

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MÉTODO PARA EXTRAÇÃO DE OBJETOS DE UMA IMAGEM DE
REFERÊNCIA ESTÁTICA COM ESTIMATIVA DAS VARIAÇÕES DE
ILUMINAÇÃO

JOZIAS PARENTE DE OLIVEIRA

TD – 08/2009

Belém, Pará

2009

JOZIAS PARENTE DE OLIVEIRA

MÉTODO PARA EXTRAÇÃO DE OBJETOS DE UMA IMAGEM DE
REFERÊNCIA ESTÁTICA COM ESTIMATIVA DAS VARIAÇÕES DE
ILUMINAÇÃO

Tese de Doutorado Submetida ao Programa de
Pós-Graduação em Engenharia Elétrica da
Universidade Federal do Pará, como parte dos
requisitos necessários para obtenção do grau de
Doutor em Engenharia Elétrica.

Orientador: Raimundo Carlos Silvério Freire, Dr.
Co-orientador: Evaldo Gonçalves Pelaes, Dr.

Belém

2009

O48m Oliveira, Jozias Parente de

Método para extração de objetos de uma imagem de referência estática com estimativa das variações de iluminação / Jozias Parente de Oliveira; orientador, Raimundo Carlos Silvério Freire.-2009.

Tese (doutorado) - Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2009.

1. Processamento de imagens – técnicas digitais. 2. FPGAs. I. Orientador. II. Título.

CDD 22. ed. 621.367

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MÉTODO PARA EXTRAÇÃO DE OBJETOS DE UMA IMAGEM DE REFERÊNCIA
ESTÁTICA COM ESTIMATIVA DAS VARIAÇÕES DE ILUMINAÇÃO

AUTOR: JOZIAS PARENTE DE OLIVEIRA

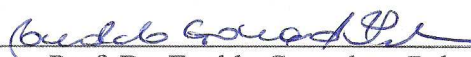
TESE DE DOUTORADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA
APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA
ADEQUADA PARA OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA
ELÉTRICA NA ÁREA DE TELECOMUNICAÇÕES.

APROVADA EM 04 / 12 / 2009

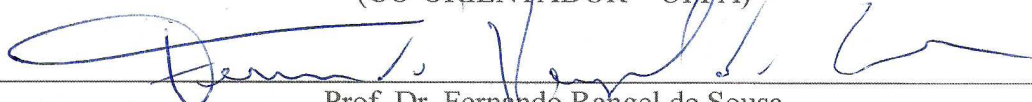
BANCA EXAMINADORA:



Prof. Dr. Raimundo Carlos Silvério Freire
(ORIENTADOR – UFCG)



Prof. Dr. Evaldo Gonçalves Pelaes
(CO-ORIENTADOR – UFPA)



Prof. Dr. Fernando Rangel de Sousa
(MEMBRO – UFSC)



Prof. Dr. Elmar Uwe Melcher
(MEMBRO – UFCG)



Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior
(MEMBRO – UFPA)

VISTO: _____

Prof. Dr. Marcus Vinicius Alves Nunes
(COORDENADOR DO PPGE/ITEC/UFPA)
UFPA / ITEC / PPGE

DEDICATÓRIA

Dedico este trabalho a minha esposa Rosiele pelos incentivos e pela compreensão da minha ausência, a minha mãe Izabel pela motivação e a memória de meu pai José Parente que sempre incentivou a minha formação.

AGRADECIMENTOS

Gostaria de expressar os meus sinceros agradecimentos à Coordenação de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará pela oportunidade do Doutorado; Aos professores Raimundo Freire e Elmar Melcher por acreditarem no meu trabalho que começou com o curso de mestrado na Universidade Federal de Campina Grande (UFCG); À memória do professor Raimundo Nazareno pela viabilidade deste trabalho; Ao professor Ivan Sebastião Souza e Silva pelo apoio recebido no início desta jornada; Ao professor Evaldo Gonçalves Pelaes pela compreensão e o suporte no final desta caminhada; Aos membros da banca examinadora: Fernando Rangel de Souza e Aldebaro Barreto da Rocha Klautau Júnior pelas críticas e sugestões; A André Printes e Marcel Bergeman pelo precioso tempo dedicado à revisão deste trabalho; Aos colegas de trabalho do Núcleo de Sistemas Embarcados (NSE) da Universidade Estado do Amazonas (UEA) pelo apoio recebido no final desta jornada.

Quando nada parece dar certo, vou ver o cortador de pedras martelando sua rocha talvez 100 vezes, sem que uma única rachadura apareça. Mas na centésima primeira martelada a pedra se abre em duas, e eu sei que não foi aquela que conseguiu isso, mas todas as que vieram antes.

Jacob Riis.

SUMÁRIO

INTRODUÇÃO.....	20
1.1 MOTIVAÇÃO E JUSTIFICATIVA	22
1.2 OBJETIVOS	25
1.3 METODOLOGIA	25
1.4 CONTRIBUIÇÕES DA TESE	26
1.5 ORGANIZAÇÃO DO DOCUMENTO.....	27
2 MÉTODOS PARA SEGMENTAÇÃO DE VÍDEO	29
2.1 INTRODUÇÃO	29
2.2 TRABALHOS RELACIONADOS	30
2.2.1 Modelo da imagem de referência.....	30
2.2.2 Atualização do modelo da imagem de referência	32
2.2.3 Tamanho da memória para armazenar o modelo da imagem de referência.....	33
2.3 CONCLUSÕES	34
3 MÉTODO PROPOSTO PARA SEGMENTAÇÃO DE VÍDEO.....	36
3.1 CONSIDERAÇÕES INICIAIS DA SOLUÇÃO PROPOSTA	36
3.2 SOLUÇÃO PROPOSTA	36
3.2.1 Modelo da Imagem de Referência	37
3.2.2 Determinação do número de quadros.....	38
3.2.3 Intervalo de confiança.....	42
3.2.4 Seleção do Nível de Limiar.....	45
3.2.5 Extração do Objeto ou classificação do <i>pixel</i>	45
3.2.6 Compensação de variações do <i>pixel</i>	46
3.2.7 Detecção de variação global	54
3.2.8 Detecção de sombras.....	57
3.3 CONCLUSÕES	58
4 AVALIAÇÃO DE MÉTODOS PARA SEGMENTAÇÃO DE VÍDEO.....	59
4.1 MÉTRICAS PARA AVALIAÇÃO.....	59
4.2 ANÁLISE DE DISTRIBUIÇÃO DAS IMAGENS DE TESTE	62
4.3 AVALIAÇÃO DO MÉTODO PROPOSTO	64
4.3.1 Avaliação da classificação dos <i>pixels</i>	65
4.3.2 Comparação dos espaços de cores	78
4.3.3 Comparação com adaptação recursiva.....	86
4.4 CONCLUSÕES	92
5 ESTRATÉGIAS PARA IMPLEMENTAÇÃO DE ALGORITMOS DE SEGMENTAÇÃO DE VIDEO EM HARDWARE	94
5.1 INTRODUÇÃO	94
5.2 ADAPTAÇÕES DO MÉTODO PROPOSTO PARA PONTO FIXO	95
5.2.1 Modelo da imagem de referência.....	95
5.2.2 Intervalo de confiança.....	96
5.2.3 Seleção dos limiares.....	98
5.2.4 Compensação de variações do <i>pixel</i>	98
5.2.5 Classificação do <i>pixel</i>	100
5.2.6 Detecção de sombras.....	100
5.2.7 Comparações entre ponto fixo e flutuante	101

5.3 COMPARAÇÃO ENTRE OPERAÇÕES EM PONTO FIXO	112
5.3.1 Valor médio	112
5.3.2 Desvio padrão	112
5.3.3 Intervalo de confiança.....	113
5.3.4 Seleção dos limiares.....	114
5.3.5 Compensação de variações do <i>pixel</i>	114
5.3.6 Classificação do <i>pixel</i>	116
5.3.7 Detecção de sombras.....	117
5.4 AVALIAÇÕES DAS OPERAÇÕES EM PONTOS FIXOS	118
5.4.1 Sequência de Imagens 01	118
5.4.2 Sequência de Imagens 02.....	119
5.4.3 Sequência de Imagens 03.....	119
5.5 REDUÇÃO DA MEMÓRIA PARA IMAGEM DE REFERÊNCIA.....	120
5.5.1 Avaliação da redução de memória.....	122
5.6 CONCLUSÕES	132
6 ARQUITETURAS PARA SEGMENTAÇÃO DE VÍDEO.....	134
6.1 INTRODUÇÃO.....	134
6.2 ARQUITETURA PROPOSTA.....	135
6.2.1 Visão Geral da Arquitetura	135
6.2.2 Unidade de Segmentação de Vídeo	137
6.3 DESEMPENHO DA ARQUITETURA	147
6.3.1 Comparação entre Hardware e MATLAB	151
6.3.2 Detecção de Sombras	152
6.3.3 Intervalos de confiança	152
6.4 RESULTADOS DA SÍNTESE.....	154
6.5 CONCLUSÕES	156
7 CONCLUSÃO E TRABALHOS FUTUROS.....	157
7.1 RESULTADOS	157
7.1.1 Aplicação de Métodos Estatísticos para Avaliação da Segmentação	157
7.1.2 Desenvolvimento do Método para Compensação das Variações dos <i>Pixels</i>	159
7.1.3 Aplicação de Estimativas Estatísticas para Determinação do Número de Quadros	161
7.1.4 Implementação da Arquitetura de Hardware para Detecção de Objetos	161
7.2 TRABALHOS FUTUROS	162
7.3 PUBLICAÇÕES	163
7.3.1 Artigos Publicados	163
7.3.2 Artigo Submetido.....	163
7.3.3 Patente – PI0204940-6.....	163
REFERÊNCIAS BIBLIOGRÁFICAS	164

LISTA DE FIGURAS

Figura 1: Diagrama em Blocos Básico de um Sistema para Detecção de Objetos	30
Figura 2: O diagrama em blocos com as etapas do método proposto para detecção de objetos.	37
Figura 3: Seqüência de imagens utilizadas para determinar o número de quadros.....	40
Figura 4: Componente Y: Média e desvio padrão estimados com 30 quadros para cada seqüência de imagem. Número de quadros (n) e margens de erro (δ) para cada seqüência de imagens.....	40
Figura 5: Componente Cb: Média e desvio padrão estimados com 30 quadros para cada seqüência de imagem. Número de quadros (n) e margens de erro (δ) para seqüência de imagens.....	41
Figura 6: Componente Cr: Média e desvio padrão estimados com 30 quadros para cada seqüência de imagem. Número de quadros (n) e margens de erro (δ) para seqüência de imagens.....	41
Figura 7: A) Imagem de referência com e sem objetos; B) Componente Y: Os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança; C) Componente Cb: Os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança; B) Componente Cr: Os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança.....	44
Figura 8: Seqüência de imagens exibindo a variação de intensidade de brilho em função das correções automáticas realizadas pela câmera.....	47
Figura 9: A) imagem de teste sem objetos; B) os pontos pretos são os <i>pixels</i> detectados abaixo dos limites inferiores dos intervalos de confiança, C) os pontos pretos são os <i>pixels</i> detectados acima dos limites superiores dos intervalos de confiança.	49
Figura 10: A) imagem de teste com objetos; B) os pontos pretos são os <i>pixels</i> detectados abaixo dos limites inferiores dos intervalos de confiança, C) os pontos pretos são os <i>pixels</i> detectados acima dos limites superiores dos intervalos de confiança.	50

Figura 11: A) imagem de teste sem objetos; B) os pontos brancos são os <i>pixels</i> classificados como objetos. Neste caso, a função de compensação de variação do <i>pixel</i> não foi utilizada; C) os pontos brancos são os <i>pixels</i> classificados como objetos. Neste caso, a função de compensação de variação do <i>pixel</i> foi utilizada.	51
Figura 12: A) imagem de teste com objetos; B) os pontos brancos são os <i>pixels</i> classificados como objetos. Neste caso, a função de compensação de variação do <i>pixel</i> não foi utilizada; C) os pontos brancos são os <i>pixels</i> classificados como objetos. Neste caso, a função de compensação de variação do <i>pixel</i> foi utilizada.	52
Figura 13: Dispersão das diferenças absolutas com objetos. BD_NC: diferenças absolutas não compensadas; BD_C: diferenças absolutas compensadas.....	52
Figura 14: Dispersão das diferenças absolutas sem objetos. BD_NC: diferenças absolutas não compensadas; BD_C: diferenças absolutas compensadas.....	53
Figura 15: Gráficos das diferenças absolutas com (BD_C) e sem (BD_NC) a função de compensação de variação do <i>pixel</i> para a seqüência de imagens de testes 03.	53
Figura 16: Gráficos das diferenças absolutas com (BD_C) e sem (BD_NC) a função de compensação de variação do <i>pixel</i> para a seqüência de imagens mostradas nesta figura.	54
Figura 17: Sequência de imagens <i>time of day</i> . Gráfico dos valores médios da luminância para cada quadro da seqüência de imagens <i>time of day</i>	55
Figura 18: Porcentagem de variação de falsos positivos com e sem a função de compensação de variação de <i>pixel</i> de acordo com a variação média de luminância.	56
Figura 19: Sequência de imagens utilizadas para avaliação da detecção de falsos positivos em um cenário com variação de iluminação. A) imagem de fundo original, B) os pontos brancos são os <i>pixels</i> classificados como objetos sem utilizar a função de compensação de variação do <i>pixel</i> , e C) os pontos brancos são os <i>pixels</i> classificados como objetos usando a função de compensação de variação do <i>pixel</i>	56
Figura 20: Exemplo de uma imagem utilizada para avaliar o método de segmentação proposto neste trabalho. a) imagem de referência; b) imagem de teste de referência (ground truth)	59
Figura 21: Resultados do teste de normalidade para a componente Y	62
Figura 22: Resultados do teste de normalidade para a componente Cb	63
Figura 23: Resultados do teste de normalidade para a componente Cr.....	63
Figura 24: Sequências de imagens utilizadas para avaliação do método de detecção de objetos.	64

Figura 25: Medida F para seqüência de imagens 01 com três tipos de classificação dos *pixels*. Teste de Kruskal-Wallis ($p = 0,0034$); Teste de igualdade de variância ($p = 0,0048$). Classificação do *Pixel* 01 – função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

..... 65

Figura 26: Valores das medianas das medidas F para a seqüência de imagens 01 com três tipos de classificação dos *pixels*. Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

..... 66

Figura 27: Resultados com a seqüência de imagens 01 com três tipos de classificação dos *pixels*. A) imagem de referência com objeto; B) Classificação do *Pixel* 01; C) Classificação do *Pixel* 02; e D) Classificação do *Pixel* 03.

..... 67

Figura 28: Medida F para seqüência de imagens 02 com três tipos de classificação dos *pixels*. Teste de Kruskal-Wallis ($p = 0,1124$); Teste de igualdade de variância ($p = 0,0038$). Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

..... 68

Figura 29: Valores das medianas das medidas F para a seqüência de imagens 02 com três tipos de classificação dos *pixels*. Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

..... 69

Figura 30: Resultados com a seqüência de imagens 02 e três tipos de classificação dos *pixels*. A) imagem de referência com objeto; B) Classificação do *Pixel* 01; C) Classificação do *Pixel* 02; e D) Classificação do *Pixel* 03.

..... 70

Figura 31: A) imagem de referência com objeto; B) os pontos pretos são os <i>pixels</i> detectados abaixo do limite inferior do intervalo de confiança, C) os pontos pretos são os <i>pixels</i> detectados acima do limite superior do intervalo de confiança.....	70
Figura 32: Medida F para sequência de imagens 03 com três tipos de classificação dos <i>pixels</i> . Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,0875$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.	71
Figura 33: Valores das medianas das medidas F para a sequência de imagens 03 com três tipos de classificação dos <i>pixels</i> . Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.	72
Figura 34: Resultados com a sequência de imagens 03 e três tipos de classificação dos <i>pixels</i> . A) imagem de referência com objeto; B) Classificação do <i>Pixel</i> 01; C) Classificação do <i>Pixel</i> 02; e D) Classificação do <i>Pixel</i> 03.....	73
Figura 35: Medida F para sequência de imagens 04 com três tipos de classificação dos <i>pixels</i> . Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.	73
Figura 36: Valores da mediana das medidas F para a sequência de imagens 04 com três tipos de classificação dos <i>pixels</i> . Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.	74

Figura 37: Resultados com a seqüência de imagens 04 e três tipos de classificação dos <i>pixels</i> . A) imagem de referência com objeto; B) Classificação do <i>Pixel</i> 01; C) Classificação do <i>Pixel</i> 02; e D) Classificação do <i>Pixel</i> 03.....	75
Figura 38: Medida F para seqüência de imagens 05 com três tipos de classificação dos <i>pixels</i> . Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.	76
Figura 39: Valores da mediana das medidas F para a seqüência de imagens 01 com três tipos de classificação dos <i>pixels</i> . Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizados durante a detecção de objetos; classificação do <i>pixel</i> 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.	77
Figura 40: Resultados com a seqüência de imagens 05 e três tipos de classificação dos <i>pixels</i> . A) imagem de referência com objeto; B) Classificação do <i>Pixel</i> 01; C) Classificação do <i>Pixel</i> 02; e D) Classificação do <i>Pixel</i> 03.....	78
Figura 41: Medida F para seqüência de imagens 01 obtidas em diferentes espaços de cores. Teste de Kruskal Wallis ($p = 0,0049$); Teste de igualdade de variância ($p = 0,0011$).	79
Figura 42: Valores das medianas das medidas F para a seqüência de imagens 01 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.	80
Figura 43: Resultados com a seqüência de imagens 01 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.	81
Figura 44: Medida F para seqüência de imagens 04 obtidas em diferentes espaços de cores. Teste de Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,0011$).	81
Figura 45: Valores das medianas das medidas F para a seqüência de imagens 04 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.	82
Figura 46: Resultados com a seqüência de imagens 04 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.	83
Figura 47: Medida F para seqüência de imagens 05 obtidas em diferentes espaços de cores. Teste de Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$).	84

Figura 48: Valores das medianas das medidas F para a seqüência de imagens 05 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.	85
Figura 49: Resultados com a seqüência de imagens 05 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.	85
Figura 50: Medida F para seqüência de imagens 03 obtidas utilizando-se a função de compensação de variação do <i>pixel</i> e a adaptação recursiva com diferentes taxas de atualização (TA). Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$)...	87
Figura 51: Valores das medianas das medidas F para seqüência de imagens 03 obtidas utilizando-se a função de compensação de variação do <i>pixel</i> e a adaptação recursiva com diferentes taxas de atualização (TA).	87
Figura 52: Medida F para seqüência de imagens 04 obtidas utilizando-se a função de compensação de variação do <i>pixel</i> e a adaptação recursiva com diferentes taxas de atualização (TA). Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$)...	88
Figura 53: Valores das medianas das medidas F para sequência de imagens 04 obtidas utilizando-se a função de compensação de variação do <i>pixel</i> e a adaptação recursiva com diferentes taxas de atualização (TA).	89
Figura 54: Medida F para sequência de imagens 05 obtidas utilizando-se a função de compensação de variação do <i>pixel</i> e a adaptação recursiva com diferentes taxas de atualização (TA). Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$)...	90
Figura 55: Valores das medianas das medidas F para seqüência de imagens 05 obtidas utilizando-se a função de compensação de variação do <i>pixel</i> e a adaptação recursiva com diferentes taxas de atualização (TA).	90
Figura 56: Resultados com as seqüências de imagens 03, 04 e 05 obtidas utilizando-se o método gaussiano com adaptação recursiva e taxa de atualização igual a 0,0025 (B) e o método gaussiano com função de compensação de variação do <i>pixel</i> (C).....	91
Figura 57: Medida F para seqüência de imagens 01 obtidas utilizando-se o método proposto com operações em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p = 0,103$); Teste de igualdade de variância ($p = 0,0118$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.	102

Figura 58: Valores das medianas das medidas F obtidas para a seqüência de imagens 01 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante. A) Classificação do <i>Pixel</i> 03 – Ponto Fixo; B) Classificação do <i>Pixel</i> 03 – Ponto Flutuante; C) Classificação do <i>Pixel</i> 01 – Ponto Fixo; D) Classificação do <i>Pixel</i> 01 – Ponto Flutuante;.....	103
Figura 59: Medida F para seqüência de imagens 02 obtidas utilizando-se o método proposto em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p = 0,1865$); Teste de igualdade de variância ($p = 0,0003$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.....	104
Figura 60: Valores das medianas das medidas F obtidas para a seqüência de imagens 02 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante. A) Classificação do <i>Pixel</i> 03 – Ponto Fixo; B) Classificação do <i>Pixel</i> 03 – Ponto Flutuante; C) Classificação do <i>Pixel</i> 01 – Ponto Fixo; D) Classificação do <i>Pixel</i> 01 – Ponto Flutuante;.....	105
Figura 61: Medida F para seqüência de imagens 03 obtidas utilizando-se o método proposto em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p = 0,0136$); Teste de igualdade de variância ($p = 0,056$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.....	106
Figura 62: Valores das medianas das medidas F obtidas para a seqüência de imagens 03 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante. A) Classificação do <i>Pixel</i> 03 – Ponto Fixo; B) Classificação do <i>Pixel</i> 03 – Ponto Flutuante; C) Classificação do <i>Pixel</i> 01 – Ponto Fixo; D) Classificação do <i>Pixel</i> 01 – Ponto Flutuante;.....	107
Figura 63: Medida F para seqüência de imagens 04 obtidas utilizando-se o método proposto em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.....	108

Figura 64: Valores das medianas das medidas F obtidas para a seqüência de imagens 04 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante. A) Classificação do <i>Pixel</i> 03 – Ponto Fixo; B) Classificação do <i>Pixel</i> 03 – Ponto Flutuante; C) Classificação do <i>Pixel</i> 01 – Ponto Fixo; D) Classificação do <i>Pixel</i> 01 – Ponto Flutuante.....	109
Figura 65: Medida F para seqüência de imagens 05 obtidas utilizando-se o método proposto em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.....	110
Figura 66: Valores das medianas das medidas F obtidas para a seqüência de imagens 05 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante. A) Classificação do <i>Pixel</i> 03 – Ponto Fixo; B) Classificação do <i>Pixel</i> 03 – Ponto Flutuante; C) Classificação do <i>Pixel</i> 01 – Ponto Fixo; D) Classificação do <i>Pixel</i> 01 – Ponto Flutuante.....	111
Figura 67: Medida F para seqüência de imagens 01 obtidas utilizando-se o método proposto em ponto fixo. Teste de Kruskal-Wallis ($p = 0,09$); Teste de igualdade de variância ($p = 0,0118$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.....	118
Figura 68: Medida F para seqüência de imagens 01 obtidas utilizando-se o método proposto em ponto em ponto fixo. Teste de Kruskal-Wallis ($p = 0,5$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.....	119
Figura 69: Medida F para seqüência de imagens 01 obtidas utilizando-se o método proposto em ponto em ponto fixo. Teste de Kruskal-Wallis ($p = 0,18$). Classificação do <i>Pixel</i> 01 – a função de compensação de variação do <i>pixel</i> e detecção de sombras são utilizadas durante a detecção de objetos; classificação do <i>pixel</i> 03 – somente a função de compensação de variação do <i>pixel</i> é usada durante a detecção de objetos.....	120

Figura 70: Medida F para a seqüência de imagens 01 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p = 0,9945$); Teste de igualdade de variância ($p = 0,9848$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.....	123
Figura 71: Medida F para a seqüência de imagens 02 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p = 0,7626$); Teste de igualdade de variância ($p = 0,7842$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.....	124
Figura 72: Medida F para a seqüência de imagens 03 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,0493$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.....	125
Figura 73: Valores das medianas das medidas F para a seqüência de imagens 03 com diferentes métodos para calcular os valores da média e do desvio padrão. Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.	126
Figura 74: Medida F para a seqüência de imagens 04 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,9688$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.....	127
Figura 75: Valores das medianas das medidas F para a seqüência de imagens 04 com diferentes métodos para calcular os valores da média e do desvio padrão. Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.	128

Figura 76: Medida F para a seqüência de imagens 05 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,9676$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.....	129
Figura 77: Valores das medianas das medidas F para a seqüência de imagens 04 com diferentes métodos para calcular os valores da média e do desvio padrão. Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.	130
Figura 78: Resultados da detecção de objetos utilizando-se o método proposto com as modificações para redução do tamanho da memória para armazenar a imagem de referência.	131
Figura 79: Diagrama em blocos da arquitetura para segmentação de vídeo	136
Figura 80: Diagrama em blocos da unidade de segmentação de vídeo	137
Figura 81: Diagrama em blocos do módulo média aritmética.....	140
Figura 82: Módulos utilizados para calcular o desvio padrão. 1) Módulo média aritmética; 2) Módulo Diferença Absoluta; 3) Módulo desvio padrão.....	141
Figura 83: Diagrama em blocos do modulo denominado Compensação das variações dos <i>pixel</i> e detecção de sombras. Cálculo do numerador da equação 31 para o Y.....	142
Figura 84: Diagrama em blocos do módulo denominado Compensação das variações dos <i>pixel</i> e detecção de sombras. Cálculo dos limites superiores e inferiores dos intervalos de confiança.	143
Figura 85: Diagrama em blocos do módulo denominado Compensação das variações dos <i>pixel</i> e detecção de sombras. Detecção de sombras, cálculo dos resultados finais das equações 31 e 33 para o Y.	144
Figura 86: Diagrama em blocos do módulo denominado Compensação das variações dos <i>pixel</i> e detecção de sombras. Cálculo dos resultados finais das equações 31 e 33 para as componentes Cb e Cr.....	145
Figura 87: Diagrama em blocos do módulo denominado classificação do <i>pixel</i>	146
Figura 88: Seqüência de imagens utilizada para avaliar o desempenho da arquitetura implementada em hardware em um ambiente com variações de iluminação.....	147

Figura 89: Resultados da segmentação realizada com a arquitetura implementada no kit Altera. Em todos os casos as lâmpadas estão desligadas. 1) a função de compensação de variação do <i>pixel</i> está ativa para as components YCbCr; 2) a função de compensação de variação do <i>pixel</i> para a componente Y está desligada; 3) a função de compensação de variação do <i>pixel</i> está ligada para a componente Y e desligada para as componentes Cb/Cr; 4) idem ao 1).	148
Figura 90: Resultados segmentação realizada com a arquitetura implementada em hardware. 1) Uma das lâmpadas está ligada e a função de compensação de variação do <i>pixel</i> está ligada para YCbCr; 2) A outra lâmpada foi ligada e a função de compensação continua ligada; 3) As duas lâmpadas estão ligadas e a função de compensação está desligada para a componente Y; 4) Uma das lâmpadas foi desligada e a função de compensação continua desligada; 5) idem ao 1).	149
Figura 91: Resultados segmentação realizada com a arquitetura implementada em hardware.	150
Figura 92: Comparação entre as implementações em Hardware e MATLAB. A) imagem de referência com objeto; b) resultado da implementação em MATLAB; c) resultado da implementação em hardware.	151
Figura 93: Resultados da detecção de sombras obtidos com a implementação em hardware. A) imagem de referência com objeto; b) resultado sem detecção de sombras; c) resultado com a detecção de sombras.	152
Figura 94: Resultados da identificação dos <i>pixels</i> pertencentes aos intervalos de confiança obtidos com a implementação em hardware. A) imagem de referência sem objetos; b) os pontos brancos são os <i>pixels</i> pertencentes aos intervalos de confiança sem objetos na imagem de referência; c) os pontos brancos são os <i>pixels</i> pertencentes aos intervalos de confiança com objetos (mão) na imagem de referência;	153
Figura 95: “Floorplane” do método de detecção de objetos mapeado no FPGA do kit DE-2.	155
Figura 96: Métodos utilizados para detecção de objetos que modelam a imagem de referência com um modelo gaussiano.	159
Figura 97: Exemplos de aplicações que efetuam a redução da resolução da imagem para atender aos requisitos de tempo real.	160
Figura 98: Autores que utilizam a adaptação recursiva para compensar as variações de iluminação.	160

Figura 99: Velocidade de processamento de arquiteturas implementadas em hardware para detecção de objetos.....	162
--	-----

LISTA DE TABELAS

Tabela 1: Número de elementos de imagens com valores compreendidos no intervalo de confiança para cada componente de cor do espaço YCbCr em cada seqüência de imagem de teste ilustrada na Figura 7.....	43
Tabela 2: Parâmetros utilizados para obtenção das métricas de avaliação de métodos de detecção de objetos. TP: verdadeiro positivo; TN: verdadeiro negativo; FP: Falso positivo; FN: Falso Negativo.....	60
Tabela 3: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 01 com três tipos de classificação dos <i>pixels</i>	66
Tabela 4: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 02 com três tipos de classificação dos <i>pixels</i>	68
Tabela 5: Comparação dos pares de parâmetros para o teste de Kruskal Wallis (p^*) das medidas F para a seqüência de imagens 03 com três tipos de classificação dos <i>pixels</i>	71
Tabela 6: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 04 com três tipos de classificação dos <i>pixels</i>	74
Tabela 7: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 05 com três tipos de classificação dos <i>pixels</i>	76
Tabela 8: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 01 com quatro espaços de cores.....	79
Tabela 9: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 04 com quatro espaços de cores.....	82
Tabela 10: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 05 com quatro espaços de cores.....	84
Tabela 11: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 01 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante.....	102

Tabela 12: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 02 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante.....	104
Tabela 13: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 03 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante.....	106
Tabela 14: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 04 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante.....	108
Tabela 15: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 05 com dois tipos de classificação dos <i>pixels</i> e operações em ponto fixo e flutuante.....	110
Tabela 16: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 03 com diferentes métodos para calcular os valores da média e do desvio padrão.	125
Tabela 17: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 04 com diferentes métodos para calcular os valores da média e do desvio padrão.	127
Tabela 18: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 05 com diferentes métodos para calcular os valores da média e do desvio padrão.	129
Tabela 19: Resultados das métricas obtidas com as implementações em hardware e MATLAB utilizando-se a seqüência de imagens ilustrada na Figura 92.....	151
Tabela 20: Resultados das métricas obtidas com a implementação em hardware utilizando-se a seqüência de imagens ilustrada na Figura 93.....	152
Tabela 21: Resultados da síntese	154
Tabela 22: Comparação de métodos para detecção de objetos utilizando uma imagem de referência estática [72].....	158

RESUMO

A segmentação de vídeo é um passo fundamental em muitos sistemas de visão, tais como sistemas de vigilância e monitoramento de tráfego. O método denominado subtração da imagem de fundo é comumente utilizado para detecção de objetos em seqüências de vídeo comparando-se cada *pixel* do quadro corrente com um modelo da imagem de referência. Neste trabalho, apresenta-se uma arquitetura em hardware para segmentação de vídeo desde a etapa de implementação do algoritmo em PC até a elaboração da arquitetura em hardware. O método de segmentação de vídeo destina-se ao processamento de operações em ponto fixo e visa aprimorar o método de detecção de objetos baseado em modelos Gaussianos. Este aprimoramento é realizado por meio da aplicação de uma técnica para compensação das variações das intensidades dos *pixels* que objetiva reduzir os falsos positivos ocasionados por ruídos ou variações de iluminação. Primeiramente, o algoritmo foi validado em MATLAB em ponto flutuante e em ponto fixo. Em seguida, foi implementado em um arranjo de portas programáveis em campo (FPGA), utilizando um kit desenvolvimento da Altera (DE-2). A arquitetura opera com uma frequência igual a 100 MHz e processa 30 quadros por segundo com resolução igual a 640 x 507. A capacidade do sistema é demonstrada com várias imagens de teste.

Palavras-chaves:

Segmentação de vídeo, FPGA, tempo real, detecção de objetos.

ABSTRACT

Video segmentation is a fundamental step in many vision systems including video surveillance and traffic monitoring. Background subtraction is a method typically used to segment moving regions in video sequences taken from a static camera by comparing each new frame to a model of the scene background. In this paper, a hardware system for video segmentation is proposed from algorithm to hardware architecture level. The video segmentation algorithm is aimed at fixed-point operations and improves a Gaussian background model by applying a two-stage linear compensation procedure to remove the undesirable subtraction results from noise and illumination changes. First, the algorithm was validated in MATLAB. Then, it was prototyped on an Altera field-programmable gate array platform (DE-2). At a clock rate of 100 MHz, the architecture can process 30 frames per second, where the image resolution is 640 x 507 pixels. The capability of the system is demonstrated for several video sequences.

Keywords:

Video Segmentation, FPGA, real time, object detection.

INTRODUÇÃO

O processamento de vídeo em tempo real é uma área multidisciplinar e tem experimentado um crescimento significativo nas últimas décadas com aplicações em sistemas de inspeção industrial, produtos de eletrônica de consumo, tais como câmeras digitais e telefones celulares, e sistemas de monitoramento e segurança [1, 2, 3]. Neste contexto, tempo real é definido como o intervalo de tempo necessário para o sistema processar cada *pixel*¹ dentro do tempo disponível entre duas amostras sucessivas de *pixels* em cada quadro. Além do tempo de processamento de cada *pixel*, o tempo necessário para transferência de cada quadro e execução de outras operações relacionadas com a transferência de dados para memória, também devem ser considerados, pois são fatores críticos em muitos sistemas práticos. [4]

O processamento de vídeo em tempo real requer o processamento de uma quantidade significativa de dados, em função dos sinais multidimensionais envolvidos, e uma quantidade significativa de memória [5]. Por exemplo, uma câmera digital de vídeo capturando imagens coloridas com uma resolução igual a 640 x 480 *pixels* e uma taxa de 30 quadros por segundo demanda um sistema capaz de processar 27 milhões de *pixels* por segundo (640x480x30x3). Com as tendências de aumento da resolução e da taxa de quadros em aplicações com imagens em alta resolução, a quantidade de *pixels* a ser processada deverá ser cada vez maior e os sistemas devem lidar com este crescente aumento na quantidade de dados. Uma abordagem utilizada para contornar este problema é o processamento paralelo dos dados. A maioria dos sistemas desenvolvidos para aplicações em processamento de vídeo em tempo real, concentra-se em explorar as diferentes formas de paralelismo no algoritmo para implementação do hardware e software para atender aos requisitos do projeto. [6, 7]

A complexidade das operações de processamento de vídeo podem ser classificadas em baixo, intermediário e alto, diferenciando-se pela relação entre os dados de entrada e saída [6, 8, 9, 10]. Os sistemas que realizam operações de nível baixo capturam imagens na entrada e produzem imagens na saída. Exemplos de tais operações incluem as conversões entre espaços de cores, correção gama, filtragem, redução de ruídos, transformações no domínio da frequência, etc. O principal objetivo destas operações é enfatizar determinadas características da imagem, preparando-as para visualização humana, ou para extração de características a

¹ *Pixel* – Derivado da expressão em inglês *picture element* (elemento de imagem).

serem processadas no nível intermediário. As operações de nível baixo podem ser classificadas como pontuais, locais e globais. [11, 8]

Nas operações pontuais, cada *pixel* de saída é função apenas do *pixel* de entrada. Neste caso, incluem-se as operações aritméticas, lógicas, e as operações que utilizam tabelas de consultas. Nas operações locais, o *pixel* de saída é função do *pixel* de entrada e da sua vizinhança. Exemplos de operações locais são convolução, filtragem, e a suavização.

Nas operações globais, cada *pixel* de saída depende de uma operação realizada em todos os *pixels* de um quadro. Os sistemas que realizam operações em baixo nível aplicam um operador pontual, local ou global nos *pixels* do quadro em processamento, o que pode ser realizado de modo paralelo.

As operações em nível intermediário processam uma imagem de entrada e geram na saída atributos ou características da imagem. Neste caso, há geralmente uma redução na quantidade de dados de saída em relação à entrada. As operações em nível intermediário incluem a segmentação de imagem ou vídeo, detecção de bordas, detecção de linhas, detecção de contornos, ou outros atributos tais como as características estatísticas de cada quadro.

As operações em nível alto interpretam os dados abstratos do nível intermediário para efetuar a classificação e o reconhecimento de objetos ou o controle de decisões baseado na extração de parâmetros da imagem. Um sistema completo para aplicações em processamento de vídeo combina os três níveis de operações, isto é, captura dos quadros, pré-processamento (redução de ruídos), segmentação, análise (extração de características) e interpretação (reconhecimento).

Entre as aplicações mais crescentes na área de processamento de vídeo estão as que realizam a segmentação de vídeo para detecção de objetos, comumente utilizada em sistemas de vigilância e rastreamento de pessoas [12, 13, 14, 15, 16]. Os sistemas desenvolvidos para estas aplicações devem realizar o processamento em tempo real e lidar com os problemas de variações das intensidades dos *pixels* causados por ruídos gerados pela câmera ou pela iluminação do ambiente.

Os arranjos de lógica programável em campo (FPGA) têm sido utilizados em diversas aplicações na área de processamento de vídeo em tempo real, tais como detecção de bordas [17], compressão de dados [18], detecção de faces [19], segmentação de vídeo [20], entre outras. Um FPGA permite a elaboração de estruturas paralelas flexíveis que podem executar operações em tempo real.

O desenvolvimento de sistemas de processamento de vídeo em tempo real é uma tarefa desafiadora. Determinados compromissos de projeto devem ser realizados durante a

transição de um algoritmo de um ambiente de desenvolvimento para uma operação em tempo real propriamente dita. Aspectos práticos importantes, tais como velocidade, desempenho, robustez, adaptabilidade, flexibilidade, e o custo total do sistema devem ser considerados para o desenvolvimento do projeto. [21]

1.1 MOTIVAÇÃO E JUSTIFICATIVA

A segmentação de vídeo é uma etapa fundamental em sistemas destinados à detecção ou extração de objetos de uma seqüência de imagens. Uma outra aplicação da segmentação de vídeo ocorre em estúdios de produção de vídeo em que o método denominado *chroma key* é muito utilizado. O objetivo do *chroma key* é extrair um objeto de uma imagem de fundo uniforme e homogênea conhecida e inserir os *pixels* do objeto em uma segunda imagem [22].

Um dos métodos utilizados para detecção de objetos em uma seqüência de imagens é denominado subtração da imagem de fundo [23, 24, 25]. Um modelo da imagem de fundo ou referência é construído durante uma fase de aprendizado enquanto a cena, imagem de fundo ou referência, está sem nenhum objeto a ser detectado. Na fase de classificação, o objeto é detectado *pixel a pixel* sempre que difere significativamente da distribuição dos valores esperados contidos no modelo da imagem de referência.

O método escolhido para a detecção de objetos deve considerar alguns fenômenos que comumente ocorrem nos ambientes não controlados em que o sistema esteja inserido. Os fenômenos mais comuns são [26]: mudança de iluminação, sombras e reflexões, mudanças de objetos na imagem de referência, e identificação da imagem de referência em ambientes com objetos em movimento.

As variações de iluminação e as compensações efetuadas pela câmera podem dificultar a detecção dos objetos, pois mudanças nas intensidades dos *pixels* geram a detecção de falsos positivos, ou seja, *pixels* que pertencem à imagem de referência podem ser erroneamente classificados como objetos. Portanto, deve-se incluir, no modelo da imagem de referência, uma técnica para adaptação ou compensação das variações de iluminação do ambiente.

Uma das técnicas utilizadas para compensar as variações de iluminação consiste em computar continuamente os parâmetros estatísticos de cada quadro e efetuar a atualização do modelo da imagem de referência após um tempo pré-estabelecido [27, 28, 29]. Desta maneira, os parâmetros são armazenados em uma memória cujo tamanho depende do número de

quadros a ser processado. O tempo necessário para aquisição dos novos parâmetros determina a velocidade de atualização do sistema.

Uma outra técnica utilizada para compensar as variações de iluminação consiste em efetuar, para cada quadro processado, uma soma ponderada entre os parâmetros do modelo da imagem de referência e os parâmetros calculados para os valores correntes dos *pixels*. O resultado desta operação é utilizado para atualizar o modelo da imagem de referência. A velocidade de atualização depende do peso atribuído aos parâmetros do modelo da imagem de referência [30, 31, 32, 33]. Esta técnica é comumente utilizada em sistemas de segmentação de vídeo cujo modelo da imagem de referência é paramétrico (gaussiano). Nestes casos, o valor médio e o desvio padrão de cada *pixel* são atualizados quadro a quadro.

As técnicas de atualização supracitadas podem ser inadequadas para aplicações em sistemas de vigilância, pois uma pessoa, dependendo da sua velocidade de deslocamento, pode ser integrada à imagem de referência, não sendo detectada como objeto. Para solucionar este problema alguns métodos efetuam uma atualização seletiva, em que os *pixels* detectados previamente como objetos não são atualizados no modelo da imagem de referência [34]. No entanto, se forem detectados falsos positivos, o sistema poderá permanentemente excluí-los da atualização, ocasionando a exibição de *pixels* pertencentes à imagem de referência.

Em função das limitações apresentadas dos métodos pesquisados, a problemática estabelecida para esta pesquisa foi a seguinte: de que maneira é possível compensar as variações de luminosidade decorrentes da iluminação ambiente ou de correções automáticas efetuadas pela câmera processando-se o sinal de vídeo na taxa de 30 quadros por segundo?

Neste trabalho é apresentado um método paramétrico (gaussiano) para detecção de objetos de uma seqüência de imagens que utiliza uma técnica para compensação quadro a quadro das variações das intensidades dos *pixels*. Os parâmetros do modelo da imagem de referência são os valores médios e os desvios padrões com os respectivos intervalos de confiança para cada *pixel*. Antes da etapa de detecção dos objetos, o valor corrente de cada *pixel* é comparado com o seu respectivo intervalo de confiança, e os que são identificados fora deste intervalo tem a sua diferença absoluta reduzida por uma função definida empiricamente. Os resultados experimentais demonstram uma redução na detecção de falsos positivos em ambientes com variação de iluminação.

A maioria dos métodos propostos para detecção de objetos são desenvolvidos para implementações em software e tem uma complexidade computacional relativamente alta para aplicações em tempo real (30 quadros/s) e de custo relativamente baixo [15, 16]. A aplicação do método apresentado neste trabalho visa o mercado de eletrônica de consumo com

aplicações, por exemplo, na área de segurança para detecção de intrusos em ambientes internos; de entretenimento inserido em jogos eletrônicos; e em sistemas de edição de vídeo em ambientes internos. A aplicação que inspirou o desenvolvimento desta pesquisa foi na área de entretenimento em aparelhos de Videokê para inserção do cantor no videoclipe. No final de 2003, um instituto privado de pesquisas desenvolveu um algoritmo para avaliação de canto para sistemas de karaokê. Este algoritmo foi embarcado em um aparelho de videoke e teve uma boa aceitação no mercado. Na época, cogitou-se a possibilidade de inserção do cantor no videoclipe como aprimoramento do sistema. Desta maneira, os requisitos essenciais são os seguintes:

- a) Custo relativamente baixo: No mercado de eletrônica de consumo um produto é competitivo se oferecer ao consumidor um custo relativamente baixo e uma boa qualidade de desempenho. Em função do alto volume de produção, alguns centavos no custo de fabricação são significativos no preço do produto final. Para atender ao requisito de custo relativamente baixo, elaborou-se um método para detecção de objetos que utiliza operações de baixa complexidade computacional para implementação em hardware.
- b) Simplicidade de uso: O procedimento para operação do produto deve ser o mais simples possível afim de que todos os usuários possam utilizar facilmente todas as funções do produto. Para atender ao requisito de simplicidade de uso, o sistema implementado neste trabalho requer apenas que o usuário pressione uma chave para iniciar as etapas de aprendizagem da imagem de referência e detecção do objeto.
- c) Robustez a variações de iluminação do ambiente: A iluminação do ambiente pode variar em função da simples abertura ou do fechamento de uma janela no ambiente em que o sistema esteja inserido, ou ainda, devido a compensações automáticas realizadas pela câmera. O método implementado neste trabalho se propõe a minimizar os erros de detecção gerados por variações de iluminação.

Nos artigos pesquisados ao longo do desenvolvimento desta tese, há alguns trabalhos que utilizam estruturas paralelas em hardware para implementação de sistemas para detecção de objetos em tempo real. As estruturas comumente utilizadas são: dispositivos de lógica programável (FPGA) [35, 36, 37, 38, 20] ou circuitos integrados com aplicação específica [39, 40]. Nesta tese é apresentada uma arquitetura implementada em FPGA para validação do

método de segmentação de vídeo para detecção de objetos. Foram feitas algumas simplificações no método de segmentação visando à redução da quantidade de operações e o número de acessos à memória externa, onde é armazenado o modelo da imagem de referência.

1.2 OBJETIVOS

Os objetivos estabelecidos para este trabalho são os seguintes:

- a) Desenvolver um método de segmentação de vídeo para detecção de objetos, considerando uma imagem de referência estática e câmera fixa, com tolerância às sombras e variações de iluminação para aplicações em tempo real (30 quadros por segundo).
- b) Validar em hardware o método de segmentação de vídeo para detecção de objetos utilizando um dispositivo de lógica programável;

1.3 METODOLOGIA

Para atingir os objetivos estabelecidos neste trabalho, procedeu-se da seguinte maneira:

- a) Primeiramente, desenvolveu-se o método de segmentação de vídeo para detecção de objetos efetuando-se operações em ponto flutuante em MATLAB. Esta versão foi avaliada em relação à detecção dos objetos utilizando-se imagens de outros autores e imagens próprias capturadas em diversos cenários. Além disso, comparou-se também o seu desempenho utilizando-se diferentes espaços de cores, quais sejam RGB, YCbCr, HSV e YUV [11]. Finalmente, os resultados foram comparados com os obtidos de um método que atualiza continuamente os parâmetros do modelo da imagem de referência;
- b) Após a validação do método com operações em ponto flutuante, foram feitas modificações para execução de operações em ponto fixo. O desempenho da implementação em ponto fixo foi comparado com os resultados obtidos anteriormente com as operações em ponto flutuante;
- c) A partir da versão do método proposto em ponto fixo, foram feitas algumas simplificações para redução da quantidade de memória necessária para capturar e armazenar o modelo da imagem de referência. Os resultados obtidos com estas

simplificações foram comparados com os anteriores obtidos com operações em ponto fixo.

- d) Com o método validado com operações em ponto fixo e com as simplificações foram feitas a definição e elaboração da arquitetura de hardware necessária para implementação do método proposto utilizando-se um dispositivo de lógica programável. Foram realizados teste de desempenho da arquitetura em um cenário com variações de iluminação.

1.4 CONTRIBUIÇÕES DA TESE

As principais contribuições apresentadas nesta tese são descritas a seguir:

- a) Desenvolvimento de um método para compensação das variações das intensidades dos *pixels*, devido às mudanças de iluminação no cenário, ruído e compensações automáticas efetuadas pela câmera. A utilização desta compensação em aplicações em que a imagem de referência é estática dispensa a atualização contínua do modelo da imagem de referência. De acordo com os testes realizados, para uma variação média global de luminância igual a 25%, a porcentagem de falso positivo é reduzida de 70% para 12% com a utilização da função de compensação de variação do *pixel*;
- b) Aplicação de métodos estatísticos para análise dos parâmetros do método proposto e para comparação de desempenho. Os métodos de avaliação encontrados na literatura são baseados em métricas que levam em consideração o número de falsos positivos e negativos obtidos para cada quadro processado. A fim de comparar os resultados obtidos em cada cenário de teste e as diversas variáveis envolvidas em cada avaliação de modo objetivo, foram utilizados métodos não paramétricos de análise de variância para verificar se as diferenças observadas entre as métricas de avaliação são estatisticamente significantes. Isto permitiu a análise de desempenho de métodos de segmentação de vídeo em diversos cenários e com diferentes parâmetros.
- c) Aplicação de estimativas estatísticas para determinação do número de quadros para determinar os parâmetros do modelo da imagem de referência. O número de quadros é um ponto crucial na estimativa do valor médio e do desvio padrão de cada *pixel*. Além disso, o número de quadros também determina o tempo

necessário para determinação do modelo da imagem de referência. Utilizando-se as estimativas aplicadas para determinação do número de amostras, foi possível estabelecer um número mínimo de quadros, levando-se em consideração um compromisso entre o erro da estimativa e o tempo necessário para aquisição do modelo imagem de referência. Com base nestas estimativas foi possível reduzir o tempo de aquisição do modelo da imagem de referência para menos que 1 segundo.

- d) Elaboração e implementação de uma arquitetura de hardware utilizando o método desenvolvido para detecção de objetos. A arquitetura elaborada foi organizada de tal modo que cada bloco de hardware recebe um conjunto de parâmetros de entrada e disponibiliza na saída o resultado do processamento juntamente com os parâmetros recebidos na entrada. Desta maneira, não foi necessário utilizar *buffers* para armazenamento dos resultados intermediários em cada etapa do processamento. Cada bloco processa simultaneamente as três componentes do espaço de cores YCbCr. A implementação em hardware permitiu validar o método proposto para detecção de objetos em tempo real (30 quadros por segundo) e verificar a quantidade de elementos lógicos necessários para esta aplicação. Esta informações são necessárias para a otimização da implementação em um sistema embarcado.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este documento é dividido em sete capítulos, além das referências bibliográficas.

No capítulo 2 são apresentados os métodos encontrados na literatura citada na referência bibliográfica para detecção de objetos que utilizam a técnica denominada subtração da imagem de fundo.

No capítulo 3 é apresentado o método proposto neste trabalho para segmentação de vídeo com operações em ponto flutuante. É feita uma análise da função de compensação, proposta para minimizar os efeitos causados pela variação de iluminação.

No capítulo 4 são descritas as métricas utilizadas para avaliação dos resultados. O método proposto é avaliado com diferentes abordagens para detecção de objetos. Além disso, faz-se também avaliação do desempenho do método proposto em diferentes espaços de cores, isto é, YCbCr, RGB, YUV e HSV. Finalmente, compara-se o método proposto com um método que utiliza atualização contínua dos parâmetros do modelo da imagem de referência.

No capítulo 5 são apresentadas as adaptações realizadas para implementação do método proposto com operações em ponto fixo e os resultados das comparações com as implementações em ponto flutuante e ponto fixo. Além disso, são apresentadas as simplificações para redução do tamanho da memória necessária para aquisição e armazenamento do modelo da imagem de referência.

No capítulo 6 é apresentada a arquitetura em hardware utilizada para implementar cada operação do método proposto para segmentação de vídeo. Os resultados de desempenho e da síntese desta arquitetura também são descritos neste capítulo.

No capítulo 7 são mencionadas as conclusões do trabalho e as sugestões para trabalhos futuros.

2 MÉTODOS PARA SEGMENTAÇÃO DE VÍDEO

Neste capítulo apresentam-se os métodos pesquisados, ao longo do desenvolvimento deste trabalho, para detecção de objetos e as suas principais características.

2.1 INTRODUÇÃO

A detecção de objetos de um sinal de vídeo é o primeiro passo relevante para extração de informação em muitas aplicações de visão computacional, incluindo sistemas de segurança, rastreamento de pessoas e monitoramento de veículos. Nestas aplicações, a robustez do rastreamento de objetos na cena depende da detecção dos objetos em tempo real que pode ser caracterizada pela flexibilidade a diferentes cenários e condições de iluminação.

Considera-se neste trabalho cenas adquiridas por uma câmera em posição fixa, cuja imagem de fundo pode ser considerada estacionária e cujos objetos de interesse podem estar em movimento ou parados. Em tais cenas, o movimento dos objetos é percebido observando-se a variação dos valores dos seus *pixels* quadro a quadro. Nestas condições, a variação dos valores dos *pixels* pode ser consequência de diferentes causas, isto é, um objeto realmente em movimento ou uma possível mudança na intensidade de brilho da imagem de fundo devido a variações de iluminação no ambiente ou compensações realizadas pela câmera de vídeo.

Neste trabalho considera-se que o modelo, atributos, forma e textura dos objetos alvos, bem como seus movimentos são desconhecidos, para deste modo atingir a máxima independência da aplicação. Além disso, não se assume o conhecimento prévio do ambiente e das suas condições de iluminação. Na ausência de qualquer conhecimento a priori do objeto e do ambiente, o método adotado para detecção de objetos em movimento é baseado na subtração da imagem de fundo.

Na Figura 1 é mostrado o diagrama em blocos básico de um sistema para detecção de objetos baseado na subtração da imagem de fundo [41, 42, 43]. Primeiramente, as estatísticas da imagem de referência são modeladas para cada *pixel* e armazenadas em um modelo de referência. A partir das características do modelo de referência são estabelecidos os limiares ou o limiar para classificação de cada *pixel*. Na fase de classificação, a imagem do primeiro espaço é detectada *pixel a pixel* sempre que difere significativamente da distribuição dos valores esperados contidos no modelo da imagem de referência. A imagem de referência, também citada nos artigos pesquisados como imagem de fundo, pode ser estática ou dinâmica.

No caso da estática, não é realizada nenhuma retirada ou inserção de objetos após a aquisição do modelo da imagem de referência. Em cenários com imagens dinâmicas, como por exemplo, em um estacionamento, acontecem modificações após a aquisição do modelo da imagem de referência.

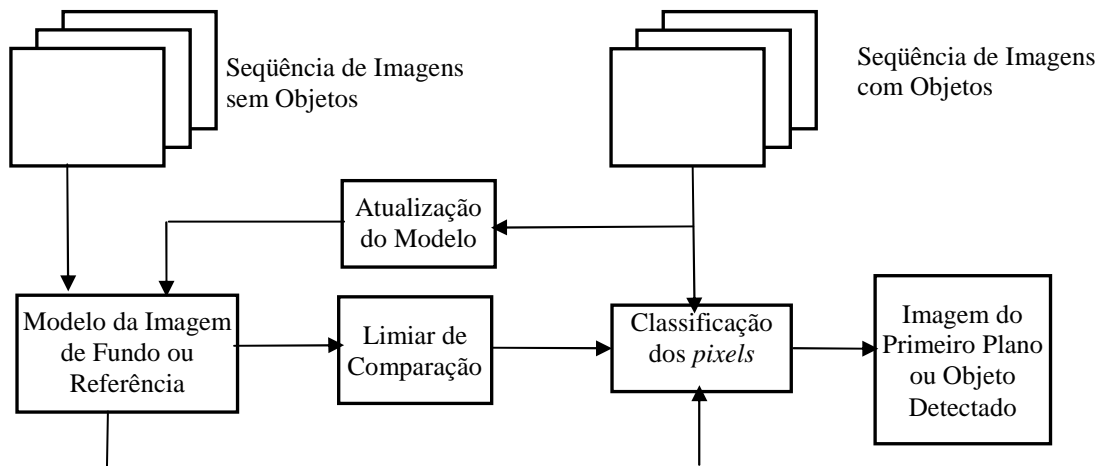


Figura 1: Diagrama em Blocos Básico de um Sistema para Detecção de Objetos

2.2 TRABALHOS RELACIONADOS

As características dos métodos pesquisados, ao longo do desenvolvimento desta tese, para detecção de objetos são as seguintes: modelo da imagem de referência, atualização do modelo da imagem de referência e tamanho da memória para armazenar os parâmetros da imagem de referência. A seguir são descritas as características dos métodos pesquisados.

2.2.1 Modelo da imagem de referência

Quanto ao modelo da imagem de referência, os métodos pesquisados estão classificados em gaussianos [41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52], misturas de modelos gaussianos [53, 54, 55, 31, 32, 56, 57, 58, 59] e não paramétricos [60, 61, 27, 47, 33, 62]. Estes métodos tendo sido utilizado nos últimos dez anos (1999 à 2009) para detecção de objetos utilizando a técnica denominada de subtração da imagem de fundo.

Os métodos de detecção de objetos que modelam a imagem de referência utilizando um modelo gaussiano consideram que cada *pixel* no espaço de cores tem uma distribuição normal. Desta maneira, o modelo da imagem de referência é construído com os valores médios e os desvios padrões, para cada *pixel*, calculados com um determinado número de

quadros. Os valores médios de cada *pixel* são utilizados como estimativas pontuais dos verdadeiros valores médios da população. Os desvios padrões expressam a dispersão dos valores de cada *pixel* em função de ruídos e variações de iluminação no ambiente em que o sistema esteja inserido. Estas estimativas dependem do número de quadros utilizados (N). Nos artigos pesquisados, o cálculo do valor de N não é mencionado, o que pode sugerir uma determinação empírica deste parâmetro. De acordo com os resultados obtidos em [63], para ambientes internos a utilização de um modelo gaussiano é suficiente para modelagem da imagem de referência. Durante a fase de detecção dos objetos, calcula-se a variação de intensidade de cada *pixel* em relação ao valor médio armazenado no modelo da imagem de referência e, em seguida, compara-se esta variação com o limiar baseado no desvio padrão. Os *pixels* cujos valores da variação estejam acima dos seus limiares, são classificados como objetos.

Os métodos de detecção de objetos que modelam a imagem de referência utilizando K modelos gaussianos consideram que cada *pixel* no espaço de cores tem uma mistura de gaussianas. Desta maneira, cada *pixel* é modelado por um número K de valores médios e desvios padrões. Estes métodos são utilizados em ambientes com constantes mudanças na imagem de referência, como por exemplo, um cenário com uma árvore cujos galhos estejam balançando. O tempo de computação dos parâmetros da imagem de referência depende das mudanças que estejam ocorrendo no cenário no instante da aquisição dos quadros para modelagem da imagem de referência, pois o método de segmentação deve ser capaz de distinguir entre um objeto em movimento, que não pertence à imagem de referência, e um outro que pertence à imagem de referência. O número K de modelos gaussianos pode variar entre 3 e 5 [31]. Na fase de segmentação, as intensidades de cada *pixel* corrente são comparadas com os desvios padrões das distribuições gaussianas do modelo da imagem de referência. Os *pixels* cujas intensidades estejam 2.5 vezes acima de um dos desvios padrões armazenados no modelo da imagem de referência são classificados como objetos. [31]

Os métodos de detecção de objetos que modelam a imagem de referência utilizando modelos não-paramétricos consideram que cada *pixel* no espaço de cores não tem uma distribuição conhecida e fazem uma estimativa da distribuição de cada *pixel*. Um dos métodos não-paramétricos utilizados para detecção de objetos modela a imagem de referência com 04 parâmetros, isto é, valor médio, desvio padrão, distorção de brilho e distorção de cor [61]. As distorções de brilho e de cor correspondem às diferenças de brilho e de cor de cada *pixel* corrente em relação aos valores médios previamente calculados. Além disso, a partir dos valores médios das distorções de brilho e cor são gerados os seus respectivos histogramas

utilizados para determinar automaticamente os limiares de comparação. Durante a fase de detecção dos objetos são calculadas as distorções de brilho e cor, para cada *pixel*, e estes valores são comparados com os limiares previamente determinados. Um *pixel* corrente é classificado como objeto se a sua distorção de cor for maior que o limiar estabelecido. Caso o *pixel* apresente uma distorção de cor e de brilho inferiores aos limiares de comparação, ele é classificado como pertencente à imagem de referência.

2.2.2 Atualização do modelo da imagem de referência

Uma das técnicas utilizada para compensar as variações de iluminação e mudanças de objetos na imagem de referência é conhecida como atualização recursiva [30, 31, 32, 33, 42, 45, 51, 54]. Esta técnica consiste em efetuar a atualização os parâmetros da imagem de referência, quadro a quadro, para cada *pixel*. Considerando um modelo gaussiano, os parâmetros seriam atualizados da seguinte maneira:

$$\begin{cases} \mu_t = (1 - \alpha)\mu_{t-1} + \alpha I^t \\ \sigma_t^2 = (1 - \alpha)\sigma_{t-1}^2 + \alpha(I^t - \mu_t)^2 \end{cases} \quad (1)$$

Em que:

- a) μ_{t-1} é o valor médio de cada *pixel* no modelo da imagem de referência;
- b) σ_{t-1} é o desvio padrão de cada *pixel* no modelo da imagem de referência;
- c) I^t é o valor corrente de cada *pixel*;
- d) α determina a taxa de atualização dos parâmetros no modelo da imagem de referência.

A taxa de atualização (α) deve ser escolhida de acordo com a velocidade de movimento dos objetos em cena, pois objetos que estejam se movimentando com uma velocidade inferior a taxa de atualização, ou que estejam parados são incorporados ao modelo da imagem de referência. Para solucionar este problema alguns métodos efetuam uma atualização seletiva [34, 64], em que os *pixels* detectados previamente como objetos não são atualizados no modelo da imagem de referência. Neste caso, se houver falsos positivos, o

sistema poderá permanentemente excluí-los da atualização, ocasionando a exibição de *pixels* pertencentes à imagem de referência.

Outra técnica utilizada para compensar as variações de iluminação e mudanças de objetos na imagem de referência consiste em computar continuamente os parâmetros estatísticos de cada quadro e efetuar a atualização do modelo da imagem de referência após um determinado número de quadros [27, 28, 29, 56]. Desta maneira, é necessária a utilização de uma memória para armazenar os parâmetros durante um tempo preestabelecido. O número de quadros a ser processado determina o tamanho da memória e, por conseguinte, a velocidade de atualização do sistema. Esta técnica de atualização é utilizada para aplicações em ambientes externos cujos objetos da imagem de referência estejam em constante movimento. No entanto, se a taxa de atualização for inferior ao tempo de variação dos objetos, falsos positivos serão detectados, isto é, os objetos pertencentes à imagem de fundo que estejam em movimento são exibidos na tela.

As técnicas de atualização supracitadas podem ser inadequadas para aplicações em sistemas de vigilância, pois uma pessoa parada é integrada a imagem de referência, não sendo detectada como objeto.

2.2.3 Tamanho da memória para armazenar o modelo da imagem de referência

O modelo da imagem de referência determina o tamanho da memória necessário para armazenar os parâmetros para efetuar a detecção dos objetos. As estimativas do tamanho da memória, apresentadas a seguir, foram efetuadas considerando imagens coloridas com uma resolução igual a 640 x 480 *pixels*, uma taxa de 30 quadros por segundo, e 8 bits para representação de cada *pixel*. Assim sendo, o tamanho da memória (TM) em bytes pode ser obtido da seguinte maneira:

$$TM = RESH \times RESV \times NP \times NC \quad (2)$$

Em que:

- a) RESH é a resolução horizontal da imagem;
- b) RESV é a resolução vertical da imagem;
- c) NP é o número de parâmetros a ser armazenado;
- d) NC é o número de componentes do espaço de cores.

O tamanho da memória necessário para armazenar um modelo paramétrico (gaussiano), isto é, os valores médios e os desvios padrões dos *pixels*, é igual a 1,8 Mbytes (640 x 480 x 3 x 2). A quantidade de quadros necessária para estimar os parâmetros da imagem de referência determina o tempo de modelagem da imagem de referência. Por exemplo, considerando 300 quadros para aquisição dos parâmetros, o tempo seria igual a 10 s. Outro aspecto importante é a quantidade de acessos à memória. Se o método de detecção de objetos realiza a atualização recursiva, para cada *pixel* processado é necessário efetuar a leitura da média e do desvio padrão da memória e a escrita dos valores atualizados da média e do desvio padrão. Supondo que o sinal de vídeo está sendo capturado com o padrão M, cada linha horizontal tem 63,5 μ s [65, 66]. Considerando-se 640 *pixels* por linha, o sistema deve processar os 640 *pixels* durante 63,5 μ s, isto é, efetuar as leituras e escritas na memória e as demais operações para segmentação de vídeo.

Os métodos de detecção de objetos baseados em três modelos gaussianos necessitam de uma memória com tamanho igual a 5,5 Mbytes para armazenar o modelo da imagem de referência. Neste caso, considerando 300 quadros para aquisição de cada modelo gaussiano, o tempo necessário para modelagem da imagem de referência seria igual a 30 s. Além disso, se os parâmetros estatísticos forem computados continuamente, para a atualização os modelos gaussianos, é necessária uma outra memória com tamanho igual a 5,5 Mbytes para armazenar os 03 modelos gaussianos mais recentes.

O método não-paramétrico apresentado em [61] e utilizado por outros autores [27, 47, 62], requer uma memória com tamanho igual a 3,6 Mbytes para armazenar os parâmetros do modelo da imagem de referência, isto é, os valores médios, desvios padrões, médias das distorções de brilho e médias das distorções de cor. Se for utilizada a atualização recursiva, para cada *pixel* processado, são necessários oito acessos à memória, isto é, quatro para leitura e quatro para escrita dos parâmetros.

2.3 CONCLUSÕES

Neste capítulo foram apresentados os métodos pesquisados, ao longo do desenvolvimento deste trabalho, para detecção de objetos e as suas principais características, bem como o método proposto neste trabalho. As características dos métodos pesquisados são as seguintes: modelo da imagem de referência, atualização do modelo da imagem de referência e tamanho da memória para armazenar os parâmetros da imagem de referência.

Quanto ao modelo da imagem de referência, os métodos pesquisados estão classificados gaussianos, misturas de modelos gaussianos e não paramétricos. Considerando imagens coloridas com uma resolução igual a 640×480 *pixels*, uma taxa de 30 quadros por segundo, e 8 bits para representação de cada *pixel*, o tamanho da memória necessário para armazenar o modelo a imagem de referência baseado em modelos gaussianos é igual a 1,8 MBytes. Entre os métodos pesquisados que utilizam a técnica de subtração de fundo para detecção de objetos, os baseados em modelos gaussianos apresentam a menor complexidade computacional, ou seja, tamanho da memória para armazenar os parâmetros do modelo da imagem de referência e quantidade de operações realizadas para detecção dos objetos.

As técnicas comumente utilizadas para compensar as variações de iluminação e mudanças de objetos na imagem de referência, atualizam continuamente o modelo da imagem de referência e podem ser inadequadas para aplicações em sistemas de vigilância, pois uma pessoa parada é integrada a imagem de referência, não sendo detectada enquanto não se deslocar ou se a sua velocidade de movimento for inferior à taxa de atualização do modelo da imagem de referência.

No próximo capítulo é apresentada a solução proposta para detecção de objetos desenvolvida ao longo desta pesquisa.

3 MÉTODO PROPOSTO PARA SEGMENTAÇÃO DE VÍDEO

Neste capítulo é apresentada a solução proposta para detecção de objetos. Em relação ao contexto do atual estado da arte, o método proposto está inserido entre os que utilizam um modelo gaussiano para modelagem da imagem de referência.

3.1 CONSIDERAÇÕES INICIAIS DA SOLUÇÃO PROPOSTA

Em função das características dos métodos pesquisados e visando, sobretudo, a implementação em hardware de um sistema para detecção de objetos para inserção do cantor em um videoclipe, foram estabelecidos os seguintes requisitos para o método implementado neste trabalho:

- a) O sistema utilizará uma única câmera fixa;
- b) A imagem de fundo ou referência será estática;
- c) A fase de aprendizado, isto é, a aquisição do modelo da imagem de referência deverá ser efetuada sem a presença de objetos que posteriormente modifiquem a sua posição inicial;
- d) O modelo da imagem de referência será gaussiano. No entanto, serão elaboradas algumas simplificações para redução da quantidade de memória necessária para armazenar os parâmetros do modelo da imagem de referência;
- e) O modelo da imagem de referência não será atualizado segundo as técnicas apresentadas na seção anterior. Para cada quadro processado, a variação de cada *pixel* será reduzida antes da etapa de detecção de objetos para minimizar a detecção de falsos positivos em função de ruídos e variações de iluminação.

3.2 SOLUÇÃO PROPOSTA

Nesta seção são descritos os fundamentos do método proposto para detecção de objetos. As etapas para segmentação de vídeo são as seguintes: modelagem da imagem de referência, seleção do nível de limiar, compensação de variações, detecção de sombras e extração do objeto. Além disso, é feito um monitoramento constante da variação de luminância da imagem corrente em relação à imagem de referência a fim de se detectar

mudanças de iluminação que demandem uma re-aquisição do modelo da imagem de referência. O diagrama em blocos com as etapas do método proposto é ilustrado na Figura 2.

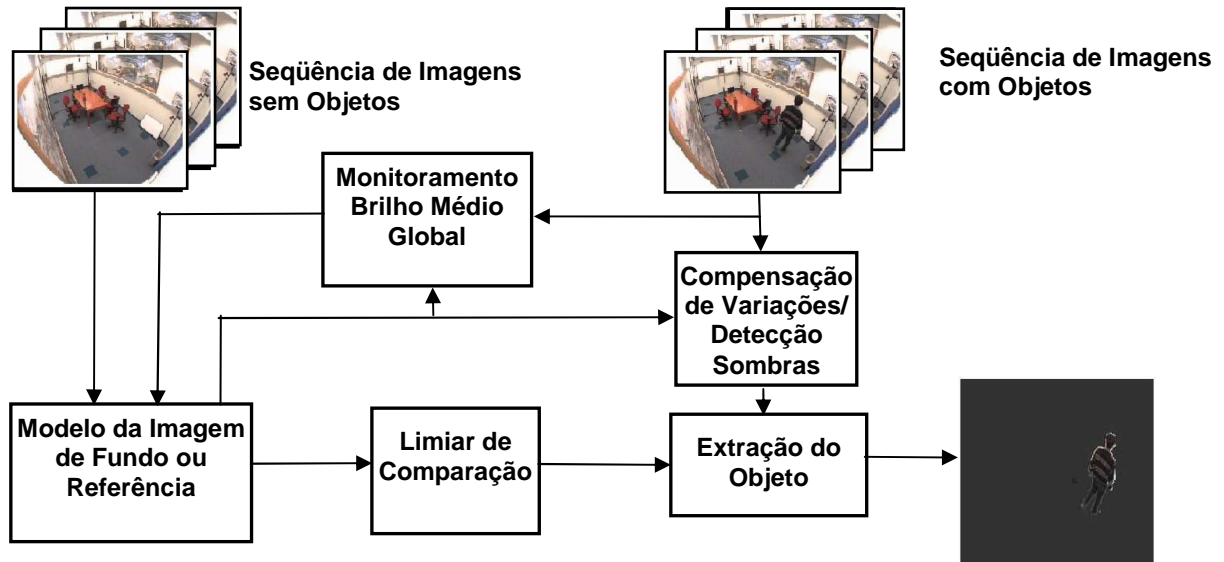


Figura 2: O diagrama em blocos com as etapas do método proposto para detecção de objetos.

3.2.1 Modelo da Imagem de Referência

O método proposto para segmentação de vídeo necessita de uma fase de aprendizado que consiste na aquisição de alguns quadros da imagem de fundo para obtenção de parâmetros estatísticos que expressam as características da cada *pixel* e constituem o modelo individual do *pixel* nesta imagem. Esta fase de aprendizado é realizada sem a presença de objetos móveis na imagem de fundo. Os parâmetros calculados durante esta fase são os seguintes: valor médio e desvio padrão amostral para cada *pixel*.

O valor médio amostral é a média dos valores obtida para cada um dos *pixels* $k(x,y)$ em cada componente de cor Y , Cb , e Cr , para um determinado número de quadros (N), segundo a equação 3:

$$\bar{x}_{(k)}(Y, Cb, Cr) = \frac{1}{N-1} \sum_{i=1}^N I_{i(k)}(Y, Cb, Cr) \quad (3)$$

Em que:

- i é o número do quadro em processamento;
- k é a coordenada (x,y) de cada *pixel*;

c) $I_{i(k)}(Y,Cb,Cr)$ é o valor corrente de cada *pixel* no espaço de cores YCbCr.

A média amostral é utilizada como uma estimativa pontual da verdadeira média da população (μ). [67]

Em função de variações na iluminação do ambiente e dos ruídos gerados pela câmera, ainda que a imagem seja estática, o mesmo *pixel* não terá valores idênticos ao longo do tempo [61]. Tendo em vista as variações de iluminação e do sensor da câmera é necessário computar as variações de cada *pixel* k em N quadros em cada componente Y , Cb e Cr . Estas variações são expressas pelo desvio padrão amostral segundo a equação 4:

$$s_{(k)}(Y, Cb, Cr) = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (I_{i(k)}(Y, Cb, Cr) - \bar{x}_{i(k)}(Y, Cb, Cr))^2} \quad (4)$$

Em que:

- a) i é o número do quadro em processamento;
- b) k é a coordenada (x,y) de cada *pixel*;
- c) $I_{i(k)}(Y,Cb,Cr)$ é o valor corrente de cada *pixel* no espaço de cores YCbCr;
- d) $\bar{x}_{i(k)}(Y, Cb, Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr.

O desvio padrão amostral é uma estimativa não-tendenciosa do desvio padrão da população (σ) e o seu valor tende a σ para valores de N muito grandes, isto é, maiores ou iguais a 30 segundo o teorema do limite central. [68]

3.2.2 Determinação do número de quadros

O dimensionamento amostral consiste na determinação da quantidade mínima de quadros necessários para estimar a média e o desvio padrão de cada *pixel*. O número de quadros é um fator crucial para determinar o tempo necessário para aquisição da imagem de referência. A variância da média amostral diminui, por um fator igual a $1/N$, com o aumento do tamanho da amostra de 1 para N [68]. Isto significa que para se obter um intervalo de confiança com maior probabilidade de conter a verdadeira média da população, quanto maior o número de quadros melhor é a estimativa. É necessário estabelecer um compromisso entre o tempo de aquisição da imagem de referência e o erro de estimativa dos parâmetros. Supondo-

se que o valor limite do erro da estimativa não ultrapasse um determinado valor d , pode-se expressar o erro padrão da seguinte maneira:

$$Z_u \frac{\sigma}{\sqrt{n}} \leq d \quad (5)$$

Em que:

- a) σ é o desvio padrão da população em estudo;
- b) n é o número de amostras da população;
- c) Z_u é o percentil com valor u da distribuição normal padrão. [68]

Os percentis são pontos estimativos de uma distribuição de frequência que determinam uma dada porcentagem de observações que se localizam abaixo ou acima deles [69]. Considerando-se a margem de erro d uma porcentagem δ da média dos *pixels* e estabelecendo-se um grau de confiança igual a 95%, o tamanho estimado para o número de quadros pode ser obtido pela equação 6:

$$n = \left[\frac{1,96\sigma}{\delta \bar{x}} \right]^2 \quad (6)$$

O valor de n é o tamanho mínimo da amostra para que se tenha um grau de confiança 95% de que a verdadeira média não difira em mais do que a margem d da média estimada. O valor de Z_u está associado ao grau de confiança desejado para a estimativa. Os níveis clássicos do grau de confiança são 90, 95 e 99 %. Escolhendo um grau de confiança igual a 95%, o valor de Z_u resulta em aproximadamente igual a 1,96 (distribuição normal) [68]. Neste caso, temos um problema circular, pois para estimar o tamanho da amostra é necessário o valor do desvio padrão populacional que se pretende estimar. Desta forma, seu valor não pode ser conhecido. Assim, o desvio populacional (σ) deve ser substituído por uma estimativa (desvio amostral) com base em uma amostra experimental de 30 quadros.

Para cada seqüência de imagens ilustrada na Figura 3 foram calculados os valores médios e os desvios padrões para uma amostra de 30 quadros, em cada componente de cor, considerando-se uma determinada margem de erro (δ) em relação ao verdadeiro valor da média, e um grau de confiança igual a 95%. Os resultados obtidos estão ilustrados nas Figuras 4 a 6. As “caixas” centrais, nos gráficos destas figuras, representam uma distância equivalente

a dois desvios padrões e os valores médios estão no centro das “caixas”. De acordo com os resultados obtidos, a quantidade de quadros para estimar o valor médio da população em cada cenário é diferente e tem a seguinte variação: para a componente Y a variação está entre 19 a 52 quadros; para a componente Cb é de 3 a 44 quadros; e para a Cr é de 2 a 49 quadros. Estas estimativas iniciais podem ser utilizadas para fixar o número de quadros para aquisição da imagem de referência em cada cenário.

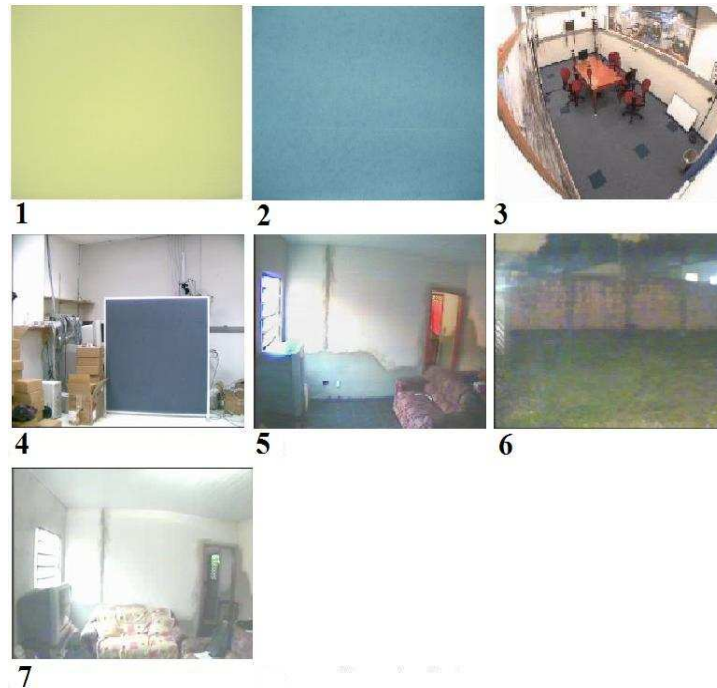


Figura 3: Seqüência de imagens utilizadas para determinar o número de quadros

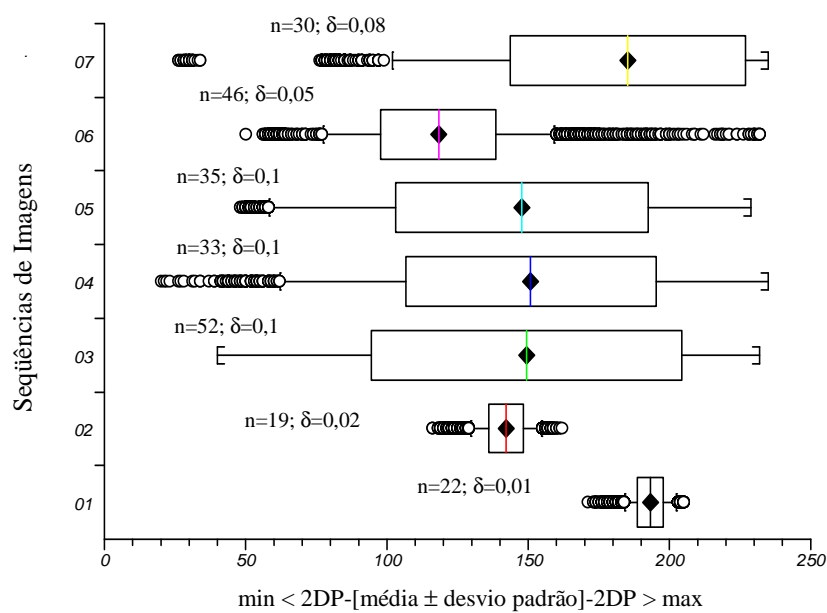


Figura 4: Componente Y: Média e desvio padrão estimados com 30 quadros para cada seqüência de imagem. Número de quadros (n) e margens de erro (δ) para cada seqüência de imagens.

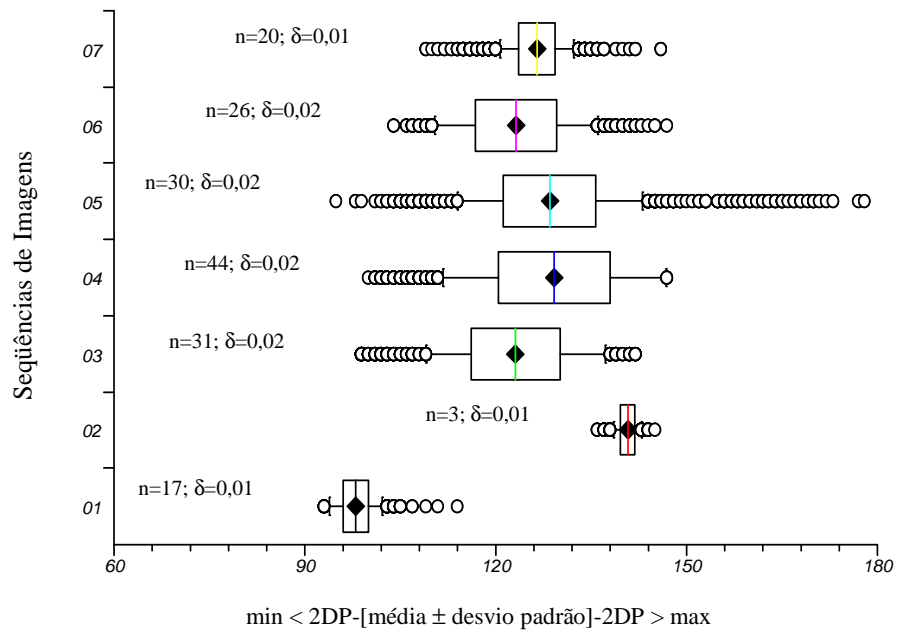


Figura 5: Componente Cb: Média e desvio padrão estimados com 30 quadros para cada seqüência de imagem. Número de quadros (n) e margens de erro (δ) para seqüência de imagens.

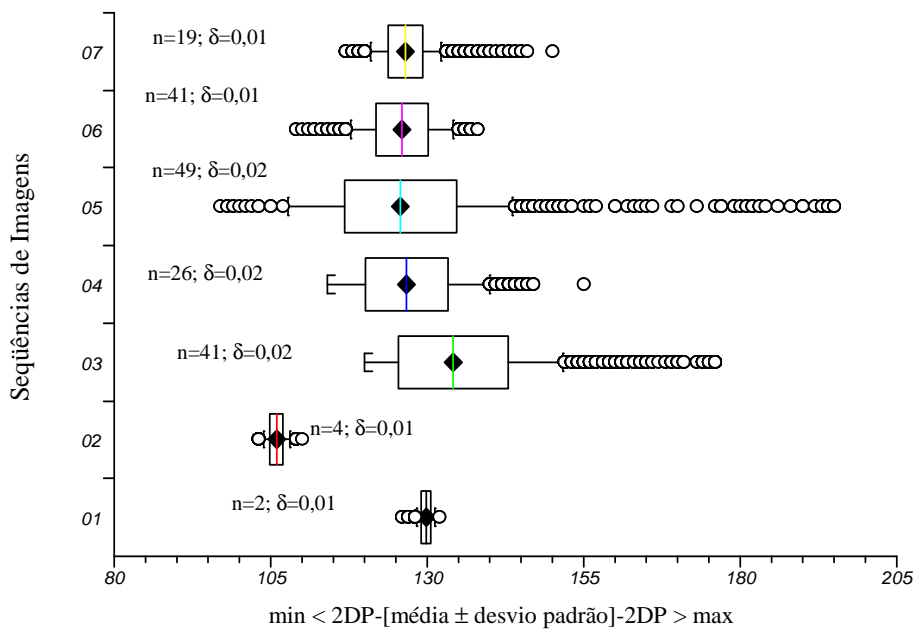


Figura 6: Componente Cr: Média e desvio padrão estimados com 30 quadros para cada seqüência de imagem. Número de quadros (n) e margens de erro (δ) para seqüência de imagens.

3.2.3 Intervalo de confiança

Uma estimativa pontual é um valor numérico único utilizado para estimar um parâmetro da população [67]. Por exemplo, a média amostral é uma estimativa pontual da média da população. Embora seja possível calcular a média amostral, como se trata de uma estimativa pontual, não se sabe o verdadeiro valor da média da população [67]. Intervalos de confiança são utilizados para estimar a média populacional com certo grau de confiança. Convencionalmente, o grau de confiança utilizado é 95%. Considerando-se uma variância desconhecida, o intervalo de confiança é obtido de acordo com a equação 7 [68]:

$$\bar{x} - t_u \frac{s}{\sqrt{N}} < \mu < \bar{x} + t_u \frac{s}{\sqrt{N}} \quad (7)$$

De acordo como a equação 7, pode-se afirmar com um certo grau de confiança que a média populacional encontra-se nesse intervalo; t_u é o percentil com valor u da distribuição t-student com $N - 1$ graus de liberdade. O intervalo de confiança é utilizado para identificar os *pixels* correntes com uma variação significativa em relação aos seus valores médios na imagem de referência. A hipótese estabelecida é que sem variações de iluminação ou objetos em cena, a maioria dos *pixels* tem seus valores compreendidos nos seus respectivos intervalos de confiança. Os valores dos limites superiores (UL) e inferiores (LL) são definidos de acordo com as equações 8 e 9, respectivamente:

$$UL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) + t_u \frac{s_{(k)}(Y, Cb, Cr)}{\sqrt{N}} \quad (8)$$

$$LL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) - t_u \frac{s_{(k)}(Y, Cb, Cr)}{\sqrt{N}} \quad (9)$$

Em que:

- a) $s_{(k)}(Y, Cb, Cr)$ é o desvio padrão de cada *pixel* no espaço de cores YCbCr;
- b) N é o número de quadros;
- c) t_u é o percentil com valor u da distribuição t-student;
- d) $\bar{x}_{(k)}(Y, Cb, Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr.

Para verificar a quantidade de elementos de imagens compreendidos dentro dos intervalos de confiança, utilizaram-se as seqüências de imagens 01, 02, 03 e 04 ilustradas na

Figura 7. Para cada seqüência de imagens foram calculados os valores médios e os desvios padrões de cada *pixel* considerando-se 16 quadros ($N = 16$). Em seguida, verificou-se o número de elementos de imagens com intensidades compreendidas nos intervalos de confiança. O valor do N igual a 16 resulta em uma estimativa pontual dos valores médios e dos desvios padrões, para maioria dos elementos de imagens das seqüências de imagens ilustradas na Figura 7, com margens de erro menores que 3% segundo os resultados obtidos na seção 2.4.2.

De acordo com os resultados exibidos na Tabela 1, quando a cena não contém objetos, a quantidade de *pixels*, cujos valores estão compreendidos nos intervalos de confiança, é maior em comparação quando um objeto é inserido na cena. O intervalo de confiança apresenta a faixa de valores cujas intensidades de cada *pixel* costumam variar em cada cenário analisado.

Tabela 1: Número de elementos de imagens com valores compreendidos no intervalo de confiança para cada componente de cor do espaço YCbCr em cada seqüência de imagem de teste ilustrada na Figura 7

Seqüências de Imagens	Número de Elementos de Imagens com Valores Compreendidos nos Intervalos de Confiança					
	Y		Cb		Cr	
	Sem Objeto	Com Objeto	Sem Objeto	Com Objeto	Sem Objeto	Com Objeto
01	49099	2789	37452	21966	60824	42914
02	33256	2763	41628	24968	40204	21030
03	28815	18721	46506	39454	47019	40772
04	27314	11349	47214	22762	52468	36887

Na Figura 7 são mostrados os resultados obtidos para cada seqüência de imagem em relação à contagem do número de elementos de imagens com valores compreendidos nos intervalos de confiança com e sem objetos na cena. As seqüências de imagens estão organizadas da seguinte maneira: a) imagem de referência com e sem objetos; b) componente Y: os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança; c) componente Cb: os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança; d) componente Cr: os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança. De acordo com os resultados obtidos, quando um objeto é inserido na cena os valores dos *pixels*, correspondentes ao objeto e também ao seu redor, variam de modo tal que

ficam fora dos intervalos de confiança. No caso da componente Y, observa-se uma grande quantidade de elementos de imagens que não pertencem ao objeto com valores fora dos intervalos de confiança. Os *pixels* fora dos intervalos de confiança são os pontos em branco nas imagens B, C e D de cada seqüência ilustrada na Figura 7. Para minimizar a detecção de falsos positivos é necessário utilizar limiares com valores acima das variações dos *pixels* pertencentes à imagem de referência ou reduzir as variações de cada *pixel* antes da etapa de classificação.

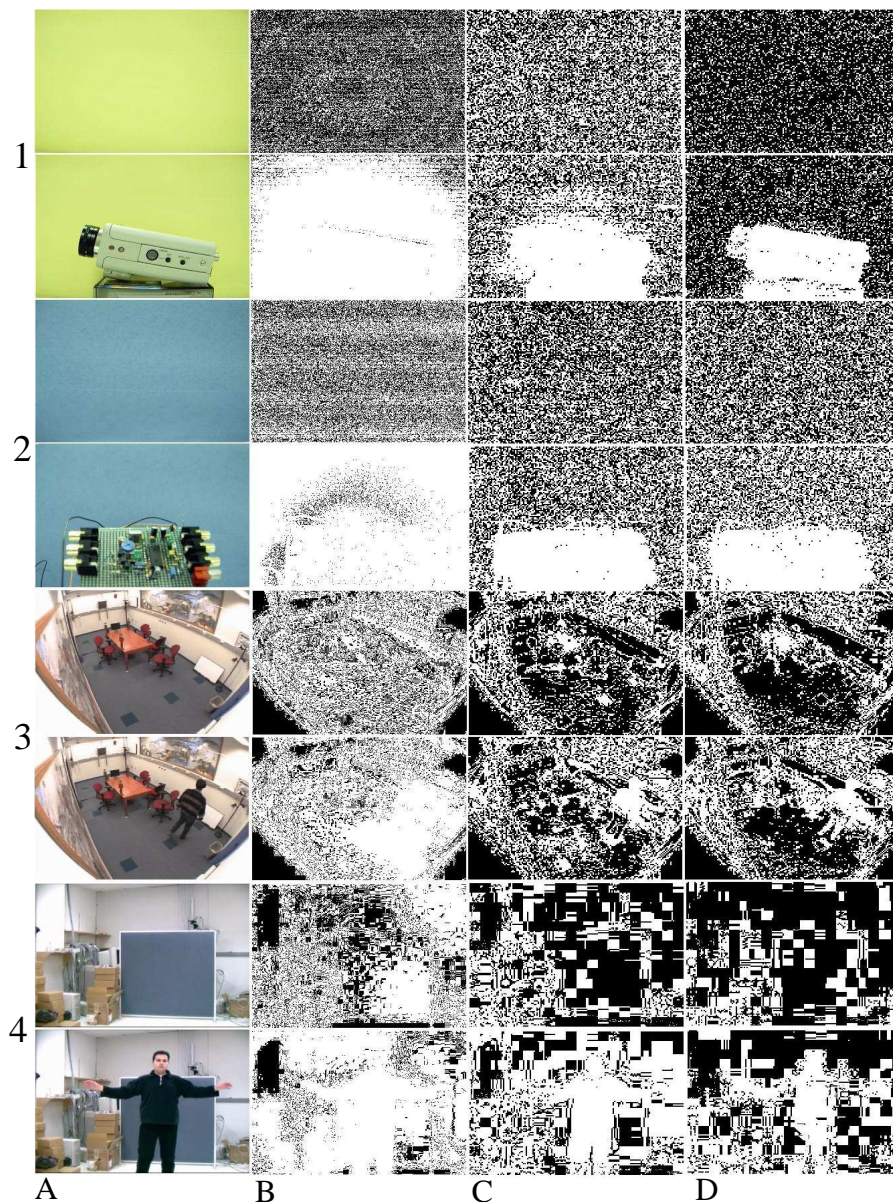


Figura 7: A) Imagem de referência com e sem objetos; B) Componente Y: Os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança; C) Componente Cb: Os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança; D) Componente Cr: Os pontos em preto são os elementos de imagens cujos valores estão compreendidos nos intervalos de confiança.

3.2.4 Seleção do Nível de Limiar

Na etapa de seleção do nível de limiar são determinados os limiares de comparação para classificação dos *pixels* na etapa de extração do objeto. A partir do desvio padrão obtido para cada *pixel* em cada componente de cor Y, Cb, Cr são determinados os limiares $T_{(k)}(Y)$, $T_{(k)}(Cb)$ e $T_{(k)}(Cr)$ da seguinte maneira:

$$\begin{cases} T_{(k)}(Y) = \alpha_Y \times s_{(k)}(Y) \\ T_{(k)}(Cb) = \alpha_{Cb} \times s_{(k)}(Cb) \\ T_{(k)}(Cr) = \alpha_{Cr} \times s_{(k)}(Cr) \end{cases} \quad (10)$$

Em que:

- $s_{(k)}(Y,Cb,Cr)$ é o desvio padrão de cada *pixel* no espaço de cores YCbCr;
- $\alpha(Y,Cb,Cr)$ são as constantes utilizadas para determinar os valores dos limiares de comparação.

Os valores de α_Y , α_{Cb} e α_{Cr} são determinados automaticamente logo após a aquisição do modelo da imagem de referência da seguinte maneira: os valores destas constantes são inicialmente iguais a 1 e procede-se com a classificação dos *pixels* correntes contando-se a quantidade de *pixels* classificados como objetos em cada componente de cor. Se o número de *pixels* classificados como objeto for maior que 50, o valor do α correspondente é incrementado em uma unidade. Este procedimento é repetido quadro a quadro até que o número de *pixels* classificados como objeto seja menor que 50 em cada componente YCbCr. Isto equivale a 0,01% do tamanho da imagem com resolução igual a 640 x 507 *pixels*.

3.2.5 Extração do Objeto ou classificação do *pixel*

Na fase de extração do objeto cada *pixel* é classificado como pertencente ao objeto ou à imagem de referência. Para isto, calcula-se o valor da diferença absoluta (BD), devidamente multiplicada pelo valor da função de compensação do *pixel* (PCF), *pixel a pixel* em cada componente de cor, de acordo com a equação 11:

$$F_k(Y,Cb,Cr) = BD_{(k)}(Y,Cb,Cr) \times PCF_{(k)}(Y,Cb,Cr) \quad (11)$$

Sendo que $F_k(Y,Cb,Cr)$ é o valor da diferença absoluta (BD) de cada *pixel* atenuado pela função de variação de compensação do *pixel* (PCF). Comparando-se os valores obtidos da equação 10 com os limiares $T_{(k)}(Y,Cb,Cr)$, cada *pixel* é classificado como pertencente ao objeto (F) ou a imagem de referência (B) de acordo com a seguinte máscara:

$$Máscara \left\{ \begin{array}{l} F, F_{(k)}(Y) > T_{(k)}(Y) \\ ou \\ F_{(k)}(Cb) > T_{(k)}(Cb) \\ ou \\ F_{(k)}(Cr) > T_{(k)}(Cr) \\ B, Se nenhuma das anteriores \end{array} \right. \quad (12)$$

3.2.6 Compensação de variações do *pixel*

Em ambientes externos e internos ocorrem variações de iluminações locais, afetando apenas uma região da imagem, ou globais, afetando toda a imagem. Estas variações alteram a intensidade dos *pixels* e, por conseguinte, se estas variações não forem levadas em consideração no modelo da imagem de referência, os *pixels* afetados podem ser erroneamente classificados como pertencentes ao objeto. Tais variações podem ocorrer em função de uma mudança na iluminação ambiente (ligar/desligar uma lâmpada) ou quando um objeto é inserido no ambiente em estudo, pois ele pode modificar a intensidade dos *pixels* no local e na sua vizinhança.

Na Figura 8 é exibida uma seqüência de imagens em que apesar da iluminação ambiente não sofrer variações, quando uma pessoa entra no cenário a câmera efetua correções automáticas modificando as intensidades dos *pixels*. Comparando-se a Figura 8-a com as demais, percebem-se diferenças de brilho entre as imagens, ou seja, as imagens das Figuras 8-b a 8-d estão mais claras.

Para lidar com as variações de intensidade dos *pixels*, definiu-se empiricamente uma função para compensar tais variações e, conseqüentemente, reduzir o número de *pixels* incorretamente classificados como objetos. A função de compensação de variação dos *pixels* (PCF) é definida de acordo com a equação 13.

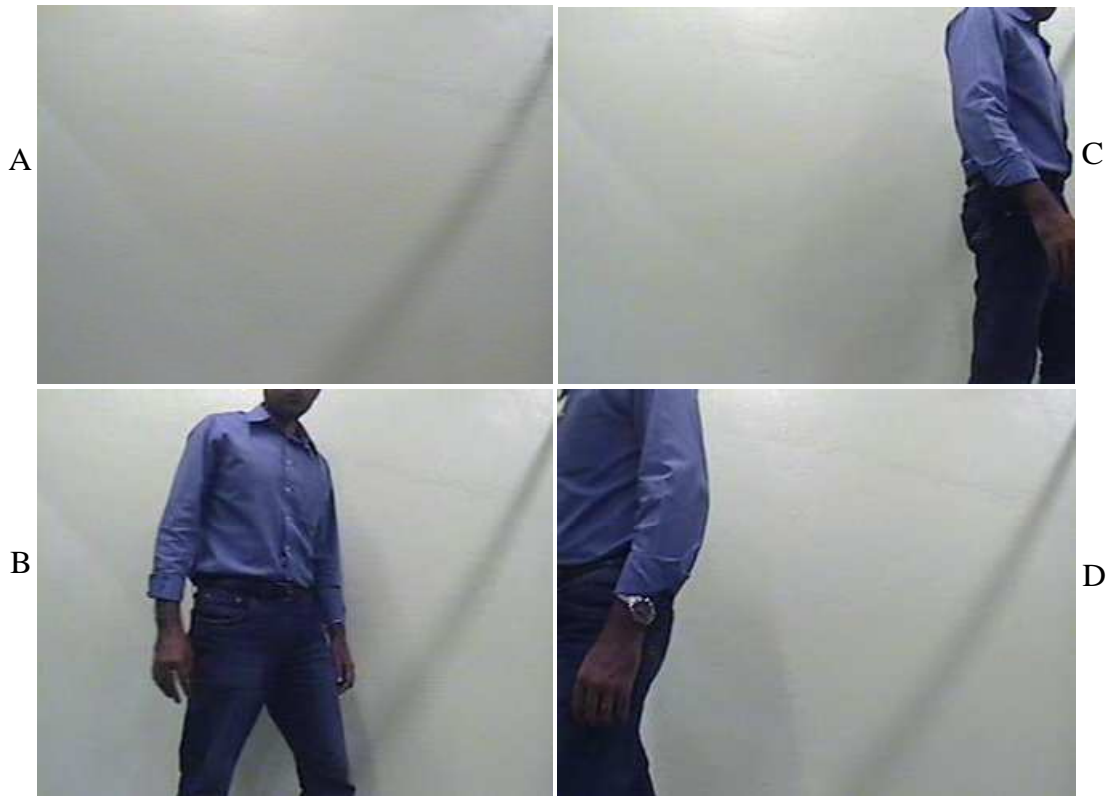


Figura 8: Seqüência de imagens exibindo a variação de intensidade de brilho em função das correções automáticas realizadas pela câmara.

$$\left\{ \begin{array}{l}
 \text{Se } I_{(k)}(Y, Cb, Cr) > UL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)|}{I_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } I_{(k)}(Y, Cb, Cr) < LL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)|}{\bar{x}_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } LL_{(k)}(Y, Cb, Cr) \leq I_{(k)}(Y, Cb, Cr) \leq UL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = 1
 \end{array} \right. \quad (13)$$

Em que:

- $I_{(k)}(Y, Cb, Cr)$ é o valor de cada *pixel* no espaço de cores YCbCr durante a fase de detecção dos objetos;
- $UL_{(k)}(Y, Cb, Cr)$ e $LL_{(k)}(Y, Cb, Cr)$ são os limites superior e inferior do intervalo de confiança, respectivamente, de cada *pixel* no espaço de cores YCbCr.

Esta abordagem é baseada na avaliação empírica de resultados de diversos vídeos de teste. Quando uma pessoa entra em cena, a variação da intensidade de alguns *pixels* pode ficar fora do intervalo de confiança, ocasionando a detecção de falsos positivos. Identificando-se os *pixels*, acima do limite superior do intervalo de confiança ou abaixo do limite inferior do intervalo de confiança, é possível minimizar a detecção de falsos positivos reduzindo-se o espalhamento dos valores das diferenças absolutas de cada *pixel* obtida na fase de segmentação. Calculando-se o valor de $F_k(Y, Cb, Cr)$, equação 11, levando-se em consideração a equação 13 e fazendo-se as simplificações necessárias, obtêm-se as seguintes expressões:

$$\left\{ \begin{array}{l} \text{Se } I_{(k)}(Y, Cb, Cr) > UL_{(k)}(Y, Cb, Cr) : \\ F_{(k)}(Y, Cb, Cr) = I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr) \left[2 - \frac{\bar{x}_{(k)}(Y, Cb, Cr)}{I_{(k)}(Y, Cb, Cr)} \right] \\ \text{ou} \\ \text{Se } I_{(k)}(Y, Cb, Cr) < LL_{(k)}(Y, Cb, Cr) : \\ F_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) - I_{(k)}(Y, Cb, Cr) \left[2 - \frac{I_{(k)}(Y, Cb, Cr)}{\bar{x}_{(k)}(Y, Cb, Cr)} \right] \end{array} \right. \quad (14)$$

Conforme a equação 14, quando os valores dos *pixels* estão acima do limites superiores dos intervalos de confiança, os valores médios de cada *pixel* no modelo da imagem de referência são multiplicados pelo seguinte fator de compensação:

$$\left[2 - \frac{\bar{x}_{(k)}(Y, Cb, Cr)}{I_{(k)}(Y, Cb, Cr)} \right]$$

O fator de compensação aumenta os valores médios de cada *pixel*, antes de efetuar o calculo das diferenças absolutas, para compensar o aumento dos seus respectivos valores correntes. Caso os valores correntes dos elementos de imagens estejam muito próximos dos seus valores médios correspondentes, no modelo da imagem de referência, o fator de compensação será aproximadamente igual a 1, ou seja, as diferenças absolutas não são compensadas. De acordo com a equação 14, quando os valores correntes dos *pixels* estão abaixo dos limites inferiores dos intervalos de confiança, eles são multiplicados pelo seguinte fator de compensação:

$$\left[2 - \frac{I_{(k)}(Y, Cb, Cr)}{\bar{x}_{(k)}(Y, Cb, Cr)} \right]$$

O fator de compensação aumenta os valores correntes, de cada *pixel* antes de efetuar o cálculo das diferenças absolutas, para compensar as suas respectivas reduções. Caso os valores correntes dos elementos de imagens estejam muito próximos dos seus correspondentes no

modelo da imagem de referência, o fator de compensação será aproximadamente igual a 1, ou seja, as diferenças absolutas não serão compensadas.

Observou-se que, em alguns casos, a intensidade do *pixel* pode ser reduzida para um valor muito baixo pela função de compensação, ocasionando a sua classificação errônea como pertencente à imagem de fundo. Para evitar este problema, introduziram-se limites inferiores para a função de compensação de acordo com a equação 15:

$$\left\{ \begin{array}{l} \text{Se } PCF_{(k)}(Y) < \alpha_{lo(Y)} : \\ PCF_{(k)}(Y) = \alpha_{lo(Y)/2} \\ e \\ \text{Se } PCF_{(k)}(Cb, Cr) < \alpha_{lo(Cb, Cr)} : \\ PCF_{(k)}(Cb, Cr) = \alpha_{lo(Cb, Cr)} \end{array} \right. \quad (15)$$

Para as imagens de testes utilizadas neste trabalho, as constantes $\alpha_{lo(Y)}$ e $\alpha_{lo(Cb, Cr)}$ foram escolhidas empiricamente com valores iguais a 0,4 e 0,6, respectivamente.

Na Figura 9 estão ilustrados os *pixels* detectados fora dos intervalos de confiança para cada componente YCbCr, sem objetos no cenário de testes. Calcularam-se os valores médios e os desvios padrões com 30 quadros e, em seguida, o intervalo de confiança para cada *pixel*. As seqüências de imagens estão organizadas da seguinte maneira: a) imagem de fundo sem objetos, b) os pontos pretos são os *pixels* detectados abaixo dos limites inferiores dos intervalos de confiança, c) os pontos pretos são os *pixels* detectados acima dos limites superiores dos intervalos de confiança.

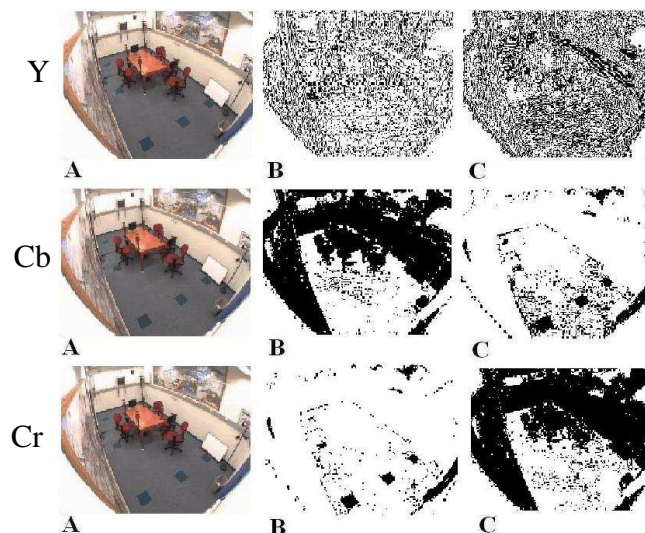


Figura 9: A) imagem de teste sem objetos; B) os pontos pretos são os *pixels* detectados abaixo dos limites inferiores dos intervalos de confiança, C) os pontos pretos são os *pixels* detectados acima dos limites superiores dos intervalos de confiança.

De acordo com os resultados, as variações dos valores dos *pixels* podem ocorrer mesmo sem nenhum objeto em cena. Na Figura 10 estão ilustrados os *pixels* detectados fora dos intervalos de confiança para cada componente YCbCr com uma pessoa na sala. Comparando-se com o resultado anterior, sem objetos em cena, pode-se perceber uma variação dos *pixels* em volta da pessoa que esta andando na sala, isto é, no chão e na parede.

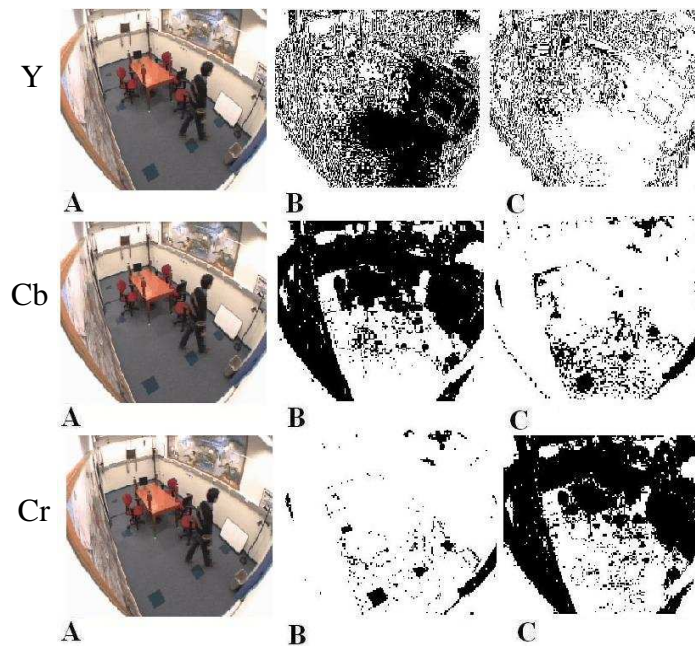


Figura 10: A) imagem de teste com objetos; B) os pontos pretos são os *pixels* detectados abaixo dos limites inferiores dos intervalos de confiança, C) os pontos pretos são os *pixels* detectados acima dos limites superiores dos intervalos de confiança.

Na Figura 11 é exibido o resultado obtido com a utilização da função de compensação de variação do *pixel* sem objetos em cena. As seqüências de imagens estão organizadas da seguinte maneira: a) imagem de referência, b) os pontos brancos são os *pixels* classificados como objetos de acordo com a equação 12. Neste caso, a função de compensação de variação do *pixel* não foi utilizada; c) os pontos brancos são os *pixels* classificados como objetos de acordo com a equação 12. Neste caso, a função de compensação de variação do *pixel* foi utilizada. De acordo com os resultados obtidos, reduzindo-se a diferença absoluta é possível reduzir o número de *pixels* falsamente classificados como pertencentes ao objeto.

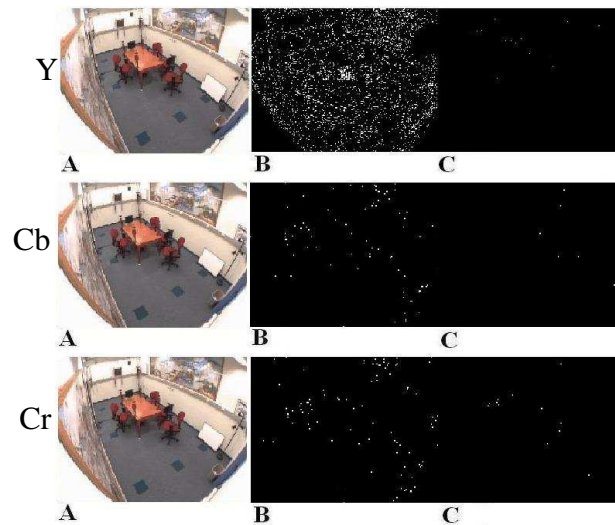


Figura 11: A) imagem de teste sem objetos; B) os pontos brancos são os *pixels* classificados como objetos. Neste caso, a função de compensação de variação do *pixel* não foi utilizada; C) os pontos brancos são os *pixels* classificados como objetos. Neste caso, a função de compensação de variação do *pixel* foi utilizada.

Na Figura 12 está ilustrado o resultado obtido com a utilização da função de compensação de variação do *pixel* com objetos em cena. As seqüências de imagens estão organizadas da seguinte maneira: a) imagem de referência com objeto, b) os pontos brancos são os *pixels* classificados como objetos de acordo com a equação 12. Neste caso, a função de compensação de variação do *pixel* não foi utilizada; c) os pontos brancos são os *pixels* classificados como objetos de acordo com a equação 12. Neste caso, a função de compensação de variação do *pixel* foi utilizada. De acordo com os resultados obtidos, sem a utilização da função de compensação do *pixel*, não somente a pessoa em cena foi detectada como objeto, mas também diversos *pixels* pertencentes à imagem de referência.

A presença da pessoa no cenário e a sua movimentação ocasionam mudanças nas intensidades dos *pixels*. A utilização de limiares de comparação maiores poderia reduzir o número de falsos positivos, porém alguns *pixels* do objeto (pessoa) seriam classificados como pertencente à imagem de fundo. Uma abordagem comumente adotada para lidar com variações de iluminação é a atualização recursiva do modelo da imagem de referência [42]. Uma desvantagem desta abordagem é que os valores passados das mudanças de cada *pixel* são incorporados ao modelo da imagem de referência, ocasionando uma modificação permanente no modelo original adquirido durante a fase de aprendizado. Portanto, para aplicações em sistemas de vigilância, um intruso que permaneça parado no cenário acaba sendo incorporado ao modelo da imagem de referência e não é detectado enquanto não modificar a sua posição na cena.

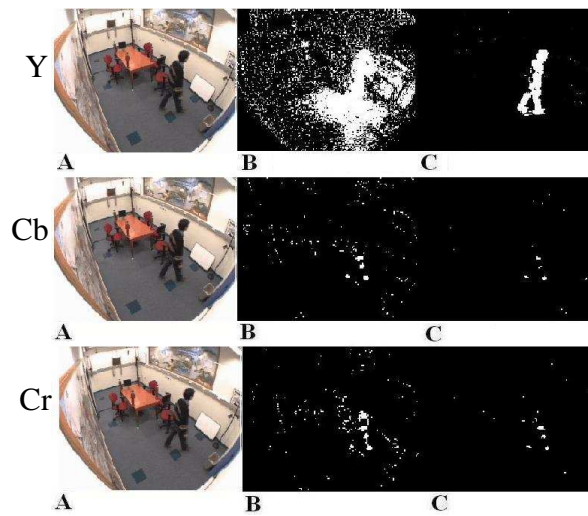


Figura 12: A) imagem de teste com objetos; B) os pontos brancos são os *pixels* classificados como objetos. Neste caso, a função de compensação de variação do *pixel* não foi utilizada; C) os pontos brancos são os *pixels* classificados como objetos. Neste caso, a função de compensação de variação do *pixel* foi utilizada.

Nas Figuras 13 e 14 estão ilustrados os gráficos de dispersão das diferenças absolutas com e sem objetos na cena, respectivamente, considerando-se a utilização da função de compensação de variação do *pixel* (BD_C) e a não utilização da função de compensação de variação do *pixel* (BD_NC). De acordo com os resultados, a utilização da função de compensação reduz o espalhamento das diferenças absolutas e isso contribui com a diminuição dos falsos positivos.

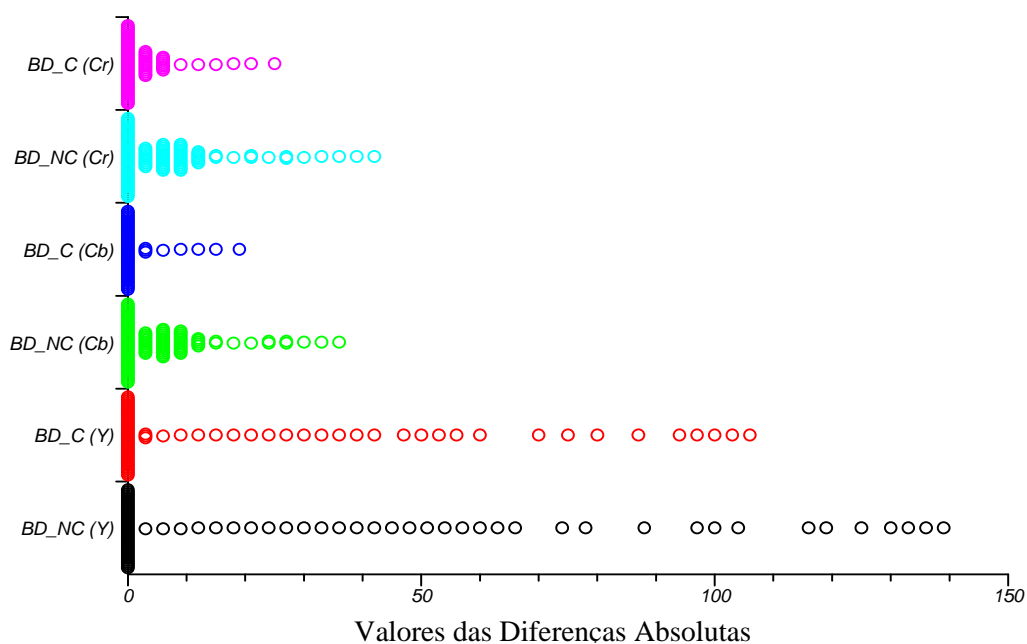


Figura 13: Dispersão das diferenças absolutas com objetos. BD_NC: diferenças absolutas não compensadas; BD_C: diferenças absolutas compensadas.

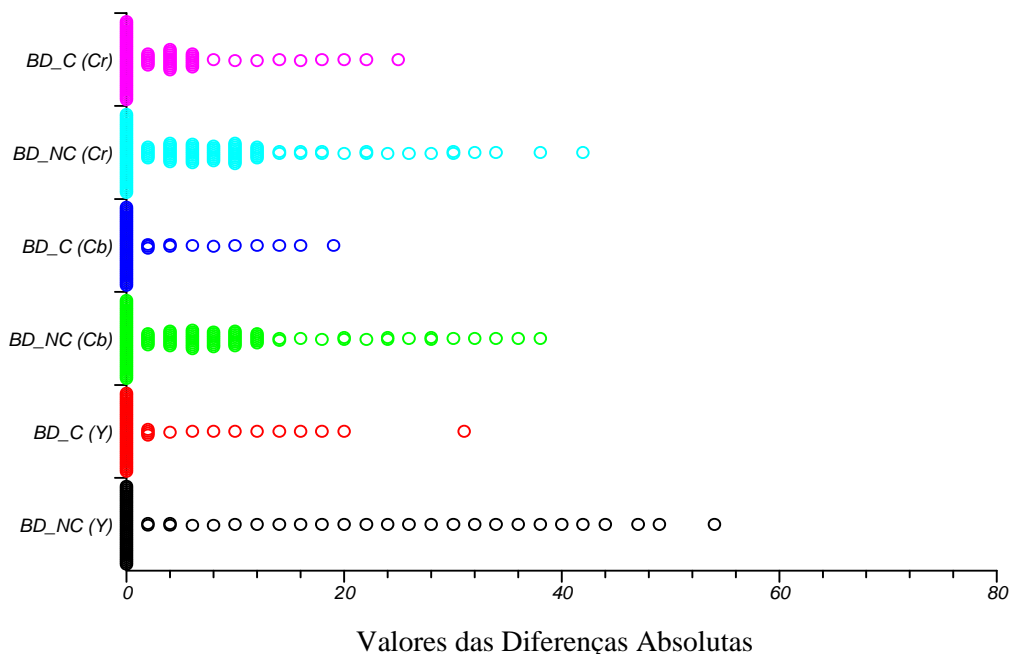


Figura 14: Dispersão das diferenças absolutas sem objetos. **BD_NC:** diferenças absolutas não compensadas; **BD_C:** diferenças absolutas compensadas.

Nas Figura 15 e 16 são mostrados os gráficos das diferenças absolutas com e sem a função de compensação de variação do *pixel*. De acordo com os resultados, a diferença absoluta obtida com a função de compensação do *pixel* é uma versão atenuada da obtida sem compensação.

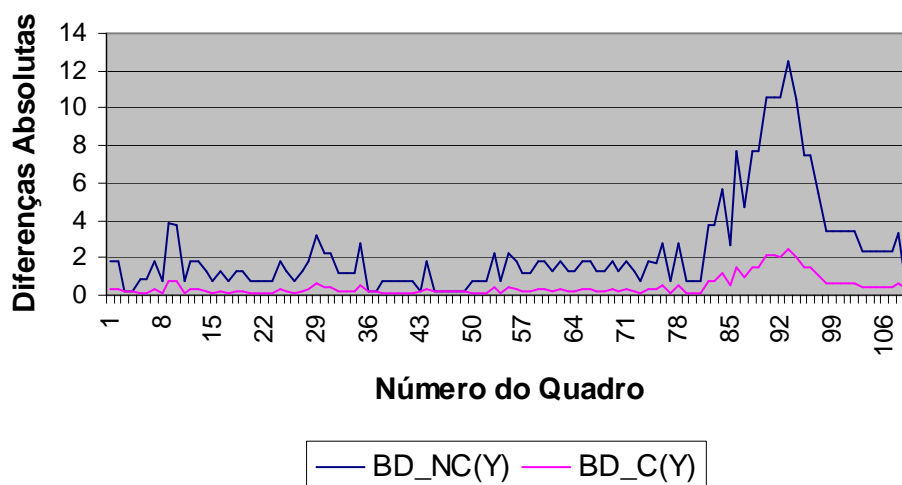


Figura 15: Gráficos das diferenças absolutas com (**BD_C**) e sem (**BD_NC**) a função de compensação de variação do *pixel* para a sequência de imagens de testes 03.

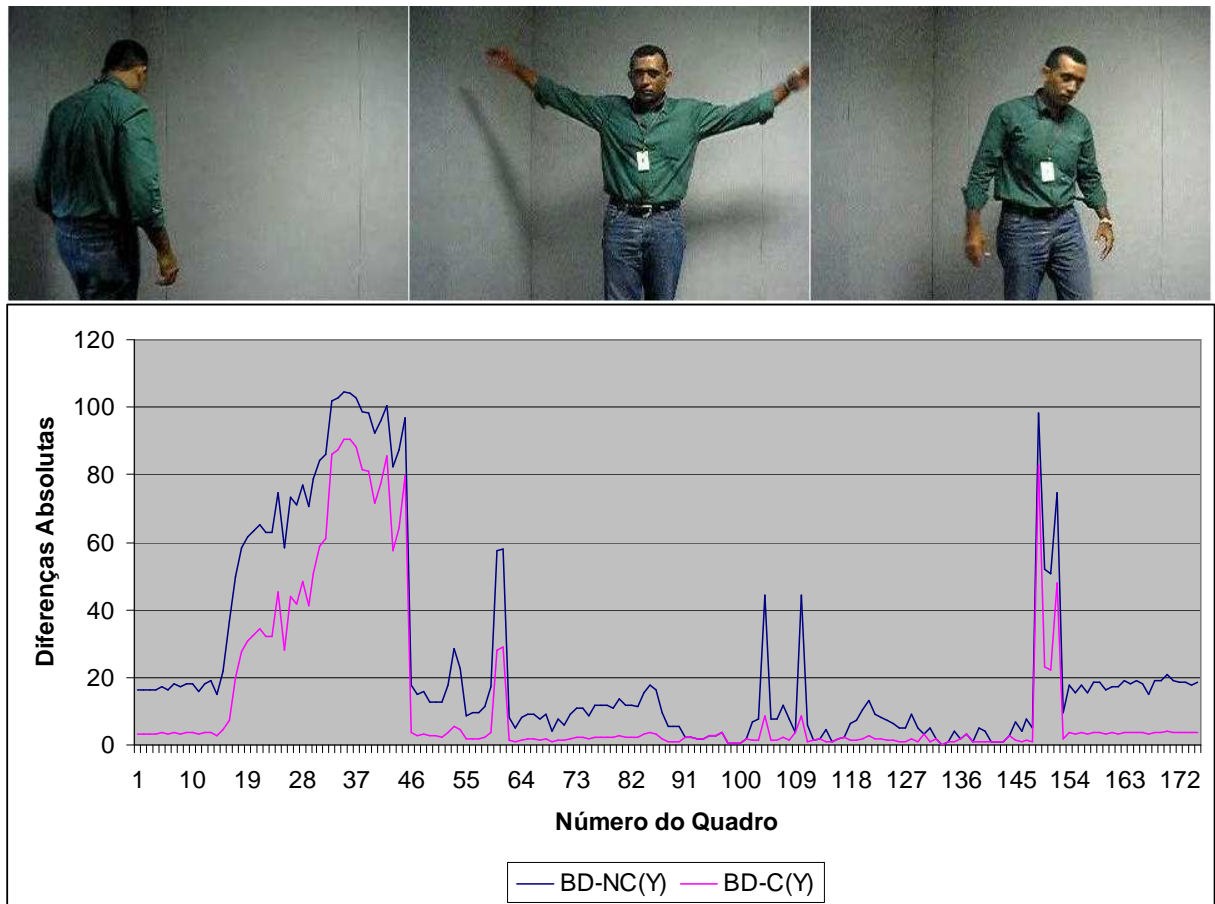


Figura 16: Gráficos das diferenças absolutas com (BD_C) e sem (BD_NC) a função de compensação de variação do *pixel* para a seqüência de imagens mostradas nesta figura.

3.2.7 Detecção de variação global

Muito embora o número de falsos positivos, causados pelas mudanças na iluminação ou por um objeto entrando em cena, pode ser reduzido com a utilização da função de compensação de variação do *pixel*, determinou-se que se a variação média global de luminância (Y) for maior que 30%, o modelo da imagem de referência deve ser readquirido. Na Figura 17 está ilustrado o gráfico com os valores médios da luminância calculados para cada quadro da seqüência de imagens denominada *time of day* [70].

Nesta seqüência de imagens, conforme exibido na Figura 17, uma sala escura é gradativamente iluminada durante um longo período de tempo. Uma pessoa entra na sala, senta no sofá, aguarda por alguns instantes e depois se retira da sala. O modelo da imagem de referência foi obtido utilizando-se os quadros de número 2500 a 2564 que apresentam uma luminância média global praticamente constante. Em seguida, realizou-se a classificação de

cada *pixel* e o cálculo da porcentagem de falsos positivos com e sem a função de compensação de variação do *pixel*. Os resultados estão ilustrados na Figura 18.

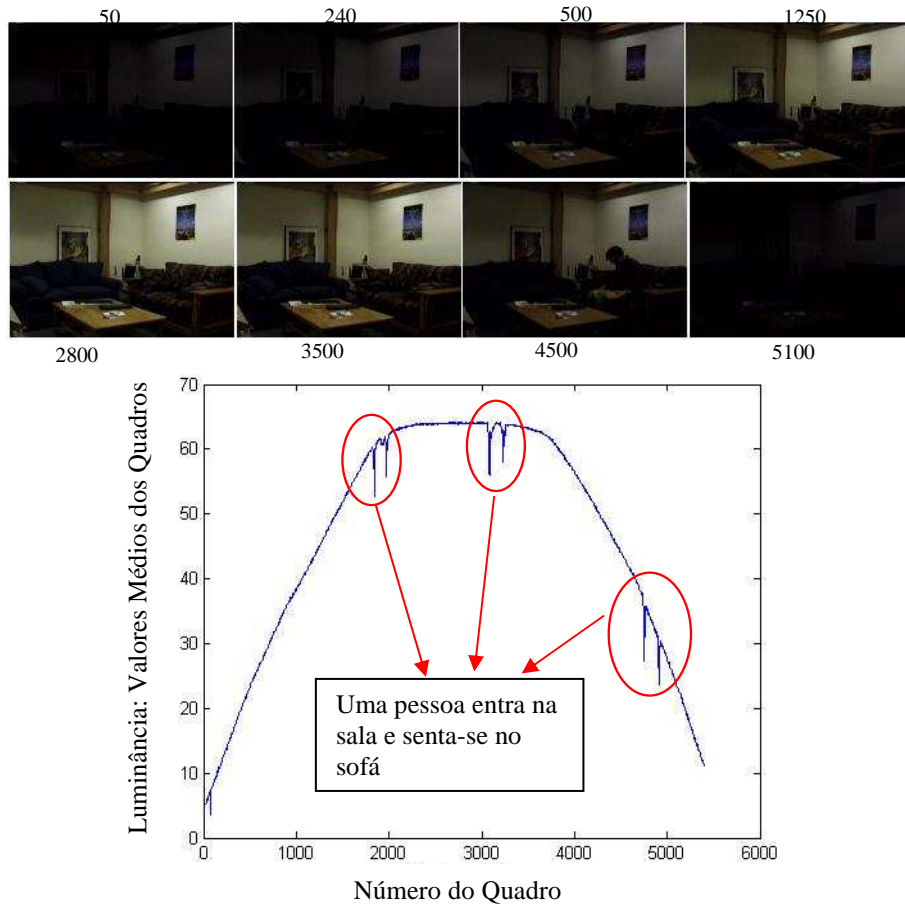


Figura 17: Sequência de imagens *time of day*. Gráfico dos valores médios da luminância para cada quadro da seqüência de imagens *time of day*.

De acordo com os resultados ilustrados na Figura 18, para uma variação média global de luminância igual a 25%, a porcentagem de falso positivo é reduzida de 70% para 12% com a utilização da função de compensação de variação do *pixel*. Para uma variação da média global de luminância maior que 30%, a porcentagem de falso positivo é maior que 70%, sem a utilização da função de compensação de variação do *pixel*, e com a função de compensação a porcentagem reduz para 32%.

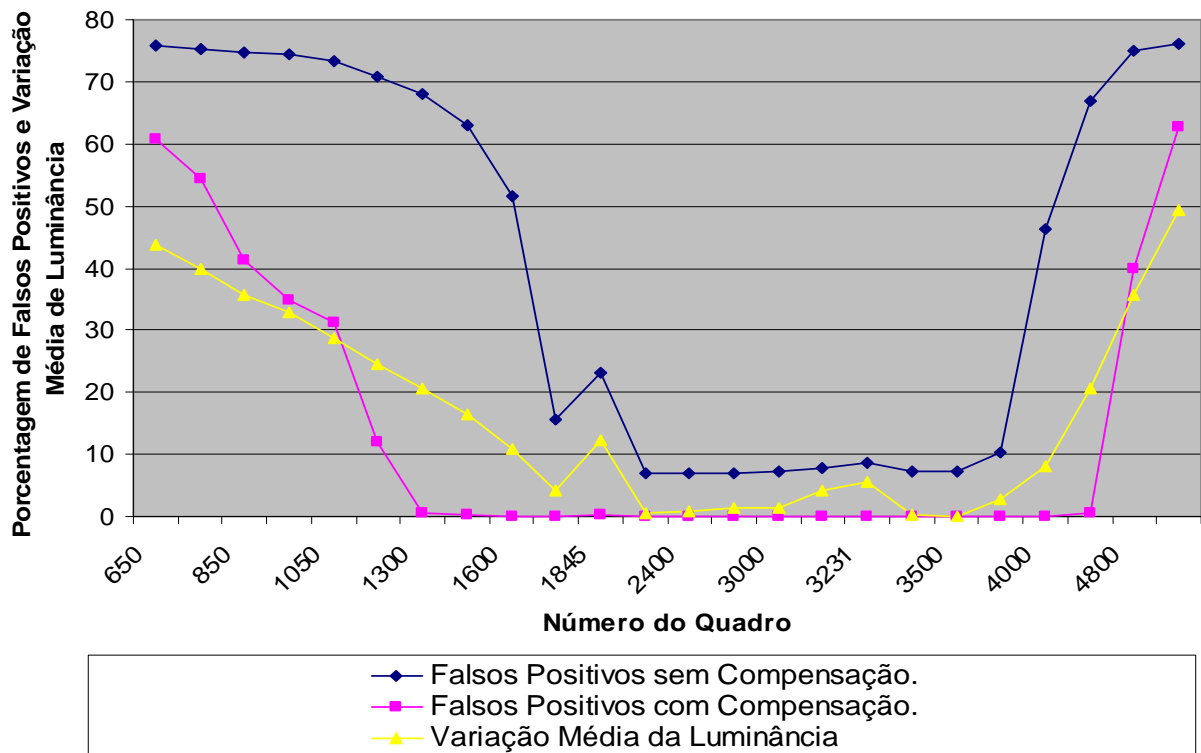


Figura 18: Porcentagem de variação de falsos positivos com e sem a função de compensação de variação de *pixel* de acordo com a variação média de luminância.

O efeito visual é ilustrado na Figura 19 cujas seqüências de imagens estão organizadas da seguinte maneira: a) imagem de referência, b) os pontos brancos são os *pixels* classificados como objetos sem utilizar a função de compensação de variação do *pixel*; c) os pontos brancos são os *pixels* classificados como objetos usando a função de compensação de variação do *pixel*.

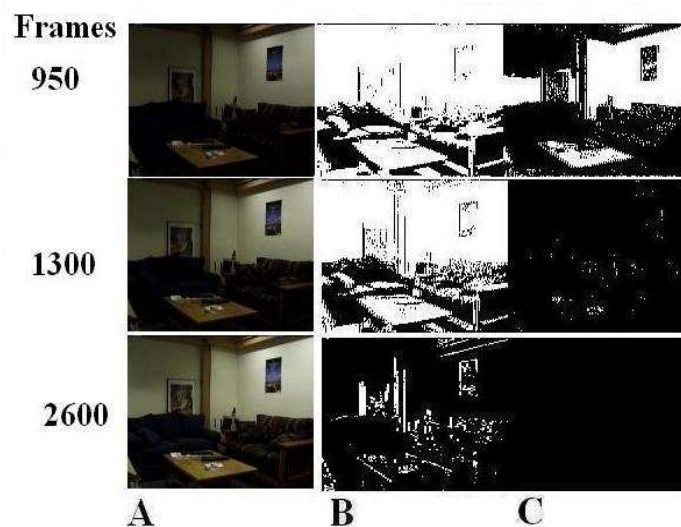


Figura 19: Seqüência de imagens utilizadas para avaliação da detecção de falsos positivos em um cenário com variação de iluminação. A) imagem de fundo original, B) os pontos brancos são os *pixels* classificados como objetos sem utilizar a função de compensação de variação do *pixel*, e C) os pontos brancos são os *pixels* classificados como objetos usando a função de compensação de variação do *pixel*.

De acordo com o gráfico da Figura 18, no quadro número 950, a porcentagem de falso positivo é maior que 30% e isto ocasiona um efeito visível, isto é, muitos *pixels* da imagem de referência são erroneamente classificados com objetos. Portanto, se a variação média global de luminância permanecer maior que 30%, o modelo da imagem de referência deve ser adquirido novamente. Para detectar a variação global de luminância, computa-se a média global da informação de luminância (componente Y) dividindo-se cada quadro em quatro regiões. A média obtida para cada região é comparada com o valor da média global da região correspondente no modelo da imagem de referência durante 300 quadros (10 segundos para uma taxa de 30 quadros por segundo). Em seguida, verifica-se se os valores das médias correntes aumentaram ou diminuíram continuamente durante este período em uma das regiões. Se for detectado um aumento ou diminuição maior que 30%, isto significa que houve uma variação significativa na iluminação e o número de falsos positivos será visível na imagem resultante da segmentação. Neste caso, o modelo da imagem de referência é adquirido novamente. Considerando-se 16 quadros para aquisição da imagem média e do desvio padrão, um novo modelo da imagem de referência estaria disponível um segundo após a detecção da variação da iluminação. Quando uma pessoa entra no cenário ilustrado na Figura 17, a média global de luminância também varia, porém esta variação é inferior a 30% conforme ilustrado no gráfico da Figura 18.

3.2.8 Detecção de sombras

Embora haja diversos fatores que podem influenciar a intensidade de um *pixel* em uma sombra, assume-se que a intensidade observada do *pixel* na sombra é diretamente proporcional à luz incidente. Consequentemente, os *pixels* pertencentes a regiões com sombras têm suas intensidades atenuadas em relação aos seus valores no modelo da imagem de referência [64, 71]. O método utilizado para detecção de sombras é baseado no pressuposto de que a cor não muda na área sombreada e somente a intensidade do *pixel* sofre uma redução. Um *pixel* é classificado como sombra se o seu valor corrente estiver no seguinte intervalo:

$$\begin{cases} I_{(k)}(Y) > LL_{(k)}(Y) \times \alpha_{lo} \\ e \\ I_{(k)}(Y) < LL_{(k)}(Y) \times \alpha_{hi} \end{cases} \quad (16)$$

Em que:

- a) $I_{(k)}(Y,Cb,Cr)$ é o valor de cada *pixel* da componente Y durante a fase de detecção dos objetos;
- b) $LL_{(k)}(Y)$ é o limite inferior do intervalo de confiança de cada *pixel* da componente Y;
- c) α_{lo} e α_{hi} são as constantes que determinam a fração dos valores médios dos pixels na imagem de referência que é esperado nas regiões sombreadas.

Nos testes realizados, os valores de α_{lo} e α_{hi} estão na faixa de 0,5 a 0,7. Foram utilizados os limites inferiores dos intervalos de confiança como referências para comparação dos valores correntes da intensidade dos *pixels*.

3.3 CONCLUSÕES

Neste capítulo foi apresentado o método proposto para detecção de objetos. Em aplicações cuja imagem de referência é estática e a câmera é fixa, não é necessária a compensação para mudanças de objetos na imagem de referência, porém é necessário compensar as variações das intensidades dos *pixels* devido às variações de iluminação ou de ruídos gerados pela câmera. Apresentou-se uma técnica para compensação quadro a quadro das variações das intensidades dos *pixels*. O modelo da imagem de referência é paramétrico (gaussiano). Além disso, são calculados também os intervalos de confiança para os valores médios de cada *pixel*. Antes da etapa de detecção dos objetos, o valor corrente de cada *pixel* é comparado com o seu respectivo intervalo de confiança e os que são identificados fora deste intervalo tem a sua diferença absoluta reduzida por uma função definida empiricamente. Os resultados experimentais demonstram uma redução na detecção de falsos positivos em ambientes com e sem variação de iluminação.

No próximo capítulo são realizadas as avaliações do método proposto implementado em MATLAB com operações em ponto flutuante.

4 AVALIAÇÃO DE MÉTODOS PARA SEGMENTAÇÃO DE VÍDEO

Neste capítulo são apresentados os resultados da avaliação do método para detecção de objetos implementado em MATLAB com operações em ponto flutuante. Primeiramente, descrevem-se as métricas utilizadas nas avaliações. Em seguida é realizada a avaliação do método de detecção de objetos em relação aos seguintes aspectos: utilização da função de compensação de variação do *pixel*; espaços de cores RGB, YCbCr, YUV, e HSV; comparação com métodos gaussianos com adaptação recursiva.

4.1 MÉTRICAS PARA AVALIAÇÃO

Para avaliar o método de detecção de objetos foram utilizadas três métricas para quantificar o quanto cada quadro segmentado se aproxima da imagem de teste de referência (*ground truth*). Além disso, a imagem de teste de referência foi construída para cada quadro das seqüências de imagens utilizadas durante as avaliações do método proposto. Na Figura 20 é ilustrado um exemplo de uma imagem de teste de referência. Na imagem de teste de referência (Figura 20-b), os *pixels* do objeto são pretos e os da imagem de referência são todos brancos. Desta forma, para calcular o número de *pixels* do objeto é feita uma contagem dos pontos pretos. O resultado desta contagem é comparado com o obtido pelo método de segmentação para avaliar o seu desempenho.

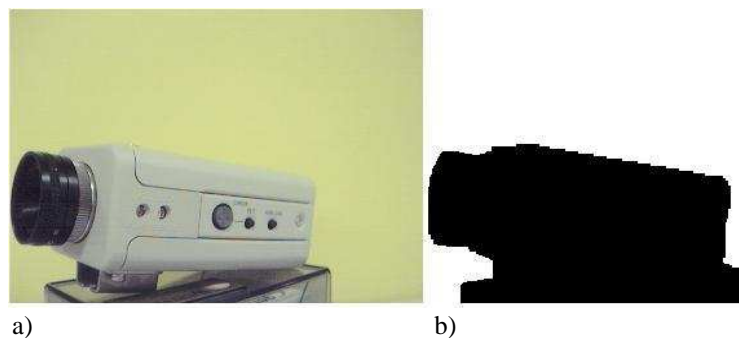


Figura 20: Exemplo de uma imagem utilizada para avaliar o método de segmentação proposto neste trabalho. a) imagem de referência; b) imagem de teste de referência (*ground truth*)

As métricas de avaliação são obtidas a partir de relações obtidas entre os parâmetros exibidos na Tabela 2:

Tabela 2: Parâmetros utilizados para obtenção das métricas de avaliação de métodos de detecção de objetos. TP: verdadeiro positivo; TN: verdadeiro negativo; FP: Falso positivo; FN: Falso Negativo.

Detecção	Imagem de teste de referência (<i>ground truth</i>)	
	Positivo	Negativo
Positivo	TP	FP
Negativo	FN	TN

O verdadeiro positivo (TP) é o número de *pixels* pertencentes ao objeto corretamente detectados pelo método de segmentação. O verdadeiro negativo (TN) é o número de *pixels* pertencentes à imagem de fundo ou referência corretamente detectado pelo método de segmentação. O falso positivo (FP) é o número de *pixels*, pertencentes à imagem de referência, erroneamente detectados como objetos. O falso negativo (FN) é o número de *pixels* do objeto erroneamente detectado como pertencente à imagem de referência. [72, 73]

As métricas utilizadas como indicadores da qualidade de desempenho do algoritmo são denominadas revocação (*Recall*), precisão² (*Precision*) e medida F (*F-measure*) [72]. Estas métricas são definidas pelas Equações 15, 16, e 17, respectivamente:

$$R = \frac{TP}{TP + FN} \quad (15)$$

$$P = \frac{TP}{TP + FP} \quad (16)$$

$$F = \frac{2PR}{P + R} \quad (17)$$

A revocação (R) é definida como a razão entre o número de *pixels* do objeto corretamente identificados pelo algoritmo ou método de segmentação e o número de *pixels* na imagem de teste de referência; a precisão (P) é definida como a razão entre o número de *pixels* corretamente identificados pelo algoritmo e o número de *pixels* detectados como

² Precisão é uma tradução do termo original (*precision*) e neste contexto é definida como a razão entre o número de elementos de imagem corretamente identificados e o número de elementos de imagem classificados como objetos pelo método de detecção de objetos.

objetos pelo algoritmo; e a medida F (F) é uma média geométrica entre a revocação e a precisão. Além de calcular as métricas supracitadas é necessário também avaliar as diferenças obtidas em diferentes cenários ou com diferentes parâmetros do método de segmentação.

Para avaliar os parâmetros do método proposto e efetuar comparações de desempenho em diferentes cenários, espaço de cores, e com outros métodos, foram utilizados testes de inferência estatística. O primeiro passo para aplicação dos testes estatísticos consiste em conhecer as naturezas das distribuições das características da população em estudo. Quando a natureza da distribuição é conhecida, aplicam-se testes denominados paramétricos nas amostras coletadas. Do contrário, devem-se utilizar testes que fazem poucas suposições sobre a natureza das distribuições, denominados testes não-paramétricos de inferência estatística [74]. Para verificar a natureza da distribuição das características das amostras estudadas, procedeu-se com o teste de normalidade de Shapiro Wilk [74]. O teste de Shapiro Wilk testa a hipótese nula de uma amostra pertencer a uma população com distribuição normal.

O procedimento geral para aplicação dos testes de inferência estatística, independente do método (paramétrico ou não-paramétrico) é o seguinte: começa-se fazendo alguma afirmação sobre as populações originais na forma de uma hipótese nula; em seguida, calcula-se o valor da estatística do teste, usando os dados contidos em uma amostra aleatória de observações. Dependendo da grandeza dessa estatística, rejeita-se ou não a hipótese nula [75]. A seguir são descritos os testes estatísticos utilizados neste trabalho para investigar a problemática formulada e, por conseguinte, atingir os objetivos estabelecidos. O nível de significância adotado em todos os testes foi de 5%, isto é, a probabilidade de rejeitar a hipótese nula mesmo que ela seja verdadeira.

Para comparar as medidas F foi utilizado o teste não-paramétrico de análise de variância denominado Kruskal-Wallis [76]. Este método compara três ou mais amostras aleatórias independentes e é uma alternativa não-paramétrica para o teste de análise de variância [75]. Nos testes realizados, as medidas F obtidas não apresentaram uma distribuição normal e por isso não foi utilizada a análise de variância para efetuar as comparações. O objetivo destas comparações é determinar se há uma diferença estatisticamente significativa, nas medidas F, quando diferentes técnicas e/ou parâmetros são utilizados para a detecção dos objetos. A hipótese nula (H_0) estabelece que as diferenças não são estatisticamente significantes entre as amostras. Se o teste for significativo ($p \leq 0,05$), a hipótese nula é rejeitada e para identificar quais os parâmetros ou técnicas apresentam as diferenças apontadas pelo teste são efetuadas comparações múltiplas com o método denominado Dwass-

Steel [77]. O valor do p ou probabilidade calculada é a probabilidade estimada de rejeitar a hipótese nula (H_0) de uma questão em estudo sendo a hipótese nula é verdadeira. A ferramenta denominada StatsDirect foi utilizada para calcular os valores das probabilidades (p) [78]. Além disso, para analisar a variância das medidas F em cada cenário de teste, foi utilizado um teste de igualdade de variância denominado teste de postos quadrados [79]. O objetivo deste teste é verificar se há uma diferença estatisticamente significativa nas variâncias das medidas F quando diferentes técnicas e/ou parâmetros são utilizados para a detecção dos objetos. Com isso é possível avaliar a dependência do método de segmentação em relação aos objetos em cena e diferentes cenários.

4.2 ANÁLISE DE DISTRIBUIÇÃO DAS IMAGENS DE TESTE

O método de detecção de objetos, implementado neste trabalho, utiliza um modelo gaussiano para modelagem da imagem de referência. Para verificar a natureza da distribuição de cada *pixel*, na seqüência de imagens de testes usada na avaliação do método, procedeu-se com o teste de normalidade de Shapiro Wilkson nas seqüências de imagens 1, 2 3, 4 e 7, ilustradas na Figura 3. Para cada seqüência de imagens foram capturados 30 quadros com resolução igual a 320 x 240 *pixels*. Desta maneira, a natureza da distribuição de cada *pixel* foi analisada com 30 amostras. Os resultados obtidos para as componentes Y , C_b e C_r estão ilustrados nas Figuras 21, 22, e 23, respectivamente.

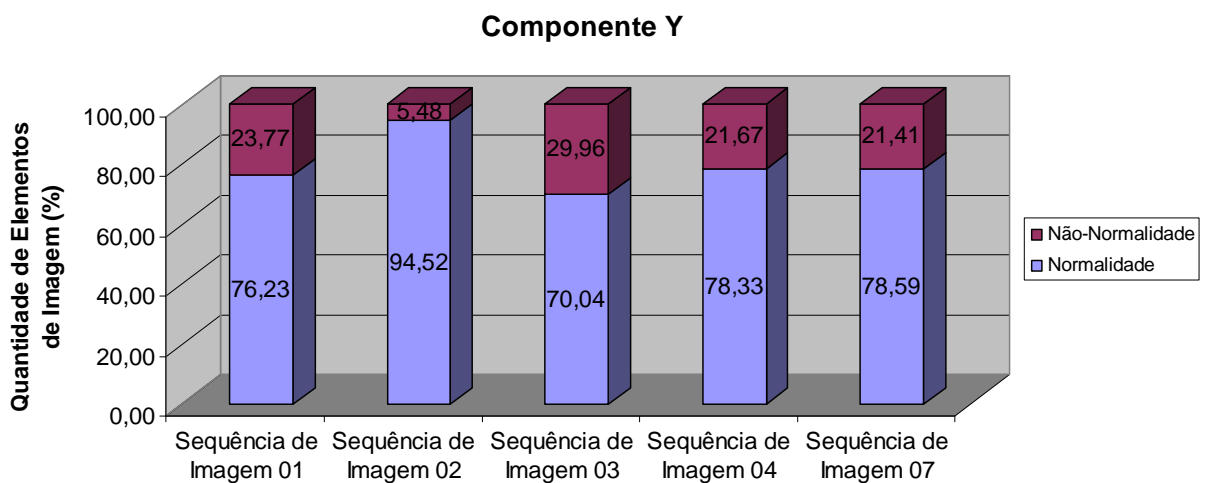


Figura 21: Resultados do teste de normalidade para a componente Y

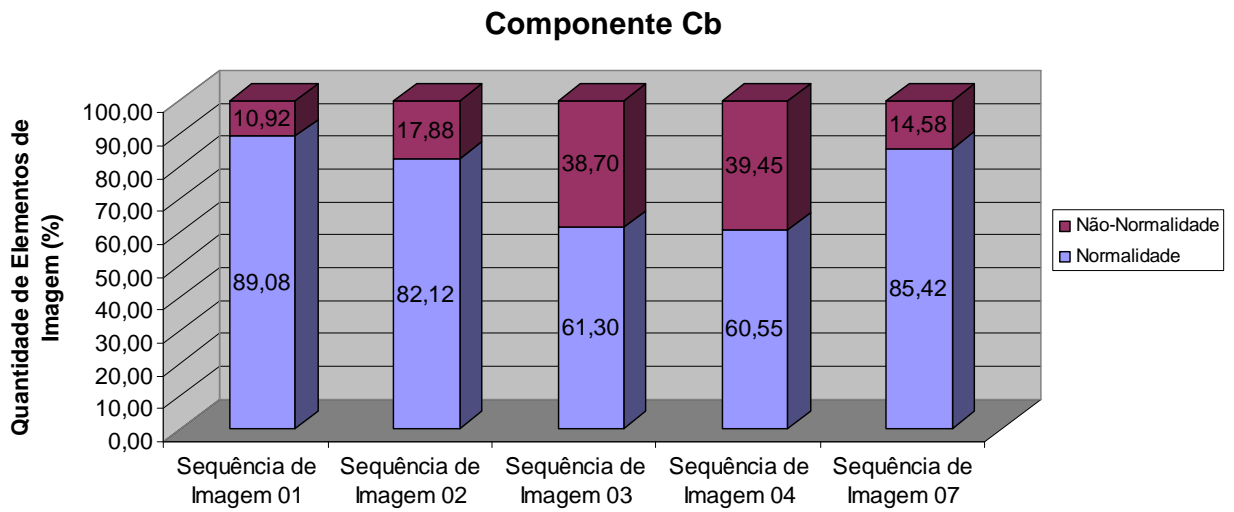


Figura 22: Resultados do teste de normalidade para a componente Cb

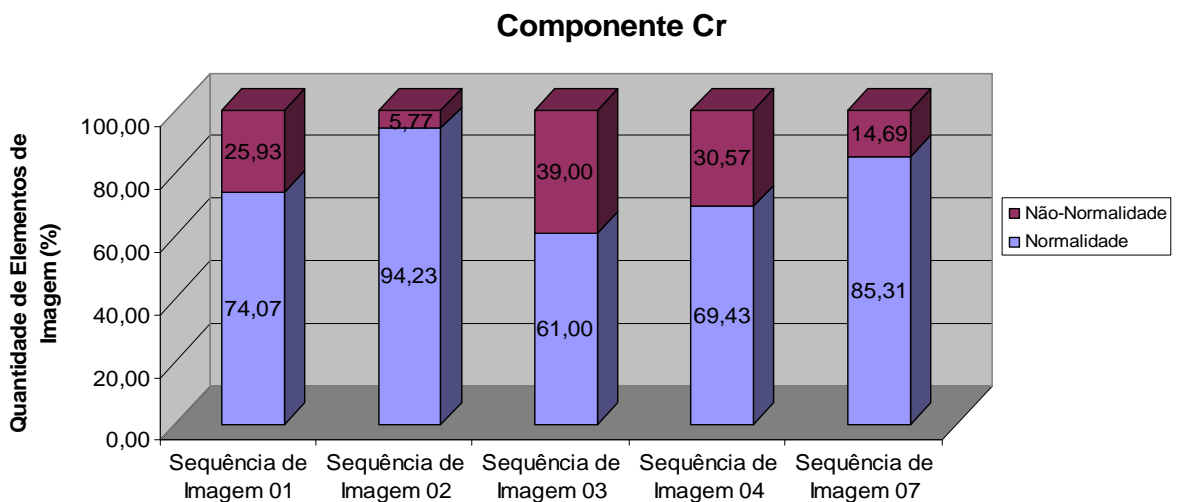


Figura 23: Resultados do teste de normalidade para a componente Cr

De acordo com os resultados obtidos, pode-se afirmar com 95% de confiança que a maioria dos *pixels* nas componentes Y, Cb, e Cr apresentam uma distribuição normal nas seqüências de imagens utilizadas para avaliação do método de segmentação. Portanto, a modelagem da imagem de referência utilizando um modelo gaussiano está de acordo com a distribuição da maioria dos elementos de imagens das seqüências de imagem de testes.

4.3 AVALIAÇÃO DO MÉTODO PROPOSTO

Nesta seção são apresentados os resultados obtidos com o método proposto implementado no MATLAB com as operações realizadas em ponto flutuante. Primeiramente, o método proposto é avaliado em relação à classificação dos *pixels* com e sem a função de compensação de variação do *pixel* e o detector de sombras. Segundo, compara-se o desempenho do método proposto em quatro espaços de cores, isto é, YCbCr, RGB, YUV, e HSV. Finalmente, é feita uma comparação do método proposto com a detecção feita com um método de detecção de objetos que utiliza a adaptação recursiva. As seqüências de imagens utilizadas para avaliação estão ilustradas na Figura 24.

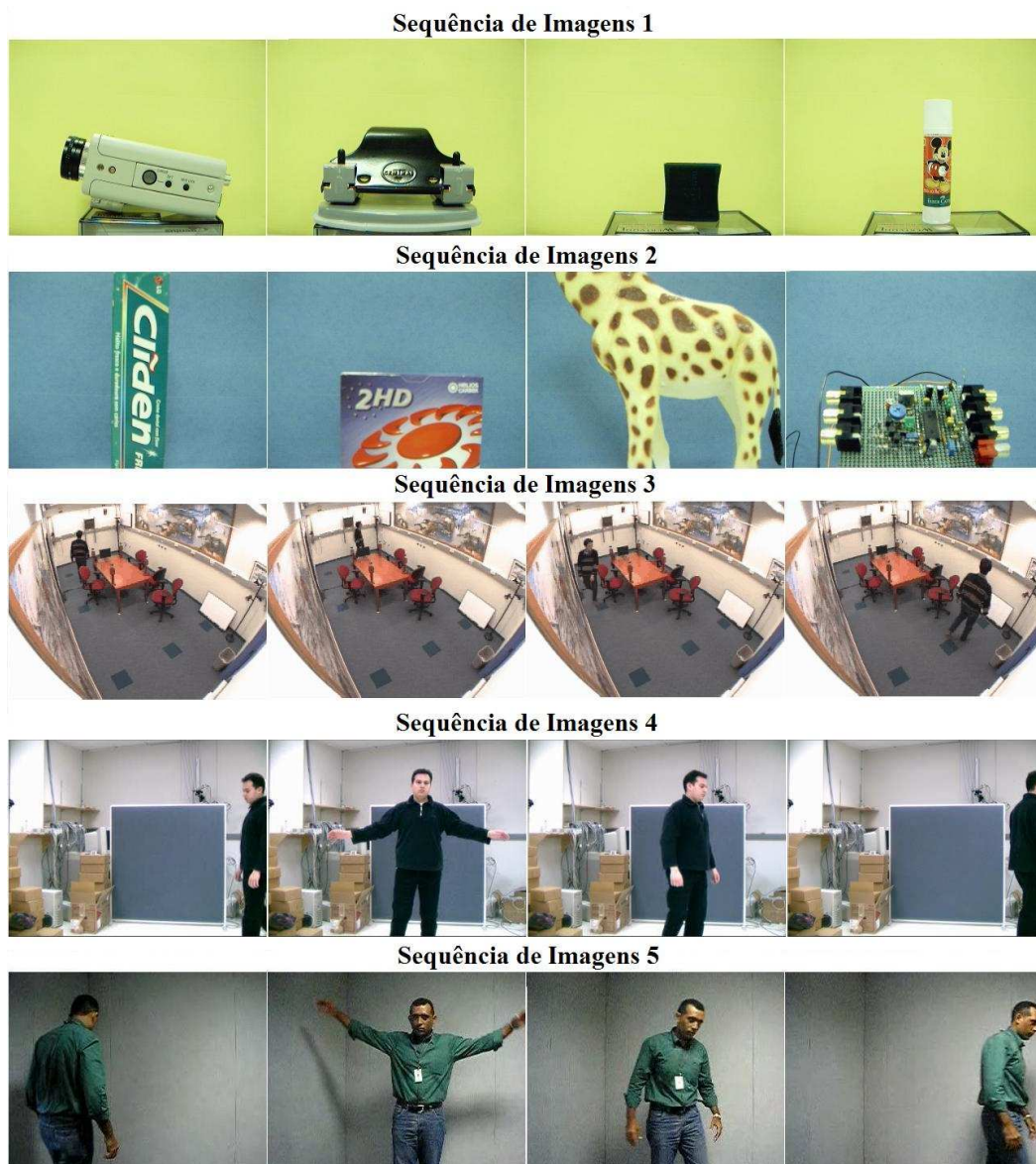


Figura 24: Seqüências de imagens utilizadas para avaliação do método de detecção de objetos.

4.3.1 Avaliação da classificação dos *pixels*

Nesta seção são apresentados os resultados da avaliação do método de detecção de objetos utilizando-se três tipos de classificações dos *pixels* denominadas Classificação do *Pixel* 01, Classificação do *Pixel* 02, e Classificação do *Pixel* 03. A Classificação do *Pixel* 01 utiliza a função de compensação de variação do *pixel* e a detecção de sombras durante a classificação dos *pixels*. A Classificação do *Pixel* 02 não utiliza a função de compensação de variação do *pixel* durante a classificação dos *pixels*, mas realiza a detecção de sombras. A Classificação do *Pixel* 03 utiliza somente a função de compensação de variação do *pixel* durante a detecção de objetos. O objetivo desta avaliação é verificar se há alguma diferença no desempenho do método de detecção de objetos quando é utilizada a função de compensação de variação do *pixel* proposta neste trabalho.

4.3.1.1 Seqüência de imagens 01

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 01 e três tipos de classificação dos *pixels* estão ilustrados na Figura 25. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p = 0,0034$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 3, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 01 e Classificação do *Pixel* 03.

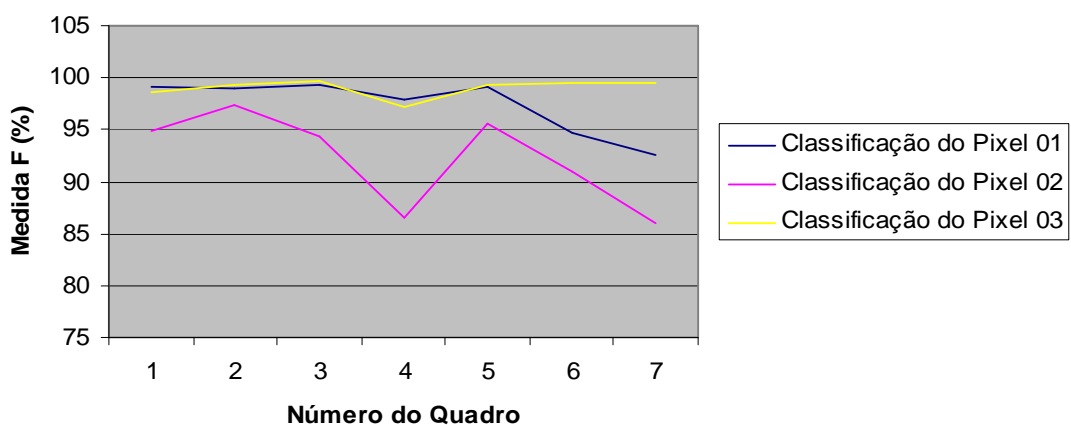


Figura 25: Medida F para seqüência de imagens 01 com três tipos de classificação dos *pixels*. Teste de Kruskal-Wallis ($p = 0,0034$); Teste de igualdade de variância ($p = 0,0048$). Classificação do *Pixel* 01 – função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 3: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{}) das medidas F para a seqüência de imagens 01 com três tipos de classificação dos *pixels*.**

Classificação do <i>Pixel</i>	p^*	p^{**}
01 vs. 02	0,0653	0,0288
01 vs. 03	0,1958	0,0447
02 vs. 03	0,0075	0,0003

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os três tipos de classificação dos *pixels*, foi rejeitada ($p = 0,0048$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 3, há uma diferença significativa entre todas as medidas F em relação à variância (p^{**}). Na Figura 26 está ilustrado o gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação do *pixel*. De acordo com o gráfico, a utilização da Classificação do *Pixel* 03 resultou na maior mediana com a menor variância. Uma menor variância na medida F significa que a detecção é menos dependente dos objetos inseridos na cena.

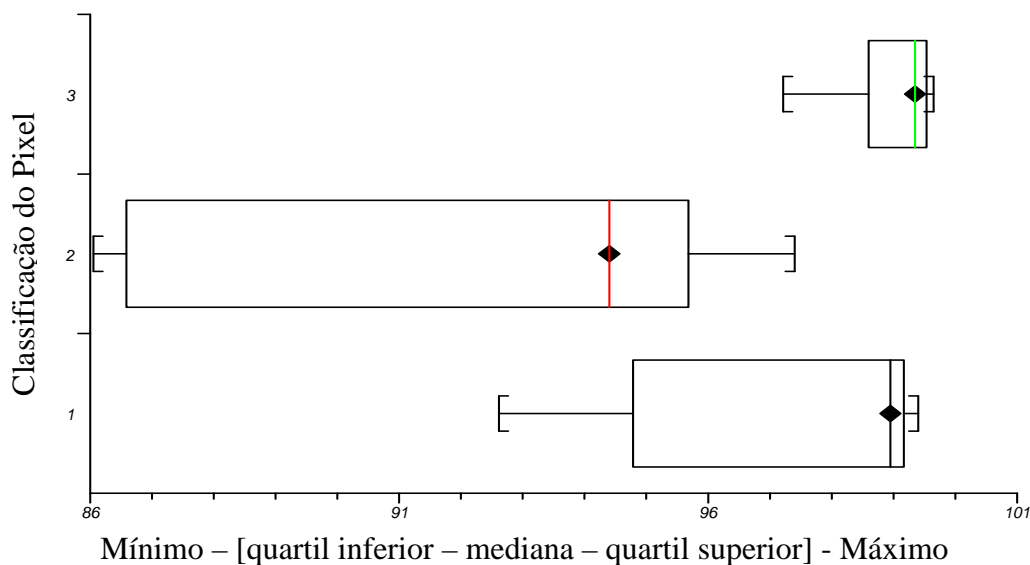


Figura 26: Valores das medianas das medidas F para a seqüência de imagens 01 com três tipos de classificação dos *pixels*. Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Na Figura 27 são mostrados os resultados obtidos com os três tipos de classificação dos *pixels*. As imagens estão organizadas da seguinte maneira: a) imagem de referência com objeto; b) Classificação do *Pixel* 01; c) Classificação do *Pixel* 02; e d) Classificação do *Pixel* 03. Utilizando-se apenas um modelo gaussiano com a detecção de sombras (Figura 27-c), alguns *pixels* pertencentes a imagem de referência foram classificados como objeto. Neste caso, nos outros dois resultados estes falsos positivos não foram detectados.

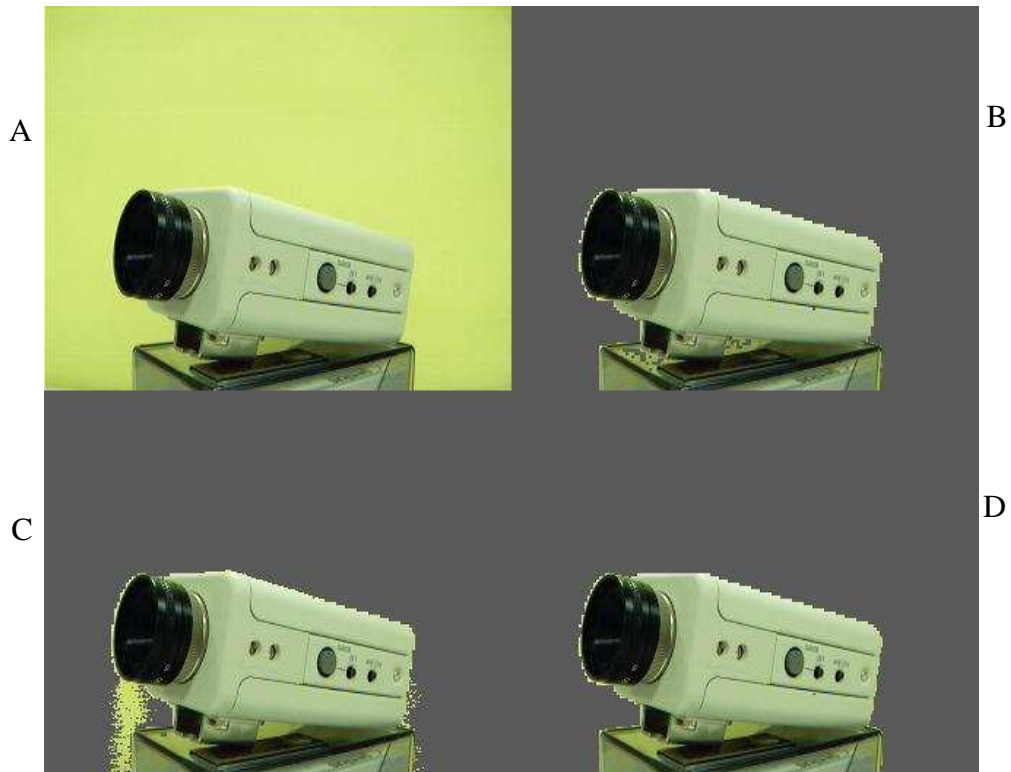


Figura 27: Resultados com a seqüência de imagens 01 com três tipos de classificação dos *pixels*. A) imagem de referência com objeto; B) Classificação do *Pixel* 01; C) Classificação do *Pixel* 02; e D) Classificação do *Pixel* 03.

4.3.1.2 Sequência de imagens 02

Os resultados da medida F utilizando o método proposto com a sequência de imagens 02 e três tipos de classificação dos *pixels* estão ilustrados na Figura 28. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F não são estatisticamente significantes ($p = 0,1124$).

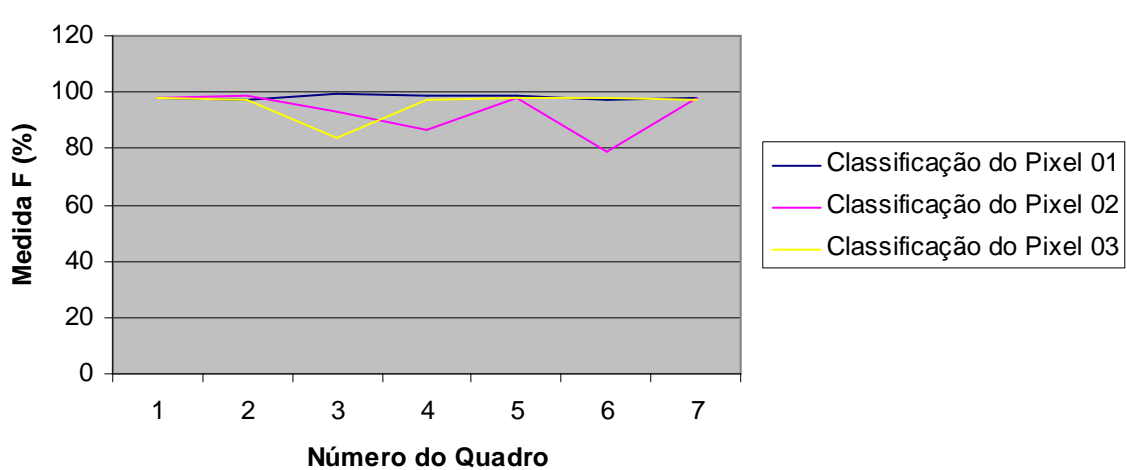


Figura 28: Medida F para sequência de imagens 02 com três tipos de classificação dos *pixels*. Teste de Kruskal-Wallis ($p = 0,1124$); Teste de igualdade de variância ($p = 0,0038$). Classificação do *Pixel 01* – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel 02* – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel 03* – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 4: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a sequência de imagens 02 com três tipos de classificação dos *pixels*.

Classificação do <i>Pixel</i>	p
01 vs. 02	0,0003
01 vs. 03	0,0151
02 vs. 03	0,083

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os três tipos de classificação dos *pixels*, foi rejeitada ($p = 0,0038$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 4, há uma diferença significativa entre todas as medidas F em relação à variância. Na Figura 29 está ilustrado o gráfico com os valores das medianas obtidos das medidas F para

cada tipo de classificação do *pixel*. De acordo com o gráfico, a utilização da Classificação do *Pixel* 01 resultou na maior mediana com uma das menores variâncias .

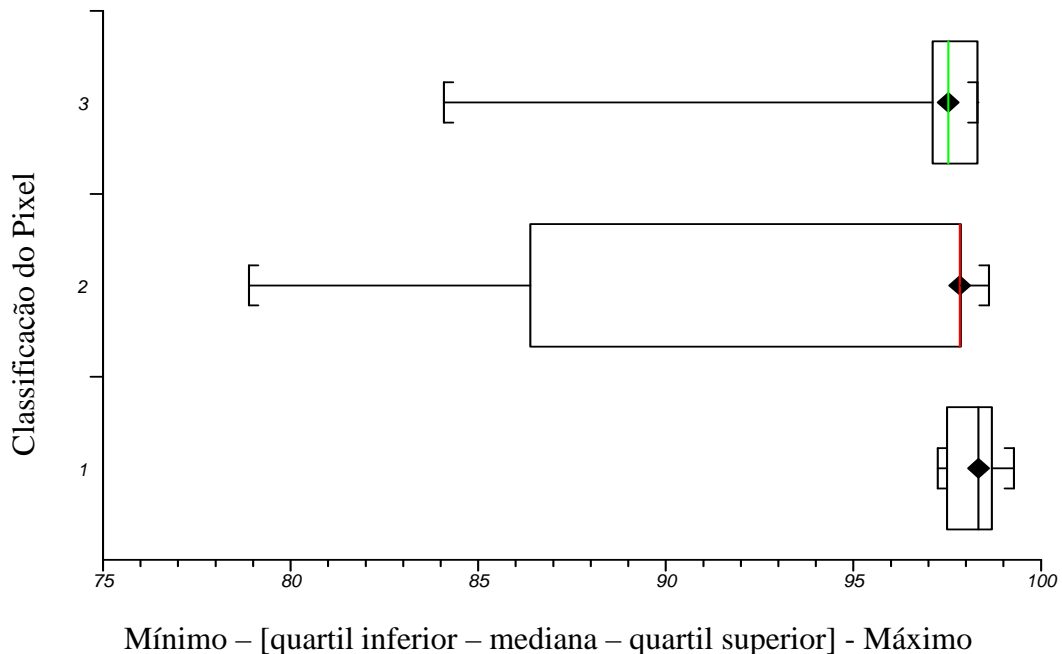


Figura 29: Valores das medianas das medidas F para a seqüência de imagens 02 com três tipos de classificação dos *pixels*. Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Na Figura 30 são mostrados os resultados obtidos com os três tipos de classificação dos *pixels*. As imagens estão organizadas da seguinte maneira: a) imagem de referência com objeto; b) Classificação do *Pixel* 01; c) Classificação do *Pixel* 02; e d) Classificação do *Pixel* 03. Utilizando-se apenas um modelo gaussiano com a detecção de sombras (Figura 30-c), diversos *pixels* pertencentes à imagem de referência foram classificados como objeto. Quando o objeto é inserido na cena, há modificações nas intensidades dos *pixels* no local e em torno do objeto conforme ilustrado na Figura 31.

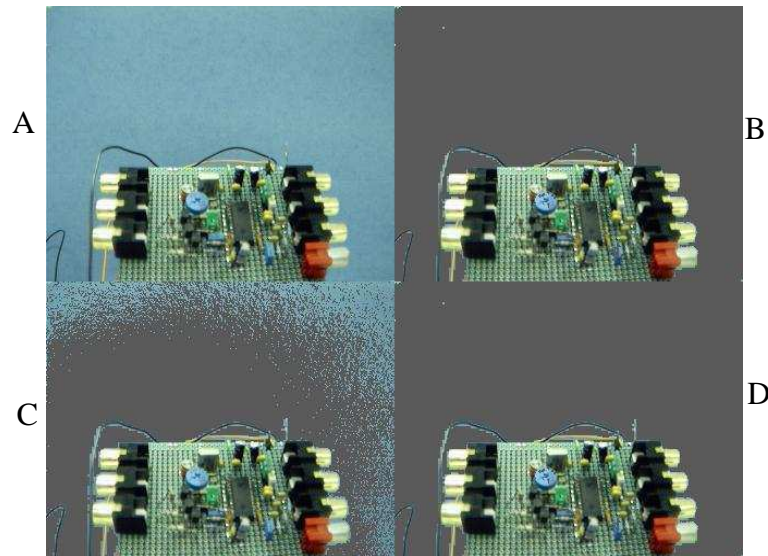


Figura 30: Resultados com a seqüência de imagens 02 e três tipos de classificação dos *pixels*. A) imagem de referência com objeto; B) Classificação do *Pixel* 01; C) Classificação do *Pixel* 02; e D) Classificação do *Pixel* 03.

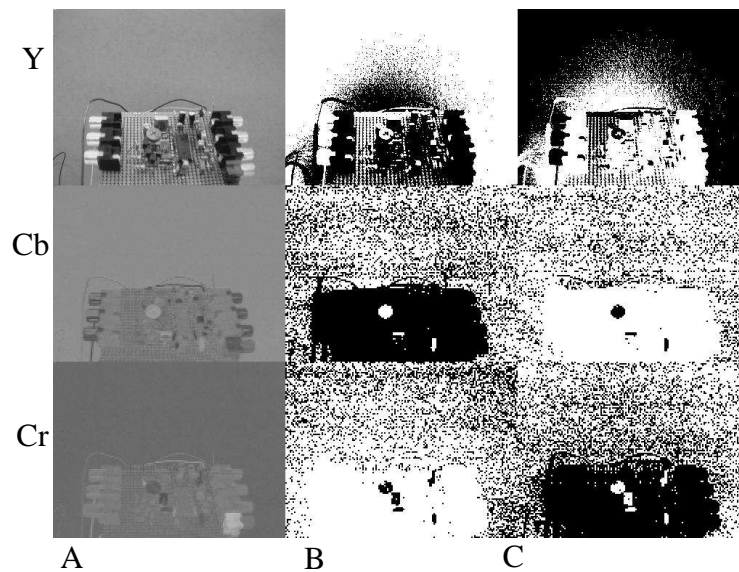


Figura 31: A) imagem de referência com objeto; B) os pontos pretos são os *pixels* detectados abaixo do limite inferior do intervalo de confiança, C) os pontos pretos são os *pixels* detectados acima do limite superior do intervalo de confiança.

Na Figura 31 estão ilustrados os *pixels* detectados fora dos intervalos de confiança para cada componente YCbCr. As seqüências de imagens estão organizadas da seguinte maneira: a) imagem de referência com objeto, b) os pontos pretos são os *pixels* detectados abaixo do limite inferior do intervalo de confiança, c) os pontos pretos são os *pixels* detectados acima do limite superior do intervalo de confiança. Desta maneira, a função de

compensação de variação do *pixel*, juntamente com os limiares de comparação, reduz a detecção dos falsos positivos ocasionados pela inserção do objeto na cena.

4.3.1.3 Sequência de imagens 03

Os resultados da medida F utilizando o método proposto com a sequência de imagens 01 e três tipos de classificação dos *pixels* estão ilustrados na Figura 32. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 5, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 01 e Classificação do *Pixel* 03.

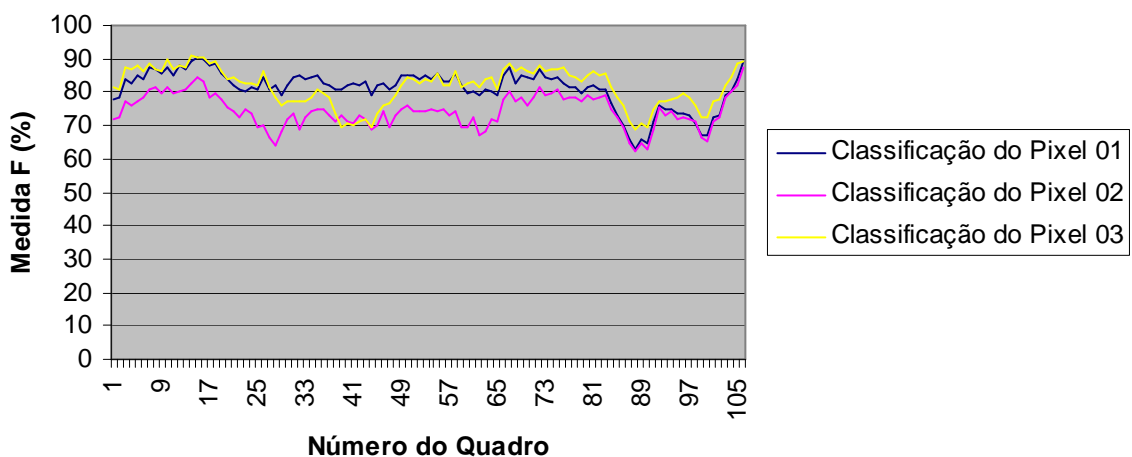


Figura 32: Medida F para sequência de imagens 03 com três tipos de classificação dos *pixels*. Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,0875$). Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 5: Comparação dos pares de parâmetros para o teste de Kruskal Wallis (p^*) das medidas F para a sequência de imagens 03 com três tipos de classificação dos *pixels*.

Classificação do <i>Pixel</i>	p^*
01 vs. 02	$< 0,0001$
01 vs. 03	0,7199
02 vs. 03	$< 0,0001$

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os três tipos de classificação dos *pixels*, foi aceita ($p = 0,0875$). Na Figura 33 está ilustrado o gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação do *pixel*. De acordo com o gráfico, a Classificação do *Pixel* 01 e 03 obtiveram a maior mediana, sendo que a menor variância foi obtida com a 01. Isto significa que a utilização da função de compensação das variações do *pixel* e da detecção de sombras resulta em um melhor desempenho na detecção dos objetos neste cenário.

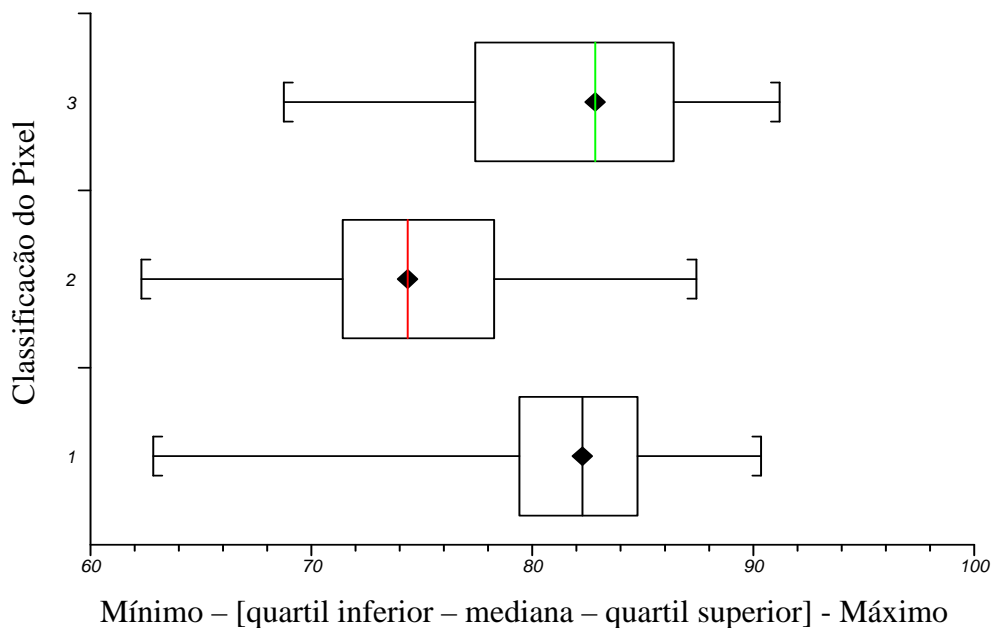


Figura 33: Valores das medianas das medidas F para a seqüência de imagens 03 com três tipos de classificação dos *pixels*. Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Na Figura 34 são mostrados os resultados obtidos com os três tipos de classificação dos *pixels*. As imagens estão organizadas da seguinte maneira: a) Imagem de referência com objeto; b) Classificação do *Pixel* 01; c) Classificação do *Pixel* 02; e d) Classificação do *Pixel* 03. Utilizando-se apenas um modelo gaussiano com a detecção de sombras (Figura 34-c), foi possível eliminar as sombras, porém alguns *pixels* pertencentes à imagem de referência foram classificados como objeto. Neste caso, foi necessário detectar as sombras e utilizar também a função de compensação das variações do *pixel* para obtenção do melhor resultado (Figura 34-b).



Figura 34: Resultados com a seqüência de imagens 03 e três tipos de classificação dos *pixels*. A) imagem de referência com objeto; B) Classificação do *Pixel* 01; C) Classificação do *Pixel* 02; e D) Classificação do *Pixel* 03.

4.3.1.4 Seqüência de imagens 04

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 04 e três tipos de classificação dos *pixels* estão ilustrados na Figura 35. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 6, há uma diferença significativa entre todas as medidas F.

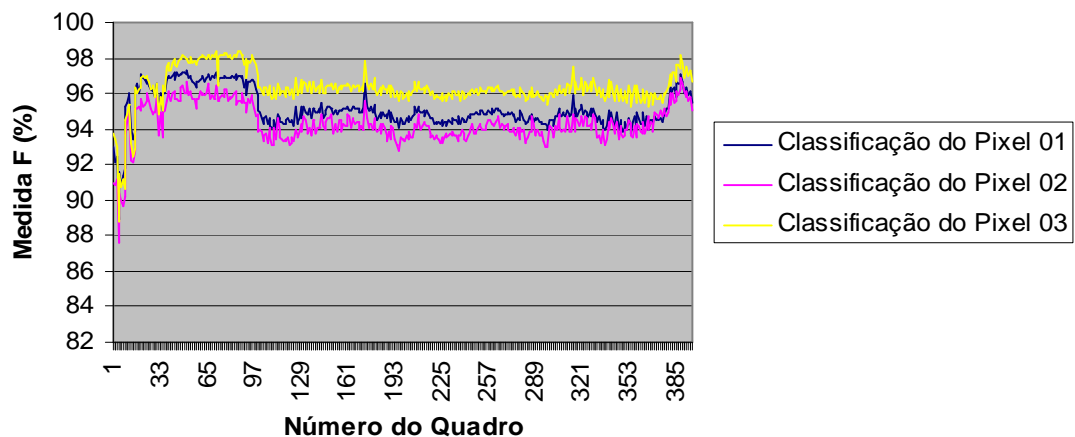


Figura 35: Medida F para seqüência de imagens 04 com três tipos de classificação dos *pixels*. Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$). Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 6: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{}) das medidas F para a seqüência de imagens 04 com três tipos de classificação dos *pixels*.**

Classificação do <i>Pixel</i>	p^*	p^{**}
01 vs. 02	< 0,0001	0,7448
01 vs. 03	< 0,0001	< 0,0001
02 vs. 03	< 0,0001	0,0001

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os três tipos de classificação dos *pixels*, foi rejeitada ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 6, não há diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 01 e 02 em relação à variância. Na Figura 36 é ilustrado o gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação do *pixel*. De acordo com o gráfico, a utilização da Classificação do *Pixel* 03 resultou na maior mediana com a menor variância. Isto significa que a utilização da função de compensação das variações do *pixel* sem detecção de sombras resulta em um melhor desempenho na detecção dos objetos neste cenário.

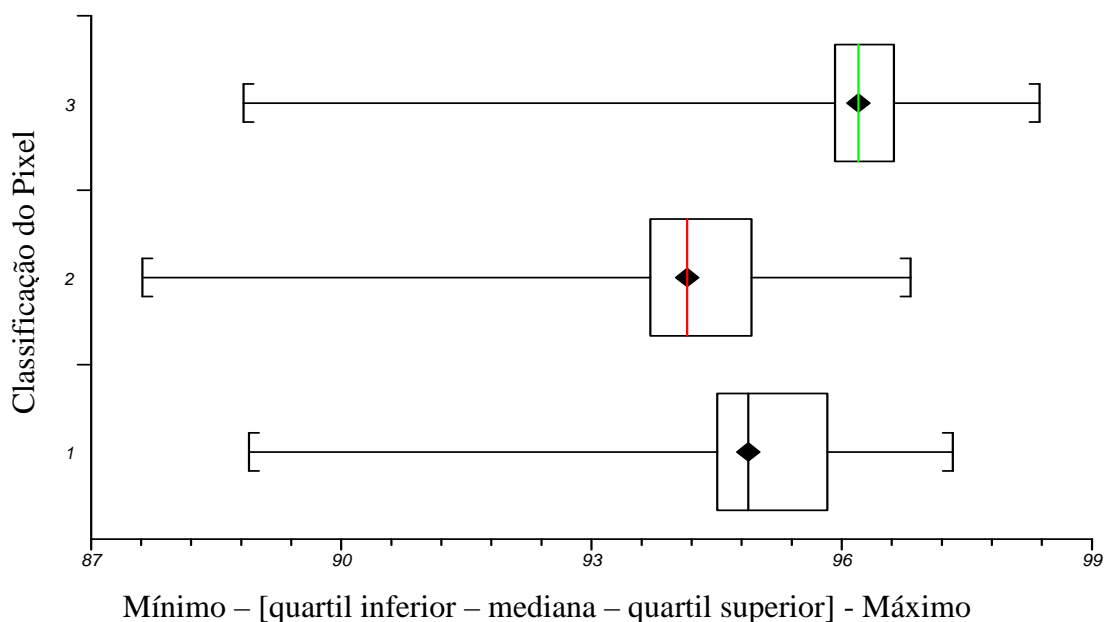


Figura 36: Valores da mediana das medidas F para a seqüência de imagens 04 com três tipos de classificação dos *pixels*. Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Os resultados obtidos com os três tipos de classificação dos *pixels* estão mostrados na Figura 37. As imagens estão organizadas da seguinte maneira: a) Imagem de referência com objeto; b) Classificação do *Pixel* 01; c) Classificação do *Pixel* 02; e d) Classificação do *Pixel* 03. Utilizando-se apenas um modelo gaussiano com a detecção de sombras (Figura 37-c), alguns *pixels* pertencentes à imagem de referência foram classificados como objeto. Neste caso, nos outros dois resultados foi detectada uma quantidade menor de falsos positivos.

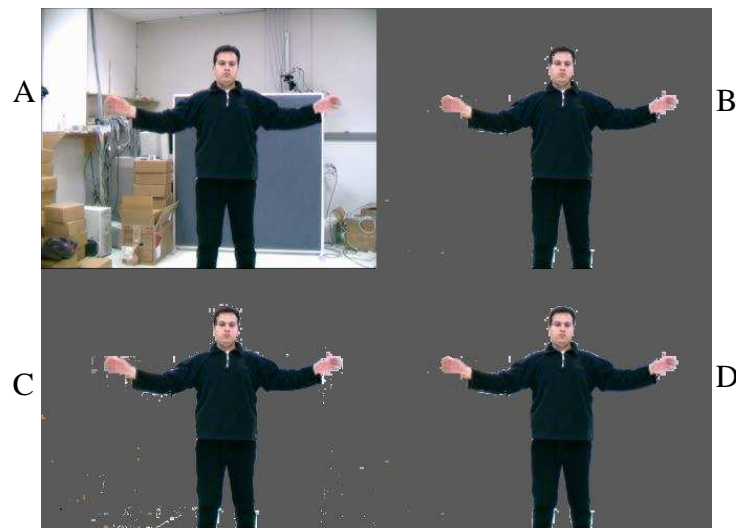


Figura 37: Resultados com a seqüência de imagens 04 e três tipos de classificação dos *pixels*. A) imagem de referência com objeto; B) Classificação do *Pixel* 01; C) Classificação do *Pixel* 02; e D) Classificação do *Pixel* 03.

4.3.1.5 Seqüência de imagens 05

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 05 e três tipos de classificação dos *pixels* estão ilustrados na Figura 38. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 7, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 01 e Classificação do *Pixel* 03.

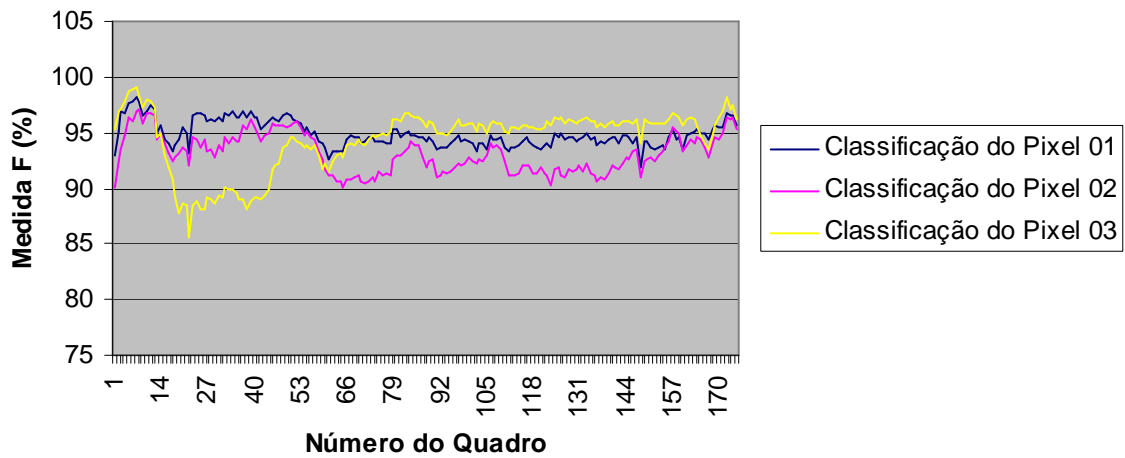


Figura 38: Medida F para seqüência de imagens 05 com três tipos de classificação dos *pixels*. Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$). Classificação do *Pixel 01* – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel 02* – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel 03* – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 7: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 05 com três tipos de classificação dos *pixels*.

Classificação do <i>Pixel</i>	p^*	p^{**}
01 vs. 02	< 0,0001	< 0,0001
01 vs. 03	0,5099	0,8915
02 vs. 03	< 0,0001	0,0034

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os três tipos de classificação dos *pixels*, foi rejeitada ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 7, há uma diferença significativa entre todas as medidas F em relação à variância. O gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação do *pixel* é exibido na Figura 39. De acordo com o gráfico, a utilização da Classificação do *Pixel 03* resultou na maior mediana, porém a menor variância foi obtida com a Classificação do *Pixel 01*.

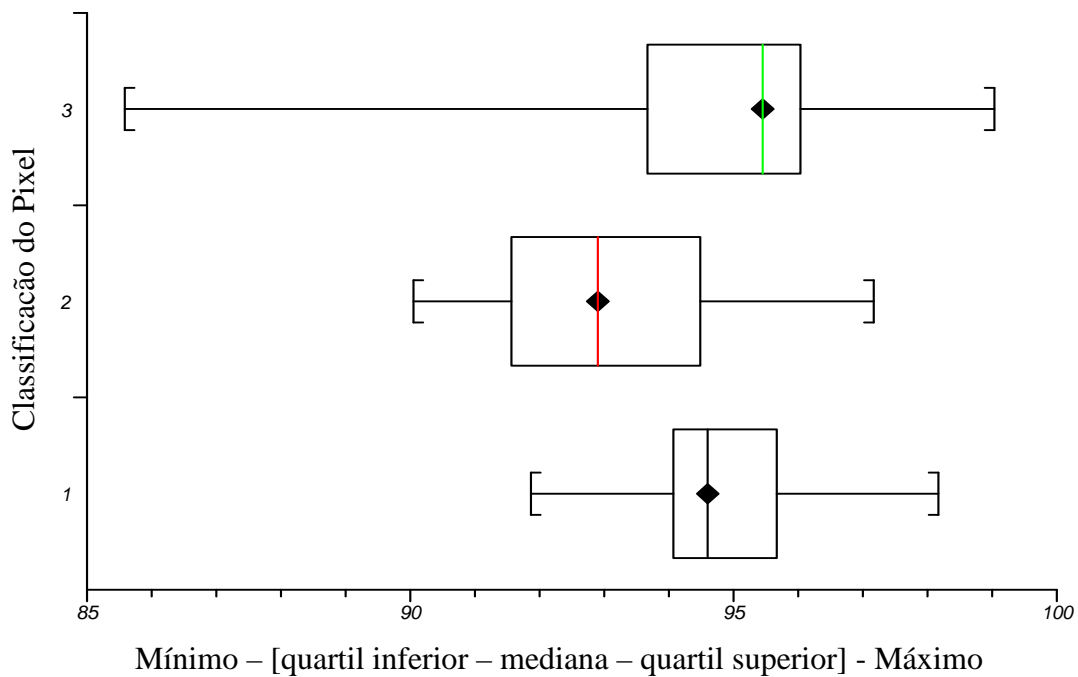


Figura 39: Valores da mediana das medidas F para a seqüência de imagens 01 com três tipos de classificação dos *pixels*. Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizados durante a detecção de objetos; classificação do *pixel* 02 – somente a detecção de sombras é utilizada durante a detecção de objetos; e classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Na Figura 40 são mostrados os resultados obtidos com os três tipos de classificação dos *pixels*. As imagens estão organizadas da seguinte maneira: a) imagem de referência com objeto; b) Classificação do *Pixel* 01; c) Classificação do *Pixel* 02; e d) Classificação do *Pixel* 03. Utilizando-se apenas um modelo gaussiano com a detecção de sombras (Figura 40-c), foi possível eliminar as sombras, porém alguns *pixels* pertencentes à imagem de referência foram classificados como objeto. Neste caso, foi necessário detectar as sombras e utilizar também a função de compensação das variações do *pixel* para obtenção do melhor resultado (Figura 40-b).

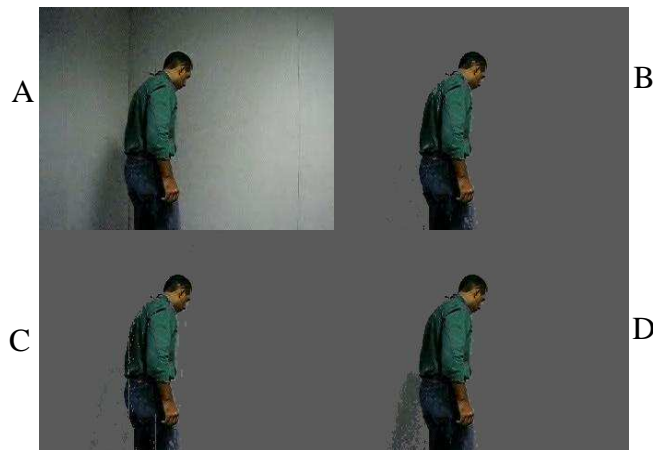


Figura 40: Resultados com a seqüência de imagens 05 e três tipos de classificação dos *pixels*. A) imagem de referência com objeto; B) Classificação do *Pixel* 01; C) Classificação do *Pixel* 02; e D) Classificação do *Pixel* 03.

4.3.2 Comparação dos espaços de cores

Nesta seção são apresentados os resultados da avaliação do método de detecção de objetos utilizando-se os espaços de cores, RGB, YCbCr, YUV, e HSV. O objetivo desta avaliação é verificar se há diferença no desempenho do método de detecção de objetos em diferentes espaços de cores.

Para realizar as comparações entre os espaços de cores, a detecção de sombras foi desligada, pois a técnica utilizada neste trabalho depende da informação de luminância (Y) e no espaço RGB esta informação deveria ser separada antes da aplicação desta técnica. Além disso, os limites inferiores utilizados para a função de compensação de variação do *pixel* foram os mesmos para os quatro espaços de cores.

4.3.2.1 Espaços de cores

O espaço de cores RGB é formado pelas cores primárias aditivas: vermelho (R), verde (G) e azul (B). Com estas três cores é possível gerar a maioria das cores do espectro visível variando o valor de cada componente RGB entre 0 e 255. O espaço de cores RGB é utilizado em dispositivos de exibição de imagens, tais como monitores e televisores e em dispositivos de captura de imagens. [65]

O espaço de cores YCbCr é formado pela componente Y que representa a informação de luminância, e pelas componentes Cb e Cr, que representam as cores azul e vermelho, respectivamente. [65]

O espaço de cores YUV é composto pela componente Y que representa a luminância e pelas componentes U e V que correspondem a crominância e descrevem os atributos de matiz e saturação da imagem. [65]

O espaço de cores HSV é formado por uma componente H que define o matiz da imagem, uma componente S que representa a saturação, e a componente V que define o valor da intensidade ou brilho em tons de cinza. [11]

4.3.2.2 Sequência de imagens 01

Os resultados da medida F utilizando o método proposto com a sequência de imagens 01 e quatro espaços de cores estão ilustrados na Figura 41. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p = 0,0049$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 8, não há uma diferença significativa entre as medidas F obtidas com os seguintes espaços: RGB, YUV, e YCbCr.

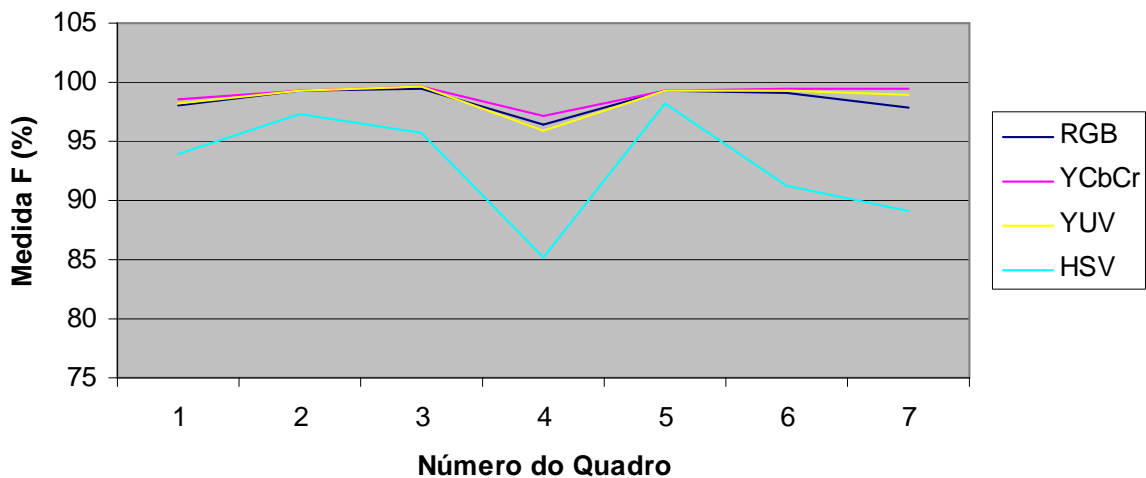


Figura 41: Medida F para sequência de imagens 01 obtidas em diferentes espaços de cores. Teste de Kruskal Wallis ($p = 0,0049$); Teste de igualdade de variância ($p = 0,0011$).

Tabela 8: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a sequência de imagens 01 com quatro espaços de cores.

Planos de Cores	p^*	p^{**}
RGB vs. YCbCr	0,1752	0,1671
RGB vs. YUV	0,5314	0,6657
RGB vs. HSV	0,0068	0,0004
YCbCr vs. YUV	0,4535	0,3333
YCbCr vs. HSV	0,0002	< 0,0001
YUV vs. HSV	0,0014	0,0001

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os quatro espaços de cores, foi rejeitada ($p = 0,0011$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 8, há uma diferença significativa entre todas as medidas F em relação à variância. O gráfico em caixa com os valores das medianas obtidos das medidas F para cada espaço de cores é exibido na Figura 42. De acordo com o gráfico, a utilização do espaço de cores HSV resultou na menor mediana com a maior variância. Isto significa que a utilização do espaço de cores HSV resulta em uma maior variação das medidas F e, por conseguinte do desempenho do método de segmentação de vídeo implementado neste trabalho, de acordo com os objetos inseridos neste cenário.

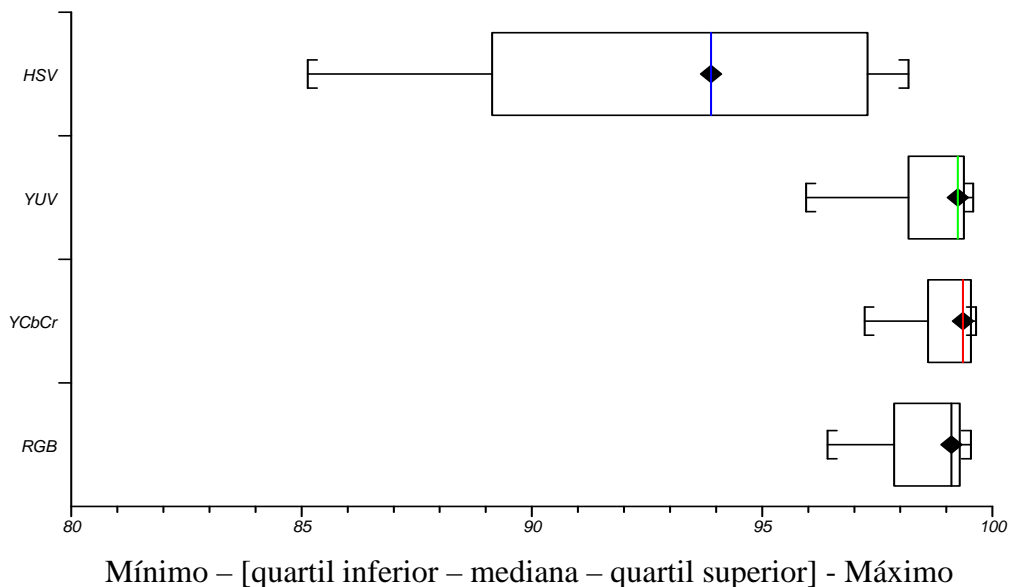


Figura 42: Valores das medianas das medidas F para a seqüência de imagens 01 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.

Na Figura 43 são mostrados os resultados obtidos com os quatro espaços de cores e a seqüência de imagens 01. Utilizando-se o HSV, alguns *pixels* em torno do objeto, pertencentes à imagem de referência, foram classificados como objeto. As intensidades dos *pixels* em torno do objeto sofreram alterações em relação a imagem de referência sem o objeto inserido. Estas variações de intensidade podem gerar os valores máximos da componente H para cada *pixel* nesta região e por isso ocorre a detecção destes falsos positivos.

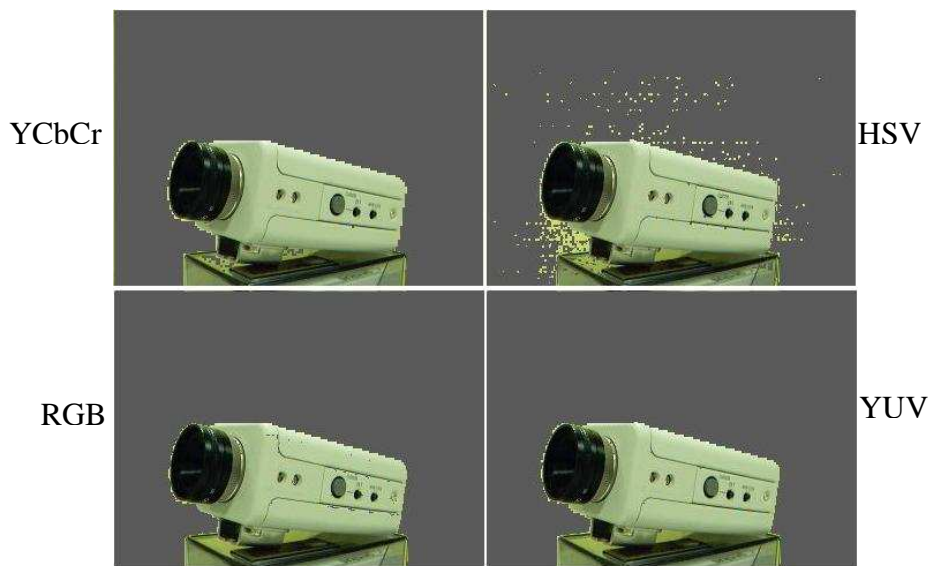


Figura 43: Resultados com a seqüência de imagens 01 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.

4.3.2.3 Sequência de imagens 04

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 04 e quatro espaços de cores estão ilustrados na Figura 44. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 9, não há uma diferença significativa entre as medidas F obtidas com os seguintes espaços: RGB e HSV.

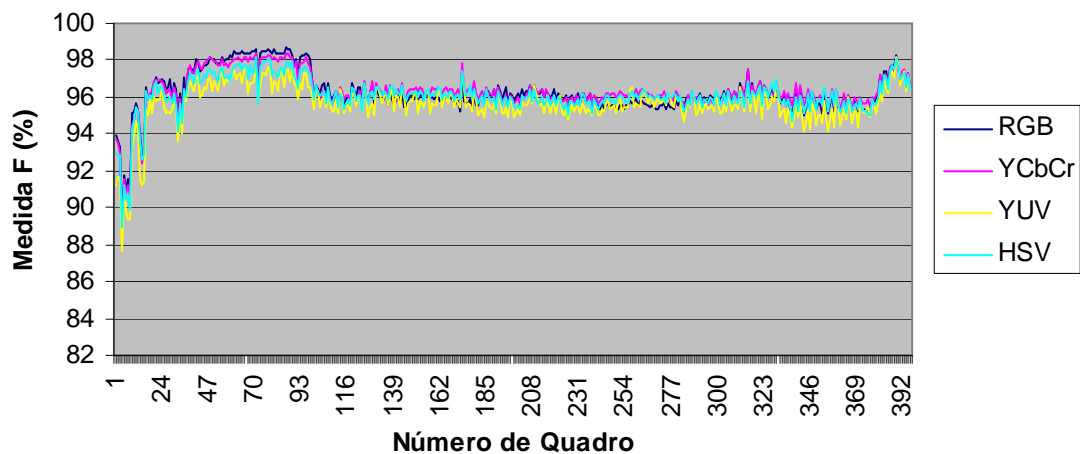


Figura 44: Medida F para seqüência de imagens 04 obtidas em diferentes espaços de cores. Teste de Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,0011$).

Tabela 9: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{}) das medidas F para a seqüência de imagens 04 com quatro espaços de cores.**

Planos de Cores	p^*	p^{**}
RGB vs. YCbCr	0,003	0,002
RGB vs. YUV	< 0,0001	0,1093
RGB vs. HSV	0,8415	0,0003
YCbCr vs. YUV	< 0,0001	0,1368
YCbCr vs. HSV	< 0,0001	0,5666
YUV vs. HSV	< 0,0001	0,0394

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os quatro espaços de cores, foi rejeitada ($p = 0,0011$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 9, não há diferenças significativas entre as medidas F, em relação à variância, obtidas com os seguintes espaços: RGB e YUV, YCbCr e YUV, YCbCr e HSV. O gráfico com os valores das medianas obtidos das medidas F para cada espaço de cores é exibido na Figura 45. De acordo com o gráfico, as medidas F obtidas em todos os planos de cores foram acima de 95 neste cenário.

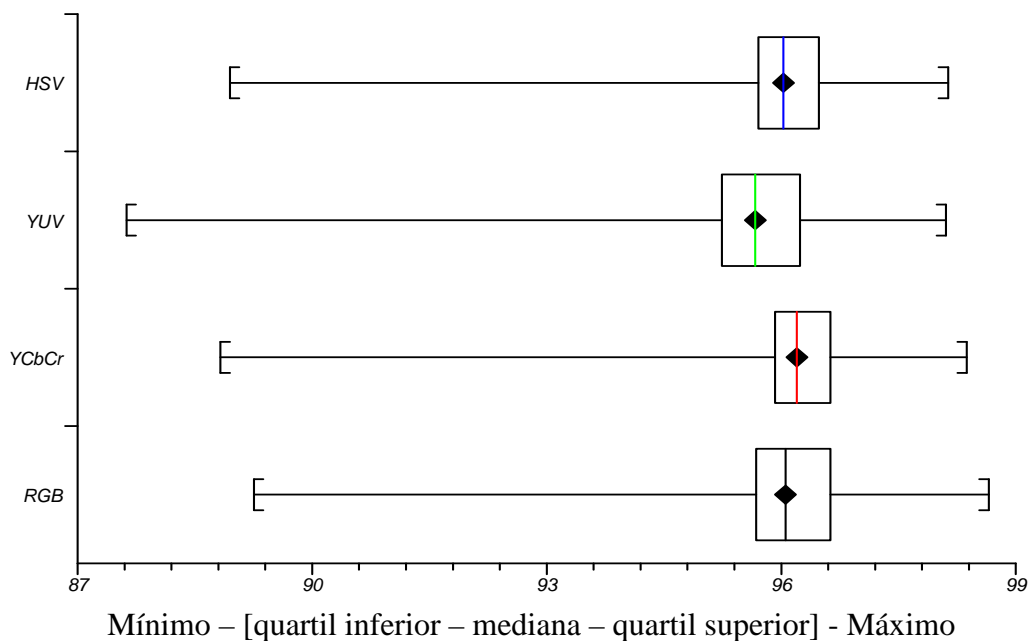


Figura 45: Valores das medianas das medidas F para a seqüência de imagens 04 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.

Na Figura 46 são mostrados os resultados obtidos com os quatro espaços de cores e a seqüência de imagens 04. Utilizando-se o YUV, alguns *pixels* em torno do objeto, pertencentes à imagem de referência, foram classificados como objeto. Neste caso, nos outros três resultados a quantidade de falsos positivos foi menor.

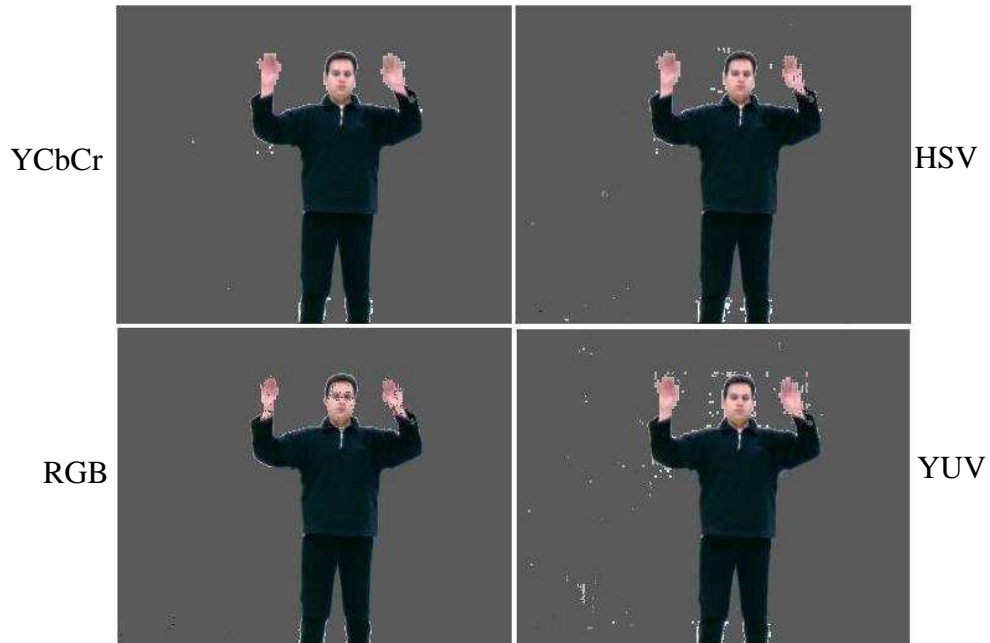


Figura 46: Resultados com a seqüência de imagens 04 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.

4.3.2.4 Sequência de imagens 05

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 05 e quatro espaços de cores estão ilustrados na Figura 47. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 10, há uma diferença significativa entre todas as medidas F obtidas com os quatro espaços de cores.

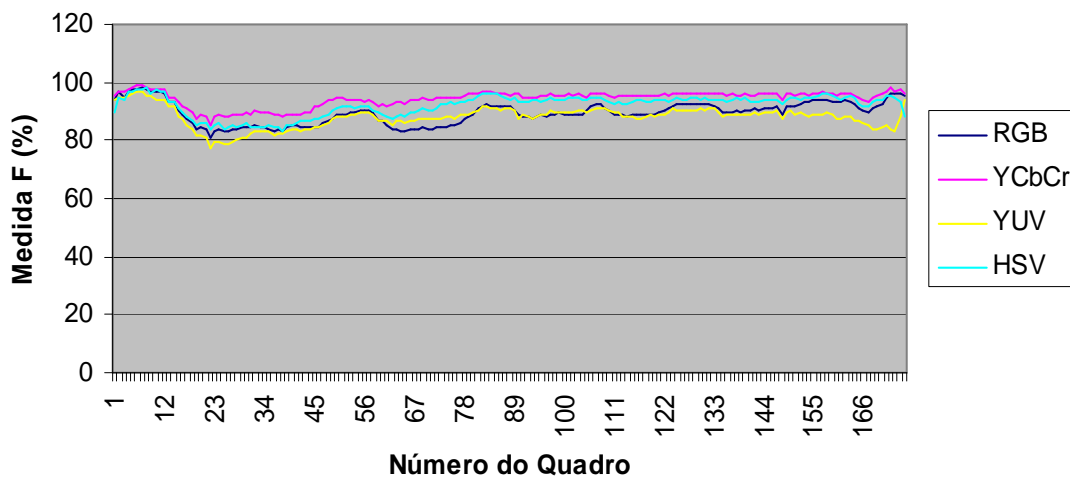


Figura 47: Medida F para sequência de imagens 05 obtidas em diferentes espaços de cores. Teste de Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$).

Tabela 10: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a sequência de imagens 05 com quatro espaços de cores.

Planos de Cores	p^*	p^{**}
RGB vs. YCbCr	$< 0,0001$	$< 0,0001$
RGB vs. YUV	0,0026	0,014
RGB vs. HSV	$< 0,0001$	0,3577
YCbCr vs. YUV	$< 0,0001$	0,0135
YCbCr vs. HSV	$< 0,0001$	$< 0,0001$
YUV vs. HSV	$< 0,0001$	0,1231

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os quatro espaços de cores, foi rejeitada ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 10, não há diferenças significativas entre as medidas F, em relação à variância, obtidas com os seguintes espaços de cores: RGB e HSV, YUV e HSV. O gráfico com os valores das medianas obtidos das medidas F para cada espaço de cores é mostrado na Figura 48. De acordo com o gráfico, a utilização do espaço de cores YCbCr resultou na maior mediana com a menor variância. Neste caso, a utilização do espaço RGB resulta em uma maior variação das medidas F e, por conseguinte, do desempenho do método de segmentação de vídeo, implementado neste trabalho, de acordo com os objetos inseridos neste cenário.

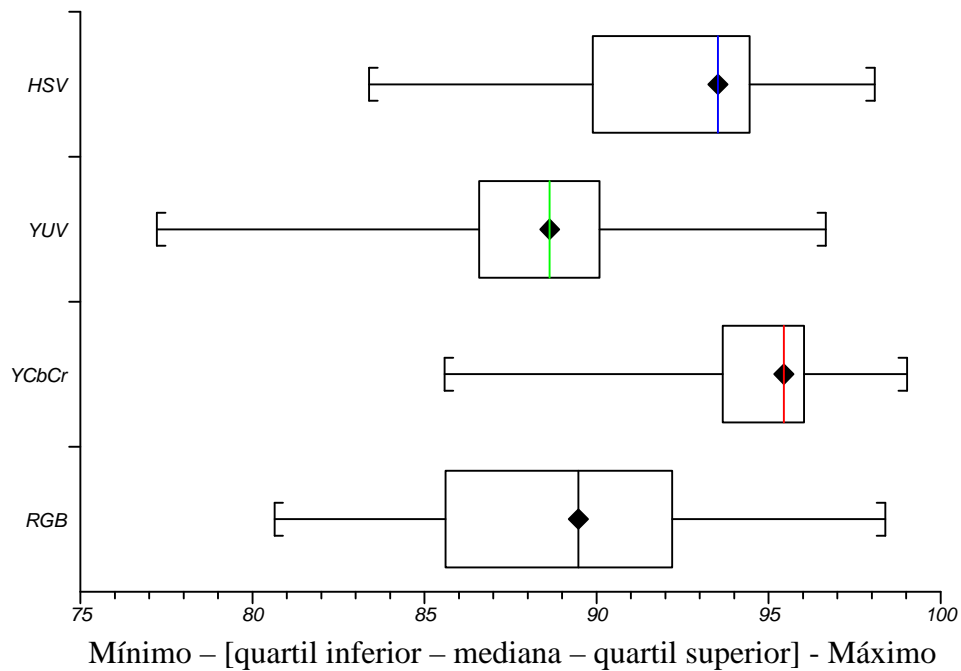


Figura 48: Valores das medianas das medidas F para a seqüência de imagens 05 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.

Na Figura 49 são mostrados os resultados obtidos com os quatro espaços de cores e a seqüência de imagens 05. Utilizando-se o HSV, alguns *pixels* em torno do objeto, pertencentes à imagem de referência, foram classificados como objeto. Neste caso, nos outros três resultados estes falsos positivos não foram detectados.

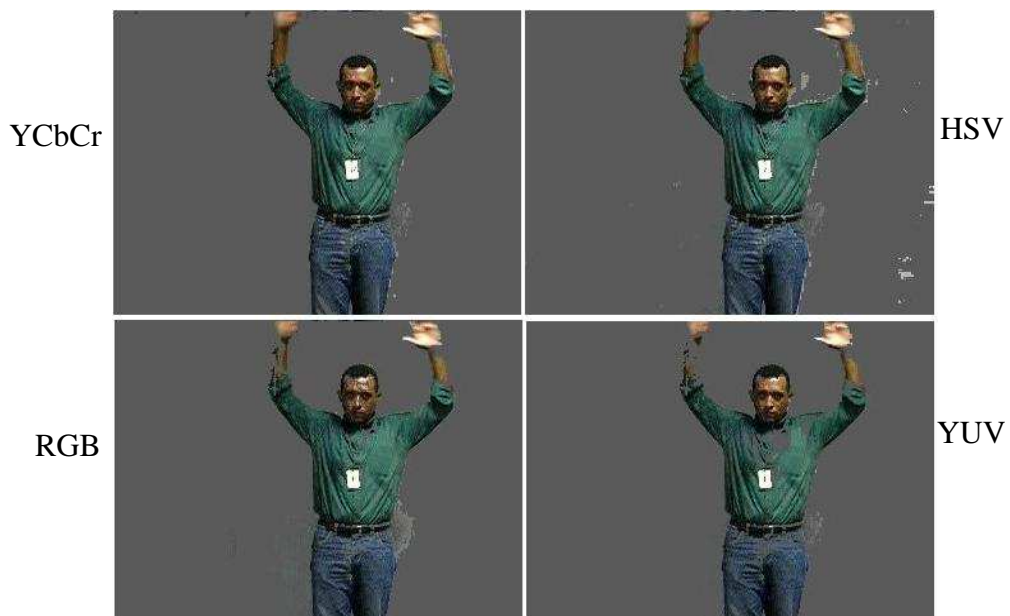


Figura 49: Resultados com a seqüência de imagens 05 com quatro espaços de cores: RGB, YCbCr, YUV, e HSV.

4.3.3 Comparação com adaptação recursiva

Nesta seção é apresentada a comparação entre o método de detecção de objetos com a função de compensação da variação do *pixel* e um método gaussiano com adaptação recursiva. Foram utilizadas três taxas de aprendizado para a adaptação recursiva, isto é, 0,01, 0,005, 0,0025. A taxa de aprendizado define a velocidade de atualização dos parâmetros do modelo da imagem de referência, ou seja, quanto maior for esta taxa, mais rapidamente os *pixels* estáticos serão incorporados ao modelo da imagem de referência.

4.3.3.1 Seqüência de imagens 03

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 01 e um método gaussiano com adaptação recursiva e diferentes taxas de aprendizado (TA) estão ilustrados na Figura 50. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p = 0,0001$). Nesta seqüência de imagens uma pessoa entra em uma sala, permanece parada por alguns instantes e depois começa a andar novamente. Conforme ilustrado nos gráficos da Figura 50, quanto maior a taxa de atualização mais rapidamente o modelo da imagem de referência é atualizado, ou seja, quando a pessoa começa a andar mais lentamente até parar, em função da atualização recursiva, ela começa a ser incorporada ao modelo da imagem de referência. Em função disso, os valores das medidas F começam a diminuir, pois uma quantidade cada vez menor de *pixels* é detectada. Quando a pessoa reinicia o seu movimento pela sala, as medidas F aumentam, pois ela começa a ser novamente detectada.

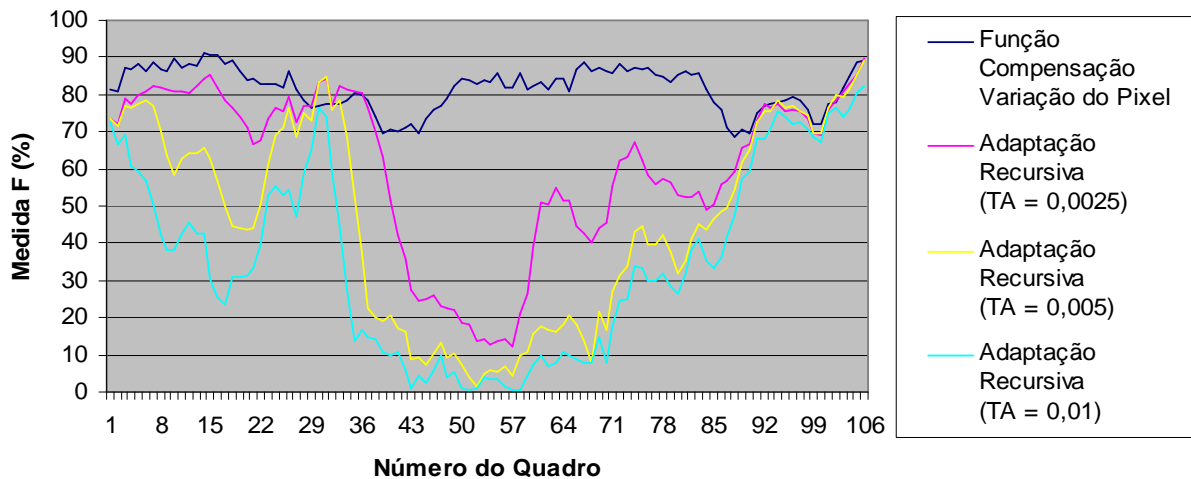


Figura 50: Medida F para seqüência de imagens 03 obtidas utilizando-se a função de compensação de variação do *pixel* e a adaptação recursiva com diferentes taxas de atualização (TA). Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$).

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F dos métodos gaussianos com adaptação recursiva e função de compensação do *pixel*, foi rejeitada ($p < 0,0001$). O gráfico com os valores das medianas obtidos das medidas F para cada método gaussiano é ilustrado na Figura 51. De acordo com o gráfico, a utilização do método implementado neste trabalho resultou na maior mediana com a menor variância.

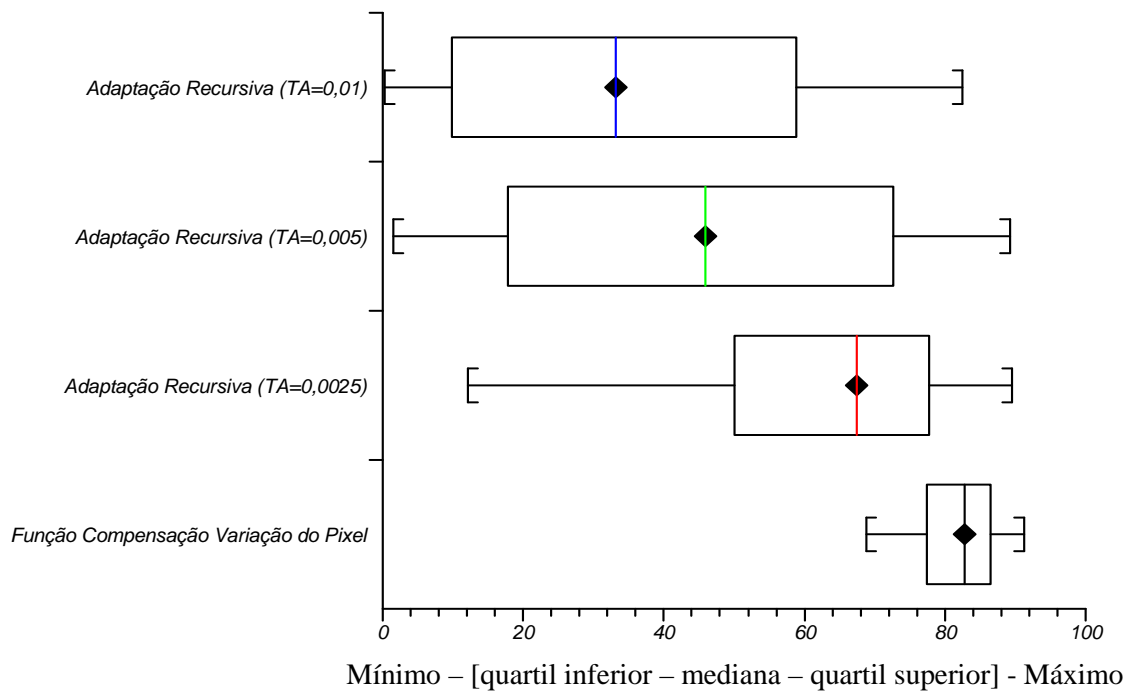


Figura 51: Valores das medianas das medidas F para seqüência de imagens 03 obtidas utilizando-se a função de compensação de variação do *pixel* e a adaptação recursiva com diferentes taxas de atualização (TA).

4.3.3.2 Seqüência de imagens 04

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 04 e um método gaussiano com adaptação recursiva e diferentes taxas de aprendizado (TA) estão ilustrados na Figura 52. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Na seqüência de imagens 04, uma pessoa entra em uma sala, permanece parada por alguns instantes movimentando os braços e depois começa a andar novamente. Conforme ilustrado nos gráficos da Figura 52, quanto maior a taxa de atualização mais rapidamente o modelo da imagem de referência é atualizado e, por conseguinte, a quantidade de elementos de imagens do objeto (pessoa) detectadas é reduzida, pois ela começa a fazer parte do modelo da imagem de referência.

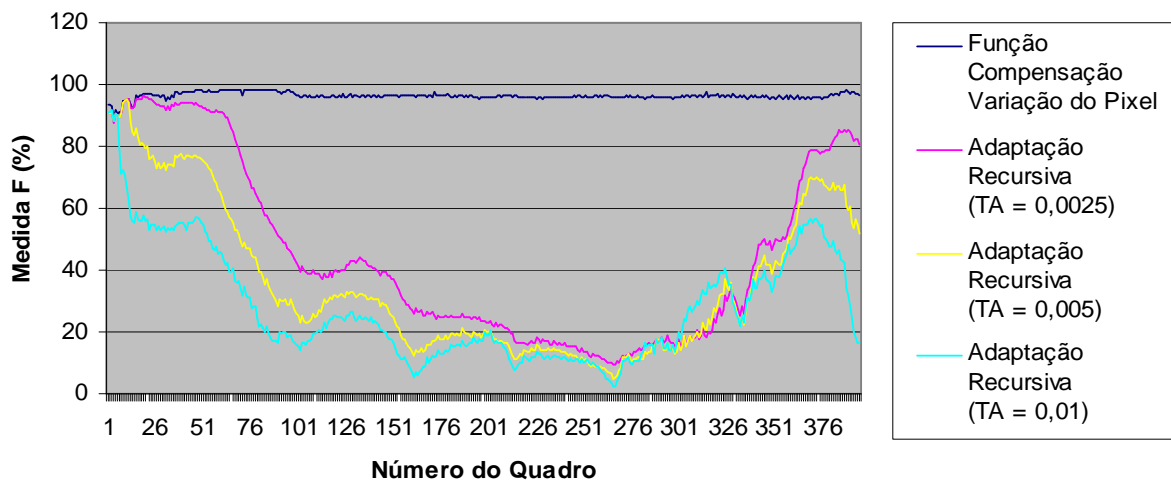


Figura 52: Medida F para seqüência de imagens 04 obtidas utilizando-se a função de compensação de variação do *pixel* e a adaptação recursiva com diferentes taxas de atualização (TA). Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$).

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F dos métodos gaussianos com adaptação recursiva e função de compensação do *pixel*, foi rejeitada ($p < 0,0001$). O gráfico com os valores das medianas obtidos das medidas F para cada método gaussiano é ilustrado na Figura 53. De acordo com o gráfico, a utilização do método implementado neste trabalho resultou na maior mediana com a menor variância. Em relação à seqüência de imagens 03, as medianas das medidas F neste cenário, com as diferentes taxas de adaptação recursiva, foram menores, pois na seqüência de imagens 04, a pessoa se movimenta mais lentamente e permanece mais tempo parada.

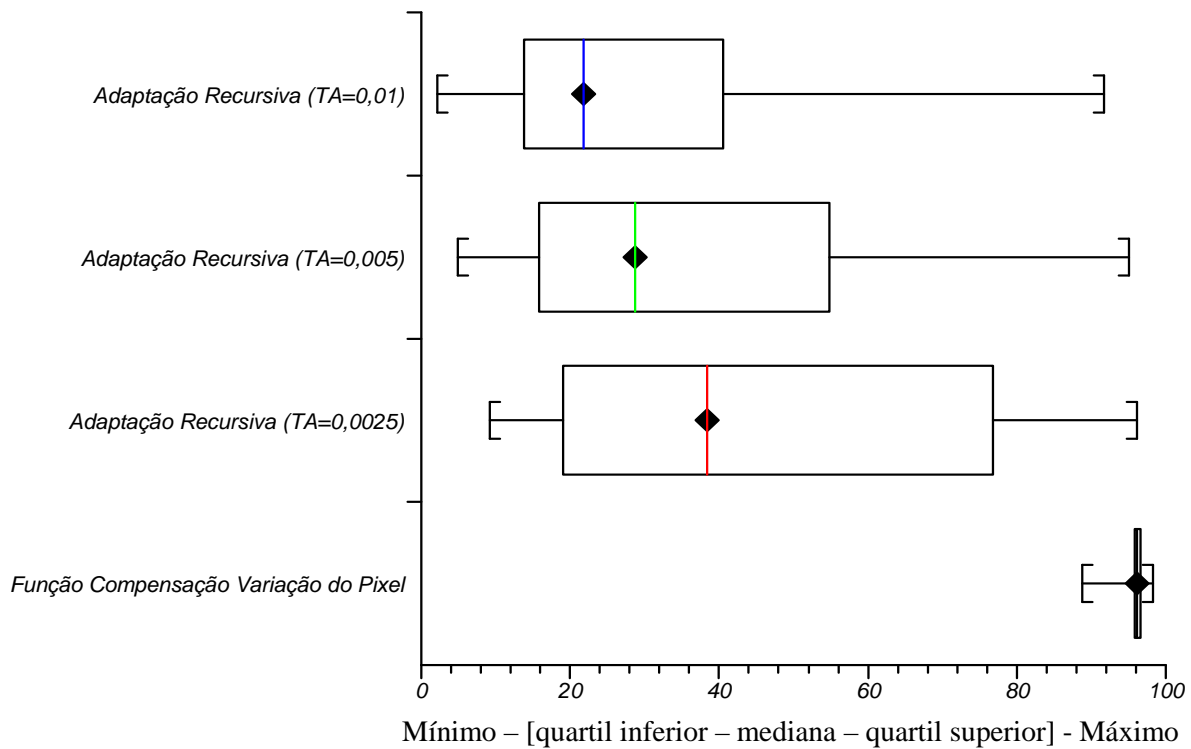


Figura 53: Valores das medianas das medidas F' para seqüência de imagens 04 obtidas utilizando-se a função de compensação de variação do *pixel* e a adaptação recursiva com diferentes taxas de atualização (TA).

4.3.3.3 Seqüência de imagens 05

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 05 e um método gaussiano com adaptação recursiva e diferentes taxas de aprendizado (TA) estão ilustrados na Figura 54. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Na seqüência de imagens 05, uma pessoa entra em uma sala, permanece parada por alguns instantes movimentando os braços e depois começa a andar novamente. Conforme ilustrado nos gráficos da Figura 54, quanto maior a taxa de atualização mais rapidamente o modelo da imagem de referência é atualizado e, por conseguinte, a quantidade de *pixels* do objeto (pessoa) detectada é reduzida, pois ela começa a fazer parte do modelo da imagem de referência.

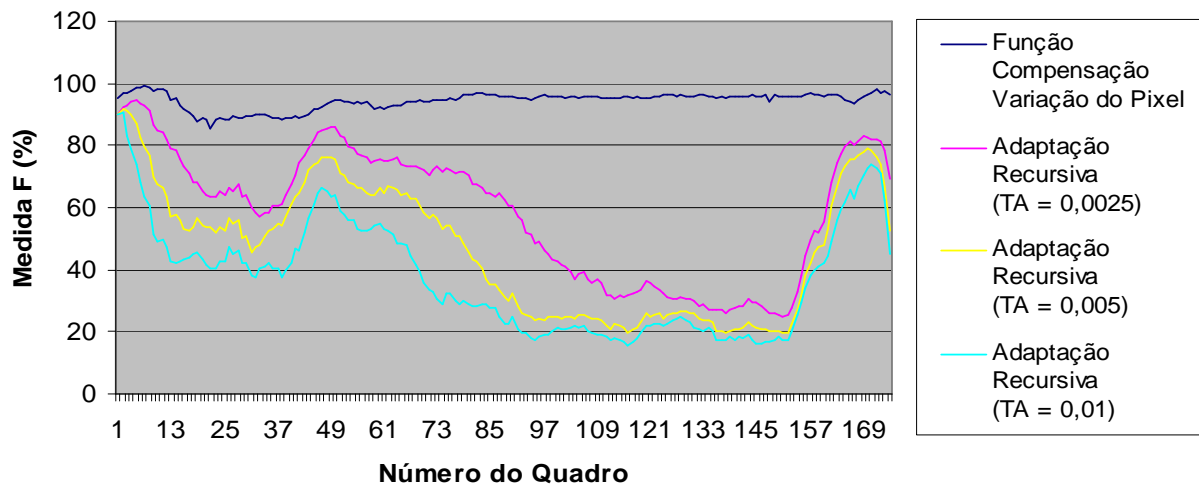


Figura 54: Medida F para seqüência de imagens 05 obtidas utilizando-se a função de compensação de variação do *pixel* e a adaptação recursiva com diferentes taxas de atualização (TA). Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$).

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F dos métodos gaussianos com adaptação recursiva e função de compensação do *pixel*, foi rejeitada ($p < 0,0001$). O gráfico com os valores das medianas obtidos das medidas F para cada método gaussiano é ilustrado na Figura 55. De acordo com o gráfico, a utilização do método implementado neste trabalho resultou na maior mediana com a menor variância.

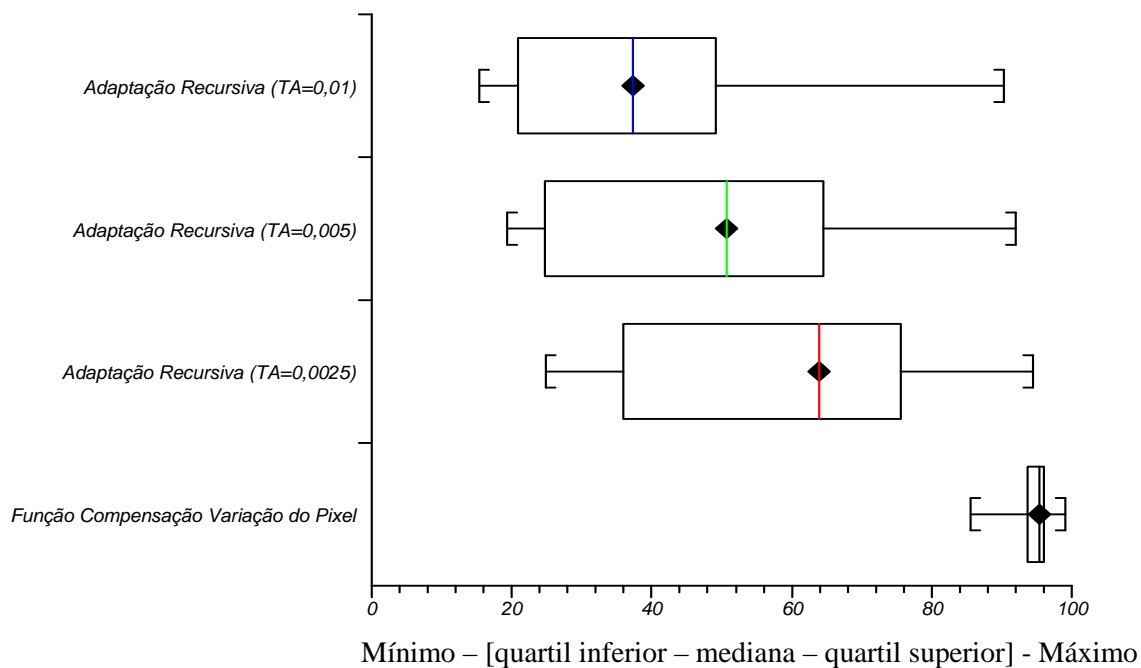


Figura 55: Valores das medianas das medidas F para seqüência de imagens 05 obtidas utilizando-se a função de compensação de variação do *pixel* e a adaptação recursiva com diferentes taxas de atualização (TA).

Na Figura 56 são mostrados os resultados obtidos com o método gaussiano implementado neste trabalho e um método gaussiano com adaptação recursiva e a taxa de aprendizado igual a 0,0025. As imagens estão organizadas da seguinte maneira: a) imagem de referência com objeto; b) método gaussiano com adaptação recursiva e a taxa de aprendizado igual a 0,0025; c) método gaussiano implementado neste trabalho. De acordo com os resultados, utilizando-se a adaptação recursiva quando a pessoa permanece parada, acaba sendo incorporada ao modelo da imagem de referência. Para a seqüência de imagens 05, apenas as mãos da pessoa em cena estão aparecendo porque são movimentados.



Figura 56: Resultados com as seqüências de imagens 03, 04 e 05 obtidas utilizando-se o método gaussiano com adaptação recursiva e taxa de atualização igual a 0,0025 (B) e o método gaussiano com função de compensação de variação do *pixel* (C).

4.4 CONCLUSÕES

Neste capítulo foram apresentados os resultados da avaliação do método para detecção de objeto implementado em MATLAB com operações em ponto flutuante. As métricas utilizadas nas avaliações foram as seguintes: precisão, recuperação e medida F. Para avaliar os parâmetros do método proposto e efetuar comparações de desempenho em diferentes cenários, espaços de cores, e com outros métodos, foram utilizados testes de inferência estatística. Para comparar as medidas F foi utilizado o teste não-paramétrico de análise de variância denominado Kruskal-Wallis juntamente com as comparações múltiplas com o método denominado Dwass-Steel. Além disso, para analisar a variância das medidas F em cada cenário de teste, foi utilizado um teste de igualdade de variância denominado teste de postos quadrados.

Para verificar a natureza da distribuição de cada *pixel* na seqüência de imagens de testes usada na avaliação do método, procedeu-se com o teste de normalidade de Shapiro Wilkson. De acordo com os resultados obtidos, pode-se afirmar com 95% de confiança que a maioria dos *pixels* nas componentes Y, Cb, e Cr apresentam uma distribuição normal nas imagens de teste utilizadas para avaliar o desempenho do método de detecção de objetos.

O método de detecção de objetos implementado neste trabalho foi avaliado utilizando-se três tipos de classificações dos *pixels* denominados Classificação do *Pixel* 01, Classificação do *Pixel* 02, e Classificação do *Pixel* 03. A Classificação do *Pixel* 01 utiliza a função de compensação de variação do *pixel* e a detecção de sombras durante a classificação dos *pixels*. A Classificação do *Pixel* 02 não utiliza a função de compensação de variação do *pixel* durante a classificação dos *pixels*, mas realiza a detecção de sombras. A Classificação do *Pixel* 03 utiliza somente a função de compensação de variação do *pixel* durante a detecção de objetos. Em alguns cenários, a utilização apenas da função de compensação do *pixel* produziu os melhores resultados.

Outra avaliação do método de detecção de objetos implementado neste trabalho foi efetuada utilizando-se os espaços de cores, RGB, YCbCr, YUV, e HSV. O objetivo desta avaliação foi verificar as diferenças no desempenho do método de detecção de objetos em diferentes espaços de cores. Os melhores resultados foram obtidos com os espaços de cores YCbCr e RGB seguidos do YUV e HSV.

Finalmente, foi feita a comparação entre o método de detecção de objetos com a função de compensação da variação do *pixel* e um método gaussiano com adaptação recursiva. Foram utilizadas três taxas de aprendizado para a adaptação recursiva, isto é, 0,01,

0,005, 0,0025. Nas seqüências de imagens de testes utilizadas para comparação dos métodos, uma pessoa entra em uma sala, permanece parada por alguns instantes e depois começa a andar novamente. De acordo com os resultados, quanto maior a taxa de atualização mais rapidamente o modelo da imagem de referência é atualizado. Quando a pessoa em cena nas seqüências de imagens de testes começa a andar mais lentamente até parar, em função da atualização recursiva, ela começa a ser integrada ao modelo da imagem de referência. Com isso, os valores da medida F diminuem, pois uma quantidade cada vez menor de *pixels*, pertencentes à pessoa em cena, são detectados. Isto demonstrou que esta técnica de atualização quando utilizada em aplicações cujo objeto deve ser detectado ainda que não esteja em movimento é inadequada.

No próximo capítulo as operações realizadas pelo método de detecção de objetos são convertidas para ponto fixo visando à implementação em hardware.

5 ESTRATÉGIAS PARA IMPLEMENTAÇÃO DE ALGORITMOS DE SEGMENTAÇÃO DE VIDEO EM HARDWARE

No capítulo anterior foram apresentados os resultados da implementação do método de detecção de objetos com operações em ponto flutuante. Neste capítulo são apresentadas as adaptações efetuadas para conversão das operações para ponto fixo visando a implementação em hardware. Além disso, fazem-se também as comparações dos resultados obtidos com operações em ponto flutuante e em ponto fixo.

5.1 INTRODUÇÃO

A primeira etapa para conversão de um algoritmo de um ambiente de desenvolvimento para aplicações em tempo real envolve a utilização de estratégias de simplificação. As principais estratégias encontradas nos artigos pesquisados são as seguintes: redução do número de operações [80, 81, 82] e a redução na quantidade de dados processados. [83, 84, 85, 86, 87, 19, 88]

Em relação à redução na quantidade de dados processados, as estratégias comumente utilizadas são: sub-amostragem [83, 84, 85, 86] e partição de imagens [19, 87, 88]. O objetivo da sub-amostragem é reduzir a quantidade de dados espaciais e temporais do sinal de vídeo processado. Em [89] a resolução do sinal de vídeo processado foi reduzida de 320 x 240 para 160 x 120 *pixels* para que a detecção de objetos fossem realizada em tempo real, isto é, 30 quadros por segundo. A partição das imagens consiste em dividir cada quadro processado em imagens menores de modo a processar paralelamente cada sub-imagem em um tempo menor que o necessário para processar um quadro completo.

A seguir são apresentadas algumas simplificações adotadas para o método de detecção de objetos, proposto neste trabalho, visando à redução da memória necessária para armazenar os resultados intermediários dos parâmetros do modelo da imagem de referência, bem como, reduzir o tamanho da memória necessária para armazenar os resultados finais.

5.2 ADAPTAÇÕES DO MÉTODO PROPOSTO PARA PONTO FIXO

A seguir são descritas as adaptações realizadas no método de detecção de objetos visando à implementação em hardware. Em hardware as operações disponíveis a um custo relativamente baixo são: soma, subtração, deslocamento para esquerda (multiplicação por dois), deslocamento para direita (divisão por dois) e as booleanas. As demais operações devem ser realizadas por blocos de hardware que executam algoritmos destas operações. As adaptações necessárias são: a utilização de números em ponto fixo, mantendo a exatidão de 7 bits fracionários, e a realização de operações de multiplicação e divisão com números de potência de dois [89, 90, 91]. A exatidão de 7 bits fracionários é obtida pela multiplicação e divisão dos termos das equações de cálculo de parâmetros pelo fator 2^7 . O fator 2^7 facilita a multiplicação e a divisão em hardware. Na seção 5.4 é feita a comparação entre duas versões do método de segmentação em ponto fixo com e sem a utilização de 7 bits fracionários nas operações visando avaliar as diferenças obtidas em caso.

5.2.1 Modelo da imagem de referência

Os parâmetros do modelo da imagem de referência são os valores médios e os desvios padrões dos elementos de imagens. De acordo com os resultados obtidos no capítulo 02, o número de quadros necessário para estimar estes parâmetros depende de cada cenário. Para atender a todos os cenários avaliados o número de quadros deve ser igual 52 para componente Y, 44 para componente Cb, e 49 para componente Cr. Adotou-se o número de quadros igual a 64 para simplificar as operações em hardware. Além disso, o desvio padrão foi multiplicado por 2^{14} para que o valor final deste parâmetro tenha uma exatidão de 7 bits fracionários.

5.2.1.1 Valor médio

A média aritmética dos *pixels* em cada componente Y,Cb,Cr é dada por:

$$\bar{x}_{i(k)}(Y, Cb, Cr) = \frac{1}{N} \sum_{i=1}^N I_{i(k)}(Y, Cb, Cr) \quad (3)$$

Substituindo-se o valor do número de quadros (N) na expressão anterior, obtêm-se:

$$\bar{x}_{i(k)}(Y, Cb, Cr) = \frac{1}{2^6} \sum_{i=1}^{64} I_{i(k)}(Y, Cb, Cr) \quad (20)$$

Lembrando que:

- a) i é o número do quadro em processamento;
- b) k é a coordenada (x,y) de cada *pixel*;
- c) $I_{i(k)}(Y,Cb,Cr)$ é o valor corrente de cada *pixel* no espaço de cores YCbCr;

5.2.1.2 Desvio padrão

O desvio padrão dos *pixels* em cada componente Y,Cb,Cr é obtido por:

$$S_k(Y, Cb, Cr) = \sqrt{\frac{1}{N} \sum_{i=1}^N [I_{i(k)}(Y, Cb, Cr) - \bar{x}_{i(k)}(Y, Cb, Cr)]^2} \quad (4)$$

Substituindo-se o valor do número de quadros (N) e multiplicando-se a expressão por 2^{14} , obtêm-se:

$$S_k(Y, Cb, Cr) = \sqrt{\frac{(2^7)^2}{2^6} \sum_{i=1}^{64} [I_{i(k)}(Y, Cb, Cr) - \bar{x}_{i(k)}(Y, Cb, Cr)]^2}$$

$$S_k(Y, Cb, Cr) = \sqrt{2^8 \sum_{i=1}^{64} [I_{i(k)}(Y, Cb, Cr) - \bar{x}_{i(k)}(Y, Cb, Cr)]^2} \quad (21)$$

Lembrando que:

- a) $I_{i(k)}(Y,Cb,Cr)$ é o valor corrente de cada *pixel* no espaço de cores YCbCr;
- b) $\bar{x}_{i(k)}(Y, Cb, Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr.

5.2.2 Intervalo de confiança

O limite superior do intervalo de confiança para cada *pixel* em cada componente Y,Cb,Cr é dado por:

$$UL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) + t_u \frac{s_{(k)}(Y, Cb, Cr)}{\sqrt{N}} \quad (8)$$

O t_u é o percentil com valor u da distribuição t-student com $N - 1$ graus de liberdade. Considerando-se o número de quadros igual a 64, o valor do percentil t_u com 63 graus de liberdade é aproximadamente igual a 2 [68]. Substituindo-se o valor de t_u e multiplicando-se a expressão do limite superior do intervalo de confiança por 2^7 , obtêm-se:

$$UL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) \times 2^7 + 2 \frac{s_{(k)}(Y, Cb, Cr)}{\sqrt{64}}$$

$$UL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) \times 2^7 + \frac{s_{(k)}(Y, Cb, Cr)}{2^2} \quad (22)$$

Lembrando que:

- a) $s_{(k)}(Y, Cb, Cr)$ é o desvio padrão de cada *pixel* no espaço de cores YCbCr;
- b) $\bar{x}_{i(k)}(Y, Cb, Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr.

O limite inferior do intervalo de confiança para cada *pixel* em cada componente Y,Cb,Cr é obtido da seguinte maneira:

$$LL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) - t_u \frac{s_{(k)}(Y, Cb, Cr)}{\sqrt{N}} \quad (9)$$

O t_u é o percentil com valor u da distribuição t-student com $N - 1$ graus de liberdade. Considerando-se o número de quadros igual a 64, o valor do percentil t_u com 63 graus de liberdade é aproximadamente igual a 2 [68]. Substituindo-se o valor de t_u e multiplicando-se a expressão do limite inferior do intervalo de confiança por 2^7 , obtêm-se:

$$LL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) \times 2^7 - 2 \frac{s_{(k)}(Y, Cb, Cr)}{\sqrt{64}}$$

$$LL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) \times 2^7 - \frac{s_{(k)}(Y, Cb, Cr)}{2^2} \quad (23)$$

Lembrando que:

- a) $s_{(k)}(Y,Cb,Cr)$ é o desvio padrão de cada *pixel* no espaço de cores YCbCr;
- b) $\bar{x}_{i(k)}(Y,Cb,Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr.

5.2.3 Seleção dos limiares

Os limiares de comparação dos *pixels* em cada componente Y,Cb,Cr são determinados da seguinte maneira:

$$\begin{cases} T_{(k)}(Y) = \alpha_Y \times s_{(k)}(Y) \\ T_{(k)}(Cb) = \alpha_{Cb} \times s_{(k)}(Cb) \\ T_{(k)}(Cr) = \alpha_{Cr} \times s_{(k)}(Cr) \end{cases} \quad (10)$$

Os valores dos desvios padrões estão com exatidão de 7 bits fracionários, logo os limiares de comparação também estão com a mesma exatidão.

5.2.4 Compensação de variações do *pixel*

Os valores de compensação das variações dos *pixels* em cada componente Y,Cb,Cr são obtidos da seguinte maneira:

$$\left\{ \begin{array}{l} \text{Se } I_{(k)}(Y, Cb, Cr) > UL_{(k)}(Y, Cb, Cr): \\ PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)|}{I_{(k)}(Y, Cb, Cr)} \\ \text{ou} \\ \text{Se } I_{(k)}(Y, Cb, Cr) < LL_{(k)}(Y, Cb, Cr): \\ PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)|}{\bar{x}_{(k)}(Y, Cb, Cr)} \\ \text{ou} \\ \text{Se } LL_{(k)}(Y, Cb, Cr) \leq I_{(k)}(Y, Cb, Cr) \leq UL_{(k)}(Y, Cb, Cr): \\ PCF_{(k)}(Y, Cb, Cr) = 1 \end{array} \right. \quad (13)$$

Multiplicando-se a expressão anterior por 2^7 , obtêm-se:

$$\left\{ \begin{array}{l}
 \text{Se } I_{(k)}(Y, Cb, Cr) \times 2^7 > UL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)| \times 2^7}{I_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } I_{(k)}(Y, Cb, Cr) \times 2^7 < LL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)| \times 2^7}{\bar{x}_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } LL_{(k)}(Y, Cb, Cr) \times 2^7 \leq I_{(k)}(Y, Cb, Cr) \times 2^7 \leq UL_{(k)}(Y, Cb, Cr) \times 2^7 : \\
 PCF_{(k)}(Y, Cb, Cr) = 1 \times 2^7
 \end{array} \right. \quad (24)$$

Lembrando que:

- $I_{(k)}(Y, Cb, Cr)$ é o valor de cada *pixel* no espaço de cores YCbCr durante a fase de detecção dos objetos;
- $UL_{(k)}(Y, Cb, Cr)$ e $LL_{(k)}(Y, Cb, Cr)$ são os limites superior e inferior do intervalo de confiança, respectivamente, de cada *pixel* no espaço de cores YCbCr.

Os valores dos limites inferiores das funções de compensação das variações dos *pixels* em cada componente Y, Cb, Cr são calculados por:

$$\left\{ \begin{array}{l}
 \text{Se } PCF_{(k)}(Y) < \alpha_{lo(y)} : \\
 PCF_{(k)}(Y) = \alpha_{lo(y)/2} \\
 e \\
 \text{Se } PCF_{(k)}(Cb, Cr) < \alpha_{lo(Cb, Cr)} : \\
 PCF_{(k)}(Cb, Cr) = \alpha_{lo(Cb, Cr)}
 \end{array} \right. \quad (15)$$

Multiplicando-se os limites inferiores por 2^7 , obtêm-se:

$$\left\{ \begin{array}{l}
 \text{Se } PCF_{(k)}(Y) < \alpha_{lo(y)} \times 2^7 : \\
 PCF_{(k)}(Y) = \alpha_{lo(y)/2} \times 2^7 \\
 e \\
 \text{Se } PCF_{(k)}(Cb, Cr) < \alpha_{lo(Cb, Cr)} \times 2^7 : \\
 PCF_{(k)}(Cb, Cr) = \alpha_{lo(Cb, Cr)} \times 2^7
 \end{array} \right. \quad (25)$$

Lembrando que, para as imagens de testes utilizadas neste trabalho, as constantes $\alpha_{lo(y)}$ e $\alpha_{lo(Cb,Cr)}$ foram escolhidas empiricamente com valores iguais a 0,4 e 0,6, respectivamente.

5.2.5 Classificação do *pixel*

Para classificar cada *pixel* como objeto ou pertencente a imagem de referência, a diferença absoluta (BD) de cada *pixel* é multiplicada pelo valor da função de compensação das variações do *pixel* (PCF) e o resultado é comparado com os valores dos limiares. As equações utilizadas para classificação de cada *pixel* estão reescritas a seguir:

$$F_k(Y, Cb, Cr) = BD_{(k)}(Y, Cb, Cr) \times PCF_{(k)}(Y, Cb, Cr) \quad (11)$$

$$Máscara \left\{ \begin{array}{l} F, F_{(k)}(Y) > T_{(k)}(Y) \\ ou \\ F_{(k)}(Cb) > T_{(k)}(Cb) \\ ou \\ F_{(k)}(Cr) > T_{(k)}(Cr) \\ B, Se nenhuma das anteriores \end{array} \right. \quad (12)$$

Considerando-se que o desvio padrão e a função de compensação do *pixel* foram multiplicados anteriormente por 2^7 , os valores dos limiares ($T_k(Y, Cb, Cr)$) estão com uma exatidão de 7 bits fracionários assim como os das diferenças absolutas compensadas ($F_k(Y, Cb, Cr)$).

5.2.6 Detecção de sombras

Um *pixel* é classificado como sombra de acordo com a seguinte equação:

$$\left\{ \begin{array}{l} I_{(k)}(Y) > LL_{(k)}(Y) \times \alpha_{lo} \\ e \\ I_{(k)}(Y) < LL_{(k)}(Y) \times \alpha_{hi} \end{array} \right. \quad (16)$$

Multiplicando-se a expressão anterior por 2^7 , obtêm-se:

$$\left\{ \begin{array}{l} I_{(k)}(Y) \times 2^{14} > LL_{(k)}(Y) \times \alpha_{lo} \times 2^7 \\ e \\ I_{(k)}(Y) \times 2^{14} < LL_{(k)}(Y) \times \alpha_{hi} \times 2^7 \end{array} \right. \quad (26)$$

Lembrando que:

- a) $I_{(k)}(Y,Cb,Cr)$ é o valor de cada *pixel* da componente Y durante a fase de detecção dos objetos;
- b) $LL_{(k)}(Y)$ é o limite inferior do intervalo de confiança de cada *pixel* da componente Y;
- c) α_{lo} e α_{hi} são as constantes que determinam a fração dos valores médios dos *pixels* na imagem de referência que é esperado nas regiões sombreadas.

5.2.7 Comparações entre ponto fixo e flutuante

Nesta seção são apresentados os resultados da avaliação do método de detecção de objetos, com operações em ponto fixo e flutuante, utilizando-se dois tipos de classificações dos *pixels* denominadas Classificação do *Pixel* 01, e Classificação do *Pixel* 03. A Classificação do *Pixel* 01 utiliza a função de compensação de variação do *pixel* e a detecção de sombras durante a classificação dos *pixels*. A Classificação do *Pixel* 03 utiliza somente a função de compensação de variação do *pixel* durante a detecção de objetos.

O objetivo desta avaliação é verificar se há alguma diferença no desempenho do método de detecção de objetos quando as operações são realizadas em ponto fixo.

5.2.7.1 Sequência de imagens 01

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 01 e dois tipos de classificação dos *pixels* com operações em ponto fixo e flutuante estão ilustrados na Figura 57. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F não são estatisticamente significantes ($p = 0,103$). Isto significa que neste cenário as diferenças observadas entre os resultados obtidos com operações em ponto fixo e ponto flutuante não refletem uma verdadeira diferença na população.

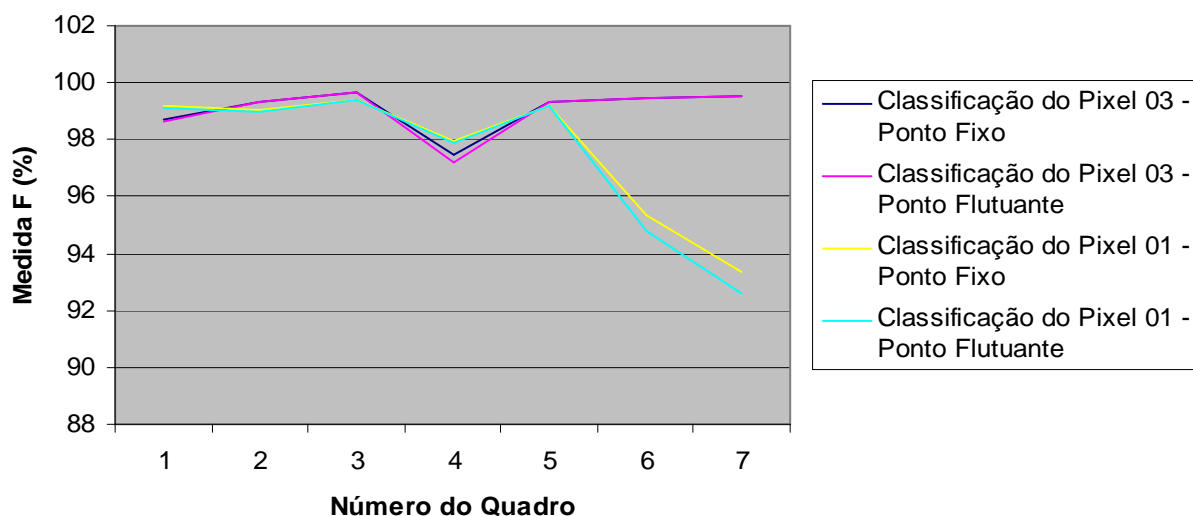


Figura 57: Medida F para seqüência de imagens 01 obtidas utilizando-se o método proposto com operações em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p = 0,103$); Teste de igualdade de variância ($p = 0,0118$). Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 11: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 01 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante.

Classificação do <i>Pixel</i> (CP)	p
CP03 – Ponto Fixo vs. CP03 – Ponto Flutuante	0,675
CP03 – Ponto Fixo vs. CP01 – Ponto Fixo	0,016
CP03 – Ponto Fixo vs. CP01 – Ponto Flutuante	0,0023
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	0,0403
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	0,0063
CP01 – Ponto Fixo vs. CP01 – Ponto Flutuante	0,4182

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os dois tipos de classificação dos *pixels* e operações realizadas em ponto fixo e flutuante, foi rejeitada ($p = 0,0118$). Baseado nas

comparações dos pares das medidas F, ilustradas na Tabela 11, não há uma diferença significativa entre as medidas F em relação à variância nos seguintes casos: Classificação do *Pixel* 03 – Ponto Fixo versus Classificação do *Pixel* 03 – Ponto Flutuante e Classificação do *Pixel* 01 – Ponto Fixo versus Classificação do *Pixel* 01 – Ponto Flutuante. Na Figura 58 está ilustrado o gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação e operação realizada para detecção dos *pixels* do objeto. De acordo com o gráfico, a utilização da Classificação do *Pixel* 03, em ponto fixo ou ponto flutuante, resultou na maior mediana com a menor variância. Uma menor variância na medida F significa que a detecção é menos dependente dos objetos inseridos na cena.

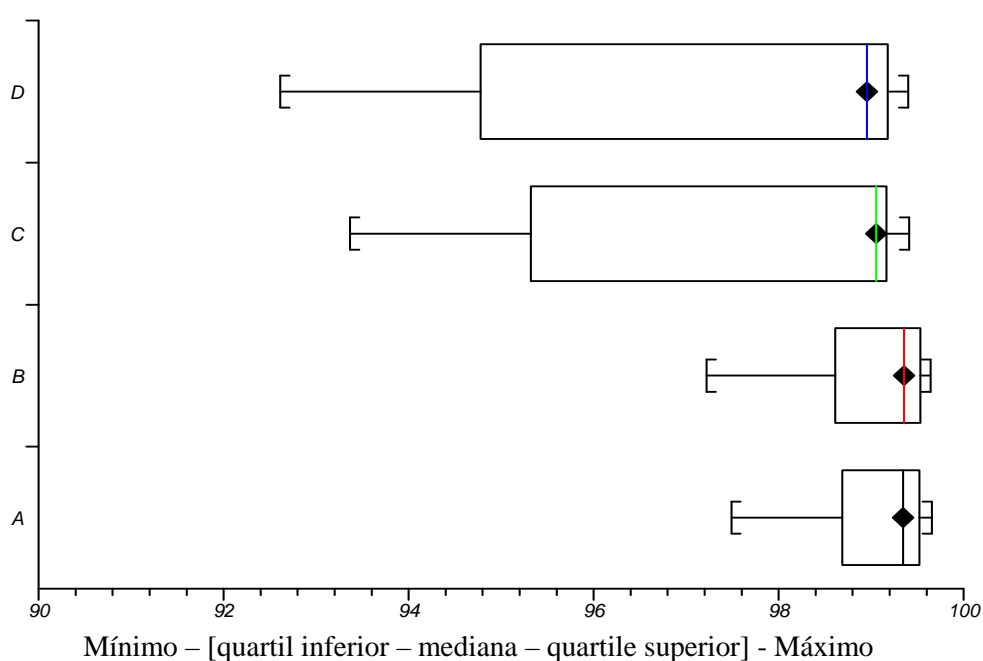


Figura 58: Valores das medianas das medidas F obtidas para a seqüência de imagens 01 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante. A) Classificação do *Pixel* 03 – Ponto Fixo; B) Classificação do *Pixel* 03 – Ponto Flutuante; C) Classificação do *Pixel* 01 – Ponto Fixo; D) Classificação do *Pixel* 01 – Ponto Flutuante;

5.2.7.2 Sequência de imagens 02

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 02 e dois tipos de classificação dos *pixels* com operações em ponto fixo e flutuante estão ilustrados na Figura 59. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F não são estatisticamente significantes ($p = 0,1865$). Isto significa que neste cenário as diferenças observadas entre os resultados obtidos com operações em ponto fixo e ponto flutuante não refletem uma verdadeira diferença na população.

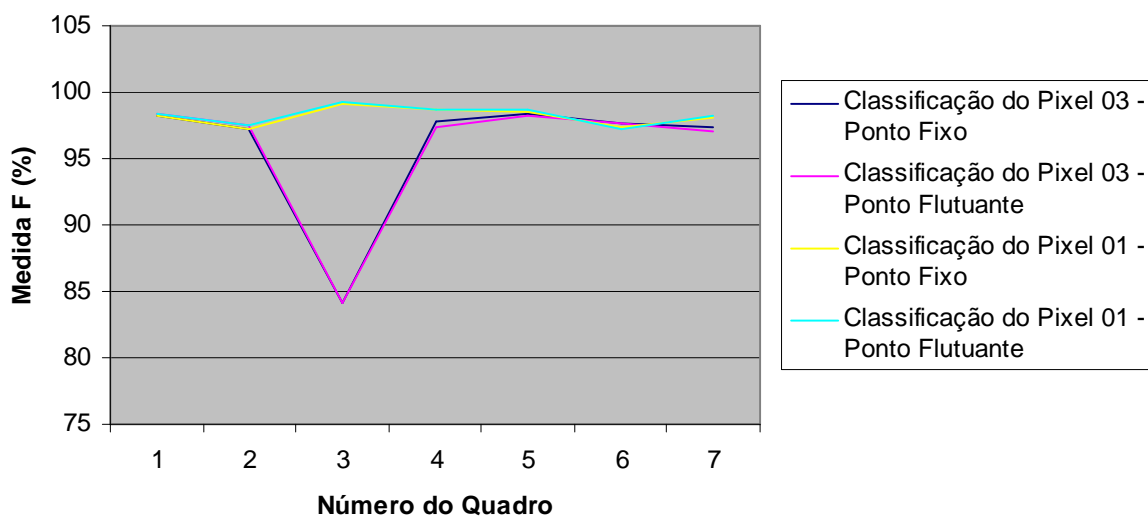


Figura 59: Medida F para seqüência de imagens 02 obtidas utilizando-se o método proposto em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p = 0,1865$); Teste de igualdade de variância ($p = 0,0003$). Classificação do *Pixel 01* – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel 03* – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 12: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 02 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante.

Classificação do <i>Pixel</i> (CP)	p
CP03 – Ponto Fixo vs. CP03 – Ponto Flutuante	0,7971
CP03 – Ponto Fixo vs. CP01 – Ponto Fixo	< 0,0001
CP03 – Ponto Fixo vs. CP01 – Ponto Flutuante	< 0,0001
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	< 0,0001
CP03 – Ponto Flutuante vs. CP01 – Ponto Fixo	< 0,0001
CP01 – Ponto Fixo vs. CP01 – Ponto Flutuante	0,9476

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os dois tipos de classificação dos *pixels*

e operações realizadas em ponto fixo e flutuante, foi rejeitada ($p = 0,0118$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 12, não há uma diferença significativa entre as medidas F em relação à variância nos seguintes casos: Classificação do *Pixel* 03 – Ponto Fixo versus Classificação do *Pixel* 03 – Ponto Flutuante e Classificação do *Pixel* 01 – Ponto Fixo versus Classificação do *Pixel* 01 – Ponto Flutuante. Na Figura 60 está ilustrado o gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação e operação realizada para detecção dos *pixels* do objeto. De acordo com o gráfico, a utilização da Classificação do *Pixel* 01, em ponto fixo ou ponto flutuante, resultou na maior mediana com uma das menores variâncias.

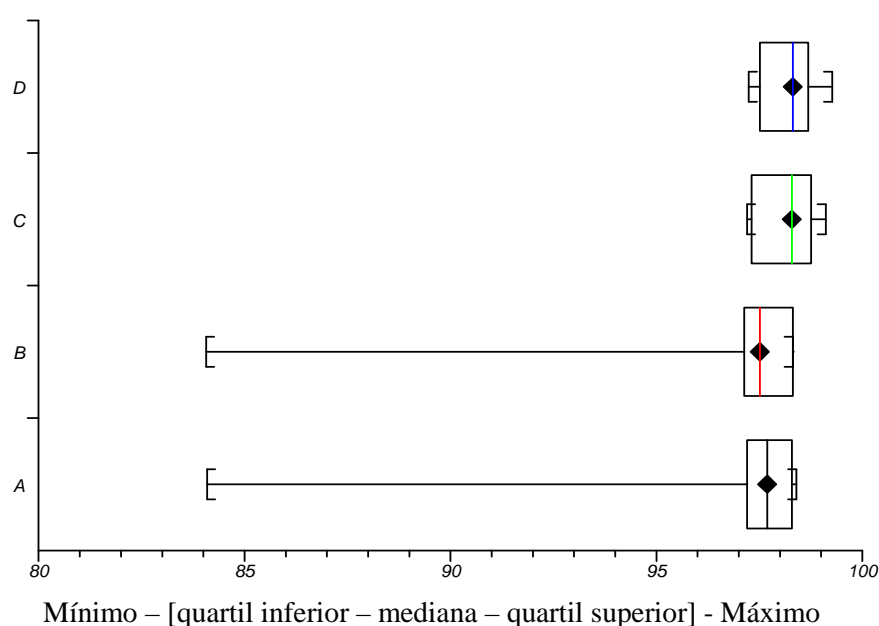


Figura 60: Valores das medianas das medidas F obtidas para a seqüência de imagens 02 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante. A) Classificação do *Pixel* 03 – Ponto Fixo; B) Classificação do *Pixel* 03 – Ponto Flutuante; C) Classificação do *Pixel* 01 – Ponto Fixo; D) Classificação do *Pixel* 01 – Ponto Flutuante;

5.2.7.3 Sequência de imagens 03

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 03 e dois tipos de classificação dos *pixels* com operações em ponto fixo e flutuante estão ilustrados na Figura 61. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p = 0,0136$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 13, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 03 realizada com operações em ponto fixo e flutuante. No entanto, há uma diferença significativa

entre as medidas F obtidas com a Classificação do *Pixel* 01 realizada com operações em ponto fixo e flutuante.

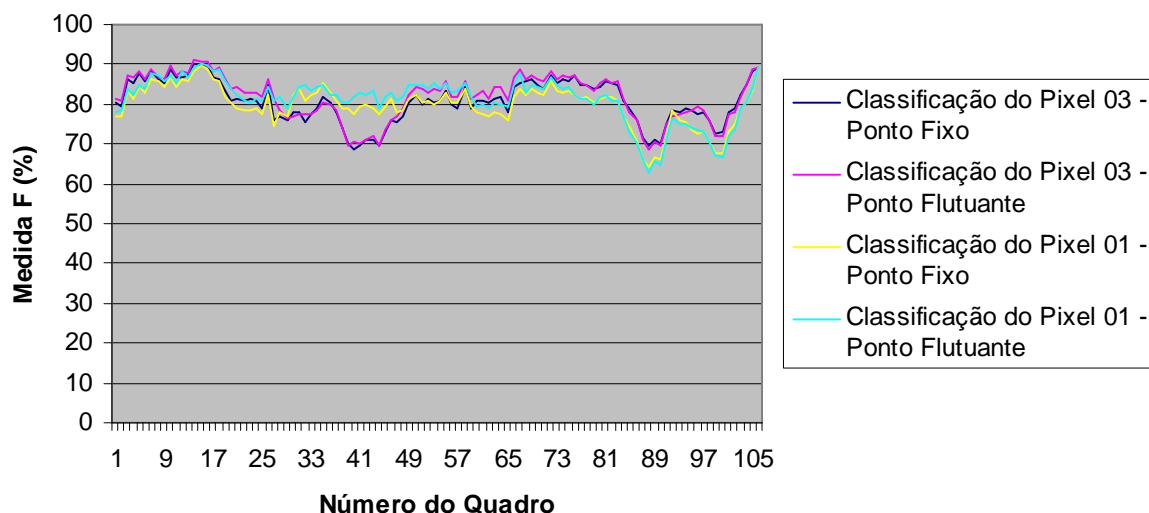


Figura 61: Medida F para sequência de imagens 03 obtidas utilizando-se o método proposto em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p = 0,0136$); Teste de igualdade de variância ($p = 0,056$). Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 13: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a sequência de imagens 03 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante.

Classificação do <i>Pixel</i> (CP)	p
CP03 – Ponto Fixo vs. CP03 – Ponto Flutuante	0,4758
CP03 – Ponto Fixo vs. CP01 – Ponto Fixo	0,5608
CP03 – Ponto Fixo vs. CP01 – Ponto Flutuante	0,7476
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	0,0206
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	0,8669
CP01 – Ponto Fixo vs. CP01 – Ponto Flutuante	0,0264

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os dois tipos de classificação dos *pixels* e operações realizadas em ponto fixo e flutuante, foi aceita ($p = 0,056$). Na Figura 62 está ilustrado o gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação e operação realizada para detecção dos *pixels* do objeto. De acordo com o gráfico, a utilização da Classificação do *Pixel* 01, em ponto fixo, resultou em uma das maiores medianas com uma das menores variâncias.

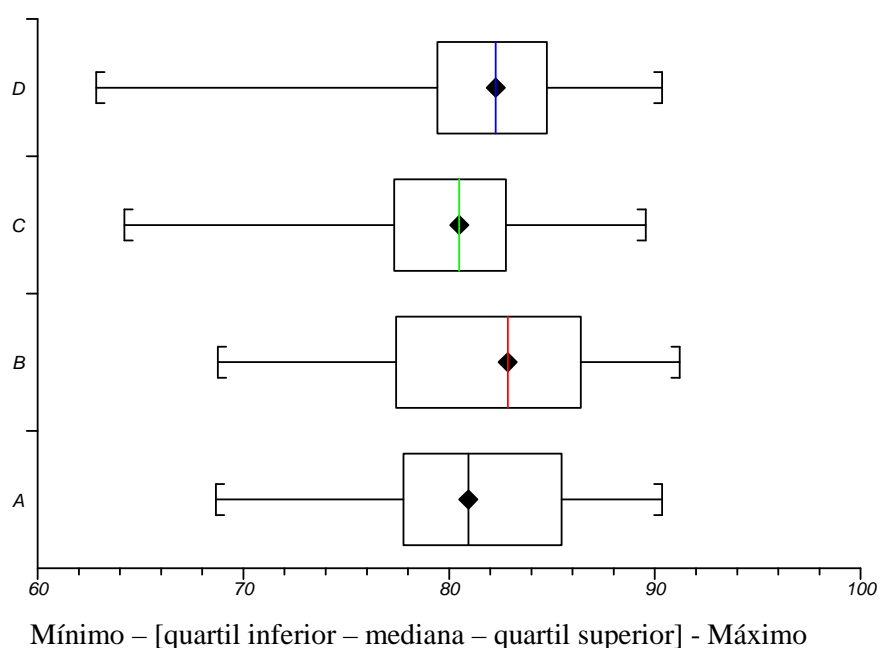


Figura 62: Valores das medianas das medidas F obtidas para a seqüência de imagens 03 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante. A) Classificação do *Pixel* 03 – Ponto Fixo; B) Classificação do *Pixel* 03 – Ponto Flutuante; C) Classificação do *Pixel* 01 – Ponto Fixo; D) Classificação do *Pixel* 01 – Ponto Flutuante;

5.2.7.4 Sequência de imagens 04

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 04 e dois tipos de classificação dos *pixels* com operações em ponto fixo e flutuante estão ilustrados na Figura 63. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 14, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 01 realizada com operações em ponto fixo e flutuante. No entanto, há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 03 realizada com operações em ponto fixo e flutuante.

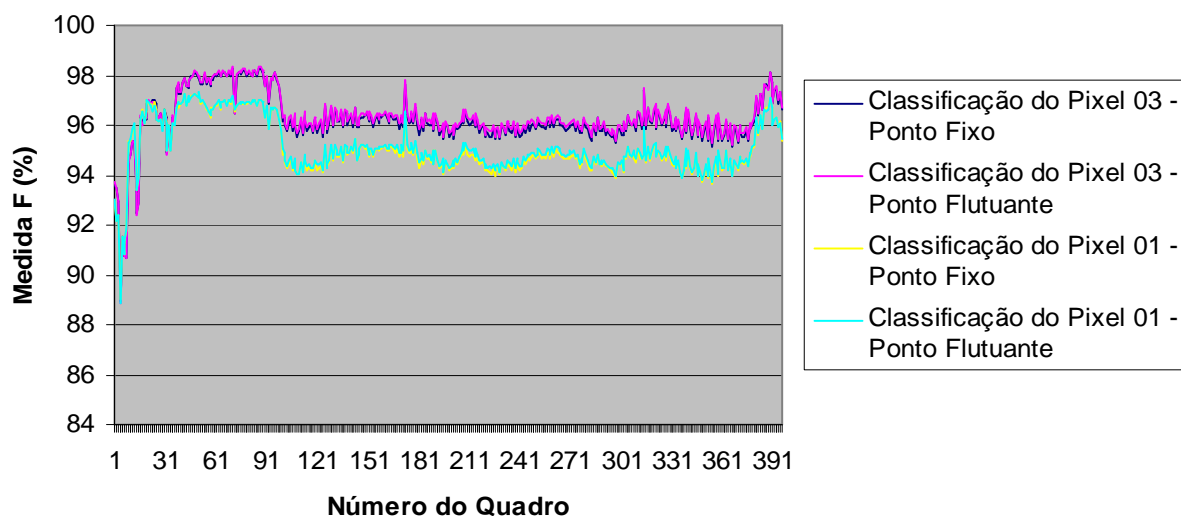


Figura 63: Medida F para seqüência de imagens 04 obtidas utilizando-se o método proposto em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$). Classificação do *Pixel 01* – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel 03* – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 14: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a seqüência de imagens 04 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante.

Classificação do <i>Pixel</i> (CP)	p^*	p^{**}
CP03 – Ponto Fixo vs. CP03 – Ponto Flutuante	0,0353	0,5182
CP03 – Ponto Fixo vs. CP01 – Ponto Fixo	< 0,0001	< 0,0001
CP03 – Ponto Fixo vs. CP01 – Ponto Flutuante	< 0,0001	0,0004
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	< 0,0001	< 0,0001
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	< 0,0001	< 0,0001
CP01 – Ponto Fixo vs. CP01 – Ponto Flutuante	0,3087	0,4718

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os dois tipos de classificação dos *pixels* e operações realizadas em ponto fixo e flutuante, foi rejeitada ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 14, não há uma diferença significativa entre as medidas F em relação à variância nos seguintes casos: Classificação do *Pixel* 01 – Ponto Fixo versus Classificação do *Pixel* 01 – Ponto Flutuante e Classificação do *Pixel* 03 – Ponto Fixo versus Classificação do *Pixel* 03 – Ponto Flutuante. Na Figura 64 está ilustrado o gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação e operação realizada para detecção dos *pixels* do objeto. De acordo com o gráfico, a utilização da Classificação do *Pixel* 03, em ponto fixo, resultou em uma das maiores medianas com uma das menores variâncias.

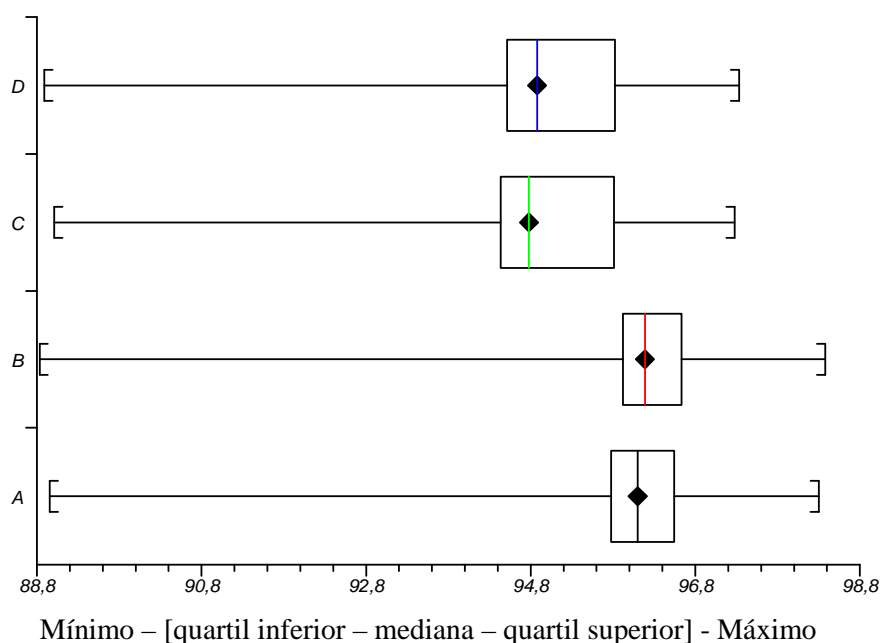


Figura 64: Valores das medianas das medidas F obtidas para a seqüência de imagens 04 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante. A) Classificação do *Pixel* 03 – Ponto Fixo; B) Classificação do *Pixel* 03 – Ponto Flutuante; C) Classificação do *Pixel* 01 – Ponto Fixo; D) Classificação do *Pixel* 01 – Ponto Flutuante.

5.2.7.5 Sequência de imagens 05

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 05 e dois tipos de classificação dos *pixels* com operações em ponto fixo e flutuante estão ilustrados na Figura 65. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 15, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 03

realizada com operações em ponto fixo e flutuante. No entanto, há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 01 realizada com operações em ponto fixo e flutuante.

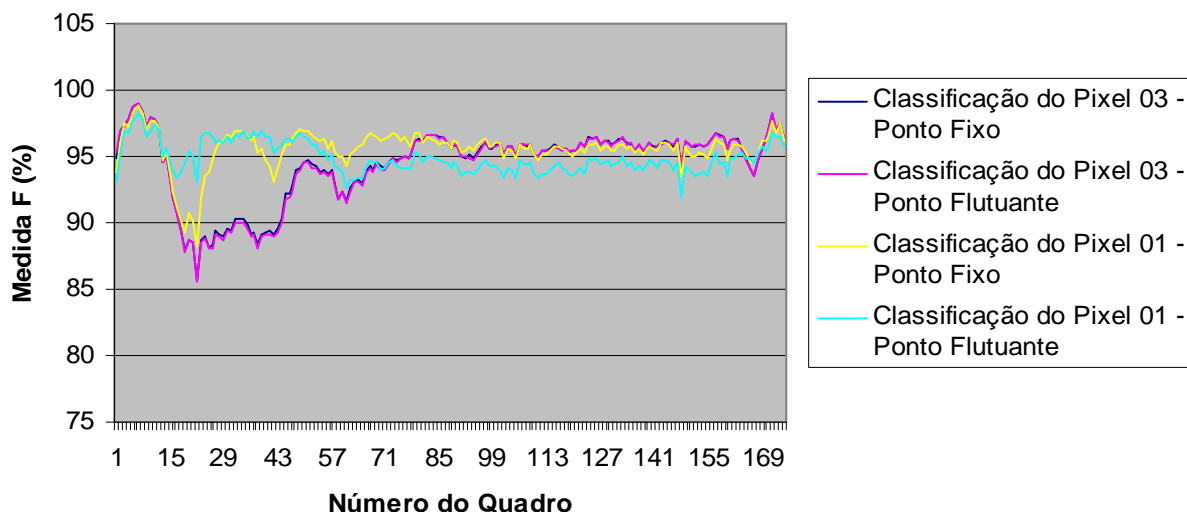


Figura 65: Medida F para sequência de imagens 05 obtidas utilizando-se o método proposto em ponto flutuante e em ponto fixo. Teste de Kruskal-Wallis ($p < 0,0001$); Teste de igualdade de variância ($p < 0,0001$). Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

Tabela 15: Comparação dos pares de parâmetros para o teste de igualdade de variância das medidas F para a sequência de imagens 05 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante.

Classificação do <i>Pixel</i> (CP)	p^*	p^{**}
CP03 – Ponto Fixo vs. CP03 – Ponto Flutuante	0,9948	0,5722
CP03 – Ponto Fixo vs. CP01 – Ponto Fixo	0,0007	< 0,0001
CP03 – Ponto Fixo vs. CP01 – Ponto Flutuante	0,509	< 0,0001
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	0,0004	< 0,0001
CP03 – Ponto Flutuante vs. CP01 – Ponto Flutuante	0,6854	< 0,0001
CP01 – Ponto Fixo vs. CP01 – Ponto Flutuante	< 0,0001	0,1256

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com os dois tipos de classificação dos *pixels* e operações realizadas em ponto fixo e flutuante, foi rejeitada ($p < 0,0001$). Baseado nas comparações dos pares das medidas F, ilustradas na Tabela 15, não há uma diferença significativa entre as medidas F em relação à variância nos seguintes casos: Classificação do *Pixel 01* – Ponto Fixo versus Classificação do *Pixel 01* – Ponto Flutuante e Classificação do *Pixel 03* – Ponto Fixo versus Classificação do *Pixel 03* – Ponto Flutuante. Na Figura 66 está ilustrado o gráfico com os valores das medianas obtidos das medidas F para cada tipo de classificação e operação realizada para detecção dos *pixels* do objeto. De acordo com o gráfico, a utilização da Classificação do *Pixel 03*, em ponto fixo, resultou em uma das maiores medianas com uma das menores variâncias.

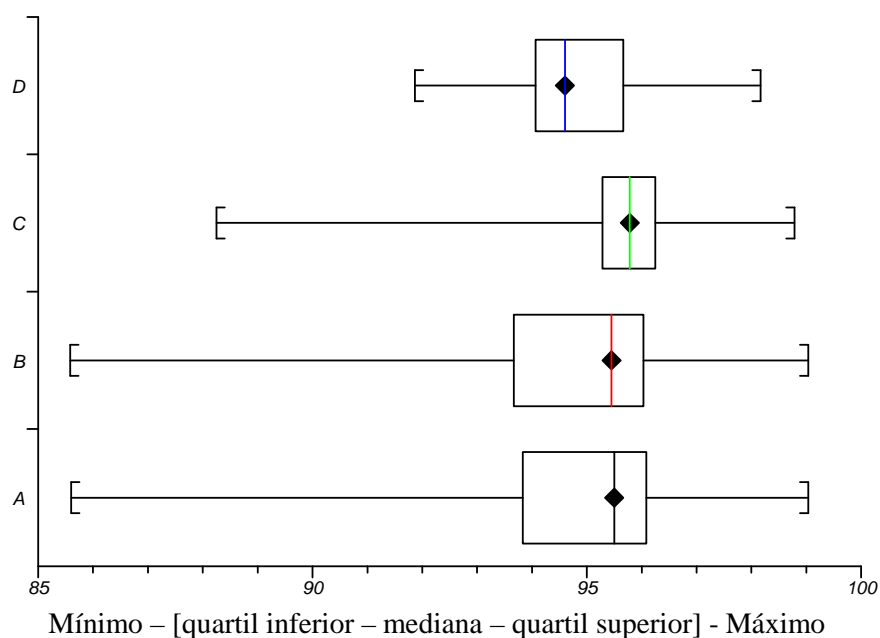


Figura 66: Valores das medianas das medidas F obtidas para a seqüência de imagens 05 com dois tipos de classificação dos *pixels* e operações em ponto fixo e flutuante. A) Classificação do *Pixel 03* – Ponto Fixo; B) Classificação do *Pixel 03* – Ponto Flutuante; C) Classificação do *Pixel 01* – Ponto Fixo; D) Classificação do *Pixel 01* – Ponto Flutuante.

5.3 COMPARAÇÃO ENTRE OPERAÇÕES EM PONTO FIXO

Na seção anterior as operações em ponto fixo foram realizadas mantendo-se a exatidão de 7 bits fracionários em todas as equações utilizadas no método de detecção de objetos implementado neste trabalho. Outra abordagem que pode ser efetuada é manter as operações em ponto fixo sem a exatidão dos bits fracionários, isto é, com 8 bits conforme mostrado a seguir.

5.3.1 Valor médio

A média aritmética dos *pixels* em cada componente Y,Cb,Cr é dada por:

$$\bar{x}_{i(k)}(Y, Cb, Cr) = \frac{1}{N} \sum_{i=1}^N I_{i(k)}(Y, Cb, Cr) \quad (3)$$

Substituindo-se o valor do número de quadros (N) na expressão anterior, obtêm-se:

$$\bar{x}_{i(k)}(Y, Cb, Cr) = \frac{1}{2^6} \sum_{i=1}^{64} I_{i(k)}(Y, Cb, Cr) \quad (27)$$

Lembrando que:

- a) i é o número do quadro em processamento;
- b) k é a coordenada (x,y) de cada *pixel*;
- c) $I_{i(k)}(Y,Cb,Cr)$ é o valor corrente de cada *pixel* no espaço de cores YCbCr;

5.3.2 Desvio padrão

O desvio padrão dos *pixels* em cada componente Y,Cb,Cr é obtido por:

$$S_k(Y, Cb, Cr) = \sqrt{\frac{1}{N} \sum_{i=1}^N [I_{i(k)}(Y, Cb, Cr) - \bar{x}_{i(k)}(Y, Cb, Cr)]^2} \quad (4)$$

Substituindo-se o valor do número de quadros (N), obtêm-se:

$$S_k(Y, Cb, Cr) = \sqrt{\frac{1}{2^6} \sum_{i=1}^{64} [I_{i(k)}(Y, Cb, Cr) - \bar{x}_{i(k)}(Y, Cb, Cr)]^2} \quad (28)$$

Lembrando que:

- a) $I_{i(k)}(Y,Cb,Cr)$ é o valor corrente de cada *pixel* no espaço de cores YCbCr;
- b) $\bar{x}_{i(k)}(Y,Cb,Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr.

5.3.3 Intervalo de confiança

O limite superior do intervalo de confiança para cada *pixel* em cada componente Y,Cb,Cr é calculado da seguinte maneira:

$$UL_{(k)}(Y,Cb,Cr) = \bar{x}_{(k)}(Y,Cb,Cr) + t_u \frac{s_{(k)}(Y,Cb,Cr)}{\sqrt{N}} \quad (8)$$

O t_u é o percentil com valor u da distribuição t-student com $N - 1$ graus de liberdade. Considerando-se o número de quadros igual a 64, o valor do percentil t_u com 63 graus de liberdade é aproximadamente igual a 2 [68]. Substituindo-se o valor de t_u na expressão anterior, obtêm-se:

$$UL_{(k)}(Y,Cb,Cr) = \bar{x}_{(k)}(Y,Cb,Cr) + 2 \frac{s_{(k)}(Y,Cb,Cr)}{\sqrt{64}}$$

$$UL_{(k)}(Y,Cb,Cr) = \bar{x}_{(k)}(Y,Cb,Cr) + \frac{s_{(k)}(Y,Cb,Cr)}{2^2} \quad (29)$$

Lembrando que:

- a) $s_{(k)}(Y,Cb,Cr)$ é o desvio padrão de cada *pixel* no espaço de cores YCbCr;
- b) $\bar{x}_{i(k)}(Y,Cb,Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr.

O limite inferior do intervalo de confiança para cada *pixel* em cada componente Y,Cb,Cr é determinado por:

$$LL_{(k)}(Y,Cb,Cr) = \bar{x}_{(k)}(Y,Cb,Cr) - t_u \frac{s_{(k)}(Y,Cb,Cr)}{\sqrt{N}} \quad (9)$$

O t_u é o percentil com valor u da distribuição t-student com $N - 1$ graus de liberdade. Considerando-se o número de quadros igual a 64, o valor do percentil t_u com 63 graus de liberdade é aproximadamente igual a 2 [68]. Substituindo-se o valor de t_u na expressão anterior, obtêm-se:

$$LL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) - 2 \frac{s_{(k)}(Y, Cb, Cr)}{\sqrt{64}}$$

$$LL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) - \frac{s_{(k)}(Y, Cb, Cr)}{2^2} \quad (30)$$

Lembrando que:

- a) $s_{(k)}(Y, Cb, Cr)$ é o desvio padrão de cada *pixel* no espaço de cores YCbCr;
- b) $\bar{x}_{i(k)}(Y, Cb, Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr.

5.3.4 Seleção dos limiares

Os limiares de comparação dos *pixels* em cada componente Y, Cb, Cr são determinados da seguinte maneira:

$$\begin{cases} T_{(k)}(Y) = \alpha_Y \times s_{(k)}(Y) \\ T_{(k)}(Cb) = \alpha_{Cb} \times s_{(k)}(Cb) \\ T_{(k)}(Cr) = \alpha_{Cr} \times s_{(k)}(Cr) \end{cases} \quad (10)$$

Lembrando que:

- a) $s_{(k)}(Y, Cb, Cr)$ é o desvio padrão de cada *pixel* no espaço de cores YCbCr;
- b) $\alpha(Y, Cb, Cr)$ são as constantes utilizadas para determinar os valores dos limiares de comparação.

5.3.5 Compensação de variações do *pixel*

Os valores de compensação das variações dos *pixels* em cada componente Y, Cb, Cr são calculados da seguinte maneira:

$$\left\{ \begin{array}{l}
 \text{Se } I_{(k)}(Y, Cb, Cr) > UL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)|}{I_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } I_{(k)}(Y, Cb, Cr) < LL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)|}{\bar{x}_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } LL_{(k)}(Y, Cb, Cr) \leq I_{(k)}(Y, Cb, Cr) \leq UL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = 1
 \end{array} \right. \quad (13)$$

Os resultados da PCF estão compreendidos entre 0 e 1. Portanto, deve-se multiplicar a expressão anterior por 2^7 para obtenção de números inteiros não negativos. Desta maneira, a expressão resulta em:

$$\left\{ \begin{array}{l}
 \text{Se } I_{(k)}(Y, Cb, Cr) > UL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)| \times 2^7}{I_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } I_{(k)}(Y, Cb, Cr) < LL_{(k)}(Y, Cb, Cr) : \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)| \times 2^7}{\bar{x}_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } LL_{(k)}(Y, Cb, Cr) \leq I_{(k)}(Y, Cb, Cr) \leq UL_{(k)}(Y, Cb, Cr) \times 2^7 : \\
 PCF_{(k)}(Y, Cb, Cr) = 1 \times 2^7
 \end{array} \right. \quad (31)$$

Lembrando que:

- a) $I_{(k)}(Y, Cb, Cr)$ é o valor de cada *pixel* no espaço de cores YCbCr durante a fase de detecção dos objetos;
- b) $UL_{(k)}(Y, Cb, Cr)$ e $LL_{(k)}(Y, Cb, Cr)$ são os limites superior e inferior do intervalo de confiança, respectivamente, de cada *pixel* no espaço de cores YCbCr.

Os valores dos limites inferiores das funções de compensação das variações dos *pixels* em cada componente Y, Cb, Cr são calculados por:

$$\left\{ \begin{array}{l} \text{Se } PCF_{(k)}(Y) < \alpha_{lo(y)} : \\ PCF_{(k)}(Y) = \alpha_{lo(y)/2} \\ e \\ \text{Se } PCF_{(k)}(Cb, Cr) < \alpha_{lo(Cb, Cr)} : \\ PCF_{(k)}(Cb, Cr) = \alpha_{lo(Cb, Cr)} \end{array} \right. \quad (15)$$

Lembrando que, para as imagens de testes utilizadas neste trabalho, as constantes $\alpha_{lo(y)}$ e $\alpha_{lo(Cb, Cr)}$ foram escolhidas empiricamente com valores iguais a 0,4 e 0,6, respectivamente. Portanto, estas constantes devem ser multiplicadas por 2^7 para obtenção de números inteiros não negativos. Desta maneira, obtêm-se:

$$\left\{ \begin{array}{l} \text{Se } PCF_{(k)}(Y) < \alpha_{lo(y)} \times 2^7 : \\ PCF_{(k)}(Y) = \alpha_{lo(y)/2} \times 2^7 \\ e \\ \text{Se } PCF_{(k)}(Cb, Cr) < \alpha_{lo(Cb, Cr)} \times 2^7 : \\ PCF_{(k)}(Cb, Cr) = \alpha_{lo(Cb, Cr)} \times 2^7 \end{array} \right. \quad (32)$$

5.3.6 Classificação do *pixel*

Para classificar cada *pixel* como objeto ou pertencente a imagem de referência, a diferença absoluta (BD) de cada *pixel* é multiplicada pelo valor da função de compensação das variações do *pixel* (PCF) e o resultado é comparado com os valores dos limiares. As equações utilizadas para classificação de cada *pixel* estão reescritas a seguir:

$$F_k(Y, Cb, Cr) = BD_{(k)}(Y, Cb, Cr) \times PCF_{(k)}(Y, Cb, Cr) \quad (11)$$

$$\text{Máscara} \left\{ \begin{array}{l} F, F_{(k)}(Y) > T_{(k)}(Y) \\ \text{ou} \\ F_{(k)}(Cb) > T_{(k)}(Cb) \\ \text{ou} \\ F_{(k)}(Cr) > T_{(k)}(Cr) \\ B, \text{ Se nenhuma das anteriores} \end{array} \right. \quad (12)$$

Os valores de $PCF_k(Y,Cb,Cr)$ estão com uma exatidão de 7 bits fracionários, pois foram multiplicados por 2^7 . Portanto deve-se dividir as diferenças absolutas compensadas ($F_k(Y,Cb,Cr)$) por 128 para manter a operação com 8 bits. Assim a equação para calcular as diferenças absolutas resulta em:

$$F_k(Y, Cb, Cr) = \frac{BD_{(k)}(Y, Cb, Cr) \times PCF_{(k)}(Y, Cb, Cr)}{2^7} \quad (33)$$

5.3.7 Detecção de sombras

Um *pixel* é classificado como sombra de acordo com a seguinte equação:

$$\begin{cases} I_{(k)}(Y) > LL_{(k)}(Y) \times \alpha_{lo} \\ e \\ I_{(k)}(Y) < LL_{(k)}(Y) \times \alpha_{hi} \end{cases} \quad (16)$$

Lembrando que:

- $I_{(k)}(Y,Cb,Cr)$ é o valor de cada *pixel* da componente Y durante a fase de detecção dos objetos;
- $LL_{(k)}(Y)$ é o limite inferior do intervalo de confiança de cada *pixel* da componente Y;
- α_{lo} e α_{hi} são as constantes que determinam a fração dos valores médios dos *pixels* na imagem de referência que é esperado nas regiões sombreadas.

De acordo com os testes realizados durante o desenvolvimento deste trabalho, os valores de α_{lo} e α_{hi} estão na faixa de 0,5 a 0,7. Portanto, estas constantes devem ser multiplicadas por 2^7 para obtenção de números inteiros não negativos. Desta maneira, obtêm-se:

$$\begin{cases} I_{(k)}(Y) \times 2^7 > LL_{(k)}(Y) \times \alpha_{lo} \times 2^7 \\ e \\ I_{(k)}(Y) \times 2^7 < LL_{(k)}(Y) \times \alpha_{hi} \times 2^7 \end{cases} \quad (34)$$

5.4 AVALIAÇÕES DAS OPERAÇÕES EM PONTOS FIXOS

A seguir são apresentados os resultados das comparações entre as operações em pontos fixos proposta nesta seção e na seção anterior. Foram utilizados dois tipos de classificações dos *pixels* denominados Classificação do *Pixel* 01, e Classificação do *Pixel* 03. A Classificação do *Pixel* 01 utiliza a função de compensação de variação do *pixel* e a detecção de sombras durante a classificação dos *pixels*. A Classificação do *Pixel* 03 utiliza somente a função de compensação de variação do *pixel* durante a detecção de objetos. As operações em ponto fixo propostas nesta seção são denominadas ponto fixo 02 e as operações em ponto fixo, da seção anterior, são denominadas ponto fixo 01.

5.4.1 Sequência de Imagens 01

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 01 e dois tipos de classificação dos *pixels* com operações em pontos fixos estão ilustrados na Figura 67. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F não são estatisticamente significantes ($p = 0,09$).

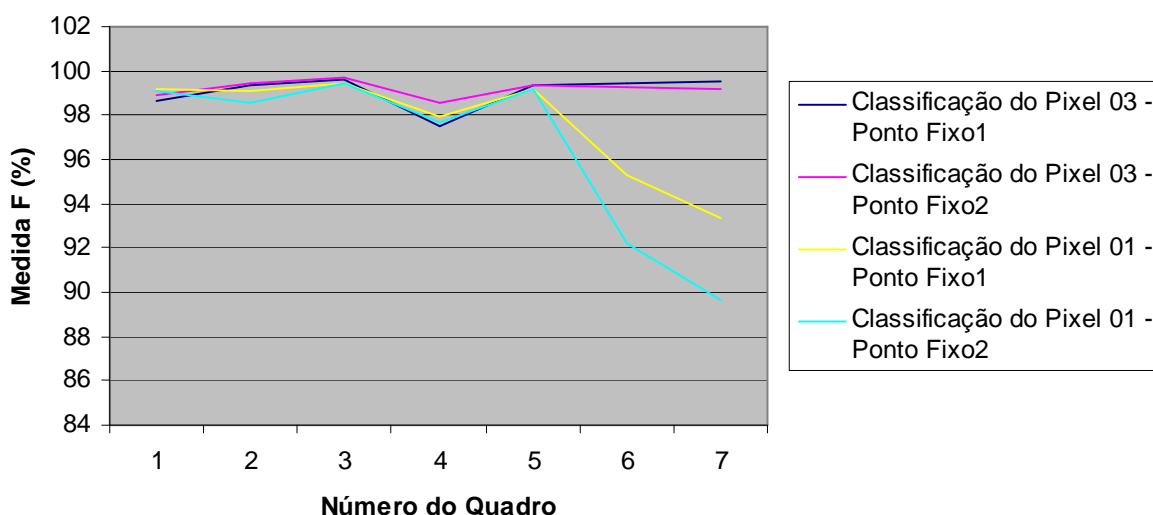


Figura 67: Medida F para seqüência de imagens 01 obtidas utilizando-se o método proposto em ponto fixo. Teste de Kruskal-Wallis ($p = 0,09$); Teste de igualdade de variância ($p = 0,0118$). Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

5.4.2 Sequência de Imagens 02

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 02 e dois tipos de classificação dos *pixels* com operações em pontos fixos estão ilustrados na Figura 68. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F não são estatisticamente significantes ($p = 0,5$).

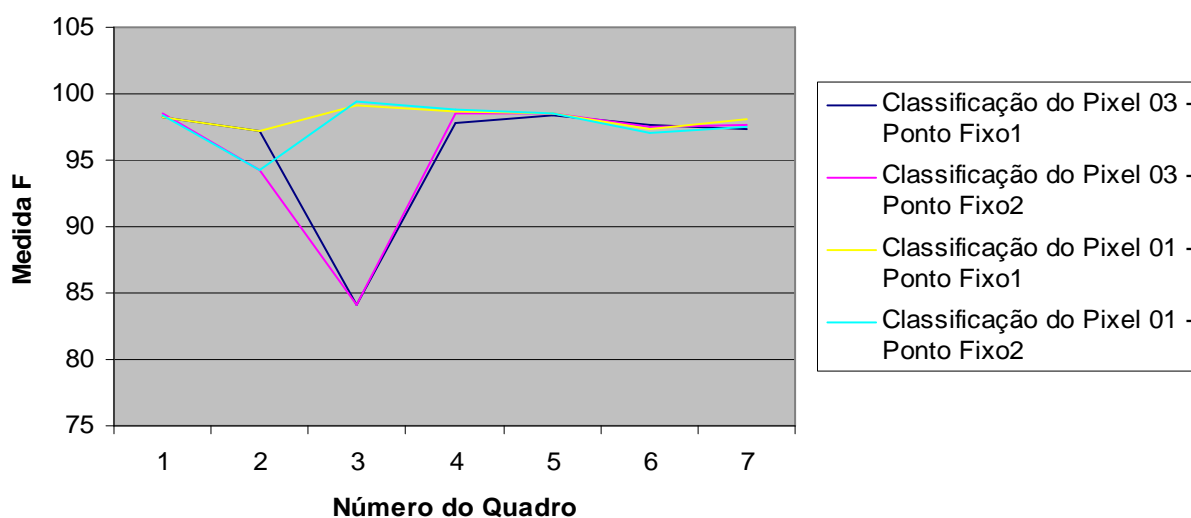


Figura 68: Medida F para seqüência de imagens 01 obtidas utilizando-se o método proposto em ponto em ponto fixo. Teste de Kruskal-Wallis ($p = 0,5$). Classificação do *Pixel 01* – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel 03* – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

5.4.3 Sequência de Imagens 03

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 03 e dois tipos de classificação dos *pixels* com operações em pontos fixos estão ilustrados na Figura 69. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F não são estatisticamente significantes ($p = 0,18$).

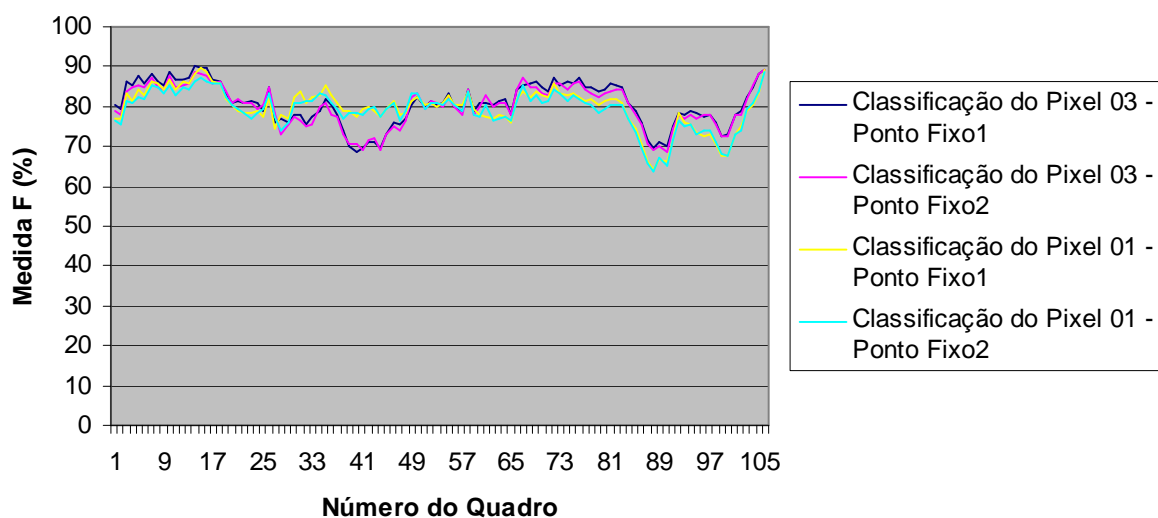


Figura 69: Medida F para sequência de imagens 01 obtidas utilizando-se o método proposto em ponto em ponto fixo. Teste de Kruskal-Wallis ($p = 0,18$). Classificação do *Pixel* 01 – a função de compensação de variação do *pixel* e detecção de sombras são utilizadas durante a detecção de objetos; classificação do *pixel* 03 – somente a função de compensação de variação do *pixel* é usada durante a detecção de objetos.

5.5 REDUÇÃO DA MEMÓRIA PARA IMAGEM DE REFERÊNCIA

Estratégias de simplificação tendo sido utilizadas para adaptar algoritmos visando implementações em tempo real. Entre estas estratégias pode-se citar a redução no número de operações e a redução na quantidade de dados a ser processada [80, 87]. O método proposto no capítulo anterior realiza a segmentação de vídeo baseada em valores médios e em desvios padrões calculados para cada *pixel* durante a fase de aprendizado. Estes parâmetros devem ser armazenados em uma memória para que sejam utilizados durante a fase de classificação e detecção dos objetos em cena.

Uma maneira de calcular os valores médios de cada *pixel* é acumular os resultados parciais em uma memória e ao final do processamento de todos os quadros, dividir os valores armazenados na memória pelo número de quadros processados para obtenção dos valores médios [89]. Esta operação deve ser realizada para cada componente do espaço de cores. O tamanho da memória (TMA), para acumular todos os valores, pode ser calculado da seguinte maneira:

$$TMA = \frac{(RESH \times RESV \times NP \times NC \times NBA)}{NB} \quad (35)$$

Em que:

- a) RESH é a resolução horizontal da imagem;
- b) RESV é a resolução vertical da imagem;
- c) NP é o número de parâmetros a ser armazenado;
- d) NC é o número de componentes do espaço de cores;
- e) NBA é o número de bits do valor acumulado;
- f) NB é o número de bits do valor final.

Para reduzir a quantidade de acesso à memória durante a fase de segmentação, os valores médios devem ser organizados na memória de modo que cada posição tenha os valores de todas as componentes de cores para cada *pixel*. Considerando-se o número de quadros igual a 16 e 8 bits para representação de cada componente de cor no espaço RGB, segundo as especificações do decodificador de vídeo, os valores acumulados poderiam atingir um valor máximo de 4080. Neste caso, seriam necessários 12 bits para acumular os valores de cada *pixel* para cada componente do espaço RGB. Supondo-se uma imagem com resolução igual a 640 x 507 *pixels*, o tamanho da memória para acumular todos os valores deveria ser igual a 1.460.160 bytes ($640 \times 507 \times 3 \times 12 / 8$).

Para reduzir o tamanho da memória utilizada para armazenar os valores médios parciais, propõe-se o armazenamento na memória dos valores parciais referentes à média entre dois quadros consecutivos. Desta maneira, quando o segundo quadro for processado, são calculados os valores médios entre o quadro anterior armazenado na memória e o atual. O resultado é armazenado na memória e utilizado para calcular os valores médios com o terceiro quadro e assim sucessivamente até que todos os quadros sejam processados. Neste caso, o valor máximo de cada *pixel* seria igual a 255 (8 bits). Procedendo-se desta maneira, o tamanho da memória é reduzido de 1.460.160 bytes para 973.440 bytes ($640 \times 507 \times 3 \times 8 / 8$), isto é, uma redução de 33%.

Outra simplificação proposta para calcular os valores médios é reduzir a resolução horizontal da imagem de 640 para 320 *pixels* somente para obtenção dos valores médios de cada *pixel*. Assim sendo, além da média temporal também seria realizada uma média espacial entre cada dois *pixels* consecutivos em cada linha. Desta maneira, o tamanho da memória para armazenamento dos valores médios dos *pixels* reduz de 973.440 bytes para 486.720 bytes.

Além das simplificações para calcular os valores médios, outra modificação proposta é a utilização de um valor global do desvio padrão para cada componente do espaço de cores. Estes valores podem ser calculados com a equação 36.

$$s(Y, Cb, Cr) = \sqrt{\frac{1}{NXY} \sum_{i=1}^N \sum_{x=1}^{XY} (I_{i(k)}(Y, Cb, Cr) - \bar{x}_{i(k)}(Y, Cb, Cr))^2} \quad (36)$$

Sendo que:

- a) $I_{i(k)}(Y, Cb, Cr)$ é o valor corrente de cada *pixel* no espaço de cores YCbCr;
- b) $\bar{x}_{i(k)}(Y, Cb, Cr)$ é o valor médio de cada *pixel* no espaço de cores YCbCr;
- c) N é o número de quadros processados.
- d) X e Y são os valores das resoluções horizontal e vertical da imagem, respectivamente.

Utilizando-se um valor global para o desvio padrão, reduz-se o tamanho da memória necessário para armazenar os valores dos desvios padrões da imagem de referência de 973440 para 3 bytes, isto é, um byte para cada componente YCbCr.

Para verificar o efeito destas simplificações na detecção dos objetos, o método proposto foi novamente testado com as seqüências de imagens ilustradas na Figura 24 e os resultados das medidas F foram comparados com os previamente obtidos sem as simplificações com operações em ponto fixo.

5.5.1 Avaliação da redução de memória

5.5.1.1 Seqüência de imagens 01

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 01 e as simplificações propostas nesta seção estão ilustrados na Figura 70. Média 01 e DP 01 são os valores médios e os desvios padrões calculados com as Equações 03 e 04, respectivamente; Média 02 e DP 02 são os valores médios e os desvios padrões calculados de acordo com as simplificações apresentadas nesta seção. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F não são estatisticamente significantes ($p = 0,9945$). Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com e sem as simplificações, foi aceita ($p = 0,9848$).

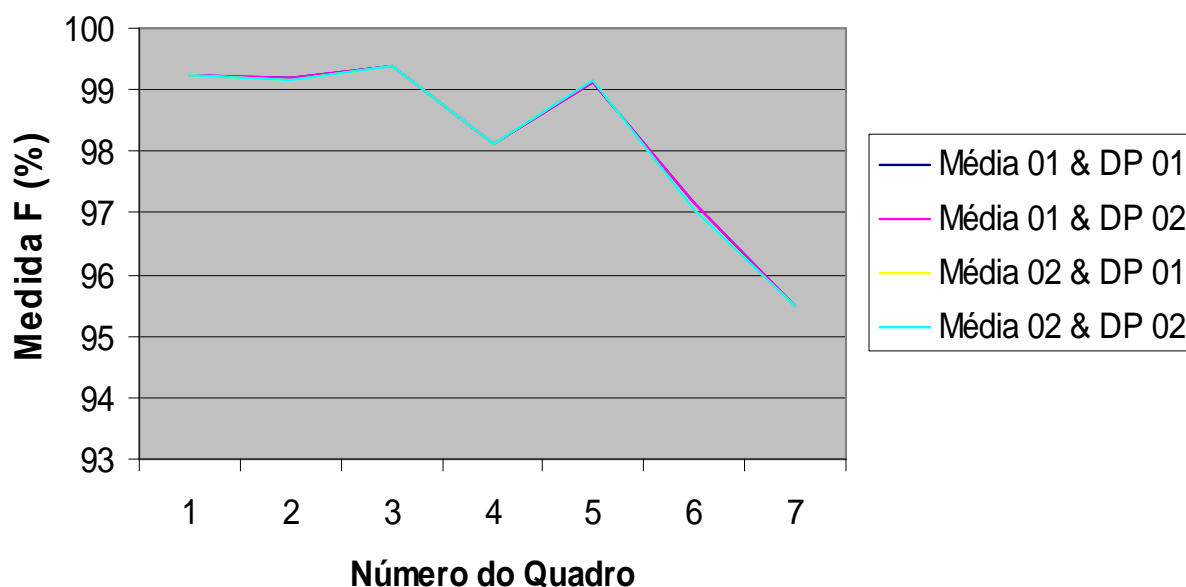


Figura 70: Medida F para a seqüência de imagens 01 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p = 0,9945$); Teste de igualdade de variância ($p = 0,9848$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.

5.5.1.2 Seqüência de imagens 02

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 02 e as simplificações propostas nesta seção estão ilustrados na Figura 71. Média 01 e DP 01 são os valores médios e os desvios padrões calculados com as Equações 03 e 04, respectivamente; Média 02 e DP 02 são os valores médios e os desvios padrões calculados de acordo com as simplificações apresentadas nesta seção. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F não são estatisticamente significantes ($p = 0,7626$). Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com e sem as simplificações, foi aceita ($p = 0,7842$).

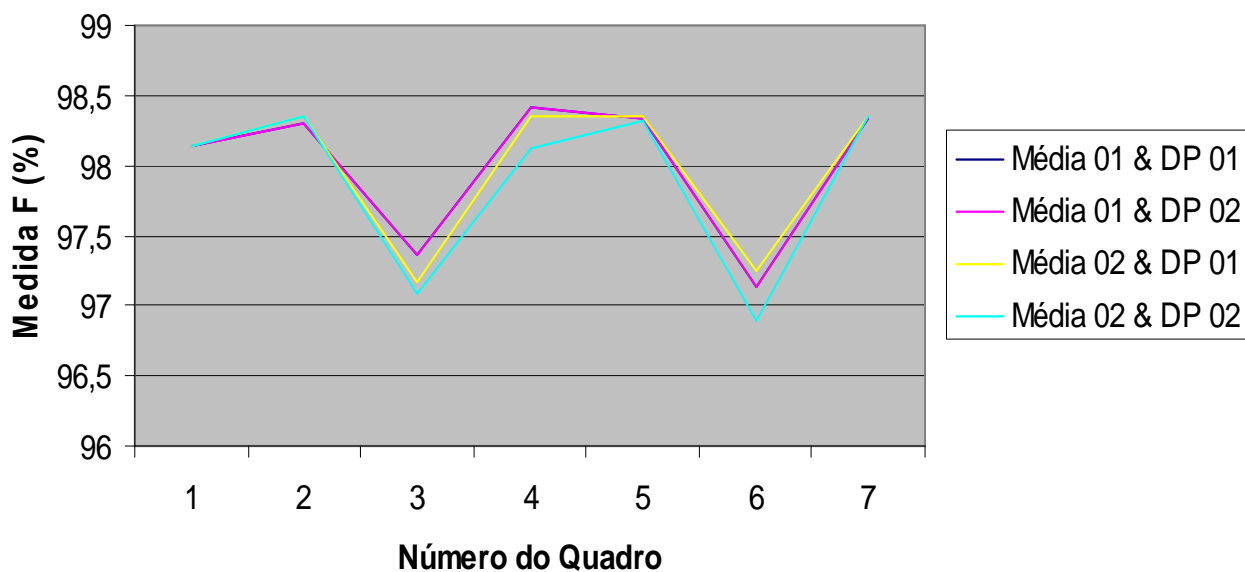


Figura 71: Medida F para a seqüência de imagens 02 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p = 0,7626$); Teste de igualdade de variância ($p = 0,7842$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.

5.5.1.3 Seqüência de imagens 03

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 03 e as simplificações propostas nesta seção estão ilustrados na Figura 72. Média 01 e DP 01 são os valores médios e os desvios padrões calculados com as Equações 03 e 04, respectivamente; Média 02 e DP 02 são os valores médios e os desvios padrões calculados de acordo com as simplificações apresentadas nesta seção. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$).

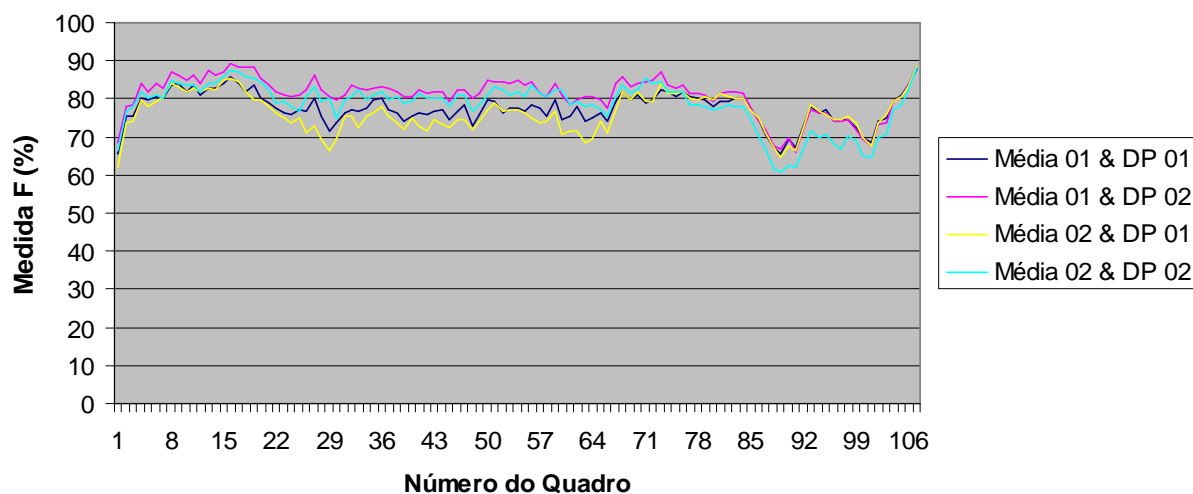


Figura 72: Medida F para a seqüência de imagens 03 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,0493$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.

Baseado nas comparações dos pares de parâmetros, ilustradas na Tabela 16, há uma diferença significativa entre todas as medidas F, exceto entre aquelas calculadas com Média 01 e DP01 versus Média 02 e DP 01.

Tabela 16: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 03 com diferentes métodos para calcular os valores da média e do desvio padrão.

Parameters	p^*	p^{**}
Média 01 & DP 01 versus Média 01 & DP 02	< 0,0001	0,3419
Média 01 & DP 01 versus Média 02 & DP 01	0,1427	0,0239
Média 01 & DP 01 versus Média 02 & DP 02	0,0139	0,0156
Média 01 & DP 02 versus Média 02 & DP 01	< 0,0001	0,1893
Média 01 & DP 02 versus Média 02 & DP 02	0,0015	0,1404
Média 02 & DP 01 versus Média 02 & DP 02	0,0008	0,8711

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com e sem as simplificações, foi rejeitada ($p = 0,0493$). Na Figura 73 está ilustrado o gráfico com os valores das medianas obtidos para as medidas F para cada par de valores da média e do desvio padrão. De acordo com o gráfico, a combinação da Média 01 com DP 02 resultou na maior mediana com a menor variação. O segundo melhor resultado foi obtido utilizando-se a Média 02 com o DP 02.

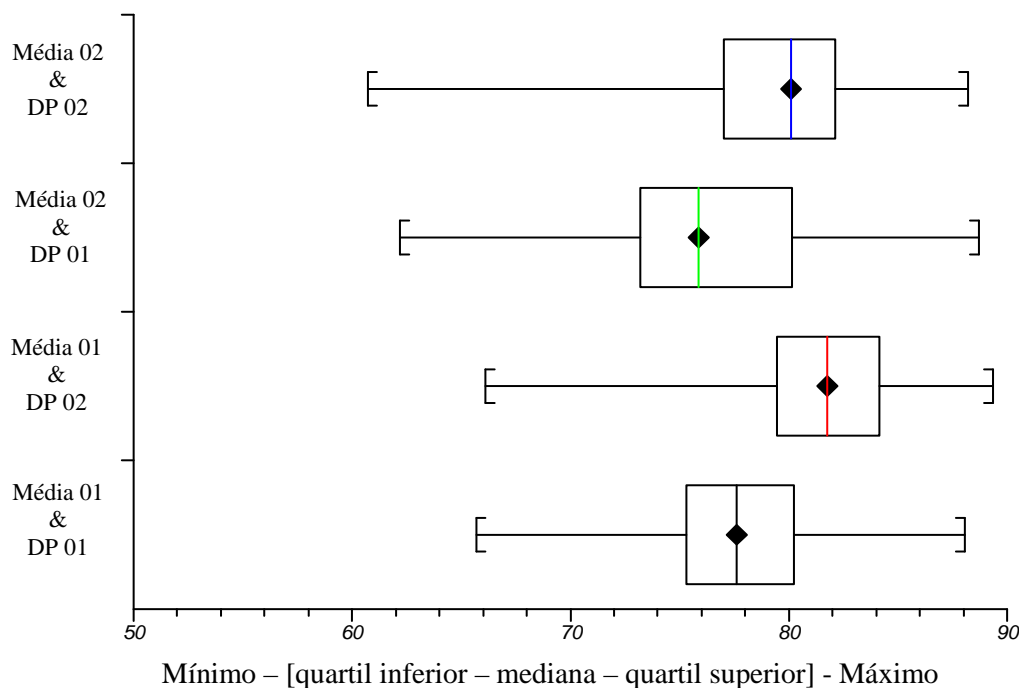


Figura 73: Valores das medianas das medidas F para a seqüência de imagens 03 com diferentes métodos para calcular os valores da média e do desvio padrão. Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.

5.5.1.4 Seqüência de imagens 04

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 04 e as simplificações propostas nesta seção estão ilustrados na Figura 74. Média 01 e DP 01 são os valores médios e os desvios padrões calculados com as Equações 03 e 04, respectivamente; Média 02 e DP 02 são os valores médios e os desvios padrões calculados de acordo com as simplificações apresentadas nesta seção. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$).

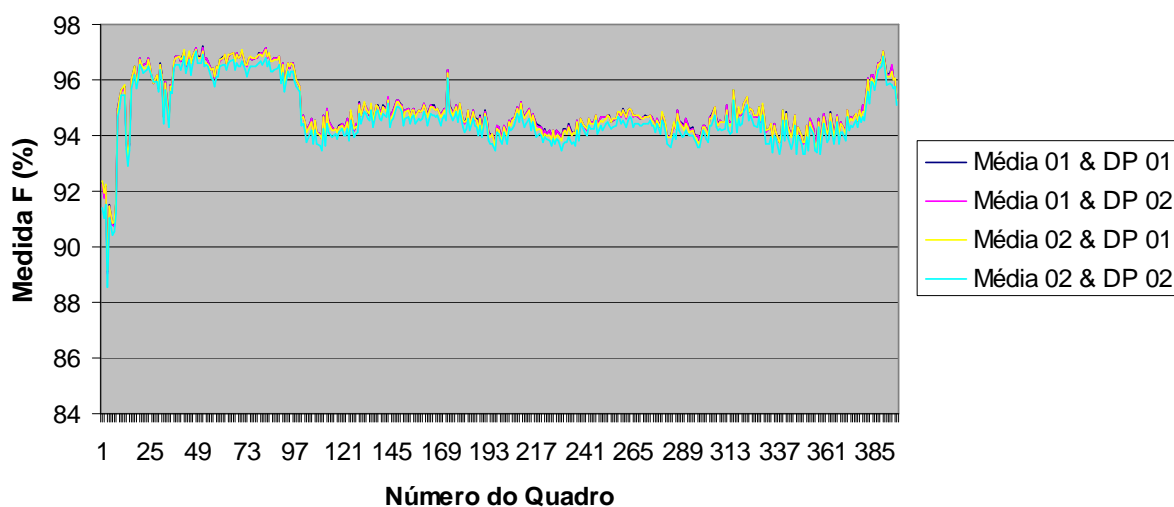


Figura 74: Medida F para a seqüência de imagens 04 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,9688$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.

Baseado nas comparações dos pares de parâmetros, ilustradas na Tabela 17, há uma diferença significativa entre todas as medidas F, exceto entre aquelas calculadas com Média 01 e DP 01 versus Média 02 e DP 01, assim como Média 01 e DP 01 versus Média 01 e DP 02.

Tabela 17: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 04 com diferentes métodos para calcular os valores da média e do desvio padrão.

Parameters	p^*
Média 01 & DP 01 versus Média 01 & DP 02	0,9959
Média 01 & DP 01 versus Média 02 & DP 01	0,7578
Média 01 & DP 01 versus Média 02 & DP 02	< 0,0001
Média 01 & DP 02 versus Média 02 & DP 01	0,861
Média 01 & DP 02 versus Média 02 & DP 02	< 0,0001
Média 02 & DP 01 versus Média 02 & DP 02	< 0,0001

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com e sem as simplificações, foi aceita ($p = 0,9688$). Na Figura 75 está ilustrado o gráfico em caixa com os valores da mediana obtidos para as medidas F para cada par de valores da média e do desvio padrão. De acordo com o gráfico, a combinação da Média 02 com DP 02 resultou em uma mediana maior que 94%. Este valor é inferior ao obtido sem as modificações propostas, porém ainda pode ser considerado um bom resultado.

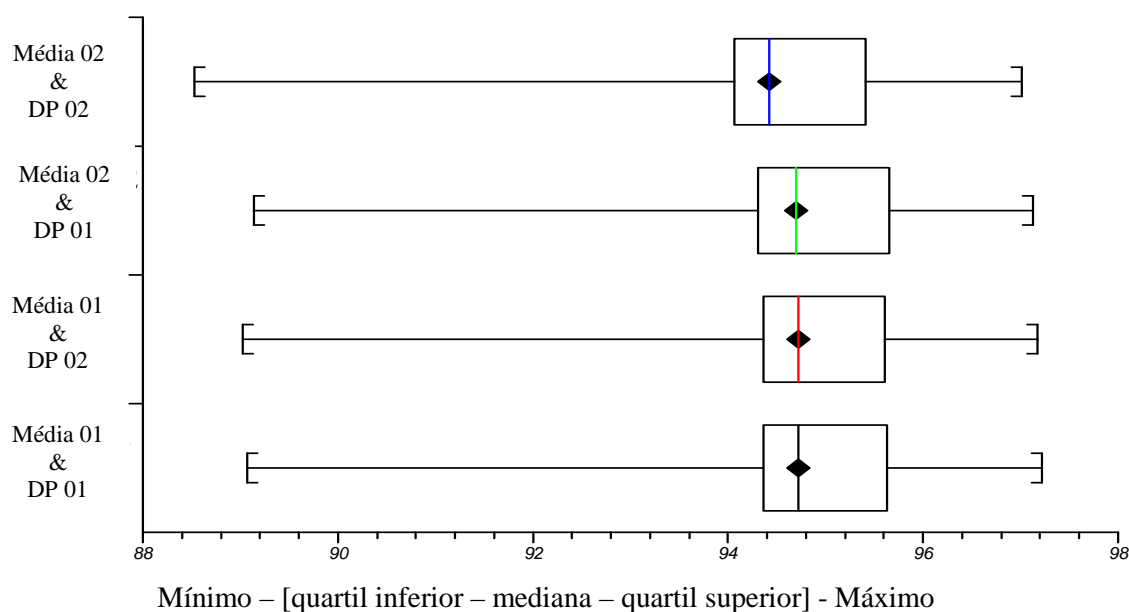


Figura 75: Valores das medianas das medidas F para a seqüência de imagens 04 com diferentes métodos para calcular os valores da média e do desvio padrão. Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.

5.5.1.5 Seqüência de imagens 05

Os resultados da medida F utilizando o método proposto com a seqüência de imagens 05 e as simplificações propostas nesta seção estão ilustrados na Figura 76. Média 01 e DP 01 são os valores médios e os desvios padrões calculados com as Equações 03 e 04, respectivamente; Média 02 e DP 02 são os valores médios e os desvios padrões calculados de acordo com as simplificações apresentadas nesta seção. De acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F são estatisticamente significantes ($p < 0,0001$).

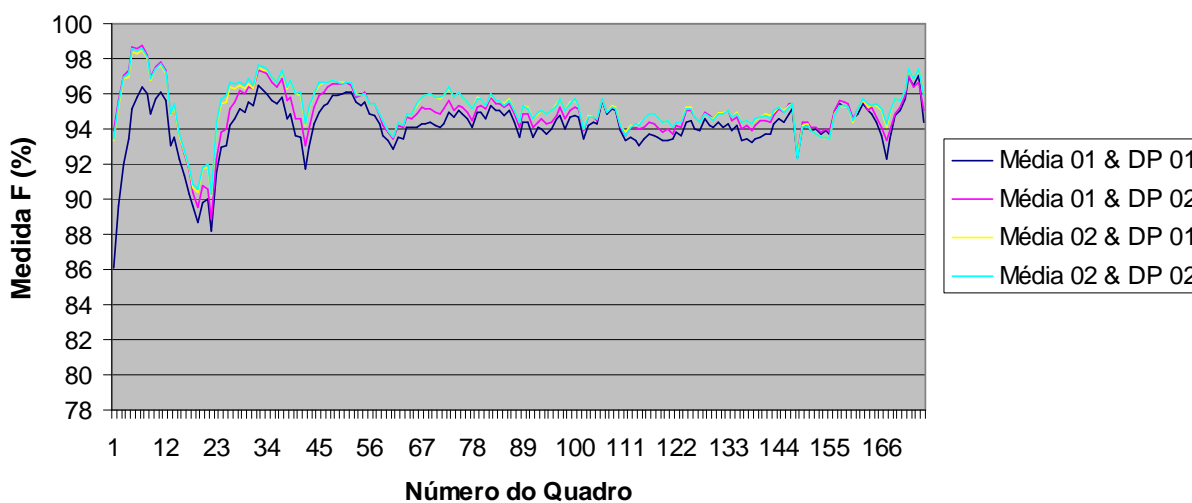


Figura 76: Medida F para a seqüência de imagens 05 com diferentes métodos para calcular os valores da média e do desvio padrão. Análise de variância Kruskal Wallis ($p < 0,0001$); Teste de igualdade de variância ($p = 0,9676$). Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.

Baseado nas comparações dos pares de parâmetros, ilustradas na Tabela 18, há uma diferença significativa entre todas as medidas F, exceto entre aquelas calculadas com Média 02 e DP01 versus Média 02 e DP 02.

Tabela 18: Comparação dos pares de parâmetros para os testes de Kruskal Wallis (p^*) e igualdade de variância (p^{**}) das medidas F para a seqüência de imagens 05 com diferentes métodos para calcular os valores da média e do desvio padrão.

Parameters	p^*
Média 01 & DP 01 versus Média 01 & DP 02	< 0,0001
Média 01 & DP 01 versus Média 02 & DP 01	< 0,0001
Média 01 & DP 01 versus Média 02 & DP 02	< 0,0001
Média 01 & DP 02 versus Média 02 & DP 01	0,0347
Média 01 & DP 02 versus Média 02 & DP 02	0,0289
Média 02 & DP 01 versus Média 02 & DP 02	0,9996

Baseado no teste de igualdade de variância, a hipótese nula de que não há diferenças significativas entre as variâncias das medidas F, com e sem as simplificações, foi aceita ($p = 0,9676$). Na Figura 77 está ilustrado o gráfico com os valores das medianas obtidos para as medidas F para cada par de valores da média e do desvio padrão. De acordo com o gráfico, a combinação da Média 02 com DP 02 resultou em uma das maiores medianas.

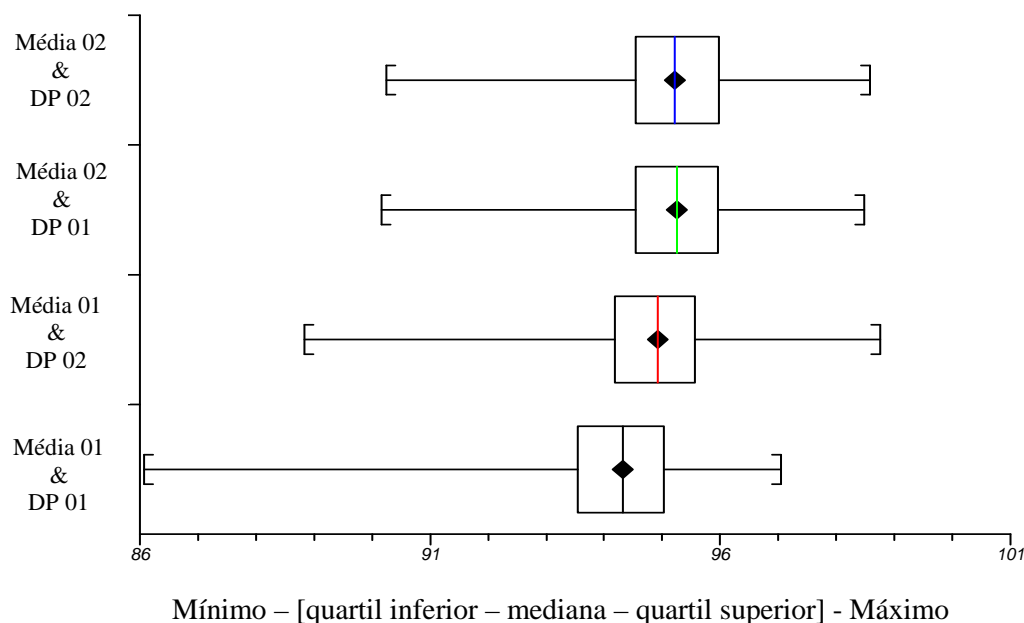


Figura 77: Valores das medianas das medidas F para a seqüência de imagens 04 com diferentes métodos para calcular os valores da média e do desvio padrão. Média 01 e DP 01 – Média e desvio padrão calculados com as equações 03 e 04, respectivamente; Média 02 e DP 02 – Média e desvio padrão calculados de acordo com as simplificações apresentadas nesta seção.

Na Figura 78 estão ilustrados os resultados obtidos com as modificações proposta nesta seção para o cálculo da média e do desvio padrão. Os pontos em torno dos objetos são os falsos positivos detectados durante a etapa de classificação dos *pixels*. De acordo com os resultados, a segmentação ainda apresenta bons resultados após as modificações efetuadas para reduzir a quantidade de memória para armazenar a imagem de referência.

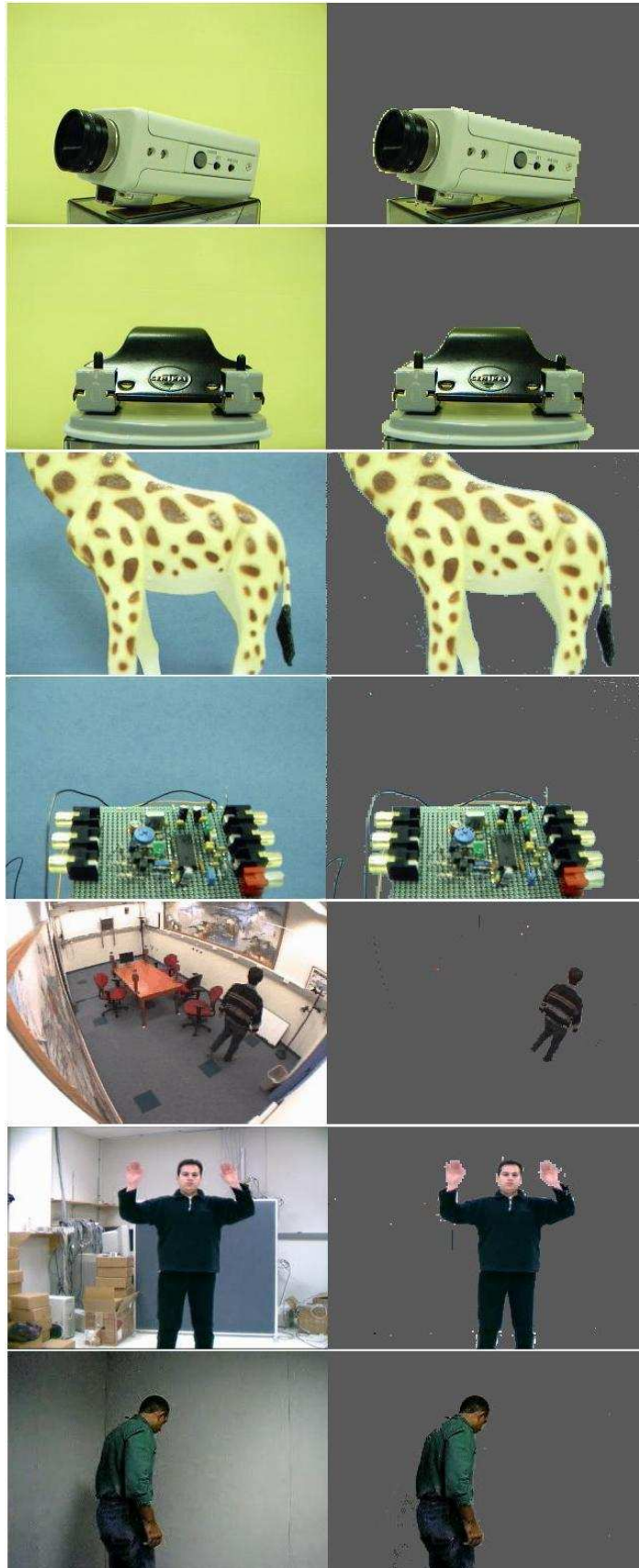


Figura 78: Resultados da detecção de objetos utilizando-se o método proposto com as modificações para redução do tamanho da memória para armazenar a imagem de referência.

5.6 CONCLUSÕES

Neste capítulo foram apresentadas as adaptações efetuadas para conversão das operações do método de detecção de objetos para ponto fixo visando a implementação em hardware. As equações foram adaptadas mantendo-se a exatidão de 7 bits fracionários, e a realização de operações de multiplicação e divisão com números de potência de dois. A exatidão de 7 bits fracionários foi obtida pela multiplicação e divisão dos termos das equações pelo fator 2^7 .

Foram apresentados os resultados da avaliação do método de detecção de objetos, com operações em ponto fixo e flutuante, utilizando-se dois tipos de classificações dos *pixels* denominadas Classificação do *Pixel* 01, e Classificação do *Pixel* 03. A Classificação do *Pixel* 01 utiliza a função de compensação de variação do *pixel* e a detecção de sombras durante a classificação dos *pixels*. A Classificação do *Pixel* 03 utiliza somente a função de compensação de variação do *pixel* durante a detecção de objetos.

Para as seqüências de imagens de testes 01 e 02, de acordo com a análise de variância feita com o método estatístico de Kruskal Wallis, as diferenças entre as medidas F, obtidas com as operações em ponto fixo e flutuante, não são estatisticamente significantes. Para a seqüência de imagens de testes 03, baseado nas comparações dos pares das medidas F, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 03 realizada com operações em ponto fixo e flutuante. Para a seqüência de imagens de testes 04, de acordo com as comparações dos pares das medidas F, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 01 realizada com operações em ponto fixo e flutuante. Para a seqüência de imagens de testes 05, baseado nas comparações dos pares das medidas F, não há uma diferença significativa entre as medidas F obtidas com a Classificação do *Pixel* 03 realizada com operações em ponto fixo e flutuante. Estes resultados demonstram que o método proposto neste trabalho, implementado em ponto fixo, pode ter um desempenho similar ao implementado com operações em ponto flutuante.

Para reduzir o tamanho da memória utilizada para armazenar os valores médios parciais de cada *pixel*, foi proposto o armazenamento na memória dos valores parciais referentes à média entre dois quadros consecutivos. Outra simplificação proposta para calcular os valores médios foi à redução da resolução horizontal da imagem de 640 para 320 *pixels* somente para obtenção dos valores médios da cada *pixel*. Assim sendo, além da média temporal também é realizada uma média espacial entre pares *pixels* consecutivos em cada linha. Desta maneira, o tamanho da memória para armazenamento dos valores médios dos

pixels reduz de 973.440 bytes para 486.720 bytes. Além das simplificações para calcular os valores médios, outra modificação proposta é a utilização de um valor global do desvio padrão para cada componente do espaço de cores.

Foram apresentados os resultados da avaliação do método de detecção de objetos com as simplificações propostas neste capítulo. Para as seqüências de imagens de testes 01 e 02 as diferenças entre as medidas F, obtidas com e sem as simplificações propostas para os valores médios e os desvios padrões, não foram estatisticamente significantes. Para as demais seqüências de imagens de testes (03, 04 e 05), o método de detecção de objetos com as simplificações propostas neste capítulo apresentou bons resultados após as modificações efetuadas para reduzir a quantidade de memória para armazenar a imagem de referência.

Uma vez validado o método de detecção de objetos com operações em ponto fixo e com as simplificações visando a redução da quantidade de memória para armazenamento do modelo da imagem de referência, no próximo capítulo é apresentada a arquitetura para implementação em hardware do método proposto neste trabalho.

6 ARQUITETURAS PARA SEGMENTAÇÃO DE VÍDEO

Neste capítulo apresenta-se a arquitetura utilizada para implementar o método de segmentação de vídeo para detecção de objetos com as adaptações apresentadas no capítulo 4. Primeiramente, faz-se uma breve descrição das arquiteturas pesquisadas na literatura para segmentação de vídeo. Em seguida, apresenta-se uma visão geral da arquitetura e os blocos de hardware utilizados. Finalmente, descrevem-se os resultados obtidos com a arquitetura implementada em um kit de desenvolvimento da Altera.

6.1 INTRODUÇÃO

Os arranjos de lógica programável em campo (FPGA) têm sido utilizados em diversas aplicações na área de processamento de vídeo em tempo, tais como detecção de bordas [17], compressão de dados [18], detecção de faces [19], e segmentação de imagens [20, 92]. Um FPGA permite a elaboração de estruturas paralelas flexíveis que podem executar operações em tempo real. Além disso, estes dispositivos têm sido utilizados como protótipo no desenvolvimento de projetos de circuitos integrados de aplicação específica (ASIC) ou em alguns casos, como produtos finais [93]. Em função destas características, alguns métodos de segmentação de vídeo em tempo real tem sido implementados em FPGAs [20, 35, 36, 37, 38, 89, 94].

As arquiteturas desenvolvidas para segmentação de vídeo, pesquisadas ao longo do desenvolvimento deste trabalho, são específicas para implementação de cada método destinado à segmentação de vídeo. Arquiteturas dedicadas são necessárias para atender aos requisitos do projeto e as arquiteturas reconfiguráveis são uma boa estratégia de implementação para atingir os objetivos do projeto com um custo e tempo razoáveis [95, 96]. Em [89], elaborou-se uma arquitetura em FPGA para implementação de um método não-paramétrico para detecção de objetos [61]. A arquitetura em hardware foi implementada em dois dispositivos FPGA com 8320 elementos lógicos cada um. Dois bancos de memória foram utilizados para armazenar os parâmetros do modelo da imagem de referência e os quadros com os resultados da segmentação. Além disso, são utilizados dois *buffers* para armazenar os *pixels* de duas linhas. Enquanto uma linha é armazenada em um dos *buffers*, a outra é lida e os valores dos *pixels* são enviados para os demais blocos do sistema. Para exibir o resultado da segmentação no codificador de vídeo são utilizados quatro *buffers* que acumulam duas linhas pares e duas ímpares. Estas linhas são lidas de uma memória externa que é utilizada para armazenar dois quadros completos com o resultado da segmentação.

Enquanto um quadro é enviado para a saída (codificador de vídeo), os valores correntes com o resultado da segmentação são armazenados em uma outra área da memória. Um sistema de gerenciamento e controle da memória é utilizado para controlar o acesso à memória. A resolução das imagens processadas foi reduzida de 320 x 240 para 160 x 120 *pixels* a fim de que o processamento fosse realizado na taxa de 30 quadros por segundos. Em função dos atrasos internos nos FPGAs utilizados nesta arquitetura, a frequência de operação obtida foi igual a 40 MHz.

6.2 ARQUITETURA PROPOSTA

A seguir apresenta-se a arquitetura implementada em hardware para detecção de objetos segundo o método desenvolvido ao longo desta pesquisa. Primeiramente, descrevem-se os principais blocos da arquitetura necessários para aquisição da informação de vídeo, segmentação e exibição dos resultados. Em seguida, os principais blocos da unidade de segmentação de vídeo são apresentados. Finalmente, mostram-se os resultados da síntese e o desempenho do método de detecção de objetos implementado em hardware.

6.2.1 Visão Geral da Arquitetura

O diagrama em blocos da arquitetura proposta para realizar a segmentação de vídeo proposta no capítulo 04 está ilustrado na Figura 79. Utilizou-se o kit educacional e de desenvolvimento DE-2 da Altera para efetuar os testes da arquitetura [97]. No FPGA foram implementados os seguintes blocos: interface I2C, interface do decodificador de vídeo, unidade de segmentação de vídeo, interface da memória SRAM, interface da memória SDRAM, e interface do codificador de vídeo.

Durante a inicialização, a interface I2C configura o decodificador de vídeo (ADV7181) para que o sinal analógico de vídeo da entrada seja corretamente digitalizado e convertido para o formato ITU-R 656. [98]

O decodificador de vídeo disponibiliza cada imagem digitalizada com uma resolução igual a 720 x 507 *pixels*. Efetuou-se uma decimação para redução da resolução horizontal de 720 para 640 *pixels* por linha. A decimação é realizada pela interface do decodificador de vídeo que também captura a informação de vídeo do protocolo ITU-R 656 e disponibiliza na saída as informações YCbCr 4:2:2, com 640 amostras do Y por linha e 320 amostras de cada

componente de cor (Cb/Cr) por linha. As componentes de vídeo, juntamente com sinal de sincronismo de *pixel* de 13,5 MHz, são enviadas para a unidade de processamento de vídeo.

A unidade de processamento de vídeo realiza todas as etapas necessárias para segmentação de vídeo de acordo com o método proposto no capítulo 04. A interface do codificador de vídeo recebe as componentes de vídeo YCbCr com o resultado da unidade de processamento de vídeo e converte estas componentes para RGB antes de enviá-las para o codificador de vídeo (ADV7123).

A interface da SRAM é utilizada para efetuar a leitura e escrita da imagem de referência armazenada na memória SRAM (IS61LV25616). A memória SDRAM é utilizada como buffer dos quadros que são enviados para o codificador de vídeo. Enquanto o quadro corrente é armazenado na memória, o anterior é lido da SDRAM e enviado para o codificador de vídeo via interface do codificador de vídeo.

Na próxima seção são descritos os blocos da unidade de processamento de vídeo que realizam a segmentação de vídeo de acordo com o método proposto nos capítulos anteriores.

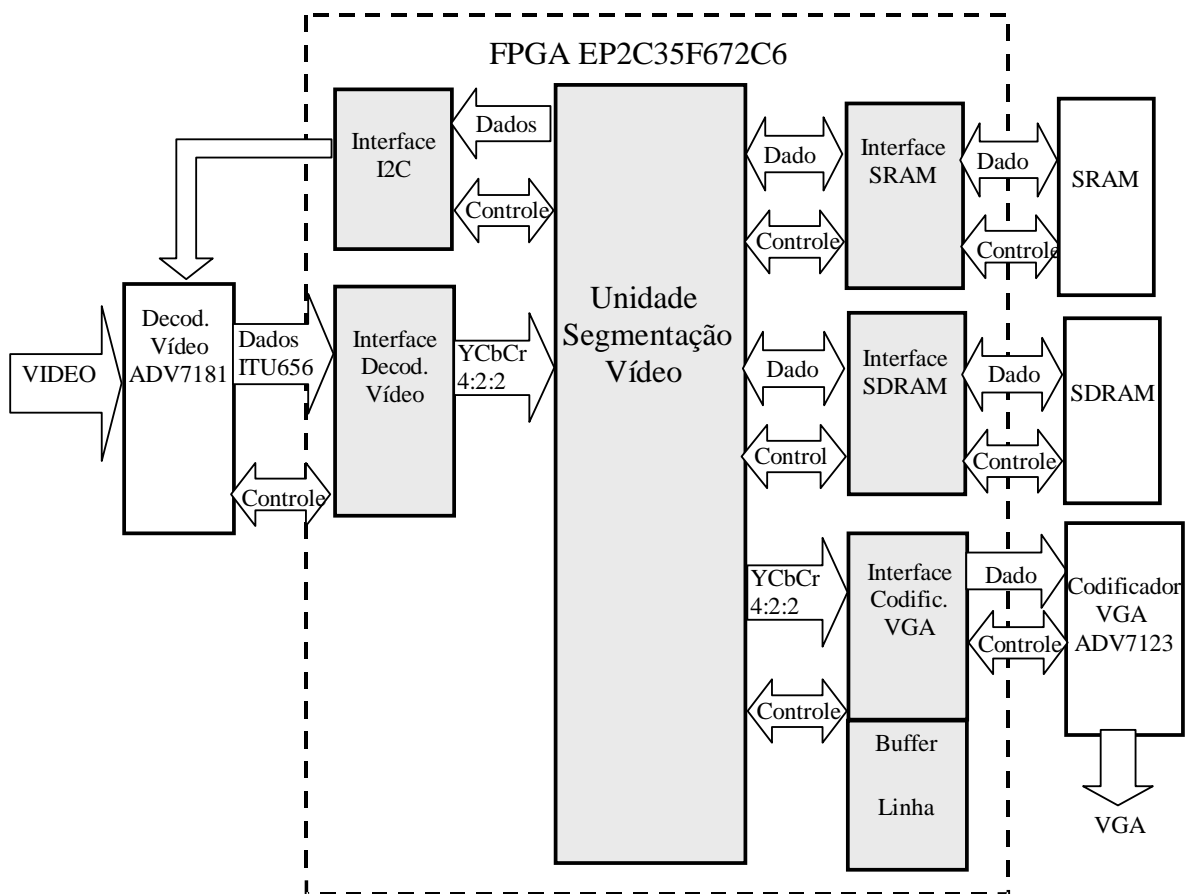


Figura 79: Diagrama em blocos da arquitetura para segmentação de vídeo

6.2.2 Unidade de Segmentação de Vídeo

O diagrama em blocos da unidade de segmentação de vídeo está ilustrado na Figura 80. A unidade de segmentação de vídeo é composta pelos seguintes blocos: gerenciamento do sistema, média aritmética, diferença absoluta, desvio padrão, detecção de variação global, detecção de sombras & compensação de variações do *pixel*, classificação do *pixel*, seleção automática dos limiares, e controle do buffer de quadro. A seguir são descritas as funcionalidades de cada bloco da unidade de segmentação de vídeo.

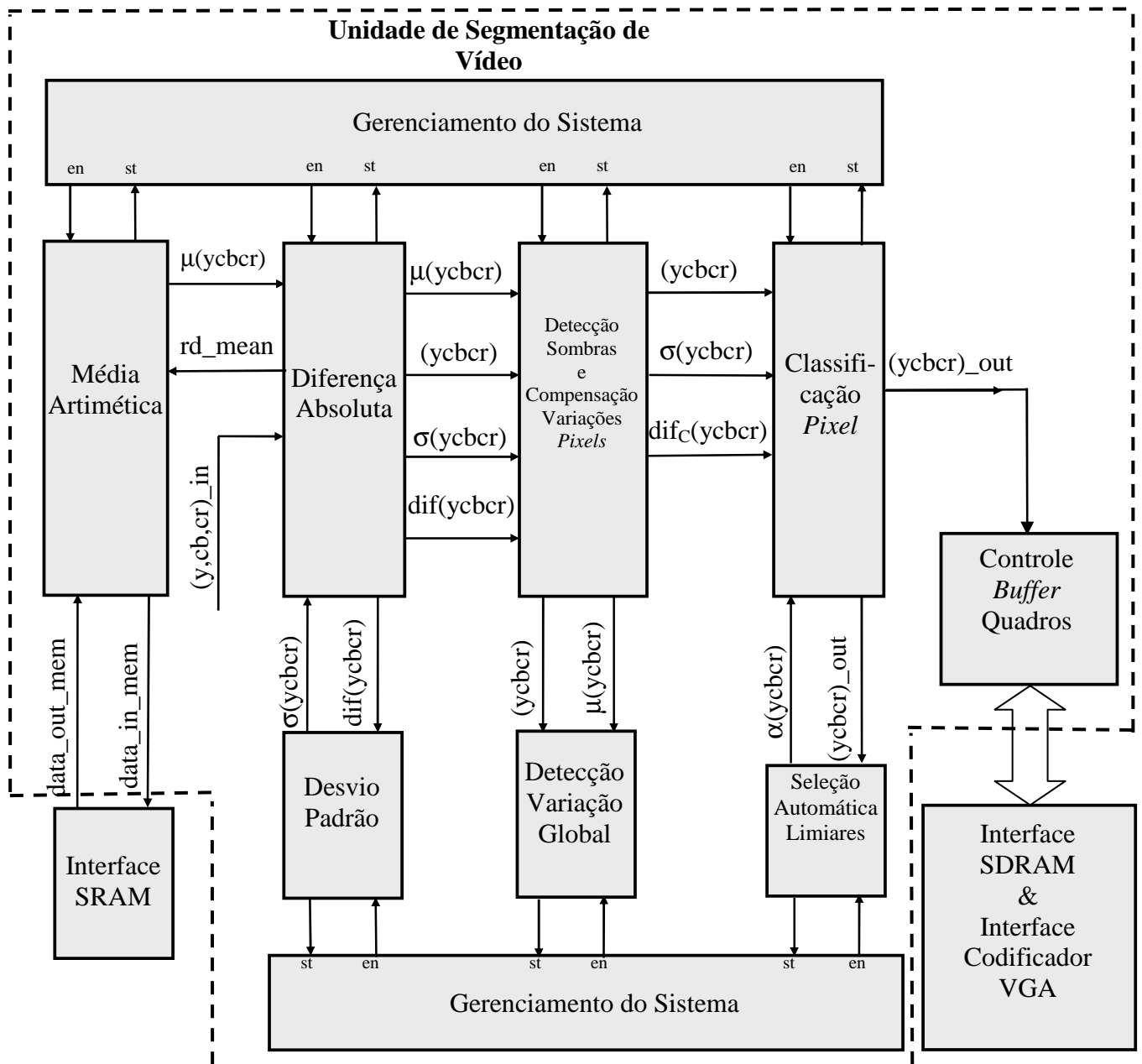


Figura 80: Diagrama em blocos da unidade de segmentação de vídeo

6.2.2.1 Gerenciamento do sistema

O bloco denominado gerenciamento do sistema controla a seqüência de processamento dos blocos da unidade de segmentação de vídeo de modo que todas as etapas para segmentação de vídeo sejam realizadas. Cada bloco recebe um conjunto de parâmetros, realiza o processamento necessário e depois disponibiliza na saída os parâmetros recebidos na entrada juntamente com os novos parâmetros resultantes do processamento realizado pelo bloco. Nenhum buffer é utilizado para armazenar as amostras de entrada, pois todas as operações são realizadas para cada dois *pixels* sucessivos, ou seja, para uma freqüência de *clock* igual a 100 MHz e a freqüência de *pixel* igual a 13,5 MHz, tem-se aproximadamente 15 ciclos para realização das operações. Todas as operações do método de segmentação de vídeo apresentado nos capítulos anteriores são realizadas com menos de 14 ciclos de operação. O processamento é dividido em duas partes: aquisição da imagem de referência, seleção automática dos limiares e extração dos objetos.

Primeiramente é feito o cálculo dos valores médios de cada *pixel*. Para isto o bloco gerenciamento do sistema habilita o módulo denominado média aritmética. Quando um quadro completo é processado, o módulo média aritmética informa ao gerenciador do sistema a qual verifica se a quantidade de quadros estabelecida já foi processada. Se não, o módulo média aritmética é novamente acionado para que um novo quadro seja processado. Quando todos os quadros são processados, os valores médios de cada *pixel* em cada componente do espaço de cores YCbCr estarão armazenados na memória SRAM externa e inicia-se o cálculo do desvio padrão.

Para calcular o desvio padrão, são acionados os módulos denominados desvio padrão e diferença absoluta. Além disso, o módulo denominado média aritmética também é habilitado para efetuar a leitura dos valores médios de cada *pixel* previamente calculados e armazenados na memória. O módulo diferença absoluta recebe os valores médios de cada *pixel* e calcula as diferenças em relação aos respectivos valores correntes. Os resultados são enviados para o módulo desvio padrão. Ao término do processamento de um quadro completo, o módulo desvio padrão sinaliza para o módulo gerenciamento do sistema, o qual verifica se todos os quadros foram processados. Quando o desvio padrão é calculado para todos os quadros estabelecidos, inicia-se o processo de classificação dos *pixels*.

Para efetuar a classificação dos *pixels* procede-se da seguinte maneira: Ao comando do módulo denominado diferença absoluta, o módulo denominado média aritmética efetua a leitura dos valores médios de cada *pixel* armazenados na memória SRAM. Os valores médios

de cada *pixel* juntamente com o respectivo desvio padrão são enviados para o módulo denominado diferença absoluta. Sincronizado com o *clock* de *pixel*, o módulo diferença absoluta disponibiliza em sua saída, para cada *pixel*, os valores médios lidos da memória, valores correntes, desvios padrões e as diferenças em relação aos valores médios. Os parâmetros na saída do módulo diferença absoluta são enviados para o módulo denominado detecção de sombras e função de compensação das variações dos *pixels* a qual identifica os elementos de imagens que pertencem a sombras e também multiplica os valores das diferenças de cada *pixel* em relação aos valores médios pelo fator de correção correspondente.

O módulo detector de sombras e função de compensação das variações dos *pixels* disponibiliza em sua saída, para cada *pixel*, os valores médios lidos da memória, os valores correntes de cada *pixel*, o desvio padrão e a diferença em relação ao valor médio multiplicada pelo respectivo fator de correção. O módulo denominado classificação do *pixel* recebe os valores dos parâmetros enviados pelo módulo detector de sombras e realiza a classificação de cada *pixel*. Os *pixels* classificados como pertencentes à imagem de fundo são exibidos em preto e os classificados como pertencentes ao objeto são substituídos pelos valores correntes oriundo dos módulos anteriores.

6.2.2.2 Média aritmética

O módulo denominado média aritmética tem duas funções: cálculo dos valores médios de cada *pixel* e a leitura destes valores, armazenados na memória SRAM, durante a fase de classificação de cada *pixel*. Na Figura 81 está ilustrado o diagrama em bloco do módulo média aritmética.

Na fase de aquisição da imagem de referência, são obtidos primeiramente os valores médios espaciais entre dois *pixels* consecutivos (Y_0Y_1 , Y_2Y_3 , Cb_0Cb_1 , e Cr_0Cr_1) e os resultados destas operações são utilizados para calcular a média temporal com os valores previamente armazenados na memória SRAM. Os valores médios obtidos são novamente armazenados na memória. Esta operação é repetida até que todos os quadros tenham sido processados.

Na fase de segmentação, o módulo média aritmética efetua a leitura dos valores médios de cada *pixel* armazenados na memória e os envia para o módulo denominado subtração de quadro.

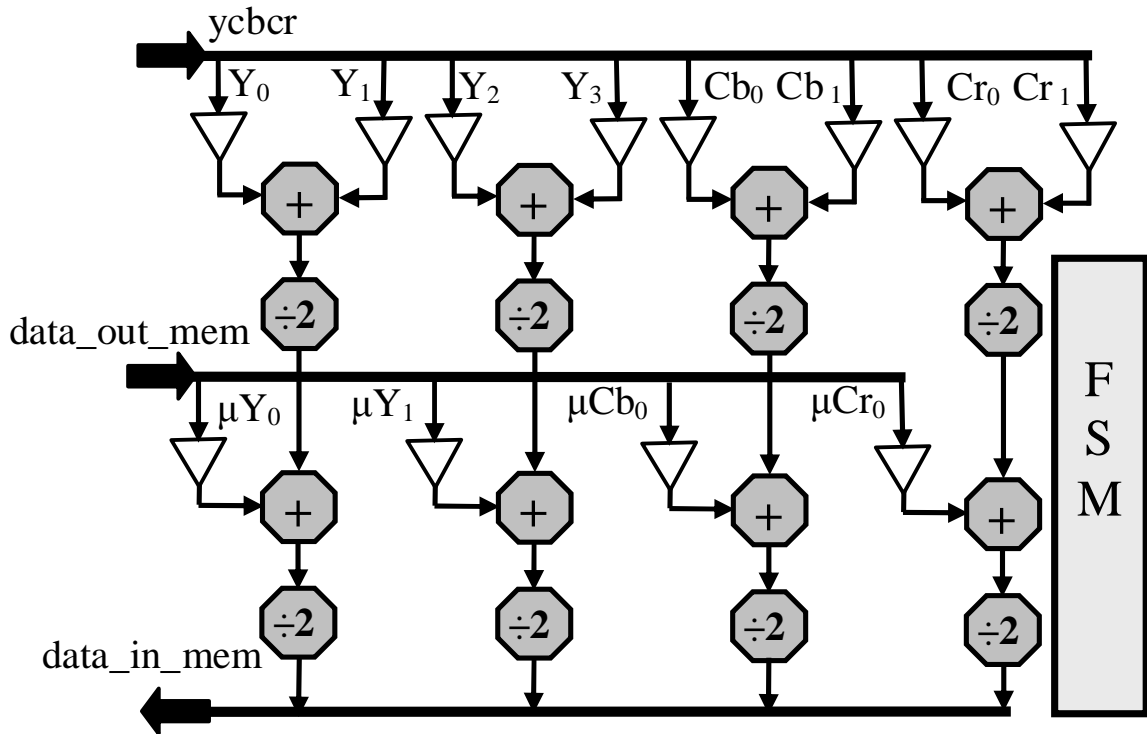


Figura 81: Diagrama em blocos do módulo média aritmética

6.2.2.3 Desvio padrão

Na Figura 82 está ilustrado o diagrama em blocos do módulo denominado desvio padrão. O módulo denominado desvio padrão é responsável pelo cálculo do desvio padrão durante a fase de aquisição da imagem de referência de acordo com a equação 36.

$$s(Y, Cb, Cr) = \sqrt{\frac{1}{NXY} \sum_{i=1}^N \sum_{x=1}^{XY} (I_{i(k)}(Y, Cb, Cr) - \bar{x}_{i(k)}(Y, Cb, Cr))^2} \quad (36)$$

Para calcular o desvio padrão são acionados também os módulos denominados média aritmética e diferença absoluta. Durante a etapa de obtenção do desvio padrão, o módulo denominado média aritmética faz a leitura dos valores médios da cada *pixel* armazenados na memória e os envia para o módulo denominado diferença absoluta. O módulo denominado subtração de quadros efetua as diferenças absolutas entre os valores correntes de cada *pixel* e os seus respectivos valores médios armazenados na memória. Os resultados obtidos são enviados para o módulo denominado desvio padrão para serem elevados ao quadrado e acumulados até que a quantidade de quadros estabelecida tenha sido processada. Ao final, faz-

se o cálculo da raiz quadrada da média dos valores acumulados para obtenção do desvio padrão para cada componente YCbCr.

Durante a etapa de classificação, os valores dos desvios padrões previamente calculados são disponibilizados para obtenção do intervalo de confiança e classificação de cada *pixel*.

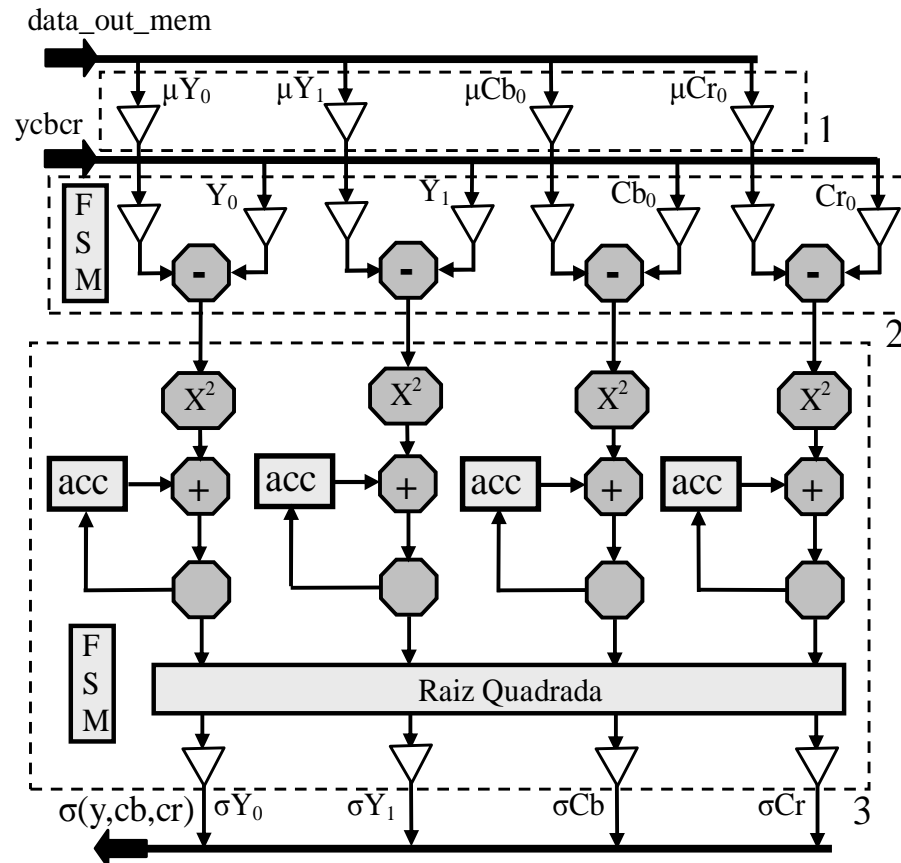


Figura 82: Módulos utilizados para calcular o desvio padrão. 1) Módulo média aritmética; 2) Módulo Diferença Absoluta; 3) Módulo desvio padrão.

6.2.2.4 Detecção de sombras e compensação de variações

O módulo denominado detecção de sombras e compensação de variações, ilustrado nas Figuras 83, 84, 85, e 86, efetua a identificação dos *pixels* cuja variação esteja fora do intervalo de confiança. Esta identificação é baseada nos valores médios, desvios padrões e número de quadros obtidos e utilizados durante a fase de aquisição da imagem de referência. Nas diferenças absolutas dos *pixels* detectados como fora do intervalo de confiança são aplicados os fatores de compensação, visando reduzir a variação detectada em função de ruídos, iluminação ou causada por ajustes da câmera como, por exemplo, íris automática. Estes fatores de correção são aplicados nas três componentes de cores. Além disso, o valor da

componente Y é comparado com uma faixa que corresponde a os pontos da imagem sombreados para identificar os *pixels* pertencentes a sombras na imagem.

O diagrama em blocos ilustrado na Figura 83 corresponde ao módulo responsável pela obtenção do numerador da equação 31, isto é, a multiplicação das diferenças absolutas por 128.

$$\left. \begin{array}{l}
 \text{Se } I_{(k)}(Y, Cb, Cr) > UL_{(k)}(Y, Cb, Cr): \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)| \times 2^7}{I_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } I_{(k)}(Y, Cb, Cr) < LL_{(k)}(Y, Cb, Cr): \\
 PCF_{(k)}(Y, Cb, Cr) = \frac{|I_{(k)}(Y, Cb, Cr) - \bar{x}_{(k)}(Y, Cb, Cr)| \times 2^7}{\bar{x}_{(k)}(Y, Cb, Cr)} \\
 \text{ou} \\
 \text{Se } LL_{(k)}(Y, Cb, Cr) \leq I_{(k)}(Y, Cb, Cr) \leq UL_{(k)}(Y, Cb, Cr): \\
 PCF_{(k)}(Y, Cb, Cr) = 1 \times 2^7
 \end{array} \right\} \quad (31)$$

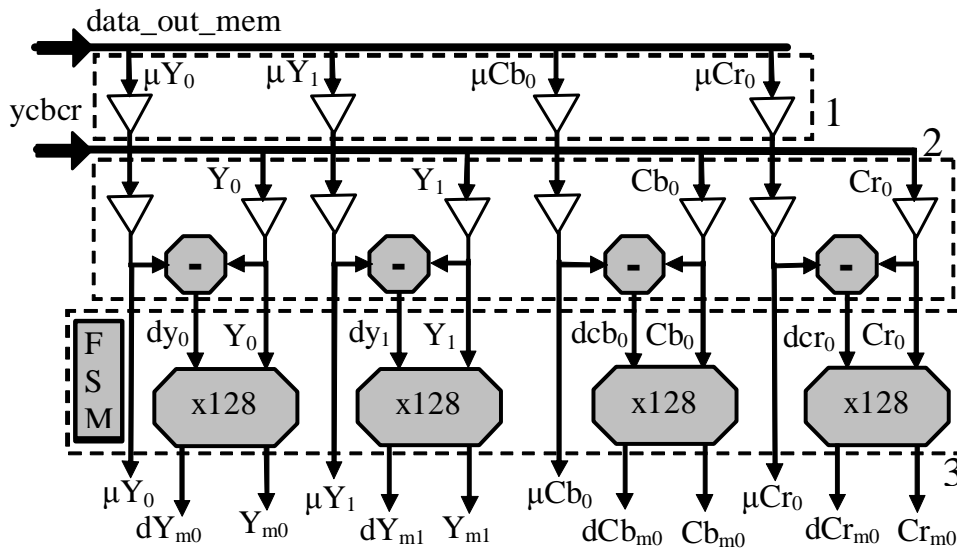


Figura 83: Diagrama em blocos do módulo denominado Compensação das variações dos *pixel* e detecção de sombras. Cálculo do numerador da equação 31 para o Y.

O diagrama em blocos mostrado na Figura 84 é do módulo responsável pelo cálculo dos limites inferiores e superiores dos intervalos de confiança para cada *pixel* de acordo com as equações 29 e 30, respectivamente.

$$UL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) + \frac{s_{(k)}(Y, Cb, Cr)}{2^2} \quad (29)$$

$$LL_{(k)}(Y, Cb, Cr) = \bar{x}_{(k)}(Y, Cb, Cr) - \frac{s_{(k)}(Y, Cb, Cr)}{2^2} \quad (30)$$

O valor do desvio padrão é dividido por 4 e o resultado desta divisão é somado e subtraído dos valores médios de cada *pixel* armazenados na memória para obtenção dos limites superiores e inferiores dos intervalos de confiança.

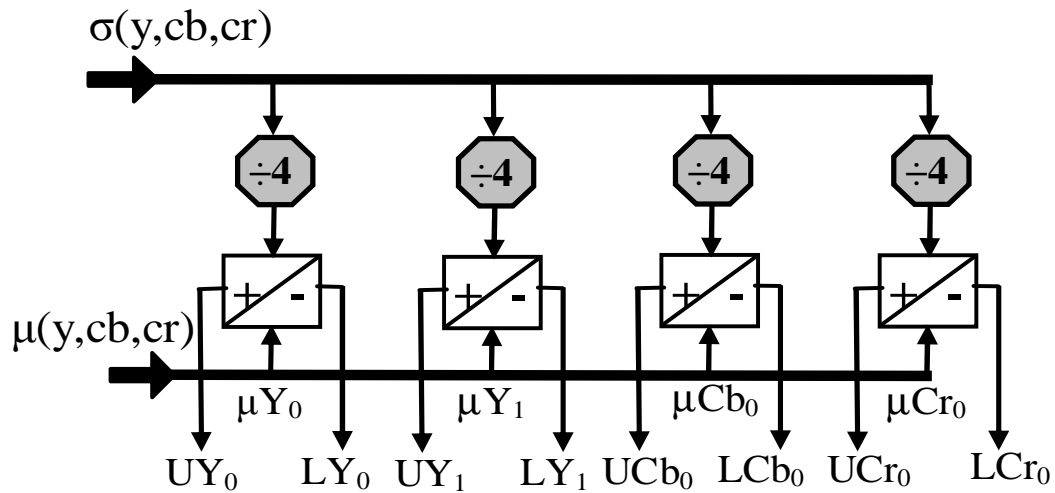


Figura 84: Diagrama em blocos do módulo denominado Compensação das variações dos *pixel* e detecção de sombras. Cálculo dos limites superiores e inferiores dos intervalos de confiança.

O diagrama em blocos da Figura 85 realiza a detecção de sombras de acordo com a equação 34 e calcula os resultados finais das equações 31 e 33.

$$F_k(Y, Cb, Cr) = \frac{BD_{(k)}(Y, Cb, Cr) \times PCF_{(k)}(Y, Cb, Cr)}{2^7} \quad (33)$$

$$\begin{cases} I_{(k)}(Y) \times 2^7 > LL_{(k)}(Y) \times \alpha_{lo} \times 2^7 \\ e \\ I_{(k)}(Y) \times 2^7 < LL_{(k)}(Y) \times \alpha_{hi} \times 2^7 \end{cases} \quad (34)$$

O módulo denominado COMP 1 compara os valores dos *pixels* correntes com os valores dos limites do intervalo de confiança previamente calculados. O resultado da comparação é utilizado para seleccionar o sinal de entrada do multiplexador 1, isto é, o denominador da equação 31. Se o valor do *pixel* corrente estiver dentro do intervalo de confiança, o valor corrente da diferença absoluta do *pixel* em processamento é enviado para a saída do multiplexador 02 sem compensação de variação. Caso contrário, o valor da diferença absoluta é multiplicado pelo fator de compensação e o resultado é enviado para a saída do multiplexador 02.

O módulo denominado COMP 02 verifica se cada *pixel* é parte integrante de sombras. Se sim, a saída do multiplexador 03 recebe zero, se não recebe o valor da diferença absoluta compensada oriunda da saída do multiplexador 02.

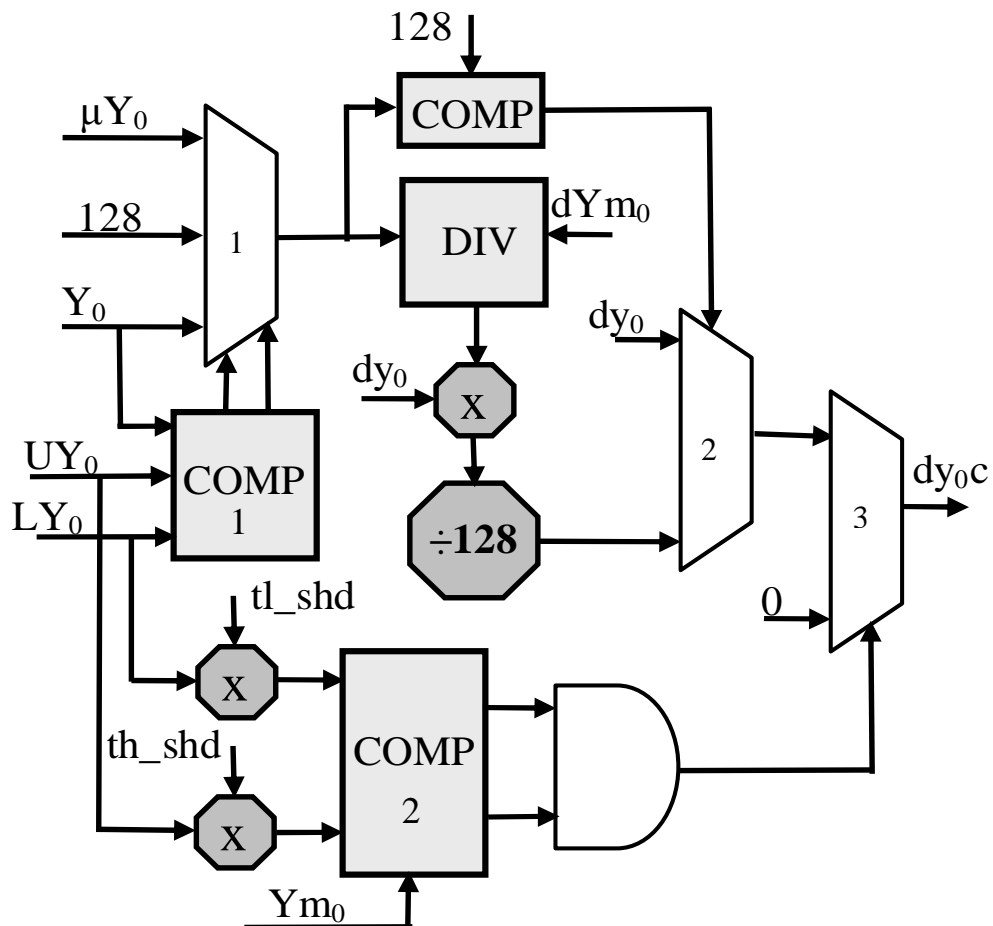


Figura 85: Diagrama em blocos do módulo denominado Compensação das variações dos *pixel* e detecção de sombras. Detecção de sombras, cálculo dos resultados finais das equações 31 e 33 para o Y .

No diagrama em blocos da Figura 86 está ilustrado o módulo responsável pelo cálculo da equação 31 e os resultados finais das Equações 31 e 33 para as componentes de cores Cb e Cr. Esta operação é similar à realizada na Figura 85, porém neste caso não efetuada a detecção de sombras.

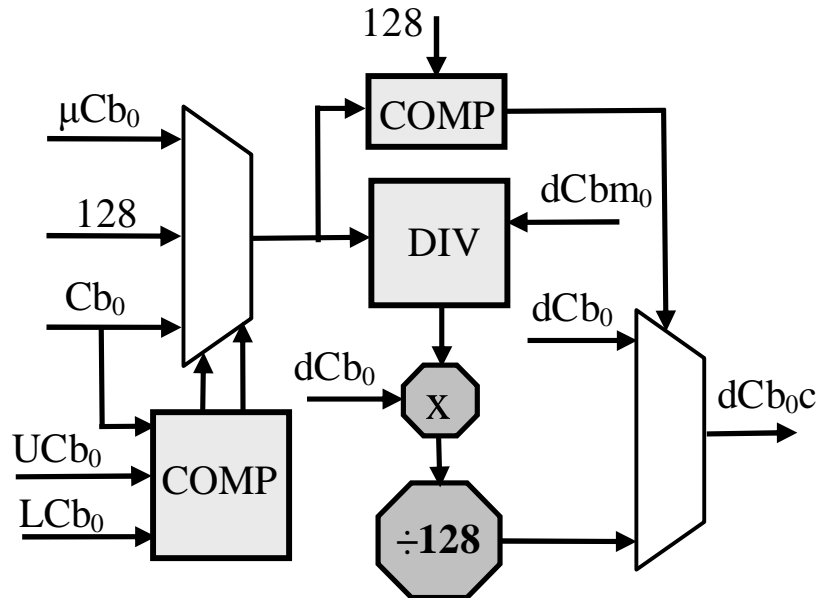


Figura 86: Diagrama em blocos do módulo denominado Compensação das variações dos *pixel* e detecção de sombras. Cálculo dos resultados finais das equações 31 e 33 para as componentes Cb e Cr.

6.2.2.5 Classificação do *pixel*

O diagrama em blocos do módulo denominado classificação do *pixel* é ilustrado na Figura 87. Para realizar a detecção dos objetos, os valores dos desvios padrões são multiplicados pelos coeficientes α e os resultados são utilizados como limiares. O módulo denominado classificação do *pixel* efetua a comparação das diferenças absolutas de cada *pixel*, enviada pelo módulo denominado detecção de sombras e compensação de variação do *pixel*, com os limiares. Se YCb ou YCr for maior que os limiares estabelecidos, o *pixel* é classificado como objeto e o seu valor na saída é o valor do *pixel* corrente. Caso contrário, o *pixel* é classificado como pertencente a imagem de fundo e o seu valor na saída será igual a 0 para o Y e 128 para Cb/Cr, isto é, a imagem resultante é preta na saída.

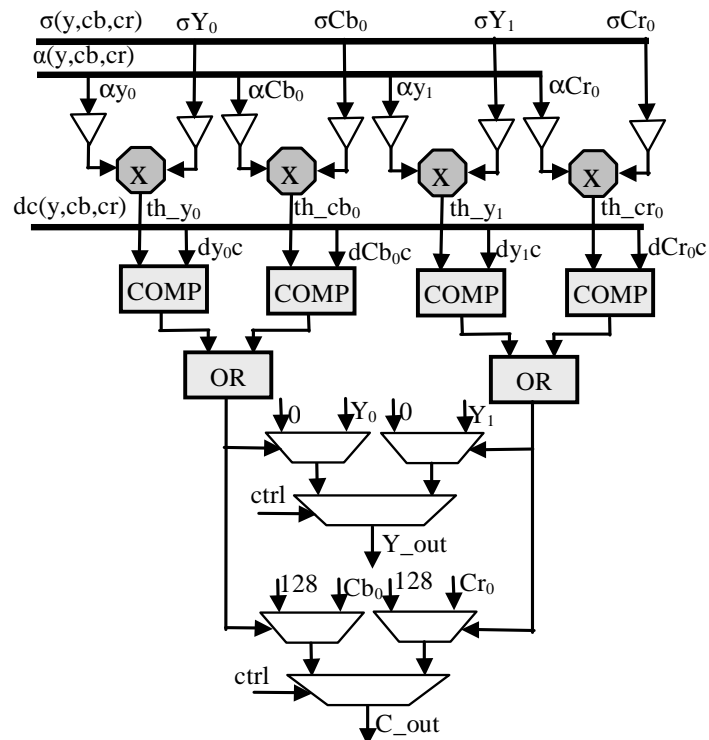


Figura 87: Diagrama em blocos do módulo denominado classificação do *pixel*

6.2.2.6 Seleção automática dos limiares

O módulo denominado seleção automática dos limiares determina os valores das constantes utilizadas como fatores de multiplicação dos desvios para classificação dos *pixels*. Logo após o término da modelagem da imagem de referência é feita a contagem do número de *pixels* detectados como objetos em cada quadro. Esta contagem é realizada para cada componente de cor (YCbCr). Se o número de *pixels* for maior que 50, a constante correspondente é incrementada por duas unidades. Este processo é repetido até que o número de *pixels* detectados seja inferior ao estabelecido. Ao final ter-se-ão os valores das constantes a serem utilizadas na detecção de objetos.

6.2.2.7 Detecção variação global

Os valores médios dos *pixels* lidos da memória e os valores correntes da saída do módulo denominado diferença absoluta são também enviados para o módulo denominado média global que efetua o monitoramento dos valores médios globais de luminância para verificar se houve uma aumento ou diminuição dos valores correntes em relação ao armazenado na memória. Quando isto é detectado um novo modelo de referência é adquirido automaticamente.

6.3 DESEMPENHO DA ARQUITETURA

Nesta seção são apresentados os resultados obtidos com a arquitetura implementada em FPGA em um ambiente com variações de iluminação e pontos saturados. Na Figura 88 está ilustrada a seqüência de imagens, utilizada para testar o desempenho da arquitetura, com e sem objetos no cenário e os respectivos histogramas da componente Y com os valores médios e o desvio padrão (D.P.).

A imagem superior esquerda é a imagem de fundo sem a presença de objetos. Há uma janela aberta que permite a entrada de intensos raios solares causando o aparecimento de pontos saturados no cenário. As próximas duas imagens são imagens de referência com uma e duas lâmpadas ligadas, respectivamente. De acordo com os histogramas, quando as duas lâmpadas são ligadas, a média global da componente Y varia de 186,4 para 208, o que corresponde a um crescimento de 11,53%.

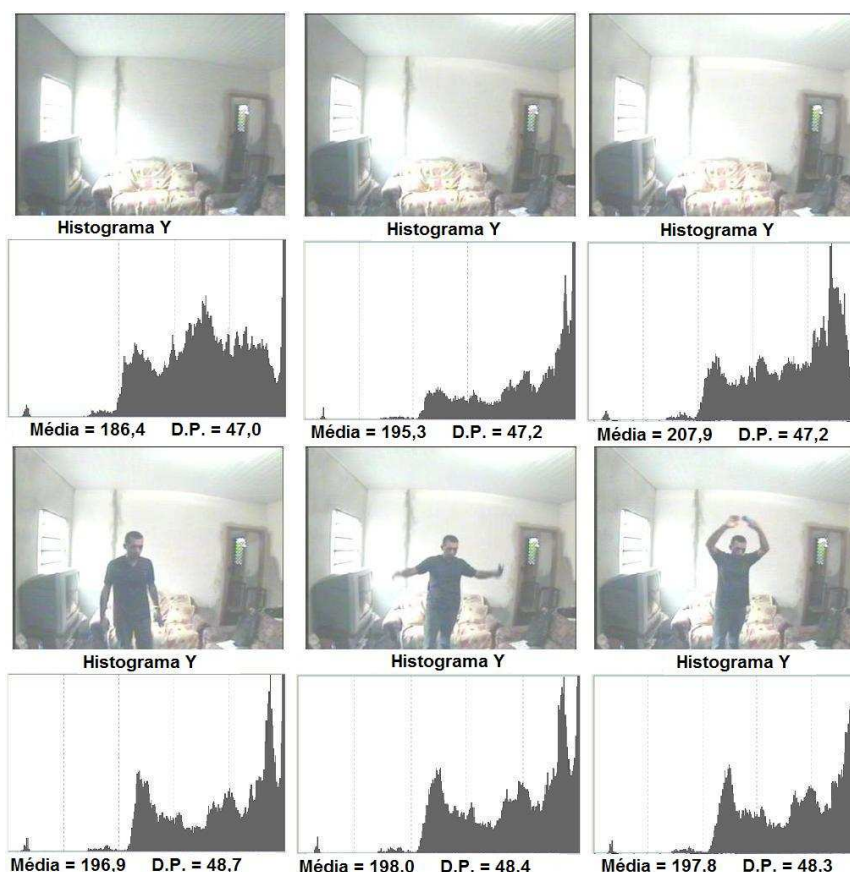


Figura 88: Seqüência de imagens utilizada para avaliar o desempenho da arquitetura implementada em hardware em um ambiente com variações de iluminação.

Na Figura 89 estão ilustrados os resultados do desempenho da arquitetura com as lâmpadas desligadas. Na seqüência de imagens 01, tanto a detecção de sombras quanto a função de compensação de variação do *pixel* foram utilizadas durante a fase de segmentação.

Os pontos em branco são os classificados como pertencentes ao objeto e os em preto são os detectados como imagem de referência. Em função da saturação em alguns pontos do cenário próximo da janela, alguns *pixels* foram erroneamente classificados como pertencentes ao objeto. De acordo com o gráfico de falsos positivos, ao lado da seqüência de imagens ilustrada na Figura 89, o número de falso positivo é menor que 1%.

Na seqüência de imagens 02, a função de compensação do *pixel* foi desligada para a componente Y. Isto causou um acréscimo no número de falsos positivos para 5%. Neste caso, os pontos saturados na janela e no sofá foram falsamente classificados como pertencentes ao objeto.

Na seqüência de imagens 03, a função de compensação do *pixel* para a componente Y foi ligada e para as componentes Cb/Cr foi desligada. Neste caso, a porcentagem de falso positivo reduziu de 5% para aproximadamente 1,5% e uma região no sofá foi falsamente detectada como parte do objeto em função da saturação das cores nesta região. Além disso, alguns pontos na janela também contribuíram para a porcentagem de falso positivo. Na última seqüência de imagens, as compensações estão ligadas para todas as componentes YCbCr e com isso a porcentagem de falso positivo reduz abaixo de 1%.

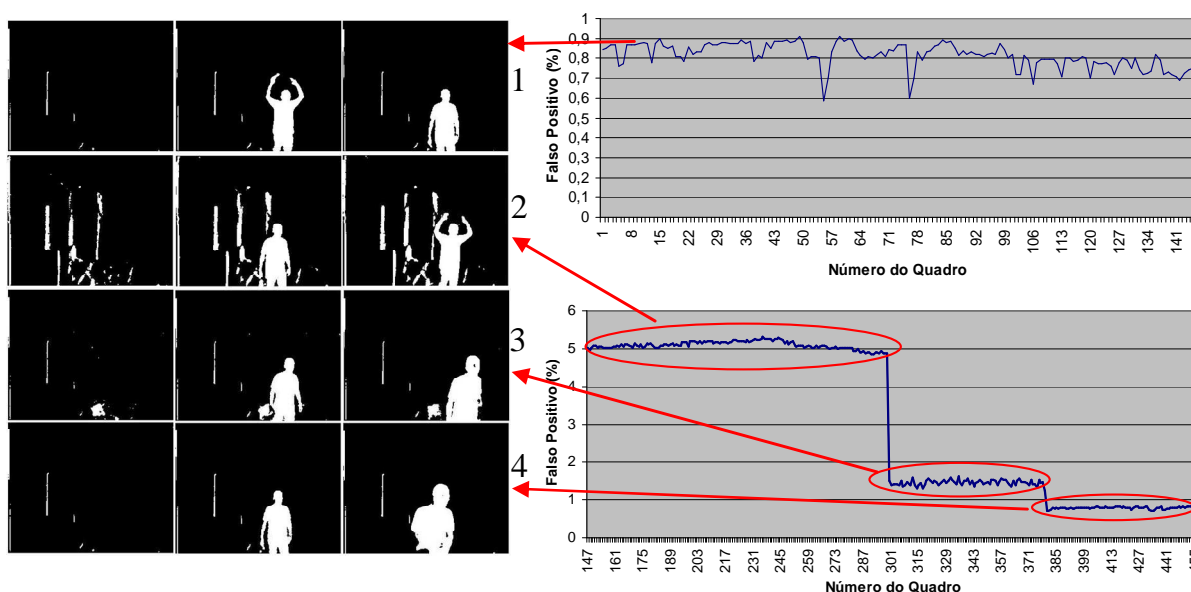


Figura 89: Resultados da segmentação realizada com a arquitetura implementada no kit Altera. Em todos os casos as lâmpadas estão desligadas. 1) a função de compensação de variação do *pixel* está ativa para as componentes YCbCr; 2) a função de compensação de variação do *pixel* para a componente Y está desligada; 3) a função de compensação de variação do *pixel* está ligada para a componente Y e desligada para as componentes Cb/Cr; 4) idem ao 1).

Na Figura 90 estão ilustrados os resultados obtidos variando-se a iluminação por meio de lâmpadas instaladas no ambiente. Na seqüência de imagens 01, uma lâmpada foi

ligada e tanto a compensação de *pixel* quanto a detecção de sombras foi utilizada durante a etapa de classificação dos *pixels*. Neste caso, o número de falso positivo está em torno de 1%.

Na seqüência de imagens 02, a outra lâmpada foi ligada e ocorreu um aumento temporário na porcentagem de falso positivo de 1 para 3,5% em função da adaptação do sensor da câmera a variação de iluminação. Em seguida, houve uma redução de 3,5 para 1,8% na porcentagem de falso positivo. Na seqüência de imagens 03, a função de compensação de variação do *pixel* foi desligada para a componente Y e aumentou a porcentagem de falso positivo para 25%. Conforme a pessoa anda na sala, os *pixels* têm suas intensidades modificadas e com isso ocorre uma variação na porcentagem de falso positivo. Na seqüência de imagens 04, uma das lâmpadas foi desligada e a função de compensação permaneceu desligada para a componente Y. Finalmente, a seqüência de imagens 05, a função de compensação foi acionada e a porcentagem de falso positivo ficou em torno de 2%.

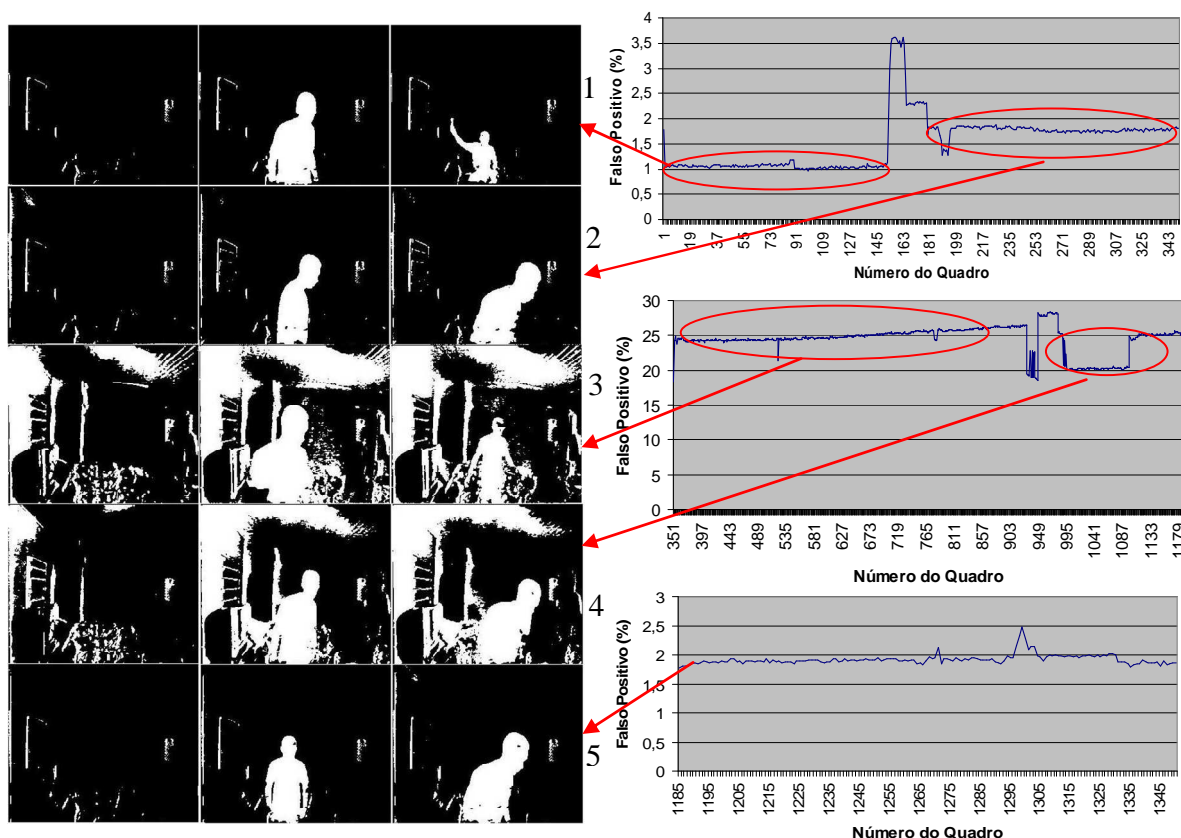


Figura 90: Resultados segmentação realizada com a arquitetura implementada em hardware. 1) Uma das lâmpadas está ligada e a função de compensação de variação do *pixel* está ligada para YCbCr; 2) A outra lâmpada foi ligada e a função de compensação contínua ligada; 3) As duas lâmpadas estão ligadas e a função de compensação está desligada para a componente Y; 4) Uma das lâmpadas foi desligada e a função de compensação contínua desligada; 5) idem ao 1).

Na Figura 91 estão exibidos os resultados obtidos em um outro cenário. Neste caso a imagem de referência não contém objetos. As seqüências de imagens estão organizadas da

seguinte maneira: A) imagem de referência com e sem objeto; B), C), e D) são resultados obtidos após a aquisição dos parâmetros da imagem de referência. Na Figura 90-D, os pontos em preto na camisa da pessoa são em função da iluminação do ambiente que gera pontos saturados que são falsamente classificados como pertencentes à imagem de referência. Neste caso, a medida F variou entre 89 a 95%.

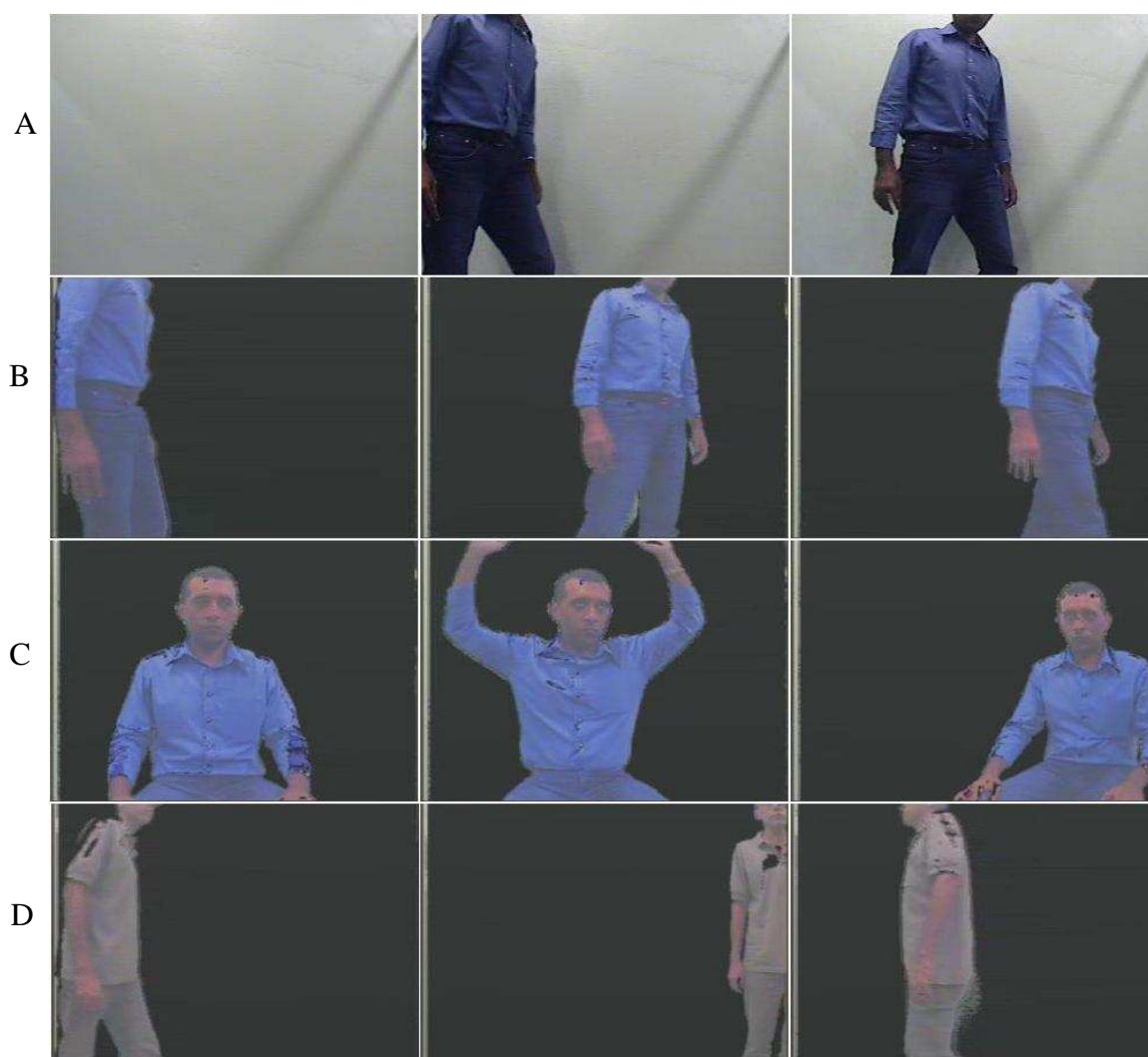


Figura 91: Resultados segmentação realizada com a arquitetura implementada em hardware.

6.3.1 Comparação entre Hardware e MATLAB

A seqüência de imagens ilustradas na Figura 92 foram utilizadas para comparar o desempenho do método de detecção de objetos implementado em MATLAB com os resultados da arquitetura implementada em hardware. As imagens estão organizadas da seguinte maneira: a) imagem de referência com objeto; b) resultado da implementação em MATLAB; c) resultado da implementação em hardware. Os resultados das métricas estão exibidos na Tabela 19. Os desempenhos das implementações em hardware e na arquitetura estão praticamente iguais. Alguns *pixels* do relógio foram classificados como pertencentes à imagem de referência, pois em função da iluminação do ambiente estes *pixels* estão saturados e ocasionaram falsos negativos.

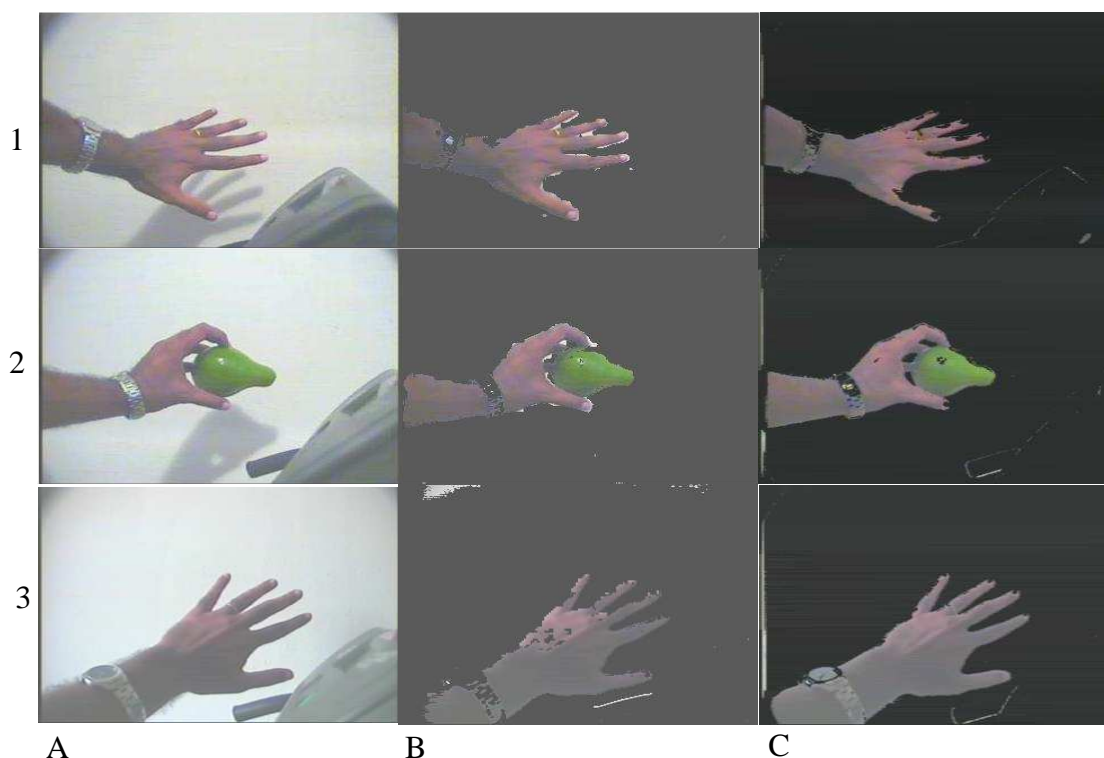


Figura 92: Comparação entre as implementações em Hardware e MATLAB. A) imagem de referência com objeto; B) resultado da implementação em MATLAB; C) resultado da implementação em hardware.

Tabela 19: Resultados das métricas obtidas com as implementações em hardware e MATLAB utilizando-se a seqüência de imagens ilustrada na Figura 92.

Métricas	Seqüência 01		Seqüência 02		Seqüência 03	
	MATLA B	Hardware	MATLA B	Hardware	MATLA B	Hardware
Precisão (P)	84%	81%	85%	83%	73%	80%
Revocação (R)	82%	78%	89%	87%	96%	92%
Medida F	83%	80%	87%	85%	83%	86%

6.3.2 Detecção de Sombras

Na Figura 93 está ilustrado o resultado obtido com a arquitetura em hardware em relação à detecção de sombras. As imagens estão organizadas da seguinte maneira: a) imagem de referência com objeto; b) resultado sem detecção de sombras; c) resultado com a detecção de sombras. Os resultados das métricas estão exibidos na Tabela 20. Na Figura 93-b observa-se que o objeto foi detectado juntamente com a sombra. No entanto, a intensidade da sombra detectada é inferior à projetada na imagem de referência (Figura 93-a) devido à função de compensação de variação do *pixel*. Na Figura 93-c, a sombra foi completamente eliminada.

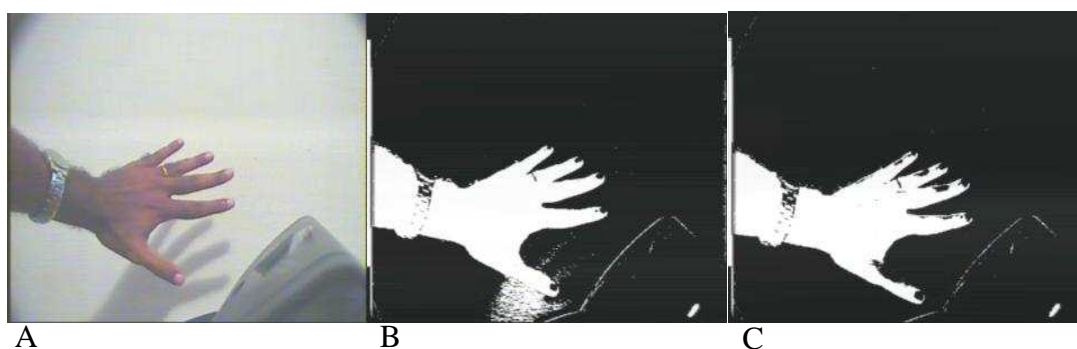


Figura 93: Resultados da detecção de sombras obtidos com a implementação em hardware. A) imagem de referência com objeto; b) resultado sem detecção de sombras; c) resultado com a detecção de sombras.

Tabela 20: Resultados das métricas obtidas com a implementação em hardware utilizando-se a seqüência de imagens ilustrada na Figura 93.

Métricas	Hardware	
	Com Sombra	Sem Sombra
Precisão (P)	76%	81%
Recuperação (R)	63%	78%
Medida F	69%	80%

6.3.3 Intervalos de confiança

Os pontos brancos na Figura 94-b são os *pixels* pertencentes aos intervalos de confiança calculados para a imagem de referência ilustrada na Figura 94-a. Neste caso, 86% dos *pixels* estão com seus valores compreendidos nos seus respectivos intervalos de confiança. Quando um objeto é inserido no cenário (Figura 94-c), a porcentagem reduz para 15% em função da variação não somente dos valores das intensidades dos *pixels* pertencentes

ao objeto, mas também daqueles ao seu redor. Na Figura 94-c os pontos pretos são os *pixels* cujas intensidades estão fora dos seus respectivos intervalos de confiança. Isto demonstra que os intervalos de confiança podem ser utilizados para detectar os *pixels* cujas intensidades estão com uma variação acima ou abaixo dos valores esperados no modelo da imagem de referência.



Figura 94: Resultados da identificação dos *pixels* pertencentes aos intervalos de confiança obtidos com a implementação em hardware. A) imagem de referência sem objetos; b) os pontos brancos são os *pixels* pertencentes aos intervalos de confiança sem objetos na imagem de referência; c) os pontos brancos são os *pixels* pertencentes aos intervalos de confiança com objetos (mão) na imagem de referência;

6.4 RESULTADOS DA SÍNTESE

Nesta seção são apresentados os resultados da síntese obtidos para a arquitetura proposta neste trabalho. A arquitetura foi implementada em um kit educacional e de desenvolvimento da Altera denominado DE-2. Este kit tem um FPGA da família Cyclone II (EP2C35F672C6) com 33.000 elementos lógicos. A arquitetura proposta foi modelada com a linguagem de descrição de hardware Verilog. A ferramenta de simulação utilizada foi o ModelSim da Mentor Graphics e o Quartus II da Altera foi utilizado para síntese e leiaute. A arquitetura implementada pode processar 30 quadros por segundo e uma imagem com resolução igual a 640 x 507 *pixels*. Na Tabela 21 estão os resultados da síntese para cada bloco da arquitetura.

Tabela 21: Resultados da síntese

Blocos FPGA	Total
Elementos Lógicos	4.833 (14,63 %)
Registradores	2758
Bits de Memória	54.208 (11 %)
Multiplicadores Embarcados	36 (51 %)
PLLs	1 (25%)
Blocos Segmentação Vídeo	Número de Elementos Lógicos
Média Aritmética	391 (8 %)
Diferença Absoluta	228 (4,7 %)
Detecção de Sombras e Compensação das Variações dos <i>Pixels</i>	1388 (28,7 %)
Classificação do <i>Pixel</i>	313 (6,4 %)
Gerenciamento do Sistema	33 (0,68 %)
Interface SRAM	61 (1,2 %)
Desvio Padrão	337 (6,9 %)
Detecção da Variação Global	460 (9,5 %)
Seleção Automática dos Limiares	200 (4,1 %)
Interface SDRAM	705 (14,5 %)
Interface Decodificador de Video	160 (3,3 %)
Interface Codificador VGA	349 (7,2 %)
Interface I2C	88 (1,8 %)
Controle do Buffer de Quadro	120 (2,4 %)
Frequência de Operação	
Interface Decodificador Video	27 MHz
Interface Codificador VGA	27 MHz
Interface I2C	27 MHz
Unidade Segmentação de Video	100 MHz
Interface SRAM	100 MHz
Interface SDRAM	100 MHz

Na Figura 95 é exibido o *floorplane* do FPGA utilizado para implementar a arquitetura para segmentação de vídeo previamente apresentada. Os elementos lógicos utilizados estão destacados em amarelo.

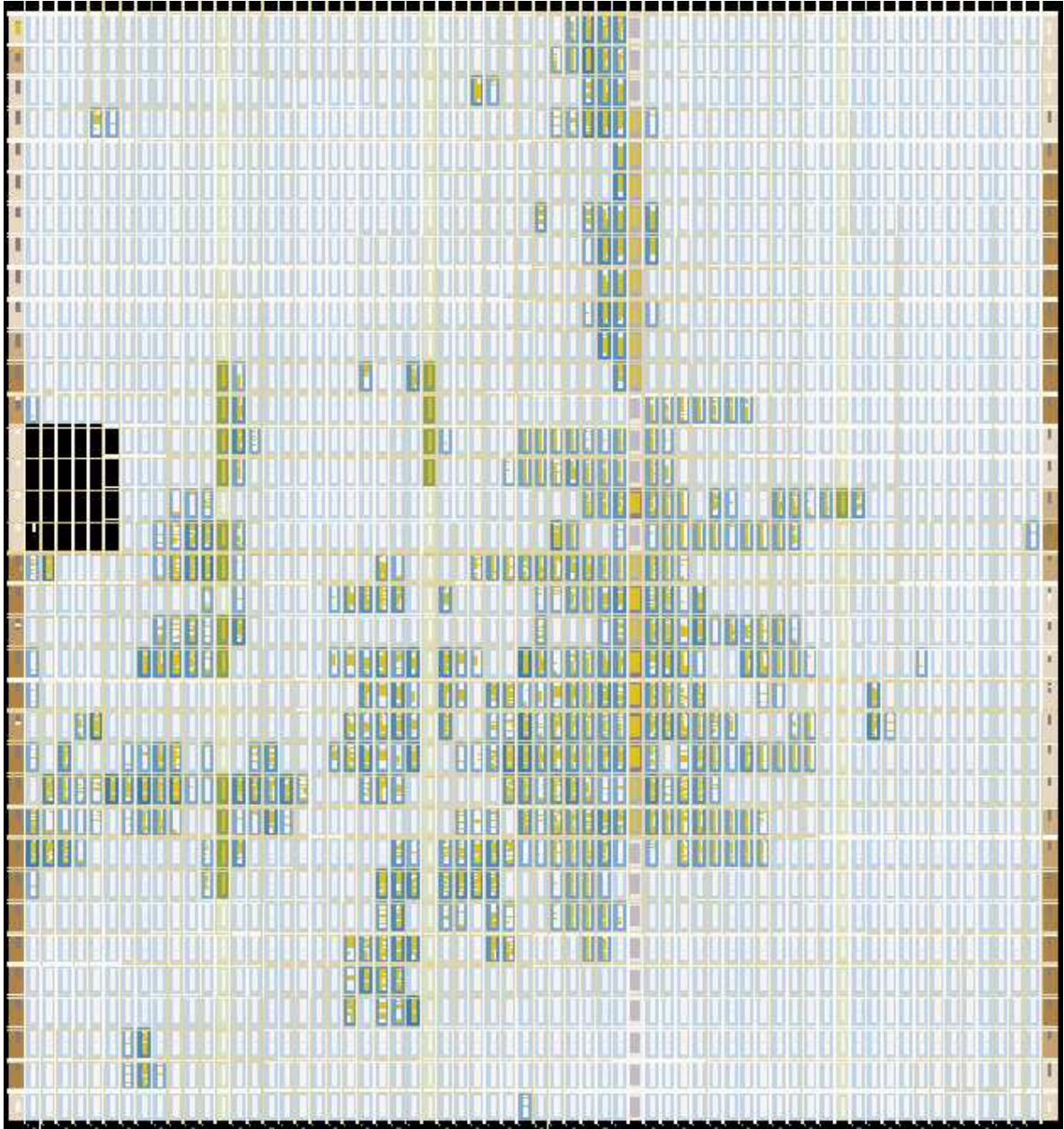


Figura 95: “Floorplane” do método de detecção de objetos mapeado no FPGA do kit DE-2.

6.5 CONCLUSÕES

Neste capítulo apresentou-se a arquitetura utilizada para implementar o método de segmentação de vídeo para detecção de objetos com as adaptações apresentadas no capítulo 4. As etapas delineadas para o desenvolvimento do sistema foram as seguintes: definição da arquitetura, modelagem em linguagem de descrição de hardware, verificação funcional e com atrasos, síntese lógica e leiaute. O kit DE-2 da Altera foi utilizado para implementação da arquitetura. A arquitetura implementada em hardware processa os elementos de imagens enviados pelo decodificador de vídeo sem armazená-los em *buffers*, o que reduz a quantidade de elementos lógicos. A memória SRAM do kit é utilizada para armazenar os valores médios de cada *pixel*. Apesar dos intervalos de confiança fazerem parte dos parâmetros do modelo da imagem de referência, eles são calculados durante a fase de classificação dos elementos de imagens e não são armazenados na memória externa. Isto também reduz o número de acessos à memória.

A arquitetura foi testada em um cenário com variações de iluminação e a utilização da função de compensação mostrou uma redução no número de falsos positivos ocasionados pelas variações de iluminação no ambiente, isto é, por lâmpadas que foram ligadas/desligadas após a aquisição do modelo da imagem de referência.

O desempenho da arquitetura também foi comparado com a implementação em MATLAB. Os resultados obtidos demonstraram que as adaptações realizadas visando à otimização da arquitetura e as operações em ponto fixo não comprometeram o desempenho do método de detecção de objetos implementado em hardware.

CONCLUSÃO E TRABALHOS FUTUROS

Os objetivos delineados neste trabalho foram os seguintes: estabelecer um método de detecção de objetos considerando uma imagem de referência estática e câmera fixa com tolerância à sombras e variações de iluminação para aplicações em tempo real (30 quadros por segundo) e implementar em hardware o método de detecção de objetos utilizando um dispositivo de lógica programável.

Para atingir os objetivos estabelecidos, primeiramente desenvolveu-se o método de segmentação de vídeo e detecção de objetos utilizando-se um modelo gaussiano com operações em ponto flutuante implementadas o MATLAB. Após a validação do método com operações em ponto flutuante, foram feitas modificações para execução de operações em ponto fixo. A partir da versão do método proposto em ponto fixo, foram feitas algumas simplificações para redução da quantidade de memória necessária para captura e armazenagem do modelo da imagem de referência. Após a validação do método com operações em ponto fixo, elaborou-se a arquitetura de hardware necessária para implementação do método proposto utilizando-se um dispositivo de lógica programável. Os resultados obtidos e as sugestões de trabalhos futuros são descritos a seguir.

7.1 RESULTADOS

As principais contribuições desta tese ao estado da arte são as seguintes: aplicação de métodos estatísticos para avaliação da segmentação; desenvolvimento de um método para compensação das variações das intensidades dos *pixels*; aplicação de estimativas estatísticas para determinação do número de quadros; implementação de uma arquitetura de hardware para detecção de objetos. A seguir cada uma destas contribuições é descrita e comparada com o encontrado nos artigos pesquisados ao longo do desenvolvimento deste trabalho.

7.1.1 Aplicação de Métodos Estatísticos para Avaliação da Segmentação

Os métodos de avaliação encontrados na literatura são baseados em métricas que levam em consideração o número de falsos positivos e negativos obtidos para cada quadro processado. Estas métricas são denominadas revocação (*Recall*), precisão (*Precision*), e medida F (*F-measure*). Nos artigos pesquisados são calculados os valores médios de cada uma destas métricas para a seqüência de imagens de teste e a análise de desempenho é baseada nos maiores valores obtidos conforme ilustrado na Tabela 22.

Tabela 22: Comparação de métodos para detecção de objetos utilizando uma imagem de referência estática [72].

Algorithm	Mean in % F measure	Recall	Precision
Image Differencing	69	73	75
Cavallaro	84	95	77
Shen	88	84	93
Haritaoglu	65	87	55
Francois	79	92	70
McKenna	81	83	80
Hong	79	77	83
Jabri	83	96	73
Horprasert	84	85	83

Analisando os resultados obtidos na Tabela 22 é relativamente subjetivo, por exemplo, afirmar que o método de Horprasert é melhor que o de Hong, apesar dos resultados obtidos com as métricas, pois não podemos inferir que uma diferença de 5% na medida-F seja realmente significativa.

A fim de comparar os resultados obtidos em cada cenário de testes e as diversas variáveis envolvidas em cada avaliação apresentada neste trabalho, foram utilizados métodos não paramétricos de análise de variância para verificar se as diferenças observadas com as métricas de revocação, precisão e medida-F eram estatisticamente significantes.

Utilizando-se os métodos de inferência estatística foi possível validar objetivamente o método proposto com diferentes estratégias para detecção de objetos e em quatro planos de cores. Além disso, efetuou-se também comparações entre as versões implementadas em ponto flutuante e em ponto fixo visando a arquitetura em hardware. Portanto, o procedimento de avaliação proposto assegura uma melhor análise de desempenho da detecção dos objetos e migração da implementação em MATLAB para uma arquitetura em hardware.

7.1.2 Desenvolvimento do Método para Compensação das Variações dos *Pixels*

Um dos métodos mais utilizados nos últimos dez anos para detecção de objetos é o denominado método estatístico paramétrico gaussiano conforme ilustrado na Figura 96. Considerando-se imagens coloridas com uma resolução igual a 640×480 *pixels*, uma taxa de 30 quadros por segundo, e 8 bits para representação de cada *pixel*, o tamanho da memória necessário para armazenar o modelo da imagem de referência baseado em modelos gaussianos é igual a 1,8 MBytes.

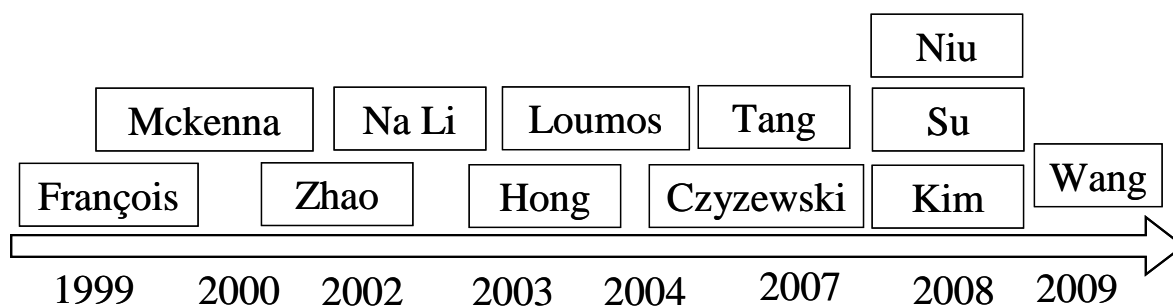


Figura 96: Autores de métodos para detecção de objetos que modelam a imagem de referência com um modelo gaussiano.

O método para detecção de objetos apresentado neste trabalho também modela a imagem de referência segundo um modelo gaussiano. No entanto, as simplificações propostas e os resultados obtidos mostram que é possível reduzir o tamanho da memória necessária para armazenar o modelo da imagem de referência de 1,8 MBytes para 486,723 Kbytes sem comprometer o desempenho da segmentação. Isto foi obtido com a redução da resolução horizontal de 640 para 320 *pixels* apenas para cálculo dos valores médios de cada *pixel* e a utilização de um desvio padrão global para cada componente de cor YCbCr. Estas simplificações foram avaliadas com as métricas de revocação, precisão e medida-F juntamente com os métodos de inferência estatística. Em alguns cenários de teste não foram encontradas diferenças estatisticamente significantes entre as versões com e sem as simplificações.

A redução da resolução da imagem é uma das técnicas utilizadas por outros autores visando atender aos requisitos de tempo real das aplicações conforme ilustrado na Figura 97. No entanto, a redução da resolução horizontal proposta neste trabalho é apenas para a obtenção dos valores médios de cada *pixel*, ou seja, durante a etapa de segmentação são processados 640×480 *pixels* a fim de garantir uma resolução final igual a das imagens de entrada do sistema de detecção de objetos.

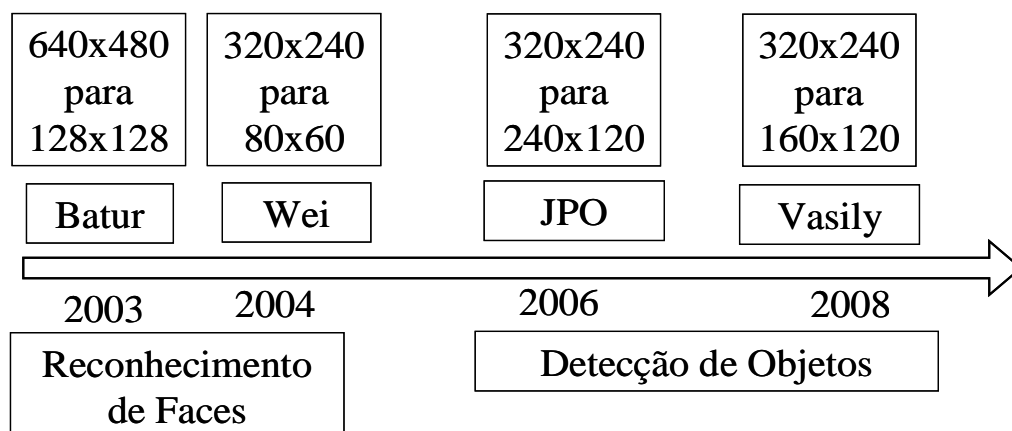


Figura 97: Exemplos de aplicações que efetuam a redução da resolução da imagem para atender aos requisitos de tempo real.

A técnica denominada atualização recursiva tem sido utilizada nos últimos onze anos para compensar as variações de iluminação e as mudanças de objetos na imagem de referência conforme ilustrado na Figura 98. A principal desvantagem desta técnica é que objetos que estejam se movimentando com uma velocidade inferior a taxa de atualização são incorporados ao modelo da imagem de referência.

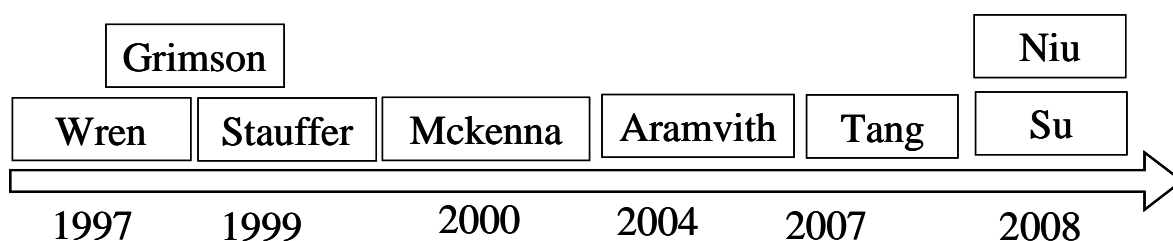


Figura 98: Autores que utilizam a adaptação recursiva para compensar as variações de iluminação.

Neste trabalho foi apresentado um método para detecção de objetos baseado em modelos gaussianos com compensação das variações das intensidades dos *pixels* a fim de tolerar as variações de iluminação do cenário e o ruído gerado pelo sistema de aquisição das imagens. A utilização desta compensação em aplicações em que a imagem de referência é estática dispensa a atualização contínua do modelo da imagem de referência. O modelo da imagem de referência é formado pelos valores médios e desvios padrões e os respectivos intervalos de confiança para cada *pixel*. Antes da etapa de detecção dos objetos, o valor corrente de cada *pixel* é comparado com o seu respectivo intervalo de confiança e os que são identificados fora deste intervalo tem a sua diferença absoluta reduzida por uma função

definida empiricamente. Os resultados experimentais demonstraram uma redução na detecção de falsos positivos em ambientes com variação de iluminação. De acordo com os testes realizados, para uma variação média global de luminância igual a 25%, a porcentagem de falso positivo é reduzida de 70% para 12% com a utilização da função de compensação de variação do *pixel*.

Em alguns cenários de teste a utilização da função de compensação do *pixel* sem a detecção de sombras produziu os melhores resultados. Isto significa que para aplicações em as sombras não sejam críticas, pode-se eliminar a etapa de processamento para detecção de sombras.

7.1.3 Aplicação de Estimativas Estatísticas para Determinação do Número de Quadros

Nos artigos pesquisados ao longo deste trabalho não foi encontrado como o número de quadros é determinado para estimar os valores médios e os desvios padrões de cada *pixel*. Neste trabalho utilizou-se o teorema do limite central para determinar um número mínimo de quadros levando-se em consideração um compromisso entre o erro e o grau de confiança da estimativa.

A determinação do número de quadros é um ponto crucial para determinação das estimativas pontuais e baseadas em intervalos de confiança dos valores médios das intensidades de cada *pixel*. Além disso, o número de quadros também determina o tempo necessário para aquisição do modelo da imagem de referência que com base nestas estimativas foi possível reduzir para menos de 1 segundo.

7.1.4 Implementação da Arquitetura de Hardware para Detecção de Objetos

O método proposto para detecção de objetos foi validado em MATLAB com operações em ponto fixo e, em seguida, foi implementada uma arquitetura em hardware. As adaptações efetuadas para conversão das operações do método de detecção de objetos para ponto fixo não comprometeram o desempenho do método. Isto demonstra que a metodologia utilizada para conversão e validação das operações em ponto fixo foi adequada.

A arquitetura foi organizada de tal modo que cada bloco de hardware recebe um conjunto de parâmetros de entrada e disponibiliza na saída o resultado do processamento juntamente com os parâmetros recebidos na entrada. Desta maneira, não foi necessário utilizar *buffers* para armazenamento dos resultados intermediários em cada etapa do processamento.

Cada bloco processa simultaneamente as três componentes do espaço de cores YCbCr. A arquitetura em hardware processa 30 quadros por segundo e cada quadro tem uma resolução igual a 640 x 507 *pixels*. A frequência de operação utilizada foi igual a 100 MHz.

Na Figura 99 estão descritas as velocidades de processamento de algumas arquiteturas implementadas em hardware para detecção de objetos. Comparando-se com os resultados obtidos nesta pesquisa, observa-se uma ligeira superioridade de processamento da arquitetura proposta neste trabalho em relação ao estado da arte.

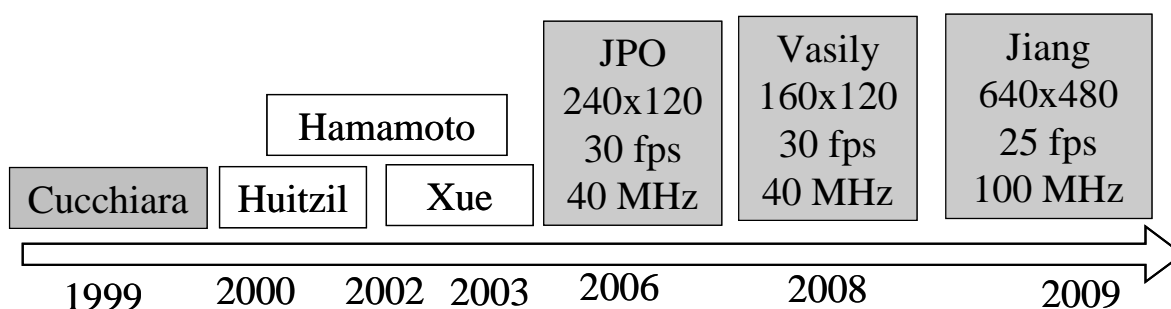


Figura 99: Velocidade de processamento de arquiteturas implementadas em hardware para detecção de objetos.

Mediante os resultados obtidos, constatou-se a tese de que é possível compensar as variações dos *pixels* em tempo real (30 quadros por segundo) sem a atualização do modelo da imagem de referência. As técnicas de atualização podem ser inadequadas para aplicações que visam à inserção do cantor em um videoclipe, pois uma pessoa parada é integrada à imagem de referência, não sendo detectada como objeto. Portanto, as aplicações de segmentação de vídeo para detecção de objetos que utilizam uma imagem de referência estática podem utilizar o método apresentado ao longo desta tese para evitar que o desempenho do sistema implementado seja dependente da velocidade de deslocamento dos objetos em cena e da taxa de atualização da imagem de referência.

7.2 TRABALHOS FUTUROS

O método desenvolvido ao longo desta pesquisa obteve melhores resultados com imagens de referência sem objetos. Por isso deve-se melhorar a detecção para ambientes internos com objetos em cena. Além disso, a função de compensação da variação do *pixel* mostrou-se eficiente na redução de falsos positivos em cenários cuja variação de brilho não seja superior a 30%. Deve-se estudar alguma técnica de reinicialização automática do modelo da imagem de referência. Outro aspecto a ser pesquisado é a aplicação em ambientes

externos. Para isso, pode-se utilizar a função de compensação em conjunto com um método que utilize mistura de modelos gaussianos. Neste caso, uma sugestão seria utilizar a função de compensação para atenuar os valores correntes dos *pixels* antes da detecção dos objetos.

7.3 PUBLICAÇÕES

A seguir são listadas as publicações efetuadas ao longo do desenvolvimento desta tese.

7.3.1 Artigos Publicados

1. Oliveira, Jozias; Sebastião, Ivan; Melcher, Elmar; Freire, Raimundo C. S.. Static Background Subtraction Using HSV Space. Proceedings of the 19th Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI, CDROM, 1 to 2 pages, October 8 - 11, 2006, Manaus, Brazil.
2. Oliveira, Jozias; Printes, André; Freire, Raimundo; Melcher, Elmar; Sebastião, Ivan. FPGA Architecture for Object Segmentation in Real Time. Proceedings of the 14th European Signal Processing Conference - EUSIPCO, CDROM, 1 to 4 pages, September 4 - 8, 2006, Florence, Italy.
3. Oliveira, Jozias; Printes, André; Freire, R.C.S; Melcher, Elmar; Silva, Ivan S. S..”FPGA Architecture for Static Background Subtraction in Real Time”. Proceedings of the 19th Annual Symposium on Integrated Circuits and Systems Design, pages 26-31, 2006.

7.3.2 Artigo Submetido

1. Oliveira, Jozias; Freire, R.C.S; Melcher, Elmar; .” FPGA Architecture for Real Time Video-Object Segmentation from Static Background with Lighting Variations”. EURASIP Journal on Advances Signal Processing, Pages 1–24, July 2009.

7.3.3 Patente – PI0204940-6

Titulo: SISTEMA ELETRÔNICO PARA EXTRAÇÃO, NÃO ASSISTIDA, DE OBJETOS EM IMAGENS DINÂMICAS COM FUNDO NÃO HOMOGÊNEO, EM TEMPO REAL.

REFERÊNCIAS BIBLIOGRÁFICAS

-
- 1 BROERS, H. et al. Architecture Study for Smart Cameras. In: *Proceedings of the European Optical Society Conference on Industrial Imaging and Machine Vision*, 2005. p. 39-49.
 - 2 GOLDING, P. *Next Generation Wireless Application*. 2.ed. Jonh Wiley & Sons, 2008. p. 1-24.
 - 3 WONG, P.; and BHAVANA, M. Hardware in Process: Mobile Handset Cameras Challenge Image Processors. *Optical Engineering Magazine*, v. 5, n. 9, p. 15–17, outubro 2005.
 - 4 QURESHI, S. *Embedded Image Processing on the TMS320C6000 DSP: Examples in Code Composer Studio and MATLAB*. Springer Verlag, 2005.
 - 5 BOVIK, A.L. *Introduction to Digital Image and Video Processing. Handbook of Image & Video Processing*. 2. ed. Amsterdam: Elsevier Academic Press, 2005. p. 3-20.
 - 6 DONG, K.; HU, M.; JI, Z.; and FANG, B. Research on Architectures for High Performance Image Processing. In: *Proceedings of the Fourth International Workshop on Advanced Parallel Processing Technologies*, 2001. p. 1-5.
 - 7 HUNTER, H.; and MORENO, J. A New Look at Exploting Data Parallelism in Embedded Systems. In: *Proceedings of the International Conference on Compilers, Architectures, and Synthesis for Embedded Systems*, 2003. p. 159–169.
 - 8 KYO, S.; OKAZAKI, S.; and ARAI, T. An Integrated Memory Array Processor Architecture for Computation of Image Moments. In: *Proceedings of the 32nd International Symposium on Computer Architectures*, 2005. p. 134-145.
 - 9 DOWNTON, A.; CROOKES, D. Parallel Architectures for Image Processing. *Electronics and Communication Engineering Journal*, v. 10, n. 3, p. 139-151, junho 1998.
 - 10 SONIAVY, C. *Embedding Data and Task Parallelism in Image Processing Applications*. The Netherlands: Delf University of Technology, 2003. Tese de Doutorado.
 - 11 GONZALES, R.; WOODS, R. *Digital Image Processing*. 3. ed. Prentice-Hall, 2008.
 - 12 BRAMBERGER, M.; BRUNNER, J.; RINNER, B.; SCHWABACH, H. Real-time Vídeo Analysis on na Embedded Smart Câmera for Traffic Surveillance. In: *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2004. p. 25-28.
 - 13 MAHLKNECHT, S.; OBERHAMMER, R.; NOVAK, G. A Real-Time Image Recognition System for Tiny Autonomous Mobile Robots. In: *Proceeding of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2004. p. 324-330.

-
- 14 DESURMONT, X.; LIENARD, B.; MEESEN, J.; DELAIGLE, J. Real-Time Optimizations for Integrated Smart Network Camera. In: *Proceedings of SPIE-SET Electronic Imaging Conference on Real-Time Imaging*, 2005. p. 85-92.
 - 15 BUTLER, D.E.; BOVE, V.M.; SRIDHARAN, S. Real-Time Adaptive Foreground-Background Segmentation. *EURASIP Journal on Applied Signal Processing*, v. 14, p. 2292-2304, janeiro 2005.
 - 16 HU, J.S.; SU, T.M. Robust Background Subtraction with Shadow and Highlight Removal For Indoor Surveillance. *EURASIP Journal on Advances in Signal Processing*, p. 1-14, 2007.
 - 17 HUSSMANN, S.; HO, T. A High-Speed Subpixel Edge Detector Implementation Inside a FPGA. *Journal of Real-Time Imaging*, v. 9, n. 5, p. 361-368, outubro 2003.
 - 18 UZUN, I.; AMIRA, A. Real-Time 2-D Wavelet Transform Implementation for HDTV Compression. *Journal of Real-Time Imaging*, v. 11, n. 2, p. 151-165, abril 2005.
 - 19 PASCHALAKIS, S.; BOBER, M. Real-Time Face Detection and Tracking for Mobile videoconferencing. *Journal of Real-Time Imaging*, v. 10, n. 2, p. 81-94, abril 2004.
 - 20 JIANG, H.; ARDO, H.; OWALL, V. A Hardware Architecture for Real-Time Video Segmentation Utilizing Memory Reduction Techniques. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 19, n. 02, fevereiro 2009.
 - 21 DAVIES, E. *Machine Vision: Theory, Algorithms, Practicalities*. San Fransisco: Morgan Kauffamann Publishers, 2005.
 - 22 TONIETTO, L. *Análise de Algoritmos para Chroma-Key*. Disponível em: < <http://www.inf.unisinos.br/~marcelow/ensino/tc/ckey/ckey.html> > Acesso em: 10 jan. 2007.
 - 23 PICCARDI, M. Background subtraction techniques: a review. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2005. p. 3099-3104.
 - 24 PARKS, D.H.; FELLS, S.S. Evaluation of Background Subtraction Algorithms with Post-processing. In: *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, 2008. p. 1-6.
 - 25 PANAH, S. et al. Evaluation of Background Subtraction Methods. In: *Digital Image Computing: Techniques and Applications (DICTA)*, 2008. p. 357-364.
 - 26 HARVILLE, M.; GORDON, G.; WOODFILL, J. Foreground Segmentation Using Adaptive Mixture Models in Color and Depth. In: *Proceedings of IEEE Workshop on Detection and Recognition of Events in Video*, 2001. p. 3-11.

-
- 27 CHEN, B.; LEI, Y. Indoor and Outdoor People Detection and Shadow Suppression by Exploiting HSV Color Information. In: *Proceedings of the Fourth International Conference on Computer and Information Technology*, 2005. p. 137–142.
 - 28 ZHANG, J.; ZHANG, L.; TAI, H.M. Efficient Video Object Segmentation Using Adaptive Background Registration and Edge-Based Change Detection. In: *IEEE International Conference on Techniques Multimedia and Expo*, 2004. p. 1467-1470.
 - 29 KARMANN, K. P.; BRANDT, A. V. Moving Object Recognition Using and Adaptive Background Memory. In: *Time Varying Image Processing and Moving Object Recognition*, 1990. p. 1-6.
 - 30 WREN, C; AZARBAYEJANI, A; DARRELL, T; PENTLAND, A. Pfunder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, p.780-785, 1997.
 - 31 STAUFFER, C.; GRIMSON, W.E.L. Adaptive Background Mixture Models for Real-time Tracking. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999. p. 249–252.
 - 32 GRIMSON, W.E.L.; STAUFFER, C; ROMANO, R.; LEE, L. Using adaptive tracking to classify and monitor activities in a site. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998. p. 22-29.
 - 33 THONGKAMWITON, T.; ARAMVITH, S.; CHALIDABHONGSE, T.H. An adaptive Real-Time Background Subtraction and Moving Shadows Detection. In: *IEEE International Conference on Multimedia and Expo*, 2004. p. 1459-1462.
 - 34 STAUFFER, C.; and GRIMSON, W.E.L. Learning Patterns of Activity Using Real Time Tracking. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000. p. 747-757.
 - 35 XUE, L.; RONGCHUN, Z.; QUING, W. FPGA Based Algorithm as Vehicle Edge Detector in VCAS. In: *International Conference on Neural Networks and Signal Processing*, 2003. p. 139–142.
 - 36 HUITZIL, C.T.; ESTRADA, M.A. An FPGA Architecture for High Speed Edge and Corner Detection. In: *Fifth IEEE International Workshop on Computer Architectures for Machine Perception*, 2000. p. 112–116.
 - 37 HAMAMOTO, T.; NAGAO, S.; AIZAWA, K. Real-time Objects Tracking by Using Smart Image Sensor and FPGA. In: *Proceedings of International Conference on Image Processing*, 2002. p. 441-444.
 - 38 CUCCHIARA, R.; ONFIANI, P.; PRATI, P.; SCARABOTTOLO, N. Segmentation of Moving Objects at Frame: A Dedicated Hardware Solution. In: *Seventh International Conference on Image Processing and its Applications*, 1999. p. 138–142.

-
- 39 KIM, J.; CHEN, T. A VLSI Architecture for Video Object Segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, p. 83-96, janeiro 2003.
- 40 REDDY, V.S.K; SENGUPTA, S.; LATHA, Y.M. New VLSI Architecture for Motion Estimation Algorithm. In: *Proceedings of World Academy of Science, Engineering and Technology*, 2007. p. 68-71.
- 41 NIU, L.; JIANG, N. A Moving Objects Detection Algorithm Based on Improved Background Subtraction. In: *Eighth International Conference on Intelligent Systems Design and Applications*, 2008. p. 604-607.
- 42 SU, S.T.; CHEN, Y.Y. Moving Object Segmentation Using Improved Running Gaussian Average Background Model. In: *Digital Image Computing: Techniques and Applications*, 2008. p. 24 – 31.
- 43 WANG, W. The Approach of Real-time Monitoring Based on Background Subtraction. In: *International Symposium on Intelligent Ubiquitous Computing and Education*, 2009. p. 46 – 49.
- 44 KIM, H.; SAKAMOTO, R.; KITAHARA, I.; TORIYAMA, T.; KOGURE, K. Background Subtraction Using Generalised Gaussian Family Model. *IEEE Electronics Letters*, v. 44, issue 3, p. 189 – 190, janeiro 2008.
- 45 TANG, Z.; MIAO, Z.; WAN, Y. Background Subtraction Using Running Gaussian Average and Frame Difference. *Springer Berlin Lecture Notes in Computer Science*, v. 4740, p. 411-414, 2007.
- 46 ALEXANDROPOULOS, T.; LOUMOS, V.; KAYAFAS, E. A block-based clustering technique for real time object detection on a static background. In: *Proceedings of 2nd International IEEE Conference Intelligent Systems*, 2004.
- 47 CZYZEWSKI, A.; DALKA, P. Visual Traffic Noise Monitoring in Urban Areas. *International Journal of Multimedia and Ubiquitous Engineering*, v. 2, p. 91 – 102, 2007.
- 48 HONG, D.; WOO, W. A Background Subtraction for a Vision-Based User Interface. In: *Proceedings of the Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, and the Fourth Pacific Rim Conference on Multimedia*, 2003. p. 263-267.
- 49 ZHAO, M.; BU, J.; CHEN, C. Robust Background Subtraction in HSV Color Space. In: *Proceedings of the International Society for Optical Engineering – Multimedia Systems and Applications V*, 2002. p. 325-332.
- 50 LI, N.; BU, J.; CHEN, C. Real-Time Video Object Segmentation Using HSV Space. In: *Proceedings of International Conference on Image Processing*, 2002. p. 85–88.

-
- 51 MCKENNA, S.J.; JABRI, S.; DURIC, Z.; ROSENFELD, A.; WECHSLER, H. Tracking Groups of People. *Computer Vision and Image Understanding*, v. 80, p. 42-56, 2000.
- 52 FRANÇOIS, A.R.J.; MEDIONI, G.G. Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams. In: *Proceedings of the International Conference on Imaging Science, Systems, and Technology*, 1999. p. 227-232.
- 53 Ying-Li Tian; Lu, M.; Hampapur, A. Robust and efficient foreground analysis for real-time video surveillance. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005. pp. 1182 – 1187.
- 54 Niu, L.; Jiang, N. A Moving Objects Detection Algorithm Based on Improved Background Subtraction. In: *Eighth International Conference on Intelligent Systems Design and Applications*, 2008. p. 604-607.
- 55 FRIEDMAN, N.; RUSSELL, S. Image Segmentation in Video Sequences: A Probabilistic Approach. In: *Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997. p. 175-181.
- 56 ZIVKOVIC, Z. Improved Adaptive Gaussian Mixture Model for Background Subtraction. In: *Proceedings of the 17th International Conference on Pattern Recognition*, 2004. p. 28-31.
- 57 LEE, D.S. Effective Gaussian Mixture Learning for Video Background Subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 5, maio 2005.
- 58 TANG, Z.; MIAO, Z. Fast Background Subtraction and Shadow Elimination Using Improved Gaussian Mixture Model. In: *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, 2007. p. 38-41.
- 59 FEI, L.; XUELI, W.; DONGSHENG, C. Drowning Detection Based on Background Subtraction. In: *International Conference on Embedded Software and Systems*, 2009. p.341-343.
- 60 JIA, L.; LIU, Y. A Novel Thresholding Approach to Background Subtraction. In: *wacv, IEEE Workshop on Applications of Computer Vision*, 2008. p.1-6.
- 61 HORPRASERT, T.; HARWOOD, D.; DAVIS, L.S. A Statistical Approach for Real Time Robust Background Subtraction. In: *IEEE ICCV'99 FRAME-RATE WORKSHOP*, 1999. p. 1-19.
- 62 ZHANG, R.; ZHANG, S.; YU, S. Moving Objects Detection Method Based on Brightness Distortion and Chromaticity Distortion. *IEEE Transactions on Consumer Electronics*, v. 53, p. 1177 - 1185, august 2007.
- 63 ZANG, Q.; KLETTE, R. Parameter Analysis for Mixture of Gaussians Model. *Communication and Information Technology Research Technical Report 188*, 2006.

-
- 64 JACQUES, C. S.; JUNG, J. R. ; MUSSE, C.S.R. A Background Subtraction Model Adapted to Illumination Changes. In: *IEEE International Conference on Image Processing*, 2006. p. 1817–1820.
- 65 IBRAHIM, K. F. *Newnes Guide to Television and Video Technology*. Elsevier, 2007. p. 1-12.
- 66 WANG, Y.; OSTERMANN, J.; ZHANG, Y.Q. *Video Processing and Communications*, Prentice-Hall, 2002. p. 2-28.
- 67 CHERNICK, M.R.; FRIIS, R.H. *Introductory Biostatistics for the Health Science – Modern Applications Including Bootstrap*. New Jersey: JonhWiley & Sons Inc, 2003. p. 121-181.
- 68 PAPOULIS, A. *Probability, Random Variables, and Stochastic Process*. 3.ed. Mc-Graw-Hill, 1991.
- 69 MONTGOMERY, D.C.; RUNGER, G.C. *Estatística Aplicada e Probabilidade para Engenheiros*, 2.ed. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 2003. p. 15-17.
- 70 TOYAMA, K. et al. Wallflower: Principles and Practice of Background Maintenance. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999. p. 255-261.
- 71 BRANCA, A. et al. Human Motion Tracking in Outdoor Environment. In: *Proceedings of Seventh International Conference on Control, Automation, Robotics, and Vision*, 2002. p. 1585 – 1590.
- 72 KARAMAN, M. et al. Comparison of Static Background Segmentation Methods. In: *Proceedings of the SPIE*, 2005. p. 2140-2151.
- 73 CHEUNG, S.C.S.; KAMATH, C.; Robust Background Subtraction with Foreground Validation for Urban Traffic Video. *EURASIP Journal on Applied Signal Processing*, v. 14, p. 2330 – 2340, 2005.
- 74 LEVINE, D.M. et al. *Statistics for Managers Using Microsoft Excel and Student CD Package*, 5.ed. Prentice Hall, 2007. p. 430 – 431.
- 75 PAGANO, M.; GAUVREAU, K. *Princípios de Bioestatística*. Traduzido por Luiz Sérgio de Castro Paiva. Thomson, 2006.
- 76 HOLLANDER, M.; WOLFE, D.A. *Nonparametric Statistical Methods*, 2.ed. Wiley-Interscience, 1999. p. 190 – 201.
- 77 RUBLIK, F. On Rank Based Multiple Comparisons for the Balance Sample Sizes. *Journal of the Institute of Measurement Science*, v. 5, section 1, 2005.

-
- 78 **StatsDirect Statistical Software**. Disponível em: < <http://www.statsdirect.com/> > Acesso em: 10 out. 2008
- 79 CONOVER, W.J. **Practical Nonparametric Statistics**, 3.ed. PwsKent, 1990. p. 300-302.
- 80 PORIKLI, F. Computationally Efficient Histogram Extraction for Rectangular Image Regions. In: **Proceedings of Electronic Imaging Conference on Real-Time Imaging SPIE**, 2005. p. 36-43.
- 81 NISHIKAWA, Y.; KAWAHITO, S.; INOUE, T. Parallel Image Compression Circuit for High-Speed Camera. In: **Proceedings of Electronic Imaging Conference on Real-Time Imaging SPIE**, 2005. p. 111-122.
- 82 KNOWLES, G. Real Time Hardware Architectures for The Bi-Orthogonal Wavelet Transform. In: **Proceedings of Electronic Imaging Conference on Real Time Imaging SPIE**, 2003. p. 102– 109.
- 83 CABELLO, E.; SÁNCHEZ, M.; DELGADO, J. A New Approach to Identify Big Rocks with Application to the Mining Industry. **Journal of Real-Time Imaging**, v. 8, n. 1, p. 1-9, february 2002.
- 84 WEI, M.; BIGDELI, A. Implementation of a Real-Time Automated Face Recognition System For Portable Devices. In: **Proceedings of the IEEE International Symposium on Communication and Information Technologies**, 2004. p. 89 – 92.
- 85 BATUR, A.; FLINCHBAUGH, B.; HAYES, M. A DSP-Based Approach for the Implementation of Face Recognition Algorithms. In: **Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing**, 2003. p. 253-256.
- 86 TSAP, L. Gesture-Tracking in Real Time with Dynamic Regional Range Computation. **Journal of Real-Time Imaging**, v. 8, n. 2, p. 115-126, abril 2002.
- 87 UZUN, I.; AMIRA, A. Real-Time 2-D Wavelet Transform Implementation for HDTV Compression. **Journal of Real-Time Imaging**, v. 11, n. 2, p. 151-165, abril 2005.
- 88 ATSALAKIS, A. et al. A. Colour Quantization Technique Based on Image Decomposition and its Embedded System Implementation. In: **Proceedings of IEE Vision, Image, and Signal Processing**, 2004. p. 511–524.
- 89 OLIVEIRA, J.P. et al. FPGA Architecture For Static Background Subtraction in Real Time. In: **Proceedings of the 19th Annual Symposium on Integrated Circuits and Systems Design**, 2006. p. 26-31.
- 90 PATEL, K. Porting PC Based Algorithms to DSPs. In: **Embedded Edge**, abril 2003.
- 91 PATEL, K. Porting and Optimization Techniques for C++ Based Image Processing Algorithms on TMS320C6x DSP. In: **TI Developer Conference**, fevereiro 2004.

-
- 92 JIANLAI, W. et al. Implementation of Otsu's Thresholding Process Based on FPGA. In: *4th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2009. p. 479–483.
- 93 MAZZETO, M. et al. Marcapasso Brasileiro MPB1 – Avaliação das Funções de Captação e Estimulação Elétrica. In: *21º Congresso Brasileiro de Engenharia Biomédica*, 2008. p. 1-4.
- 94 MORIMOTO, T. et al. An FPGA-Based Region-Growing Video Segmentation System with Boundary-Scan-Only LSI Architecture. In: *IEEE Asia Pacific Conference on Circuits and Systems*, 2006. p. 944–947.
- 95 MACLEAN, W. An Evaluation of the Suitability of FPGAs For Embedded Vision Systems. In: *IEEE Computer Society Conference on Computer Vision Pattern Recognition*, 2005, p. 131–131.
- 96 RATHA, N.; JAIN, A. Computer Vision Algorithms on Reconfigurable Logic Arrays. *IEEE Transactions on Parallel Distribution and Systems*, v. 10, n. 1, p. 29–43, janeiro 1999.
- 97 Altera development and education board DE-2. Available: <http://www.altera.com/education/univ/materials/boards/unv-de2-board.html>.
- 98 Intersil Application Note: BT.656 Video Interface for ICs. <http://www.intersil.com/data/an/an9728.pdf>