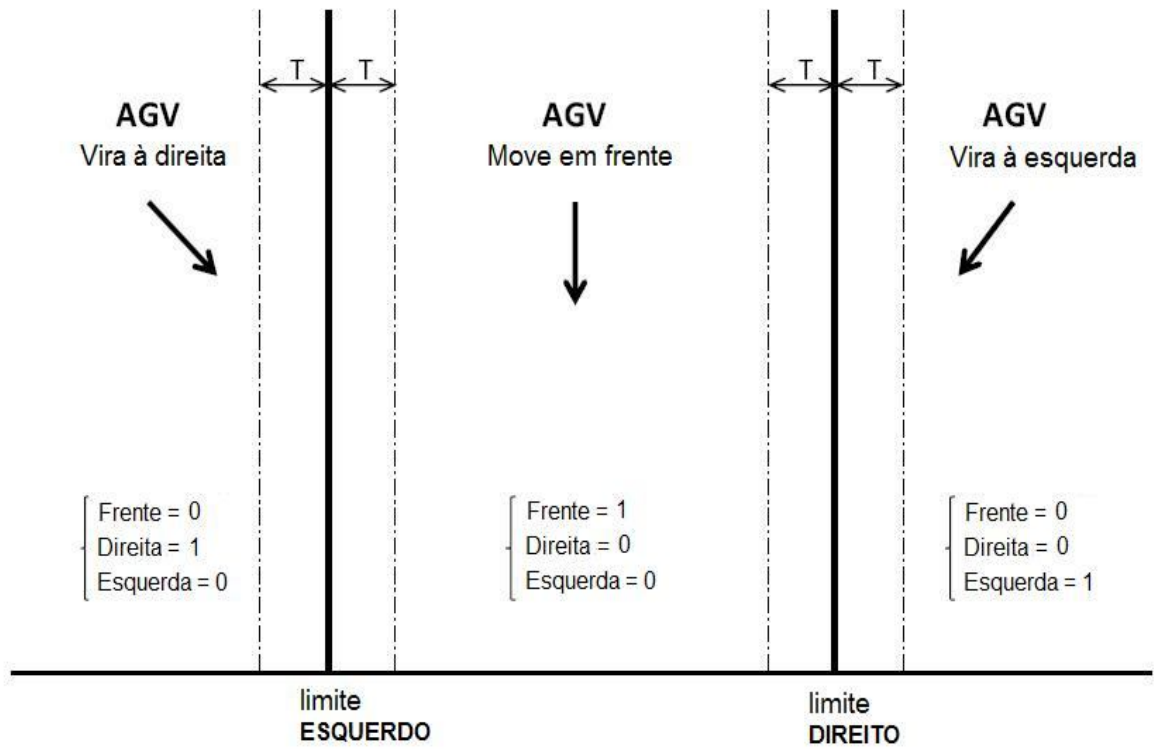


Fig. 6.19 Linhas de fronteira para seguimento de caminho AGV (T = tolerância).

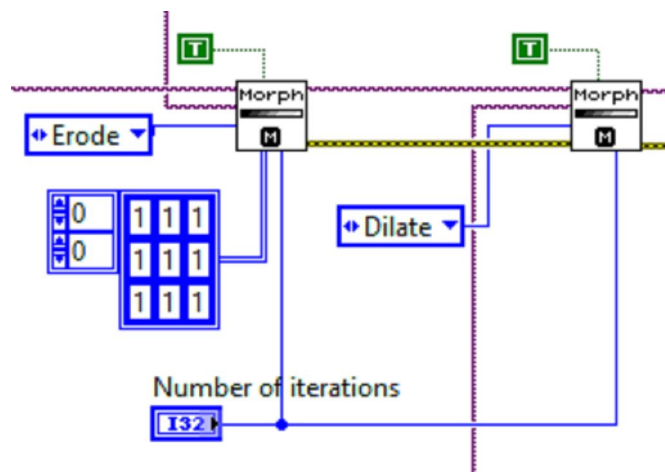


Fig. 6.20 - Processamento de morfologia da imagem.



Fig. 6.21 AGV executando seguimento de trajetória.

#### 6.2.4 Processamento de imagem para controlo do AGV por movimento da face humana

O processamento da imagem que é responsável pelo controlo remoto do AGV por movimento da face humana utiliza a metodologia já desenvolvida e apresentada no capítulo 5.

Quando o AGV está no modo de controlo remoto por face humana a aplicação do *PC-Master* entra em modo manual e em simultâneo inicia a aplicação de controlo por face humana, cuja interface gráfica de utilizador (GUI) é apresentada na figura 6.22.

Esta GUI será a interface de comando do AGV sempre que seja seleccionado o modo de controlo remoto por face humana, que enviará os 4 comandos principais (mover para frente, mover para trás, virar à esquerda, virar à direita) para aplicação *Slave*, a correr no AGV (ver 6.2.2.2), utilizando a rede Xbee.

Para se iniciar esta aplicação, deve posicionar-se o botão AGV em OFF, conforme visualizado na parte direita da fig. 6.22, de modo a proceder-se à calibração do sistema. O processo de calibração é efectuado, posicionando-se a cruz, visível na GUI (fig. 6.22), no centro da face, sendo sinalizada por um pequeno quadrado verde. Feita esta pequena calibração de posicionamento inicial da face humana o sistema está pronto a ser utilizado.

Para iniciar então o controlo do AGV por movimentação da face, deve posicionar-se o botão AGV para ON colocando o sistema em modo de navegação.

O utilizador ao mover a face, a aplicação irá calcular o vector da distância entre o centro definido no início da calibração e o centro da face nos diversos instantes de tempo sequenciais, conforme é ilustrado na figura 6.22, onde essa distância é mostrada simultaneamente em termos numéricos e em termos gráficos, com a amplitude do vector de deslocamento representada a amarelo. Na Fig. 6.22a é apresentado um pormenor da GUI ilustrada na Fig. 6.22, focando a representação do vector da distância calculada, nas suas componentes segundo os eixos coordenados x e y.

Finalmente, o algoritmo desenvolvido para controlar o movimento do AGV utiliza o valor calculado da distancia no eixo x (dx) e no eixo y (dy), comparando-os com valores limites Tx e Ty e seguindo os critérios de condução que vão determinar qual o movimento a ser realizado pelo veiculo, conforme ilustrado na fig. 6.23.

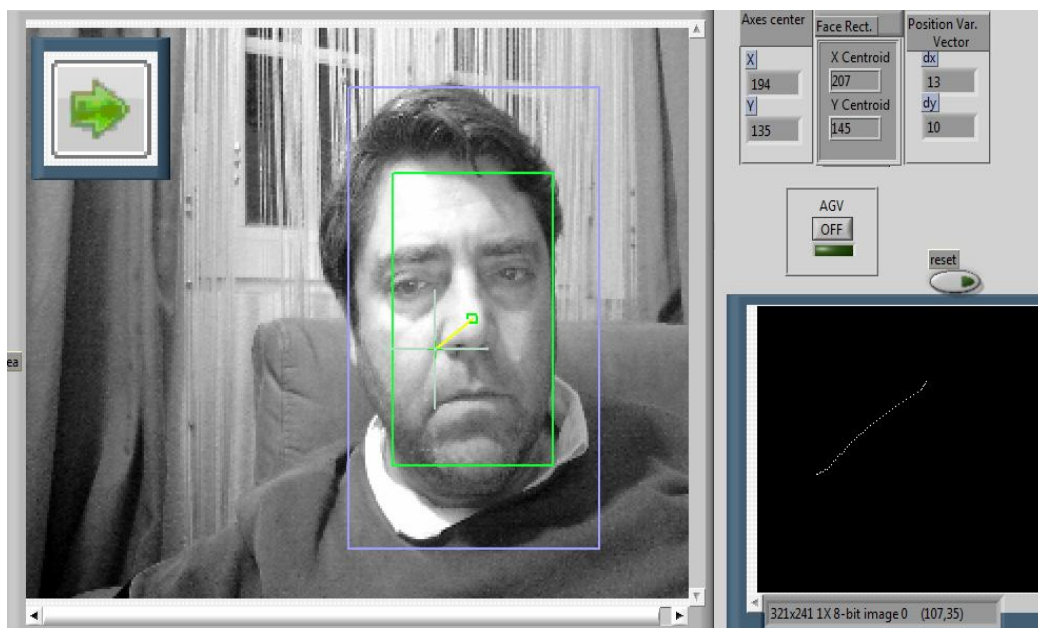


Fig. 6.22 Interface gráfica da aplicação do *PC-Master* para o controlo do AGV por movimento da face humana.

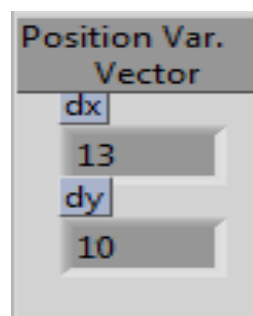


Fig. 6.22a. Pormenor do Valor do vector da distância calculada (detalhe da GUI da Fig.6.22).



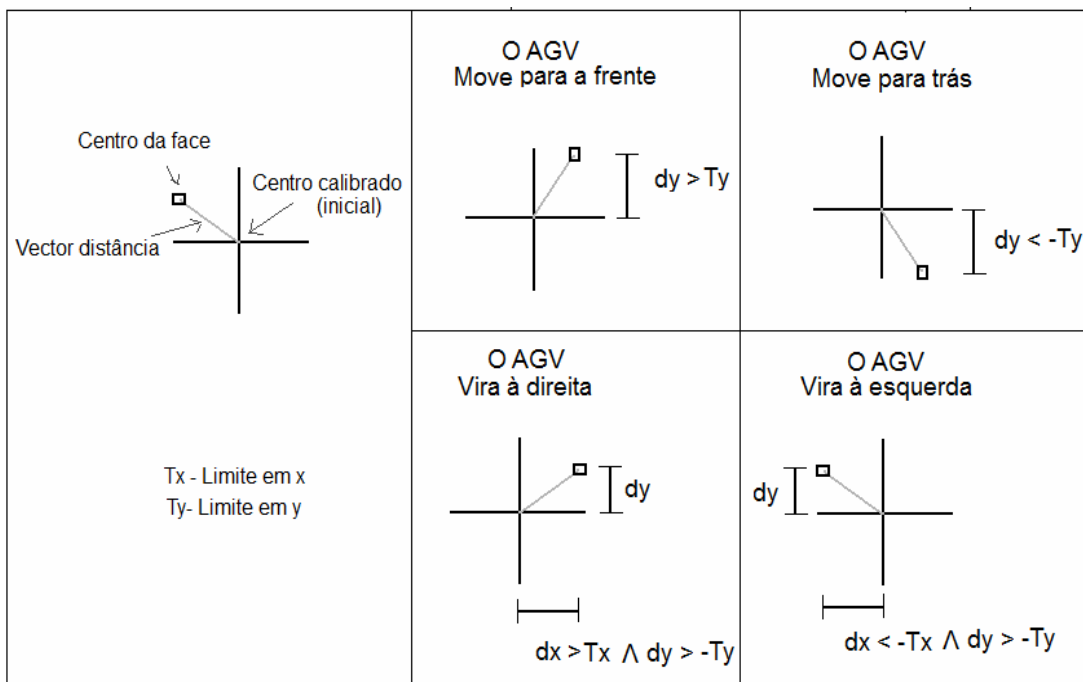


Fig. 6.23 Algoritmo para controlo do AGV (comparação com limites Tx e Ty)

Na fig. 6.24 mostra-se a parte do programa desenvolvido, para implementar a navegação do AGV usando o algoritmo de navegação.

Os valores escolhidos experimentalmente para os limites Tx e Ty tiveram em conta os aspectos práticos das velocidades relativas dos motores, no seu movimento rectilíneo, bem como no seu comportamento em curva, de modo ao AGV apresentar uma navegação regular e suave.

Finalmente na fig. 6.25 são apresentados os vários selectores alternativos da condução por face humana, cuja visualização está presente no canto superior esquerdo da GUI (fig. 6.22), recordando ao operador que tipo de condução está a comandar.

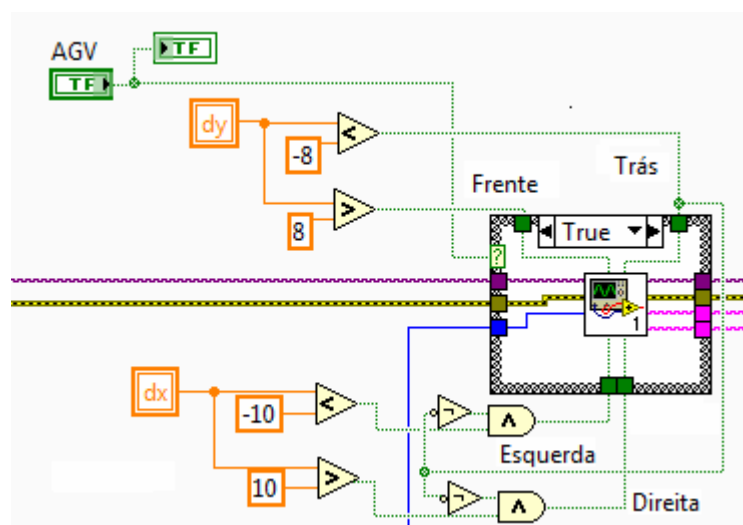


Fig. 6.24 Implementação do algoritmo de navegação (Tx=10 e Ty=8).



Fig. 6.25 Sentidos possíveis da condução na navegação.

### 6.3 Testes e Conclusões

Os testes experimentais realizados tiveram como objectivo avaliar o desempenho do AGV nos modos: i) automático e ii) controlo remoto por face humana.

No modo automático, mesmo tendo em conta que a navegação utiliza uma abordagem clássica no seguimento da fita, como orientação da trajectória, tem a característica interessante e inovadora da utilização da aquisição da imagem proveniente de uma câmara colocada no topo do AGV e não orientada para o solo, conforme verificado nos sistemas comerciais mais comuns. A localização da câmara no topo do AGV tem a grande vantagem da mesma poder ser utilizada para outras funções complementares do robô ou também no desenvolvimento de estratégias de navegação com algoritmos de integração de múltiplos sensores.

Ainda reportando-nos ao modo automático, o AGV mostrou um bom desempenho nos diversos testes *in-door* realizados, comprovando a adequabilidade da estratégia de controlo desenvolvida, nomeadamente mostrando robustez relativamente à mais séria limitação apontada à navegação por vídeo-câmara nomeadamente resolvendo a elevada sensibilidade às variações de iluminação na trajectória.

Relativamente ao modo de controlo remoto por face humana, os testes conduzidos *in-door* revelaram uma forma de navegação interessante, demonstrando flexibilidade e uma boa capacidade de adaptação. A estratégia de controlo desenvolvida com a acção directa sobre os 4 comandos principais (mover para frente, mover para trás, virar à esquerda, virar à direita) apresentou uma resposta suficientemente rápida de forma a responder à dinâmica do AGV em causa, verificando-se uma condução regular e suave, impedindo colisões com os vários obstáculos colocados no ambiente de navegação, o que também comprova a adequabilidade da rede sem fios seleccionada.

Por outro lado, o ênfase colocado ao longo de todo este trabalho na capacidade de redução dos dados a tratar no processamento de imagem, permitiu a obtenção dos resultados práticos anteriormente enunciados.

Abordando a terminologia utilizada nos capítulos anteriores, para a medição da capacidade de redução de dados para processamento, nos algoritmos desenvolvidos nos capítulos precedentes, relata-se, a título de exemplo, que num teste realizado com a duração de 15 segundos e com uma trajetória de cerca de 9 m, registou-se um factor  $Af_{\text{médio}} = 2$  durante 9 segundos e  $Af_{\text{médio}} = 8$  durante 6 segundos.

Deste modo, considerando uma taxa de 25 fps (*frames* por segundo) na aquisição da imagem com a resolução  $320 \times 240 = 76.800$  pixeis e com a duração de 15 segundos, o numero total de pixeis a tratar seria de  $2,88 \times 10^7$ .

Em face dos factores adaptativos obtidos e da sua correspondente duração, o numero de pixeis efectivamente processados foi de  $8,64 \times 10^6 + 1,44 \times 10^6 = 1,08 \times 10^7$ , ou seja, representa um redução absoluta de  $1,8 \times 10^7$  de pixéis, ou seja, 62% de redução na carga computacional associada ao processamento de imagem

Finalmente refere-se que a capacidade de condução de AGVs, através do movimento da face humana, em ambientes complexos revela-se uma solução interessante em campos muito diversificados, desde logo na domótica, acrescentando novas possibilidades ao utilizador e até possíveis soluções para utilizadores com algumas limitações físicas.

# Capítulo 7

## Conclusões e Trabalho Futuro

*Neste capítulo são apresentadas as conclusões assim como o trabalho futuro a desenvolver*

## 7.1 Conclusões

Nos modelos de processamento de imagem aplicados à domótica identificaram-se os seguintes temas fundamentais: a detecção de movimento de objectos, a detecção da face humana como elemento de comando, entre outras aplicações, e a capacidade de utilização de multi-fontes de aquisição e processamento de imagem, com diferentes condições de iluminação, não controladas, integradas num sistema completo, com diversas topologias, para aplicação preferencial na domótica e robótica

A questão da carga computacional, assim como a diminuição de dados a processar, foram questões amplamente verificadas e atingidas, pois nos algoritmos propostos, estas foram das principais características subjacentes no seu desenvolvimento.

Por outro lado, tendo em conta a trajectória de desenvolvimento futuro que se prevê para a visão artificial e processamento de imagem, nomeadamente nas aplicações domóticas, deu-se especialmente ênfase a outra importante característica, no desenvolvimento dos algoritmos, concretamente a necessidade de serem modulares e independentes dos sistemas de *software* de desenvolvimento e com a capacidade de serem implementados em blocos de *hardware* ou *software* conforme a necessidade da aplicação em causa.

Nos algoritmos desenvolvidos foi salvaguardada a capacidade dos mesmos serem executados em diferentes plataformas de processamento, com capacidade de processamento variável, com implementação possível em diversos sistemas, sistemas dedicados embebidos, com microcontroladores dedicados, ou com o recurso a sistemas de lógica programável, onde é mais sensível a capacidade de processamento (redução do volume de dados a processar e conseqüente melhoramento nas velocidades de processamento).

A utilização da capacidade de condução de AGVs, através do movimento da face humana, em ambientes complexos, revelou-se uma solução interessante na domótica assim como em vários outros campos, acrescentando novas possibilidades ao utilizador e até possíveis soluções, como no apoio a tarefas em ambientes específicos, na área da cooperação de robôs, ou no apoio a utilizadores com limitações físicas.

As 3 principais contribuições científicas que são descritas nos 3 métodos desenvolvidos, nos capítulos 3, 4, 5 e implementadas conjuntamente no capítulo 6, onde se apresenta a face humana como elemento de comando de um robô móvel, são:

- Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células, Cap 3;
- Detecção de pele em Imagens a cores por Método adaptativo, Cap. 4;

- Detecção da variação de posição da face humana por enquadramento de rectângulo áureo com Método adaptativo, Cap. 5.

### **7.1.1 Redução de Volume de dados - Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células**

No algoritmo desenvolvido no capítulo 3, a grande redução é verificada no volume de dados tratados. Efectivamente, os testes realizados mostram que, relativamente aos resultados obtidos, o número de pixels testados é reduzido em cerca de 60%, sem comprometer a função principal da detecção do movimento.

### **7.1.2 Redução de Complexidade - Detecção de movimento em imagens 2D por variação adaptativa da dimensão de células**

O volume de dados a tratar reflecte-se na carga computacional do processamento de imagem.

Relativamente ao algoritmo desenvolvido no capítulo 3, se analisarmos a sua complexidade poderemos concluir que:

na detecção de movimento por diferenciação entre imagens (*interframe differencing*) a cada pixel é aplicada a operação digital (ex.: XOR, Ou exclusivo) que apresenta um complexidade do tipo  $n$ . No algoritmo desenvolvido são adicionadas duas operações de multiplicação/divisão com complexidade do tipo  $n^2$ . No entanto, estas 2 operações só são realizadas uma vez por *frame* em comparação com a operação de lógica do algoritmo base (*interframe differencing*) que é aplicada a cada pixel, por *frame*, “sofrendo” assim de uma significativa redução, com a redução de pixels a tratar. Concretizando para as 2 resoluções típicas, obtém-se:

- VGA (720 x 480): considerando uma redução total de 207.360 pixels no algoritmo (60%), esta medida traduz-se numa redução da complexidade: redução líquida do numero de operações totais (207.356) = quantidade acrescida de operações ( $2^2$ ) – quantidade de operações reduzidas(207.360).
- Full HD/1080p (1920 x 1080:) considerando uma redução total de 1.244.160 pixels no algoritmo (60%), esta medida traduz-se numa redução da complexidade: redução líquida do numero de operações totais (1.244.154) = quantidade acrescida de operações ( $2^2$ ) – quantidade de operações reduzidas (1.244.160).

### **7.1.3 Avaliação da robustez - Detecção de pele em Imagens a cores por Método adaptativo**

No capítulo 4 a avaliação da robustez do novo algoritmo adaptativo desenvolvido, em relação às condições de iluminação externas forneceu importantes resultados.

Foi estudado o desempenho deste algoritmo adaptativo para três diferentes condições de iluminação:

- lâmpada fluorescente no tecto;
- lâmpada de halogéneo por trás;
- iluminação directa proveniente da luz do monitor do computador:

Os testes mostraram sempre melhores resultados na detecção do rosto, quando comparados com os resultados fornecidos sem o algoritmo adaptativo. O melhor desempenho é particularmente evidente quando as condições de iluminação são mais fracas.

A metodologia desenvolvida mostra ser uma boa solução na utilização de múltiplas fontes de aquisição de imagens nos diversos e complexos ambientes, comuns nas aplicações em domótica, onde não há controlo da iluminação, por parte do equipamento de aquisição de imagem.

### **7.1.4 Redução de Volume de dados e complexidade - Detecção de pele em Imagens a cores por Método adaptativo**

Nos testes realizados para os 3 cenários obtiveram-se 3 casos diferentes, tendo em conta o valor médio do factor  $A_f$ . Assim, verificou-se, para o caso mais desfavorável, (Vlr. Médio de  $A_f = 3,7$ ), um melhoramento traduzido numa redução do volume de dados de 73%.

Para o caso correspondente a Vlr. Médio de  $A_f = 6,5$ , a redução do volume de dados correspondente foi de 80%. Finalmente, e para o caso mais favorável (Vlr. Médio de  $A_f = 8,0$ ), verificou-se uma redução do volume de dados de 88%.

Se tivermos em conta 2 tipos de resolução (ver tabela 1.1):

- QVGA (*Quarter Video Graphics Array*),
- XGA (*eXtended Video Graphics Array*);

respectivamente

- 320 x 240 (76.800 pixeis),
- 1024 x 768 (786.432 pixeis)

Para a resolução QVGA temos, no melhor caso, uma redução de cerca de 67.500 pixels e no pior caso uma redução de cerca de 56.000 pixels.

Considerando a plataforma de maior resolução XGA temos, no melhor caso, uma redução de cerca de 692.000 pixels e no pior caso uma redução de cerca de 570.000 pixels.

Analisando os dois casos significativos, constata-se o enorme potencial do algoritmo desenvolvido na significativa redução de dados a processar, assim como da consequente redução substancial da carga computacional e por consequência da complexidade do processamento digital de imagem, no sistema onde o algoritmo seja implementado.

### **7.1.5 Utilização da face como elemento de comando**

A face como elemento de comando de forma isolada, ou integrada com outros elementos de comando, apresenta uma solução com grandes potencialidades em várias áreas, nomeadamente na domótica. No cap. 6 a utilização da face humana como elemento de controlo na navegação de um robô móvel em ambiente doméstico mostrou, nos testes, uma válida capacidade e, ao mesmo tempo, continua a verificar a premissa de redução do volume de dados tratados, em parte para não comprometer a resposta atempada do sistema, tendo em conta as características das diversas configurações de *hardware* dos veículos a implementar.

Finalmente há a referir que se verifica flexibilidade na solução desenvolvida aplicada a sistema com elementos heterogêneos, nas suas características, assim como na sua possível integração em sistema já existentes.

## **7.2 Trabalho Futuro**

Como trabalho futuro foram já apontados ao longo dos vários capítulos, desta tese, alguns caminhos de desenvolvimento potencial.

Porém, como trabalho de fundo, propomos a integração de um elemento de aquisição/processamento e comunicação autónomo em domótica (e não só), numa filosofia do tipo Internet das coisas (IoT).

Descrevendo resumidamente o conceito, em termos de *hardware*, teremos um processador, com uma placa tipo Raspberry Pi com o sistema operativo Linux a correr numa microSD, com um sensor de aquisição de imagem a cores, de 5 a 8 megapixels, com uma corrente de consumo inferior ou igual a 2A, com capacidade de comunicação via WiFi e Bluetooth.

Com esta plataforma, facilmente se poderá construir um sistema autónomo que apenas necessite de alimentação de 5 V DC (facilmente disponível numa instalação doméstica comum, com recurso a uma pequena fonte/adaptador 240 V AC /5 V DC).



Como conceitos tecnológicos nesta linha, encontra-se o Raspberry Pi Zero. Este pequeno equipamento (50x50x80 mm) é apresentado na Figura 7.1 e incorpora já o referido sensor sendo só necessário acrescentar a fonte/adaptador.

Para validar este primeiro conceito de teste foi construído um protótipo, no qual se desenvolveu uma aplicação escrita em Python com recurso à biblioteca de software aberto, para processamento de imagem, OpenCV.

A aplicação desenvolvida aplicou o algoritmo base de segmentação de cor para detectar pele, conforme é apresentado na figura 7.2.

A comunicação com o protótipo Raspberry Pi é feita numa rede WIFI, com acesso remoto via VNC (ver figura 7.3).

Descrevendo sucintamente esta primeira aplicação (versão v0) cujos resultados são apresentadas na figura 7.3, verifica-se no rectângulo vermelho da imagem (img2) a detecção da pele por segmentação (cor verde). Na janela do terminal apresenta-se o respectivo número de pixeis detectados. Os primeiros testes realizados mostram bons resultados, perspectivando um bom desenvolvimento do conceito definido.

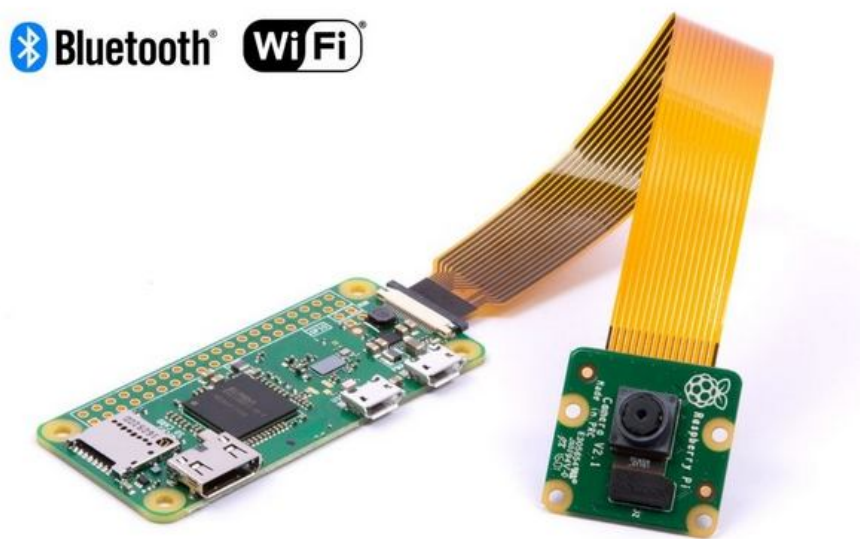


Fig. 7.1 O Raspberry Pi Zero e câmara ([www.raspberrypi.org](http://www.raspberrypi.org), acedido em Maio de 2018).

```
*camerasegf.py - /home/pi/camerasegf.py (2.7.9)*
File Edit Format Run Options Windows Help

import cv2
import numpy as np

from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2

camera= PiCamera()
camera.resolution = (320,240)
camera.framerate = 24
camera.brightness =55
rawCapture = PiRGBArray(camera, size=(320,240))

#tempo para camera acordar
time.sleep(0.1)
n=0
#captura frame
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array
    vermelhoverde_inferior = np.array([20,40,95])
    vermelhoverde_superior = np.array([255,255,255])
    repl = np.array([0,255,0])
    mascara = cv2.inRange(image, vermelhoverde_inferior, vermelhoverde_superior)
    for i in range(40,140):
        for j in range(120,220):
            if image[i,j,2] > image[i,j,0] and image[i,j,2] > image[i,j,1]
            and image[i,j,2] - image[i,j,0] < 100 and abs(image[i,j,2] - image[i,j,1]) > 5
                mascara[i,j]=1
                n=n+1
            else:
                mascara[i,j]=0
    res = cv2.bitwise_xor(image,repl,mask= mascara)
    image3 = cv2.rectangle(res,(120,40),(220,140),(0,0,255),1)

# means = cv2.mean(mascara)
print " number skin pixels", n

Ln: 29 Col: 24
```

Fig. 7.2 Aplicação em Python, versão 2.7.9.

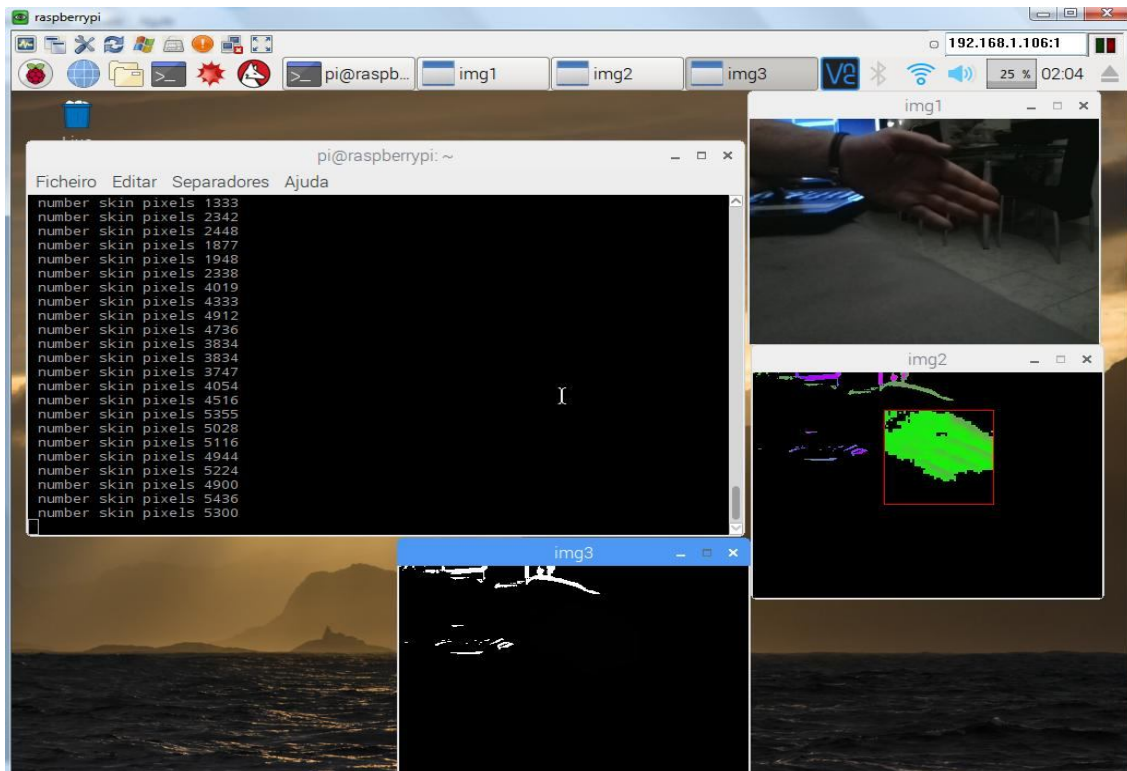


Fig. 7.3 Aplicação a correr no protótipo, com acesso via VNCviewer na rede-WIFI

Uma característica muito importante no projecto do elemento a desenvolver será a capacidade de comunicação com várias entidades externas, de forma autónoma, podendo explorar novos desafios com vários níveis de abstracção, como por exemplo, não havendo uma informação precisa da disposição/topologia dos vários elementos no ambiente de trabalho (doméstico, industrial urbano, etc), criar um mapa da disposição dos mesmos por triangulação, utilizando o elemento de comunicação WIFI e assim explorar a melhor forma de aquisição de múltiplas fontes, complementando essa informação, com a avaliação de distâncias, perspectivas e determinação precisa de caminhos e evoluções no ambiente em monitorização/ controlo.

Pensamos que este conceito tem um potencial de desenvolvimento extraordinário, no contexto da Internet das coisas (IoT), existindo a possibilidade de utilizar a capacidade de processamento de imagem autónoma do elemento, ou integrada num sistema distribuído (constituído por vários elementos), permitindo paralelamente a possibilidade de utilização de recursos externos, localizados remotamente (ex.: bases dados de acesso gratuito, elementos de processamento disponibilizados em rede - seja Internet -, “nuvem” pública, “nuvem privada”, etc).

Nesta vertente de aplicação domótica, cada um destes elementos será utilizado em pontos fixos nos edifícios e também como elementos móveis em elementos tipo AGVs, que têm a liberdade de se mover nos vários locais dos edifícios. Assim, com a capacidade de localização de cada elemento, com a capacidade de referenciação das imagens colectadas e processadas, como a detecção e identificação da face humana, para elemento de controlo, podem ser exploradas inúmeras oportunidades num contexto altamente complexo e dinâmico (Frank, et al, 2017). Nomeadamente a utilização de sistemas de inteligência artificial com capacidades de análise tendo em conta as várias posições relativas dos indivíduos nos locais com referências espacial e temporal, para obtenção das mais variadas informações que alimentam sistemas de decisão complexos, de uma forma, algo semelhante à fusão sensorial utilizada na robótica móvel.

Uma outra possível linha de investigação futura, decorrente do trabalho realizado, prende-se com a utilização de AGVs em actividades de apoio aos indivíduos, de forma semelhante à abordagem em ambiente industrial, mas neste caso, na realização de tarefas domésticas (Herrera et al, 2016; Sakamoto e tal, 2016), de forma autónoma ou numa lógica de grupo, numa abordagem cooperativa, por forma a solucionar problemas e realizar tarefas, que um elemento isolado não tenha capacidade suficiente, necessitando da abordagem cooperativa. Esta abordagem segue de perto o que vem acontecendo em vários domínios da robótica móvel, fora do ambiente doméstico, como por exemplo no futebol robótico (Pagello et al, 2006; Frank, et al, 2017).

Finalmente, propomos abordar futuramente a gestão e controlo de sistemas que incluam elementos robóticos (fixos ou móveis) que interajam com seres humanos, seja numa forma cooperativa, seja na resolução de tarefas automáticas, com capacidade de interacção, a cada momento, com diversos pontos heterogéneos de aquisição e controlo, em ambiente doméstico. Pretende implementar-se a integração de funções do tipo reconhecimento de expressões e/ ou emoções do ser humano, presentemente já utilizadas noutros sistemas, mas com grande campo de evolução no futuro (Zhang et al, 2016; Chen et al, 2018) numa óptica de integração tecnológica crescente, tirando partido das altas taxas de transmissão de dados das novas gerações de redes de interligação inter-elementos e entre os elementos e os centros de processamento/ armazenamento distribuído de dados, permitindo a produção de informação, a vários níveis, nos mais diversos campos da actividade humana, em ambiente doméstico.



## Referências Bibliográficas

Abrantes, S.A. (2000) *Processamento Adaptativo de Sinais*, Fundação Calouste Gulbenkian.

Albiol A., Torres L., Bouman C. A. and Delp E. J. (2000) 'A simple and efficient face detection algorithm for video database applications' *International Conference on Image Processing*, Vancouver, BC, Canada.

Agarwal, A., Issac, A., Dutta, M., Riha, K. and Uher, V. (2017) 'Automated skin lesion segmentation using K-Means clustering from digital dermoscopic images', *Telecommunications and Signal Processing (TSP) 2017 40th International Conference on*, pp. 743-748.

Ahmadi, Z. and Sobhani, N. (2014) 'Arbitration Management with Using Artificial Intelligence Technology (the sample: Goal-line Technology in Football)', *the Russian Federation Modeling of Artificial Intelligence 2014 Vol. 3, Issue 2*, pp. 48-58.

Ahmed, A. and Terada, K. (2012) 'A fast approach for person detection and tracking', *Intl. Journal of Computer Applications in Technology*, Vol. 44, Issue 3, pp.210–216.

Alexandre, L.; Campilho, A. (1998) 'A 2D Image Motion Detection Method Using a Stationary Camera', *RECPAD98*.

Angelopoulou, E., Molana, R. and Daniilidis, K (2001) 'Multispectral skin color modeling', *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol .2, pp. 635–642, IEEE.

Bartlett, M.S., Movellan, J.R. and Sejnowski, T.J.(2002) 'Face recognition by independent component analysis',*IEEE Transactions on Neural Networks*, Vol. 13, pp. 1450-1464. IEEE.

Bearden, C. and Thompson, P. (2017) 'Emerging Global Initiatives in Neurogenetics: The Enhancing Neuroimaging Genetics through Meta-analysis (ENIGMA) Consortium.', *Neuron*, Vol. 94, Issue 2, pp 232-236, Elsevier.

Bhimani, J., Yang, Z., Leeser, M. and Mi, N. (2017) 'Accelerating big data applications using lightweight virtualization framework on enterprise cloud', *2017 IEEE High Performance Extreme Computing Conference (HPEC)*, IEEE.

Brasil, R and Machado, A. (2017) 'Automatic Detection of Red Light Running Using Vehicular Cameras', *IEEE Latin America Transactions*, Vol. 15, Issue 1, pp. 81 – 86, IEEE.

Brown, R., Hann, C. and Chase, G. (2010) 'Vision-based 3D surface motion capture for the DIET breast cancer screening system', *Intl. Journal of Computer Applications in Technology*, Vol. 39, Issue 1–3, pp. 72–78.

Caetano T. S. and Barone D. A. C. (2001) 'A probabilistic model for the human skin color', *Proceedings. 11<sup>th</sup> International Conference on Image Analysis and Processing*, pp. 279 – 283.

Carlo, H., Vis, I. and Roodbergen, K. (2014) 'Transport operations in container terminals: Literature overview, trends, research directions and classification scheme', *European Journal of Operational Research*, Vol. 236, Issue 1, pp.1-13, Elsevier.

Chai, D. and Ngan, K.N. (1999) 'Face segmentation using skin-color map in videophone applications', *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, pp. 551–564, IEEE.

Chang, T., Comandur, B., Johnny Park and Kak, A. (2016) 'A variance-based Bayesian framework for improving Land-Cover classification through wide-area learning from large geographic regions', *Computer Vision and Image Understanding*, Vol.147, pp. 3-22

Chen, G., Wang, W.W. and Zhu, Q. P. (2008) 'A Face Detector Based on Statistical Estimation for Background'. *First International Conference on Intelligent Networks and Intelligent Systems*, pp. 548–551.

Chen, H., Ma, S., Sun, B. and Shen, Y. (2012) 'Projection ray intersection location-based multicolour pseudo-random coded projected active vision method', *Intl. Journal of Computer Applications in Technology*, Vol. 43, Issue 1, pp. 21–28.

Chen, L., Zhou, M., Wu, M., She, J., Liu, Z., Dong, F. and Hirota, K. (2018) 'Three-Layer Weighted Fuzzy Support Vector Regression for Emotional Intention Understanding in Human–Robot Interaction', *IEEE Transactions on Fuzzy Systems*, Vol. 26, Issue 5, pp. 2524 – 2538, IEEE.

Chen, X. and Lin, X. (2014) 'Big Data Deep Learning: Challenges and Perspectives', *IEEE Access*, Vol. 2, pp. 514 – 525.

Chen, Y., Argentinis, E. and Weber, G. (2016) 'IBM Watson: How Cognitive Computing Can Be Applied to Big Data Challenges in Life Sciences Research', *Clinical Therapeutics*, Vol. 38, Issue 4, pp. 688-701, Elsevier.

Chuan-Xian R., Zhen L., Dao-Qing D., and Stan Z. L. (2016) 'Enhanced Local Gradient Order Features and Discriminant Analysis for Face Recognition', *IEEE Trans. Cybernetics.*, vol.46, pp. 2656–2669, IEEE.

Codella, N. (2016) 'Identifying skin cancer with computer vision', Cognitive Computing, Healthcare, Thomas J Watson Research Center.

Correia, A., Miranda, L. and Hornung, H. (2013) 'Gesture-Based Interaction in Domotic Environments: State of the Art and HCI Framework Inspired by the Diversity', 14th International Conference on Human-Computer Interaction (INTERACT, Cape Town, South Africa.,pp.300-317, Springer.

Cui, J., Zha, H., Zhao, H. and Shibasaki, R. (2008) 'Multi-modal tracking of people using laser scanners and video camera', Image and Vision Computing, Vol. 26, pp. 240–252.

D'Andrea, R. (2012) 'A Revolution in the Warehouse: A Retrospective on Kiva Systems and the Grand Challenges Ahead', IEEE Transactions on Automation Science and Engineering, Vol. 9, Issue 4, pp.638-639, IEEE.

Digani, V., Caramaschi, F., Sabbatini, L., Secchi, C. and Fantuzzi, C. (2014), 'Obstacle avoidance for industrial AGVs', 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), IEEE.

Du, C. and Sun, D. (2008) 'Multi-classification of pizza using computer vision and support vector machine', Journal of Food Engineering, Vol. 86, pp.234–242.

Erhan, D., Szegedy, C., Toshev, A. and Anguelov, D. (2014) 'Scalable object detection using deep neural networks', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp. 2147-2154, IEEE.

Erra, U., Malandrino, D. and Pepe, L. (2017) 'A Methodological Evaluation of Natural User Interfaces for Immersive 3D Graph Explorations', Journal of Visual Languages & Computing, Vol. 44, Elsevier.

Faraji, M. and Qi, X., (2014) 'Face recognition under varying illumination based on adaptive homomorphic eight local directional patterns', IET Computer Vision, Vol. 9, Issue 2, pp. 390-399

Faraji, M. and Qi, X., (2015) 'Face recognition under illumination variations based on eight local directional patterns', IET Biometrics, Vol. 4, Issue 1, pp. 10-17.

Faraji, M. and Qi, X., (2016) 'Face recognition under varying illuminations using logarithmic fractal dimension-based complete eight local directional patterns', Neurocomputing, Vol. 199, pp. 16-30.



Fathy, M.; Siyal, M. (1995) 'An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis', *Pattern Recognition Letters*, Vol.16, Issue 12, pp. 1321- 1330.

Fei, Y., Zhuang, Y. and Wang, W. (2002) 'Mobile robot 3D map building and path planning based on multi-sensor data fusion', *Intl. Journal of Computer Applications in Technology*, Vol. 44, Issue 4, pp. 276-283.

Figueiredo, J. (2008) 'Measuring railroad wheel profiles through image processing technology', *Proc. 18th IASTED Intl. Conf. on Visualization, Imaging and Image Processing – VIIP 2008*, Palma de Mallorca, Spain, pp.1–6.

Figueiredo, J., Martins, J. (2010) 'Energy Production System Management – Renewable Energy Power Supply Integration with Building Automation System', *Energy Conversion and Management*, 51 (2010), pp. 1120-1126 – Elsevier.

Figueiredo, J., Sá da Costa J. (2012) 'A SCADA System for Energy Management in Intelligent Buildings', *Energy and Buildings* Vol. 49 (2012), pp. 85-98 - Elsevier.

Figueiredo, J., Ayala Botto, M., Rijo, M. (2013) 'SCADA System with Predictive Controller applied to Irrigation Canals', *Control Engineering Practice*, Vol. 21 (2013), pp. 870-886 – Elsevier.

Feraund, R., Bernier, O.J. and Viallet, J.-E. (2001) 'A fast and accurate face detector based on neural networks'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, pp.42–53, IEEE.

Frank, J., Krishnamoorthy, S. and Kapila, V. (2017) 'Toward Mobile Mixed-Reality Interaction With Multi-Robot Systems', *IEEE Robotics and Automation Letters*, Vol. 2, Issue 4, pp. 1901 – 1908, IEEE.

Caggianese G., Gallo L., Neroni P. (2015) 'Design and Preliminary Evaluation of Free-Hand Travel Techniques for Wearable Immersive Virtual Reality Systems with Egocentric Sensing', *International Conference on Augmented and Virtual Reality AVR 2015 Image Processing, Computer Vision, Pattern Recognition, and Graphics*, Vol 9254. Springer.

Ganesharajah, T., Hall, N. and Sriskandarajah, C. (1998). 'Design and operational issues in AGV-served manufacturing systems', *Annals of Operations Research*, Vol. 76, Issue 0, pp.109-154, Springer.

Garcia C. and Tziritas G. (1999) 'Face detection using quantized skin color regions merging and wavelet packet analysis', *IEEE Transactions on Multimedia*, Vol.1, pp. 264 – 277, IEEE.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) 'Rich feature hierarchies for accurate object detection and semantic segmentation', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, IEEE.

Gokturk, S. B., Bouguet, J. -Y., Tomasi, C. and B. Girod (2002) 'Model-based face tracking for view-independent facial expression recognition'. Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition pp. 287 – 293, IEEE.

Gonzalez R.I C. and Woods R. E. (2002) 'Digital Image Processing 2ndEdition', Prentice Hall.

Gouttière, C. and Coninck, J. (2007) 'Detection of synthetic singularities in digital mammographies using spherical filters', Proc. ECCOMAS Thematic Conf. On Computational Vision and Medical Image Processing, Porto, Portugal, pp.97–100.

Grilo, F. and Figueiredo, J. (2008) 'An industrial vision system for quality control based on a distributed strategy', Proc. Intl. Conf. on Automatic Control – CONTROLO 2008, Vila Real, Portugal, pp.1-6.

Grilo, F., Figueiredo, J., Dias, O. and Amaral, T. (2007) 'Movement Detection in 2-D Images with Variable Dimension Cells', Proc. ECCOMAS Thematic Conf. On Computational Vision and Medical Image Processing, VIP IMAGE, Porto, Portugal

Grilo, F., Figueiredo, J., Dias, O. and Amaral, T. (2007) 'Adaptive topologies for dimension of cells in 2D-images for movement detection', WISP 2007, IEEE Intl. Symposium on Intelligent Signal Processing, Alcalá de Henares (Madrid), Spain, pp.1-6, IEEE.

Grilo, F., Figueiredo, J., Dias, O. and Amaral, T. (2009) 'Adaptive method for skin detection in coloured images', Proc. IEEE Intl. Conf. on Mechatronics, Malaga, Spain, pp.1-6, IEEE.

Grilo, F., Figueiredo, J. and Dias, O. (2014) 'Adaptive method for improvement of human skin detection in colour images', Int. J. Computer Applications in Technology, Vol. 49, Issue 1, pp.1-11.

Grilo, F. and Figueiredo, J. (2018) 'Computer-Vision in Industrial Automation and Mobile Robots', Introduction to Mechanical Engineering, Ed. P. Davim, pp. 241-266 SPRINGER-Verlag Berlin, Heidelberg.

Grilo, F. and Figueiredo, J. (2018) 'Tracking Moving Faces using the Golden Ratio in Colour Images', artigo submetido a Revista Internacional – Scopus Index).

Gomez G. and Morales E. (2002) 'Automatic feature construction and a simple rule induction algorithm for skin detection', ICML Workshop on Machine Learning in Computer Vision, pp. 31-38.

Gouttière, C., Coninck, J. (2007) 'Detection of Synthetic Singularities in Digital Mammographies using Spherical Filters', Proc. ECCOMAS Thematic Conf. On Computational Vision and Medical Image Processing, Porto, Portugal.

Güzel, M. and Zakaria, W. (2013) 'A Hybrid Architecture for Vision-Based Obstacle Avoidance', Mechatronics and Embedded System, Vol. 5, SAGE Journals.

Han, M. and Wang, J. (2015) 'Vision-based Automated Guided Vehicle Control by Using Fuzzy Kohonen Clustering Network', The 14th IFToMM World Congress, Taipei, Taiwan, 2015.

Haykin, S. (1991) 'Adaptive Filter Theory. Englewood Cliffs', NJ: Prentice-Hall, 1991.

Haro, A., Flickner, M. and Essa, I (2000) 'Detecting and tracking eyes by using their physiological properties, dynamics, and appearance'. Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, pp 163-168, IEEE.

Hensman, J., Fusi, N. and Lawrence, N. (2013) 'Gaussian Processes for Big Data', Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013), pp. 282-290.

Herrera, D., Roberti, F., Toibero, M. and Carelli, R. (2016) 'Human-Robot Interaction: Legible behavior rules in passing and crossing events', IEEE Latin America Transactions, Vol. 14, Issue 6, pp. 2644 – 2650, IEEE.

Hjelmås, E. and Kee Low, B. (2001) 'Face Detection: A Survey'. Computer Vision and Image Understanding, Vol. 83, pp. 236-274.

Hlou, L., Lichioui, A. and Guennoun, Z. (2003) 'Degraded 3D-objects restoration and their envelope extraction', Intl. Journal of Robotics and Automation, Vol. 18, Issue 2, pp. 88–93.

Hossain, V. and Muhammad, G. (2018) 'Emotion-Aware Connected Healthcare Big Data Towards 5G', IEEE Internet of Things Journal, Vol. 5, Issue 4, pp 2399-2406, IEEE.

Huang, C., Ai, H., Wu, B. and Lao, S. (2004) 'Boosting nested cascade detector for multi-view face detection'. Proceedings of the 17th International Conference on Pattern, ICPR 2004, Vol. 2, pp. 415 – 418.

Huang, T. (2014) 'Surveillance Video: The Biggest Big Data' Computing Now', Vol. 7, Issue 2, IEEE Computer.

Jain, A. (1981) 'Image data compression: A review', Proceeding of IEEE, Vol. 69, Issue 3.

Jain, R.; Kasturi, R.; Schunck, B. (1995) '*Machine Vision*', McGraw Hill, 1995.

Jain, S., Jagtap, V. and Pise, N. (2015) 'Computer Aided Melanoma Skin Cancer Detection Using Image Processing', Procedia Computer Science, Vol. 48, pp. 735-740, Elsevier.

Jain, V. and Learned-Miller, E. G.(2010) "FDDB: A benchmark for face detection in unconstrained settings" UMass Amherst Technical Report

Jebara T. S. and A. Pentland (1997) 'Parameterized structure from motion for 3d adaptive feedback tracking of faces', In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 144-150, San Juan, PuertoRico, IEEE.

Jones, M. and Rehg, J. (2002) 'Statistical color models with application to skin detection', International Journal of Computer Vision, Vol. 46, Issue 1, pp. 81-96, Springer.

Jordao L., Perrone M., Costeira J. and Santos-Victor J. (1999) 'Active face and feature tracking', the 10th Intl. Conf. Image Analysis and Processing, pp. 572-577.

KaKumanu, P., Makrogiannis, S. and Bourbakis, N. (2007) 'A survey of skin-color modeling and detetin methods', Pattern Recognition, Vol. 40, Issue 3, pp. 1106-1122.

Kale, G.and Patil, V.(2018) 'A Study of Vision based Human Motion Recognition and Analysis' , International Journal of Ambient Computing and Intelligence, Vol. 7, Issue 2, pp. 75-92.

Khan, Z., Anjum, A., Soomro, K. and Muhammad, A. (2015) 'Towards cloud based big data analytics for smart future cities', Journal of Cloud Computing 2015, Vol. 4, Num. 1, pp 1, Springer.

Kim, D., Comandur, B., Medeiros, H., Elfiky, N. and Kak, A. (2017) 'Multi-view face recognition from single RGBD models of the faces', *Computer Vision and Image Understanding*, Vol. 160, pp. 114-132.

Kim, H., Lee, K., Kim, S. and Yang, H. (2010) 'A precise inspection technique for wafer pre-sawing lines using Affine transformation', *Intl. Journal of Computer Applications in Technology*, Vol. 39, Issue 1-3, pp. 46-52.

Klančar, G., Zdešar, A., Blažič, S. and Škrjanc, I. (2017) 'Wheeled mobile robotics', 1st edition Butterworth-Heinemann.

Krizhevsky A., Sutskever I. and Hinton G. (2012) 'Imagenet classification with deepconvolutional neural networks', *Advances in neural information processing systems*, pp. 1097-1105.

Lee, J. and Shinozuka, M. (2006) 'A vision-based system for remote sensing of bridge displacement', *NDT&E International*, Vol. 39, Elsevier, pp. 425-431.

Lee, S. and Yang, H. (2012) 'Navigation of automated guided vehicles using magnet spot guidance method', *Robotics and Computer-Integrated Manufacturing*, Vol. 28, Issue 3, pp.425-436, Elsevier.

Lee, Y., Kim, J., Yu, W. and Moon, K. (2016) 'An RFID-based Real-time Localization of Multiple AGV (Automatic Guided Vehicle) System for the Guided Path Navigation', 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcalá de Henares, Spain.

Li, Z. and Wang, L. (2012) 'A multiple camera coordination method based on genetic programming and material character', *Intl. Journal of Computer Applications in Technology*, Vol. 43, Issue 4, pp. 351-358.

Lim, K., Du, T. and Wang, Q. (2011) 'Partially occluded object recognition', *Intl. Journal of Computer Applications in Technology*, Vol. 40, Issue 1-2, pp.122-131.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. and Berg, A. (2016) 'SSD: Single shot multibox detector', *European conference on computer vision ECCV 2016*, pp. 21-37, Springer.

Long, J., Shelhamer, E. and Darrell, T. (2015) 'Fully convolutional networks for semantic segmentation', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015*, pp. 3431-3440, IEEE.

Loos, S., Irving, G., Szegedy, C. and Kaliszyk, C. (2017) 'Deep network guided proof search', 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-21). EPiC Series in Computing, Vol. 46, pp. 85-105.

Luimula, M., Sääskilahti, K., Partala, T., Pieskä, S. and Alaspää, J. (2009), 'Remote navigation of a mobile robot in an RFID-augmented environment', Personal and Ubiquitous Computing, Vol. 14, Issue 2, pp.125-136, Springer.

Mahmoodi, M. and Sayedi, S. (2015) 'A face detection method based on kernel probability map.', Computers & Electrical Engineering 2015 Vol.46. pp. 205-216.

Mahmoodi, M. and Sayedi, S. (2016) 'A Comprehensive Survey on Human Skin Detection', International Journal of Image, Graphics and Signal Processing 2016, Vol. 8, Issue 5, pp.1-35

Manogaran, G., Thota, C. and Lopez, D. (2018) 'Human-Computer Interaction With Big Data Analytics', HCI Challenges and Privacy Preservation in Big Data Security, IGI Global, pp 1–22

Martínez-Barberá, H. and Herrero-Pérez, D. (2010a) 'Autonomous navigation of an automated guided vehicle in industrial environments', Robotics and Computer-Integrated Manufacturing, Vol. 26, Issue 4, pp.296-311.

Martinez-Barbera, H. and Herrero-Pérez, D. (2010b). 'Development of a flexible AGV for flexible manufacturing systems', Industrial Robot: An International Journal, Vol. 37, Issue 5, pp.459-468.

Messon, C. (2010) 'Synchronisation of vision-based sensor networks with variable frame rates', Intl. Journal of Computer Applications in Technology, Vol. 39, Issue 1-3, pp.153-158.

Miljkovic, Z., Vukovic, N., Mitic, B. and Babic, B. (2013) 'New hybrid vision-based control approach for automated guided vehicles', The International Journal of Advanced Manufacturing Technology, Vol. 66, Issue 1-4, pp. 231-249, Springer.

Mitchell, J.; Pennebaker, W.; Fogg, C.; LeGall, D. (1996) '*MPEG video compression standard*', Chapman &Hall, 1996.

Moeslund, T., Hilton, A. and Krüger, V. (2016) 'A survey of advances in vision-based human motion capture and analysis', Computer Vision and Image Understanding, Vol. 104, , Issue 2, pp. 90-126.

Mohammadi, M., Al-Fuqaha, A., Sorour, S. and Guizani, M. (2018) 'Deep Learning for IoT Big Data and Streaming Analytics: A Survey', *IEEE Communications Surveys & Tutorials*, 2018. Vol. 20, Issue 4, pp. 2923-2960. IEEE.

Najafabadi, M., Villanustre, F. Khoshgoftaar, T., Seliya, N., Wald, R. and Muharemagic, E. (2015) 'Deep learning applications and challenges in big data analytics', *Journal of Big Data* 2015 2:1, Springer.

Nelson, B. (2003) 'A distributed framework for visually servoed manipulation using an active camera', *International Journal of Robotics and Automation*, Vol. 18, Issue 2, pp. 21-27.

Obermeyer, Z., Phil, M. and Emanuel, E. (2016) 'Predicting the Future — Big Data, Machine Learning, and Clinical Medicine', *The New England journal of medicine* 375(13): pp. 1216-1219.

Pagello, E., D'Angelo, A. and Menegatti, E. (2006) 'Cooperation Issues and Distributed Sensing for Multirobot Systems', *Proceedings of the IEEE*, Vol. 6, Issue 3, pp. 1370 – 1383, IEEE.

Pai, Y., Lee, L., Ruan, S., Chen, Y., Mohanty, S. and Kougianos, E. (2010) 'Honeycomb model based skin colour detector for face detection', *Intl. Journal of Computer Applications in Technology*, Vol. 39, Issue 1-3, pp. 93-100.

Peer, P., Kovac, J. and Solina, F. (2003) 'Human skin colour clustering for face detection', *EUROCON 2003 – IEEE Region 8 Interl. Conf. Computer as a Tool*, Ljubljana, Slovenia, Vol. B, pp. 144-148, IEEE.

Peng, L., Yang, Y., Qi, X. and Wang, H. (2014) 'Highly accurate video object identification utilizing hint information', *Networking and Communications (ICNC), 2014 International Conference on Computing*, 2014, pp. 317-321.

Phimoltares, S., Lursinsap, C. and Chamnongthai, K. (2007) 'Face detection and facial feature localization without considering the appearance of image context', *Image and Vision Computing*, Vol. 25, Issue 5, pp. 741-753.

Phung, S., Bouzerdoum, A. and Chai, D. (2005) 'Skin segmentation using color pixel classification: analysis and comparison', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, Issue 1, pp. 148-154, IEEE.

Plageras, A., Psannis, K., Christos, S., Wang, H. and Gupta, B. (2018) 'Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings', *Future Generation Computer Systems*, Vol. 82, pp 349-357, Elsevier.

Prakash, T. and Kak, A. (2018) 'Active learning for designing detectors for infrequently occurring objects in wide-area satellite imagery', *Computer Vision and Image Understanding*, Vol. 170, pp. 92-108.

Rein-Lien, H. and Abdel-Mottaleb, J. (2002) 'Face detection in color images', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, pp. 696-706, IEEE.

Riisgaard, S. and Blas, M. (2005). 'SLAM for dummies', 1st edition MIT OCW.

Rowley, H. A., Baluja, S. and Kanade, T. (1998) 'Neural Network-Based Face Detection'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, pp. 23-38, IEEE.

Sakamoto, D., Sugiura, Y. and Inami, T. (2016) 'Graphical instruction for home robots', *Computer*, Vol. 49, Issue 7, pp. 20-25, IEEE.

Sankari, J. and Imtiaz, R. (2016). 'Automated guided vehicle (AGV) for industrial sector', 2016 10<sup>th</sup> International Conference on Intelligent Systems and Control (ISCO), IEEE <http://ieeexplore.ieee.org/document/7726962/> (Dezembro 2018).

Santos, R., Lengerke, O. and Edwards, A. (2013) 'Mobile Ad Hoc Robots and Wireless Robotic Systems: Design and Implementation', IGI Global.

Santy, A., and Joseph, R. (2015) 'Segmentation methods for computer aided melanoma detection', 2015 Global Conference on Communication Technologies (GCCT), IEEE.

Schroff F., Kalenichenko D. and Philbin J. (2015) 'Facenet: A unified embedding for face recognition and clustering', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815-823, IEEE.

Shaik, K., Ganesan, P., Kalist, V., Sathish, B., Jenitha, J. (2015) 'Comparative study of skin color detection and segmentation in HSV and YCbCr color space', *Procedia Computer Science* 2015, Vol. 57, pp. 41-48.

Shalaby, M., Salem, M., Khamis, A. and Melgan, F. (2014) 'Geometric model for vision-based door detection', *IEEE International Conference on Computer Engineering & Systems (ICCES)*, pp. 4146, IEEE.

Shelestov, A., Lavreniuk, M., Kussul, N., Novikov, A. and Skakun, S. (2017) 'Exploring Google Earth Engine Platform for Big Data Processing: Classification of Multi-Temporal Satellite Imagery for Crop Mapping', *Frontiers in Earth Science*, Vol. 5.



- Shih, F. and Chuang, C. (2004) 'Automatic extraction of head and face boundaries and facial features', *Information Sciences* Vol. 158 (1), pp. 117-130.
- Shneier, M. and Bostelman, R. (2015). 'Literature Review of Mobile Robots for Manufacturing', <http://nvlpubs.nist.gov/nistpubs/ir/2015/NIST.IR.8022.pdf> (Dezembro 2018).
- Smith, S. and Nichols, T. (2018) 'Statistical Challenges in "Big Data" Human Neuroimaging', *Neuron*, Vol. 97, Issue 2, pp 263-268, Elsevier.
- Snášel, M., Nowaková, J., Xhafa, F. and Barolli, L. (2017) 'Geometrical and topological approaches to Big Data', *Future Generation Computer Systems*, Vol. 67, pp 286-296, Elsevier.
- Soares F. J. and Tavares J. M. R. S. (2005) 'Metodologias para Identificação de Faces em Imagens: Introdução e Exemplos de resultados', *Congresso de Métodos Numéricos en Ingeniería, Granada, España*, 14 p.
- Sobreira, H., Moreira, A., Costa, P. and Lima, J. (2016) 'Robust mobile robot localization based on a security laser: an industry case study', *Industrial Robot: An International Journal*, Vol. 43, Issue 6, pp.596-606.
- Storring M., Andersen H. J. and Granum E. (2000) 'Estimation of the illuminant colour from human skin colour, Automatic Face and Gesture Recognition', *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 64-69, IEEE.
- Sun Y., Wang X. and Tang X. (2014) 'Deep learning face representation from predicting 10,000 classes', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1891-1898, IEEE.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015) 'Going deeper with convolutions', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015*, pp. 1-9, IEEE.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Z. Wojna (2016) 'Rethinking the Inception Architecture for Computer Vision', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016*, pp. 2818-2826, IEEE.
- Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. (2017) 'Inception-v4, inception-resnet and the impact of residual connections on learning', *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, Vol. 4, pp. 4278- 4284.

Tekalp, A. (2015) 'Digital Video Processing', 2nd Prentice Hall.

Terrillon J. C., David M. and Akamatsu S. (1998) 'Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments', Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 112 -117, IEEE.

Terrillon J. C., Shirazi M. N., Fukamachi H. and Akamatsu S. (2000) 'Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images', Proceedings. Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 54 -61, IEEE.

Toshev, A. and Szegedy, C. (2014) 'DeepPose: Human pose estimation via deep neural networks', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp. 1653-1660, IEEE.

Ullrich, G. (2015) 'Automated guided vehicle systems', 2nd ed. Heidelberg, Springer.

Umbaugh S. E. (2010) 'Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPTools. 2<sup>nd</sup> Edition', CRC Press.

Vadakkepat, P., Lim, P., Silva, L., Liu, J. and Li, L. (2008) 'Multimodal approach to human-face detection and tracking', IEEE Transactions on Industrial Electronics, Vol. 55, pp. 1385 -1393. IEEE.

Vezhnevets, V., Sazonov, V. and Andreeva, A. (2003) 'A survey on pixel-based skin color detection techniques', Intl. Conf. Graphicon-2003, Moscow, pp. 1-8.

Vignesh, S., Rajesh, S. and Lingaraj, N. (2017) 'Intelligent Automated Guided Vehicle using Visual Servoing', American Journal of Engineering Research (AJER) Vol. 6, Issue 11, pp.16-20

Wang, Y. and Yuan, B. (2002) 'Fast method for face location and tracking by distributed behaviour-based agents', Vision, Image and Signal Processing, IEEE Proceedings, Vol. 149, pp.173-178, IEEE.

Widrow, B. and Stearns, S. (1985) 'Adaptive Signal Processing', Prentice Hall Inc., New Jersey.

Wu, H., Chen, Q. and Yachida, M. (1999) 'Face detection from colour images using a fuzzy pattern matching method', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, Issue 6, pp. 557-563, IEEE.

Xiaoping, L. and Bui, D. (2012) 'Functional neuroimaging of circadian fatigue', Intl. Journal of Computer Applications in Technology, Vol. 45, Issue 2-3, pp. 156-162.

Xing, W., Peihuang, L., Jun, Y., Xiaoming, Q. and Dunbing, T. (2014) 'Intersection Recognition and Guide-Path Selection for a Vision-Based AGV in a Bidirectional Flow Network', International Journal of Advanced Robotic Systems, Vol. 11, Issue 3.

Yang, C., Yu, M., Hu, F., Jiang, Y. and Li, Y. (2017) 'Utilizing Cloud Computing to address big geospatial data challenges', Computers, Environment and Urban Systems, Vol. 61, pp 120-128, Elsevier.

Yang, M., Kriegman, D. and Ahuja, N. (2002) 'Detecting faces in images: a survey', IEEE Trans. Pattern Anal. & Machine Intell., Vol. 24, pp. 34-58, IEEE.

Yang, T., Zhu, K., Ruan, Q. and Han, D. (2010) 'Moving target tracking and measurement with a binocular vision system'. Intl. Journal of Computer Applications in Technology, Vol. 39, Issue 1-3, pp. 145 -152.

Zafeiriou, S., Zhang, C., Zhang, Z. (2015) 'A survey on face detection in the wild: past, present and future', Computer Vision Image Understanding, Vol. 138, pp. 1-24.

Zhang, X., Gong, F. and Xu, L. (2012) 'Inspection of surface defects in copper strip using multivariate statistical approach and SVM', Intl. Journal of Computer Applications in Technology, Vol. 43, Issue 1, pp. 44-50.

Zhang, Y., Yang, Z., Lu, H., Zhou, X., Phillips, P., Liu, Q. and Wang, S. (2016) 'Facial Emotion Recognition Based on Biorthogonal Wavelet Entropy, Fuzzy Support Vector Machine, and Stratified Cross Validation', IEEE Access, Vol. 4, pp. 8375 – 8385, IEEE.

Zhang, Z., Chen, Z., Shi, J., Jia, F. and Dai, M. (2010) 'Surface roughness vision measurement in different ambient light conditions', Intl. Journal of Computer Applications in Technology, Vol. 39, Issue 1-3, pp. 53-57.

Zhao W., Chellappa R., Phillips P. J. and Rosenfeld A. (2003) 'Face Recognition: A Literature Survey', ACM Computing Surveys, Vol. 35, pp. 399-458.

Zunino, G. (2002) 'Simultaneous Localization and Mapping for Navigation in Realistic Environments' The Royal Institute of Technology Stockholm, <http://www.nada.kth.se/utbildning/forsk.utb/avhandlingar/lic/020220.pdf> (Dezembro 2018)

Labeled Faces in the Wild A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts  
<http://vis-www.cs.umass.edu/lfw/> (acedido 2008).

<https://www.digi.com/> (acedido em Maio 2019).

<http://www.robot-electronics.co.uk/htm/md49tech.htm> (acedido em Maio 2019).

<http://www.ni.com> (acedido em Maio 2019).

<https://www.arduino.cc> (acedido em Maio 2019).