



WILSON MITSUO UAQUIDA

Detecção de pessoas não utilizando capacetes de segurança através de sistema de visão artificial

São Paulo – Brasil

2016

WILSON MITSUO UAQUIDA

Detecção de pessoas não utilizando capacetes de segurança através de sistema de visão artificial

Dissertação apresentada ao programa de Pós-Graduação em Automação e Controle de Processos do Instituto Federal de Educação, Ciência e tecnologia de São Paulo, como requisito para a obtenção do título de mestre.

Orientador: Prof. Dr. Ricardo Pires

São Paulo – Brasil

2016

U11a Uaquida, Wilson Mitsuo.

 Detecção de pessoas não utilizando capacetes de
 segurança através de sistema de visão artificial / Wilson
 Mitsuo Uaquida. -- São Paulo: [s.n.], 2016.
 87 f.: il.

 Orientador: Prof. Dr. Ricardo Pires.

 Dissertação (Mestrado Profissional em Automação e Controle
 de Processos) - Instituto Federal de Educação, Ciência e
 Tecnologia de São Paulo, IFSP, 2016.

 1 Visão artificial 2. Segmentação por movimento
 3. Descrição por histogramas 4. Comparação de histogramas
 I. Instituto Federal de Educação, Ciência e Tecnologia de São
 Paulo. II. Título

CDU 681.0



ATA DE EXAME DE DEFESA DE DISSERTAÇÃO

Nome do Programa: **Mestrado Profissional em Automação e Controle de Processos**

Nome do(a) Aluno(a): Wilson Mitsuo Uaquida

Nome do Orientador: Prof. Dr. Ricardo Pires

Suplente do orientador:

Título do Trabalho: "Detecção de pessoas não utilizando capacetes de segurança através de sistema de visão artificial"

Abaixo o resultado de cada participante da Banca Examinadora

Nome completo dos Participantes Titulares da Banca	Sigla da Instituição	Aprovado / Não Aprovado
Prof. Dr. Ricardo Pires – Orientador	IFSP – SPO	<i>aprovado</i>
Profª. Dra. Jussara Pimenta Matos – Membro Interno	IFSP – SPO	<i>APROVADO</i>
Prof. Dr. Paulo Batista Lopes – Membro Externo	Mackenzie	<i>aprovado</i>
Nome completo dos Participantes Suplentes da Banca	Sigla da Instituição	Aprovado / Não Aprovado
Prof. Dr. Alexandre Brincalepe Campo – Membro Interno	IFSP - SPO	—
Prof. Dr. Francisco Martins Portelinha – Membro Externo	UNIFEI	—

Considerando-o: APROVADO
 NÃO APROVADO

Assinaturas

São Paulo, 07 de outubro de 2016

R. Pires

Presidente da Banca

[Signature]

Membro Interno

[Signature]

Membro Externo

Observações:

Agradecimentos

Gostaria de expressar meus agradecimentos a Deus pela abertura desta porta e de tantas outras que tem me proporcionado, pela Sua Presença em minha vida e pela Sua Proteção.

Agradeço muitíssimo a minha esposa Celina e ao meu filho William pelo amor, paciência e principalmente pelo apoio que deram para seguir sempre em frente.

Agradeço imensamente ao Instituto Federal de São Paulo pela oportunidade concedida.

Ao meu orientador Ricardo Pires, minha eterna gratidão pelo acompanhamento diligente deste trabalho e pelas palavras de incentivo, que contribuíram muito para a minha formação.

A Igreja Metodista Livre de Santana e a empresa Ale Art's pela disponibilidade e cessão de filmagens em seus domínios.

Aos amigos, colegas e a todos aqueles que por meio de discussões, sugestões e comentários colaboraram para a realização deste trabalho.

Muito obrigado,

Wilson

“Uma imagem vale mais que mil palavras”

Confúcio, literalmente Mestre Kong, foi um pensador e filósofo chinês do Período das Primaveras e Outonos.

Resumo

No Brasil, acidentes de trabalho envolvendo a cabeça e o crânio são comuns em muitas das atividades econômicas aqui desenvolvidas. De acordo com o Anuário Estatístico da Previdência Social, houve 24.561 acidentes desse tipo em 2014, representando cerca de 3,4% do total de acidentes de trabalho no país. O Anuário não revela detalhes, o que se leva a conjecturar que, pelo menos em parte, ocorreu devido a não observância quanto ao uso de equipamentos de proteção. A proposta deste trabalho é detectar automaticamente pessoas que não estão utilizando capacetes de segurança em local onde são obrigatórios. Para este fim, foi utilizado um sistema de visão artificial, cujo algoritmo é composto de uma série de etapas. A primeira etapa consiste na aquisição da imagem com filmagens. A segunda etapa, denominada segmentação, identifica imagens em movimento e delimita uma subimagem da cabeça das pessoas. A terceira e a quarta etapa referem-se a aplicação de filtros e mudança do sistema de cores. Na quinta etapa, denominada descrição, a subimagem é convertida em dados através de histograma de cores. Na última etapa, da classificação por comparação de histogramas, as pessoas sem capacetes são assinaladas por um retângulo. Os resultados de detecção atingiram valores maiores que 99,0%.

PALAVRAS-CHAVE: visão artificial; segmentação por movimento; descrição por histogramas; comparação de histogramas.

Abstract

In Brazil, industrial accidents involving the head and skull are common in many of the economic activities developed here. According to the Statistical Yearbook of Social Security, there were 24,561 such accidents in 2014, representing about 3.4% of all work accidents in the country. The Yearbook does not reveal details, which leads to conjecture that, at least in part, was due to non-compliance regarding the use of protective equipment. The purpose of this work is to automatically detect people who are not using safety helmets in where they are required. To this end, we used an artificial vision system whose algorithm is composed of a series of steps. The first step is the acquisition of the image with shooting. The second stage, called segmentation, identifies moving images and delimits a sub-image of people's heads. The third and the fourth step refers to applying filters and change the color system. In the fifth step, called description, the subpicture is converted into data through color histogram. In the last step, the rating histograms comparison, people without helmets are marked by a rectangle. The detection results reached values greater than 99.0%.

KEYWORDS: artificial vision; segmentation by movement; description by histograms; histogram comparison.

Lista de ilustrações

Figura 1: Escala de cinza (http://hammadk.com/monitor-test/ , 15/10/2015).....	23
Figura 2: Cubo RGB (https://sites.google.com/site/aimciv/modelo-rgb , 15/10/2015)...	25
Figura 3: Modelo de cores HSV (Konwar et al, 2014).....	26
Figura 4: Palmas das mãos em diferentes situações: em local fechado (canto superior esquerdo da figura), ao ar livre sob a sombra (imagem intermediária do lado esquerdo da figura) e ao ar livre sob o sol (canto inferior esquerdo da figura). Nas colunas central e direita são representados também os histogramas RGB e HSV, respectivamente (Bradski e Kaehler (2008), p. 206).....	30
Figura 5: Aplicação da máscara H sobre o pixel da imagem I. Novo valor assumido representado na imagem suavizada I' (Burger e Burge (2008) p. 90).....	32
Figura 6: Máscaras para cálculo do filtro da média. (a) 3 x 3; (b) 5 x 5; (c) 7 x 7 (Marques Filho e Vieira Neto (1999) p. 86).....	33
Figura 7: (a) Imagem original. (b) a (f) Resultados da suavização com filtro de média utilizando máscaras $n \times n$, com $n = 3, 5, 7, 17$ e 31 (Marques Filho e Vieira Neto (1999) p. 87 e 88).....	34
Figura 8: Aplicação de filtros em imagem com ruído (Gonzalez (2010) p. 103).....	35
Figura 9: Determinação da mediana de uma máscara 3 x 3 (Burger e Burge (2008) p. 107).....	36
Figura 10: (a) Imagem original; (b) imagem contaminada por ruído impulsivo (sal e pimenta); (c) resultado da filtragem pelo filtro de mediana com máscara 3x3; (d) resultado da filtragem pelo filtro de média com máscara 3 x 3 (Marques Filho e Vieira Neto (1999) p. 91 e 92).....	37

Figura 11: Representação em 3D do filtro de Gauss e sua respectiva máscara (Burger e Burge (2008) p. 97).....	39
Figura 12: (a) Imagem original; (b) Resultado da aplicação com filtro bilateral; (c) e (d) Detalhes de (a). (Tomasi e Manduchi, 1998).....	42
Figura 13: Efeitos da variação dos parâmetros σ_d e σ_r . (Tomasi e Manduchi, 1998)....	43
Figura 14: fluxo das etapas do algoritmo de visão de máquina. (Fonte: autor).....	48
Figura 15: resumo ilustrativo do algoritmo de visão de máquina. (Fonte: autor).....	59
Figura 16: segmentação por movimento MOG. (Fonte: autor).....	60
Figura 17: segmentação por movimento MOG2. (Fonte: autor).....	61
Figura 18: Geração de imagem de interesse a partir de sub imagem do segmentador MOG2. (Fonte: autor).....	61
Figura 19: exemplo de uma das máscaras utilizadas na pesquisa. (Fonte: autor).....	62
Figura 20: contingência no algoritmo MOG2 – detecção de movimento do tronco separado da cabeça. (Fonte: autor).....	62
Figura 21: imagem segmentada do topo de uma cabeça humana. (Fonte: autor).....	63
Figura 22: imagem segmentada do topo de uma cabeça humana utilizando capacete amarelo. (Fonte: autor).....	63
Figura 23: gráficos representando os histogramas de 45 intervalos de classe com a utilização de filtro bilateral, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha).....	64
Figura 24: gráficos representando os histogramas de 180 intervalos de classe com a utilização de filtro bilateral, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha).....	64
Figura 25: gráficos representando os histogramas de 45 intervalos de classe com a utilização de filtro gaussiano, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).....	66

Figura 26: gráficos representando os histogramas de 180 intervalos de classe com a utilização de filtro gaussiano, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).....	66
Figura 27: gráficos representando os histogramas de 45 intervalos de classe com a utilização de filtro de média, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).....	68
Figura 28: gráficos representando os histogramas de 180 intervalos de classe com a utilização de filtro de média, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).....	68
Figura 29: gráficos representando os histogramas de 45 intervalos de classe com a utilização de filtro de mediana, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).....	70
Figura 30: gráficos representando os histogramas de 180 intervalos de classe com a utilização de filtro de mediana, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).....	70

Lista de tabelas

- Tabela 1: Comparação do histograma H_1 gerado pela metade superior da imagem da palma em local fechado, com os histogramas H_2 referentes às imagens completas das palmas em todas as situações, pelos métodos de correlação, qui quadrado, intersecção e Bhattacharyya (Bradski e Kaehler (2008), p. 206).....**31**
- Tabela 2: comparativo entre os limites HSV da biblioteca OpenCV e a teoria apresentada tópico 2.3.2.....**52**
- Tabela 3: influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro bilateral.....**65**
- Tabela 4: influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro gaussiano.....**67**
- Tabela 5: influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro de média.....**69**
- Tabela 6: influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro de mediana.....**71**
- Tabela 7: influência dos filtros nas magnitudes dos classificadores-comparadores.....**71**
- Tabela 8: resultado da detecção de pessoas que não estão utilizando capacetes e os respectivos falsos positivos, comparando-se os modelos de cabeças com os capacetes amarelos. Foram utilizadas todas combinações possíveis entre intervalos de classe, filtros e classificadores-comparadores.....**73**
- Tabela 9: resultado da detecção de pessoas que não estão utilizando capacetes e os respectivos falsos positivos, comparando-se os modelos de cabeças com os capacetes azuis.

Foram utilizadas todas combinações possíveis entre intervalos de classe, filtros e classificadores-comparadores.....**74**

Tabela 10: resultado da detecção de pessoas que não estão utilizando capacetes e os respectivos falsos positivos, comparando-se os modelos de cabeças com os capacetes brancos. Foram utilizadas todas combinações possíveis entre intervalos de classe, filtros e classificadores-comparadores.....**75**

Lista de abreviaturas e siglas

- 3D - Modelo tridimensional
- BGR - Do inglês: “*blue*” (azul), “*green*” (verde) e “*red*” (vermelho).
- CCD - Do inglês: “*charge-coupled device*” (dispositivo de carga acoplada).
- CIM - Centro Integrado de Monitoramento
- DETECTA - Designação dada ao sistema de monitoramento criminal adquirido pelo Governo do Estado
- EPI's - Equipamentos de Proteção Individual.
- HSV - Do inglês: “*hue*” (matiz), “*saturation*” (saturação) e “*value*” (luminância).
- H - Do inglês: “*hue*” (matiz).
- MOG - Do inglês: “*Mixture of Gaussians*” (Mistura de Gaussianas).
- MOG2 - Do inglês: “*Mixture of Gaussians 2*” (Mistura de Gaussianas versão 2).
- OpenCV - Do inglês: “*Open Computer Vision*”.
- RGB - Do inglês: “*red*” (vermelho), “*green*” (verde) e “*blue*” (azul).
- S - Do inglês: “*saturation*” (saturação).

Lista de símbolos

C++ Uma das linguagens de programação existentes

σ_r Parâmetro de propagação fotométrica do filtro bilateral

σ_d Parâmetro de difusão geométrica do filtro bilateral

Sumário

1	Introdução.....	19
1.1	Formulação do problema.....	19
1.2	Objetivo geral.....	20
1.3	Objetivos específicos.....	20
1.4	Organização deste trabalho.....	20
2	Fundamentos teóricos e revisão bibliográfica.....	22
2.1	Imagem digital.....	21
2.2	Amostragem e quantização.....	23
2.3	Modelo de cores.....	24
2.3.1	O modelo de cores RGB.....	25
2.3.2	O modelo de cores HSV.....	26
2.4	Conceito e operações com histogramas.....	27
2.4.1	Correlação entre dois histogramas.....	27
2.4.2	Qui quadrado entre dois histogramas.....	28
2.4.3	Intersecção entre dois histogramas.....	29
2.4.4	Distância Bhattacharyya entre dois histogramas.....	29
2.4.5	Comparação entre os métodos.....	30

2.5	Filtragem digital	32
2.5.1	Conceito.....	32
2.5.2	Filtro de média.....	33
2.5.3	Filtro de mediana.....	35
2.5.4	Filtro de Gauss.....	38
2.5.5	Filtro bilateral.....	39
2.6	Segmentação	43
2.6.1	Segmentação por movimento.....	44
2.6.1.1	Métodos de segmentação por movimento.....	44
2.7	Etapas de um sistema de visão artificial	46
2.8	Biblioteca OpenCV	47
3	Materiais e Métodos	48
3.1	Materiais	48
3.2	Métodos	48
3.2.1	Aquisição da imagem.....	49
3.2.2	Segmentação.....	50
3.2.3	Conversão do sistema de cores RGB para HSV.....	52
3.2.4	Filtragem.....	53
3.2.5	Descrição – cálculo de histogramas.....	54

3.2.6	Classificação.....	55
3.2.7	Integração filtragem-descrição-classificação.....	58
4	Resultados e discussão.....	60
4.1	Etapa da segmentação.....	60
4.2	Etapa de definição do número de intervalos de classes de histogramas, filtros e classificadores-comparadores.....	63
5	Conclusões e sugestões para trabalhos futuros.....	76
6	Referências bibliográficas.....	78
7	APÊNDICE A - código fonte.....	81

1. Introdução

1.1 Formulação do problema:

Os sistemas de monitoramento por câmeras têm sido amplamente utilizados e difundidos nos últimos anos, principalmente nas áreas de segurança pública e segurança patrimonial. A evolução da tecnologia e a consequente redução de custos operacionais têm contribuído para estes fatos. Porém o fator preponderante é a eficácia do sistema.

Conforme Números (2016), desde que o Centro Integrado de Monitoramento (CIM) foi inaugurado, em 2014, na cidade de São Bernardo do Campo, o sistema tem inibido a ocorrência de crimes. O CIM conta com 400 câmeras e já se tornou ferramenta essencial para a política de segurança pública da cidade. Exemplo da eficiência do sistema são os números apresentados antes e depois do monitoramento por câmeras da Avenida Senador Fláquer, na Vila Euclides. De abril de 2013 a março de 2014, antes do CIM começar a funcionar, foram registrados 28 casos de furtos de veículos na via. Esse número foi reduzido para oito casos de abril de 2014 a março de 2015, ou seja, queda de 71,43%. Já no período de abril 2015 a março de 2016, outra redução, dessa vez para cinco ocorrências, ou de 35,5%. O monitoramento é realizado por operadores de câmeras treinados, que conseguem identificar situações suspeitas de delito.

O governo do Estado de São Paulo também investiu, no ano de 2014, em um sistema de monitoramento inteligente por câmera, denominado DETECTA. Conforme SP (2016), o primeiro sistema DETECTA foi desenvolvido pela empresa Microsoft em parceria com a polícia da cidade de Nova Iorque. A ferramenta foi idealizada para ações de contraterrorismo na cidade americana e passou também a ser utilizada no trabalho contra outros tipos de crimes. Com o DETECTA, são emitidos alarmes automáticos de acordo com os 10 mil padrões de crimes, que foram desenvolvidos durante a experiência em Nova Iorque. Esses alertas foram alterados para a realidade brasileira. Exemplificando: o sistema alerta automaticamente um padrão de suspeita quando um pedestre transita entre veículos parados no congestionamento. Isso permite que as informações de inteligência sejam recebidas sem a necessidade de operar o sistema a todo momento.

Conforme argumentos expostos, as ações de monitoramento são eficazes quando há a necessidade de proteção de algo ou alguém. O problema levantado nesta dissertação é também

concernente a proteção de pessoas, porém no âmbito da segurança do trabalho, especificamente envolvendo a cabeça e o crânio dos trabalhadores.

No Brasil, de acordo com o Anuário (2016, p. 590), houve 24.561 acidentes relacionados com cabeça e crânio em 2014 (dados revisados em maio de 2016), representando cerca de 3,4% do total de acidentes de trabalho no país . O Anuário (2016, p. 590) não revela detalhes e em que circunstâncias aconteceram estes casos, o que se leva a conjecturar que, pelo menos uma parte, ocorreu devido a não observância quanto ao uso de EPI's (Equipamentos de Proteção Individual). Um monitoramento por câmera, aliado a um sistema que consiga detectar automaticamente trabalhadores que não estão utilizando capacetes de segurança, poderá ser utilizado como uma ferramenta informativa, a exemplo do CIM e DETECTA, dando subsídios para que a engenharia de segurança do trabalho possa atuar de forma a mitigar o número de óbitos relacionados.

1.2 Objetivo geral:

Desenvolver um sistema de visão artificial no intuito de detectar a presença de pessoas que não estão utilizando capacetes de segurança em locais onde são obrigatórios.

1.3 Objetivos específicos:

- Identificar o algoritmo de detecção de movimento mais adequado.
- Descobrir quais parâmetros da combinação filtro-descritor-classificador apresentam maior precisão.

1.4 Organização do trabalho:

O sistema de visão artificial para detecção de pessoas não utilizando capacetes de segurança é tratado neste documento ao longo das seguintes seções:

- **FUNDAMENTOS TEÓRICOS E REVISÃO BIBLIOGRÁFICA:** aqui é encontrada uma revisão dos fundamentos teóricos do processamento digital de imagens concomitantemente a revisão bibliográfica dos algoritmos utilizados no desenvolvimento deste estudo.
- **MATERIAIS E MÉTODOS:** nesta seção são apresentados os materiais utilizados e a metodologia empregada para a obtenção dos resultados.
- **RESULTADOS E DISCUSSÃO:** aqui é visto um relato dos experimentos realizados e os resultados obtidos.
- **CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS:** nesta seção são colocadas as conclusões, as devidas considerações e sugestões para trabalhos futuros.
- **REFERÊNCIAS BIBLIOGRÁFICAS:** lista de publicações utilizadas ao longo do presente estudo.

2. Fundamentos teóricos e revisão bibliográfica

Os conceitos teóricos apresentados a seguir, fazem-se necessários para melhor entendimento da pesquisa efetuada.

2.1 Imagem digital

Conforme Gonzalez e Woods (2010), uma imagem é definida como sendo pontos de uma função bidimensional, $f(x, y)$, em que x e y são coordenadas espaciais e a amplitude de f é chamada de intensidade de luz naquele ponto. Os valores a serem assumidos devem ser positivos e finitos, como mostra a inequação 1:

$$0 \leq f(x, y) < \infty \quad (1)$$

A luz é caracterizada por duas componentes: a iluminação e a reflectância. A iluminação refere-se à quantidade de luz incidindo diretamente sobre o objeto. A reflectância corresponde a quantidade de luz refletida por ele. Serão representadas por $i(x, y)$ e $r(x, y)$, respectivamente. O produto destas funções resultará em $f(x, y)$ conforme expressões 2,3 e 4:

$$f(x, y) = i(x, y)r(x, y) \quad (2)$$

Onde:

$$0 \leq i(x, y) < \infty \quad (3)$$

E

$$0 \leq r(x, y) < 1 \quad (4)$$

Se for considerada imagem monocromática, $f(x, y)$ assumirá níveis de cinza em cada ponto, de forma que ficará restrito ao intervalo:

$$L_{min} \leq f(x, y) \leq L_{max} \quad (5)$$

Sendo que L_{min} assumirá valores não negativos e L_{max} , valores que sejam finitos. Este intervalo será denominado escala de cinza. No trabalho a ser apresentado, será utilizado o intervalo $[0,255]$, onde $L = 0$ é considerado negro e $L = 255$, branco. Os valores intermediários são tons de cinza, próximos do negro ou do branco, conforme sua proximidade numérica do valor 0 ou 255, respectivamente. A figura 1 elucida melhor este conceito.

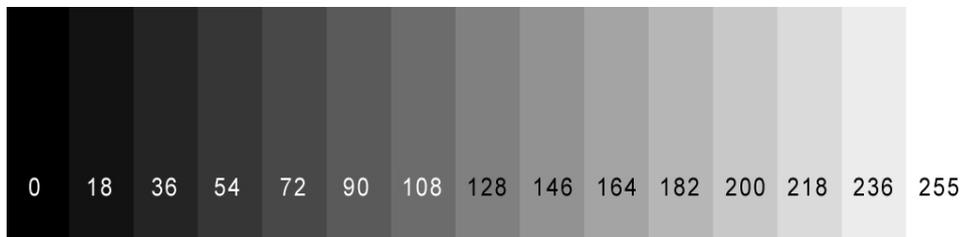


Figura 1: Escala de cinza (<http://hammadk.com/monitor-test/>, 15/10/2015).

2.2 Amostragem e quantização

Conforme Gonzalez e Woods (2010), para que uma imagem seja processada computacionalmente, é necessário que a função $f(x, y)$, que a representa, seja digitalizada tanto espacialmente quanto na sua amplitude. O termo amostragem refere-se à digitalização de suas coordenadas espaciais e quantização, a digitalização da amplitude em níveis de cinza.

Para melhor exemplificar, ao se supor uma matriz $N \times M$ representando uma imagem, obtém-se a equação 6.

$$F = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M - 1) \\ f(1,0) & f(1,1) & \dots & f(1, M - 1) \\ \dots & \dots & \dots & \dots \\ f(N - 1,0) & f(N - 1,1) & \dots & f(N - 1, M - 1) \end{bmatrix} \quad (6)$$

Cada elemento da matriz F , onde i representa a linha e j a coluna, representa um elemento de uma imagem digital, ou *pixel*, ou ainda *pel* (abreviações do inglês: “Picture elements”).

Os parâmetros N , M e o número de níveis de cinza discretos, que será chamado de G , devem ser definidos antes do processamento da imagem digital. Desta forma, o cálculo para a obtenção do número de bits (b) necessários para armazenar uma imagem digital é apresentado pela equação 7:

$$b = N \cdot M \cdot G \quad (7)$$

Exemplificando: uma imagem de 256 x 256 pixels com 128 níveis de cinza requerem 8.388.608 bits.

Outro conceito importante: a resolução, definida como sendo o grau de detalhes que torna a imagem discernível. Depende dos parâmetros N , M e G . É importante salientar que a resolução, dependendo do projeto, não necessita atender o discernimento do olho humano e sim atender de forma mais otimizada possível a fase final da visão de máquina: o reconhecimento de padrões.

2.3 Modelo de cores

Conforme Gonzalez e Woods (2000), os modelos utilizados objetivam aplicações no hardware. Em monitores coloridos e câmeras de vídeo, é largamente usado o RGB (do inglês: “red” (vermelho), “green” (verde) e “blue” (azul)); para impressoras coloridas, utiliza-se o

CMY (do inglês “cyan”(ciano), “magenta”(magenta), “yellow”(amarelo)); para transmissão de TV colorida, o YIQ (Y (luminância), I (em fase), Q (quadratura)). No caso de manipulação de imagens coloridas são frequentemente usados HSI (matiz, saturação e intensidade) e o HSV (matiz, saturação e valor).

Devido às características do presente trabalho, somente os modelos RGB e HSV foram utilizados. O modelo RGB foi aplicado na etapa de aquisição de imagens efetuada através de câmera de vídeo, enquanto que o modelo HSV na etapa de descrição e comparação de imagens.

2.3.1 O modelo de cores RGB

Este modelo é baseado num sistema de coordenadas cartesianas tridimensional, cujos eixos são bases para as cores primárias vermelho, verde e azul; enquanto que em outras arestas estão as cores secundárias ciano, magenta e amarelo; a cor preta está na origem e o branco no ponto mais distante da origem. Na figura 2 é apresentado o cubo de cores RGB para melhor elucidação. Uma observação importante: os pontos ao longo da diagonal principal do cubo têm valores de cinza desde preto na origem até branco no ponto (1,1,1).

Para se ter ideia da ordem de grandeza da quantidade de cores no cubo RGB, se forem consideradas 256 tonalidades (8 bits) para cada cor primária, seriam obtidas 16.777.216 cores distintas.

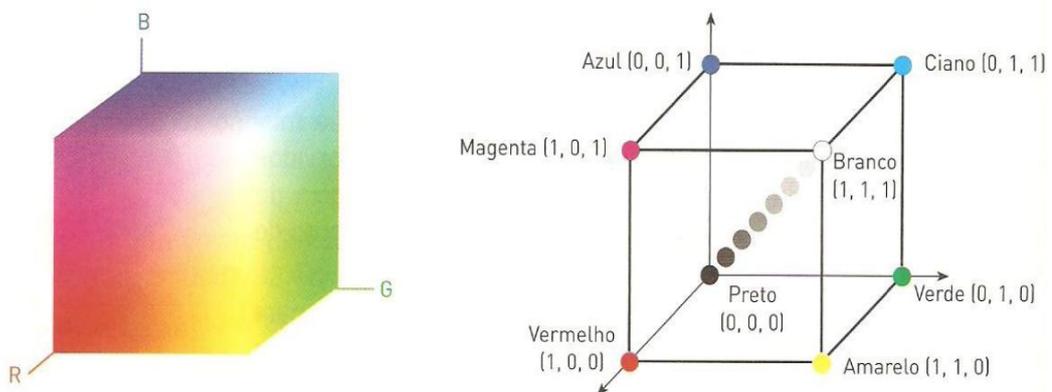


Figura 2: Cubo RGB (<https://sites.google.com/site/aimciv/modelo-rgb> em 15/10/2015).

2.3.2 O modelo de cores HSV

Este modelo foi promovido por Smith (1978). É o modelo que melhor representa a sensação humana ao se ver um objeto, descrito em termos de matiz, saturação e valor (luminância). Na figura 3 é apresentado graficamente o espaço de cores HSV.

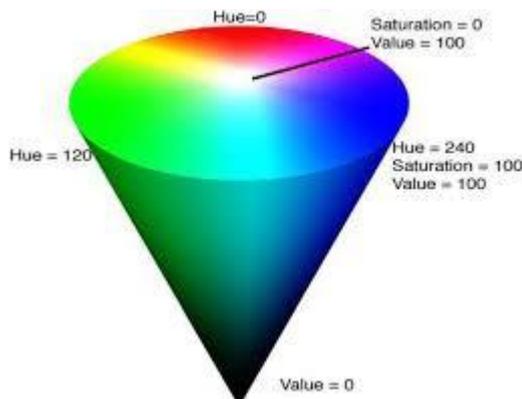


Figura 3: Modelo de cores HSV (Konwar et al, 2014)

Matiz é um atributo que descreve uma cor pura. Quando um observador define um objeto como sendo laranja, ele está especificando a cor predominante, ou seja, a matiz. Em termos de magnitude, varia de 0° a 360° .

Saturação é o grau de diluição da cor pura com os tons de cinza. Varia entre 0% a 100% ou 0 a 1.

Valor (luminância) refere-se ao brilho e varia com a saturação da cor. Normalmente, sua magnitude é trabalhada em percentuais, de 0 a 100%. Quando o valor é zero no espaço de cores, representa ausência de brilho e caracterizará a cor preta.

Este modelo permite que o usuário especifique as cores de um pixel em termos de matiz e saturação. A informação de brilho pode ser suprimida.

2.4 Conceito e operações com histogramas

Histograma é uma ferramenta comumente utilizada no campo da estatística para apresentar uma distribuição de frequências de determinados dados ou faixa de dados. No processamento de imagens, conforme Bradski e Kaebler (2008), pode ser aplicado para mostrar a distribuição de cores de um objeto, ou ainda auxiliar em uma utilização mais sofisticada como se um determinado objeto está ou não presente em uma imagem, através de distribuição de probabilidades de borda.

Um dos recursos mais utilizados é a normalização de histogramas. Esta operação consiste em distribuir os dados nas respectivas classes, de forma que representem fração do número total de dados de todo o histograma, imprescindível para se comparar dois histogramas de imagens de tamanhos distintos, cujo conceito foi introduzido por Swain e Ballard (1991) e generalizado posteriormente por Schiele e Crowley (1996 e 1998).

Bradski e Kaebler (2008), descrevem quatro métodos disponíveis de comparação de histogramas, comentados a seguir:

2.4.1 Correlação entre dois histogramas

É calculada conforme a equação 8:

$$d_{correl}(H_1, H_2) = \frac{\sum_i H_1'(i).H_2'(i)}{\sqrt{\sum_i H_1'^2(i).H_2'^2(i)}} \quad (8)$$

Onde:

H_i = Histograma referente a imagem 1.

$H_2 =$ Histograma referente a imagem 2.

$i =$ Intervalo de classe do histograma.

$H'_k(i) = H_k(i) - (1/N) (\sum_j H_k(j))$, sendo N definido como a quantidade de classes do histograma.

Os valores de d_{correl} próximos de 1, indicam grande similaridade; próximos a -1, incompatibilidade e próximos do valor zero, correlação quase inexistente.

2.4.2 Qui-quadrado entre dois histogramas

É regido pela equação 9:

$$d_{\text{qui-quadrado}}(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)} \quad (9)$$

Onde:

$H_1 =$ Histograma referente a imagem 1.

$H_2 =$ Histograma referente a imagem 2.

$i =$ Intervalo de classe do histograma.

Os valores de $d_{\text{qui-quadrado}}$ próximos de 0 indicam grande similaridade; para histogramas totalmente incompatíveis, os valores podem tornar-se ilimitados, dependendo somente do tamanho do histograma.

2.4.3 Intersecção entre dois histogramas

É calculada conforme a equação 10:

$$d_{intersecção}(H_1, H_2) = \sum_i \min(H_1(i), H_2(i)) \quad (10)$$

Onde:

H_1 = Histograma referente a imagem 1.

H_2 = Histograma referente a imagem 2.

I = Intervalo de classe do histograma.

Os valores de $d_{intersecção}$ que indicam grande compatibilidade estão próximos de 1. Em contrapartida, valores próximos de zero expressam histogramas incompatíveis.

2.4.4 Distância Bhattacharyya entre dois histogramas

É calculada conforme a equação 11:

$$d_{Bhattacharyya}(H_1, H_2) = \sqrt{1 - \sum_i \frac{\sqrt{H_1(i) \cdot H_2(i)}}{\sqrt{\sum_i H_1(i) \cdot \sum_i H_2(i)}}} \quad (11)$$

Onde:

H_1 = Histograma referente a imagem 1.

H_2 = Histograma referente a imagem 2.

i = Intervalo de classe do histograma.

Histogramas similares resultam em valores de $d_{Bhattacharyya}$ próximos de zero. Por outro lado, valores próximos de 1 expressam histogramas distantes.

2.4.5 Comparação entre os métodos

Bradski e Kaebler (2008) exemplificam e comparam os métodos propostos, conforme ilustrado na figura 4, através das imagens das palmas das mãos em diferentes situações: em local fechado (canto superior esquerdo da figura), ao ar livre sob a sombra (imagem intermediária do lado esquerdo da figura) e ao ar livre sob o sol (canto inferior esquerdo da figura). Nas colunas central e direita, a distribuição das cores é representada graficamente no contexto do cubo RGB e do cone HSV, respectivamente. Na tabela 1 são aplicados os métodos, comparando o histograma H_1 gerado pela metade superior da imagem da palma em local fechado, com os histogramas H_2 referentes às imagens completas das palmas em todas as situações.

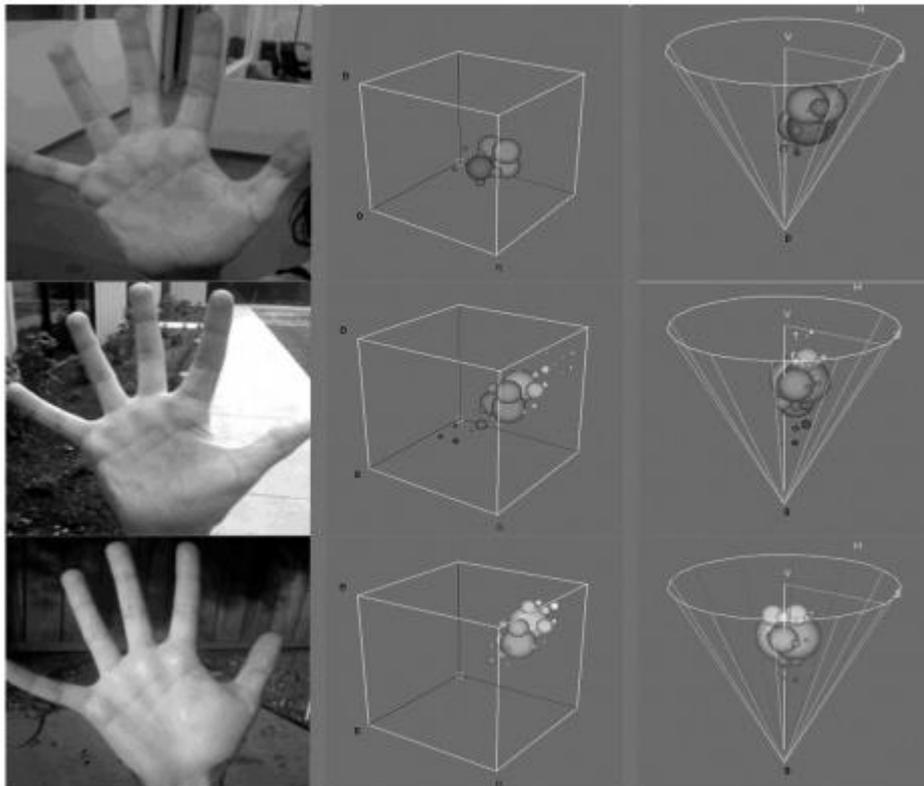


Figura 4: Palmas das mãos em diferentes situações: em local fechado (canto superior esquerdo da figura), ao ar livre sob a sombra (imagem intermediária do lado esquerdo da figura) e ao ar livre sob o sol (canto inferior esquerdo da figura). Nas colunas central e direita, a distribuição das cores é representada graficamente no contexto do cubo RGB e do cone HSV, respectivamente. (Bradski e Kaehler (2008), p. 206).

Tabela 1: Comparação do histograma H_1 gerado pela metade superior da imagem da palma em local fechado, com os histogramas H_2 referentes às imagens completas das palmas em todas as situações, pelos métodos de correlação, qui-quadrado, intersecção e Bhattacharyya (Bradski e Kaehler (2008), p. 206).

Método Imagem	Correlação	Qui-quadrado	Intersecção	Bhattacharyya
Palma da mão em local fechado	0,96	0,14	0,82	0,20
Palma da mão ao ar livre e sob a sombra	0,09	1,57	0,13	0,80
Palma da mão ao ar livre e sob o sol	0,00	1,98	0,01	0,99

Conforme esperado, os dados da tabela 1 indicam que existe uma maior afinidade entre o histograma H_1 gerado pela metade superior da imagem da palma em local fechado, com o histograma H_2 referente à mesma imagem da palma em local fechado, porém 100% completa. Considerando, por exemplo, a magnitude de 0,96 obtida pelo método da correlação, observa-se que é muito próximo do valor 1,00 - indicativo de comparação de histogramas idênticos. Esta diferença é gerada principalmente entre as diferenças dos fundos das imagens. Na imagem referente a H_1 nota-se uma proporção menor de pixels de cor escura, se comparadas a imagem referente a H_2 . Quanto aos outros métodos, mostram a mesma tendência ao indicar em suas magnitudes a proximidade destes histogramas.

Ao analisar a segunda comparação, entre o histograma H_1 gerado pela metade superior da imagem da palma em local fechado, com o histograma H_2 referente a imagem da palma ao ar livre e sob a sombra; foi obtido a magnitude de 0,09 pelo método da correlação. O valor é próximo de 0,00 - indicativo de correlação inexistente. Esta diferença é gerada entre as diferenças dos fundos das imagens e da mudança de cor da palma da mão. Na imagem referente a H_1 nota-se que as cores do fundo são intermediárias, se comparadas ao fundo da imagem referente a H_2 , que contém pixels de cores claras e escuras. Quanto a palma da mão, na imagem referente a H_2 houve uma mudança para cores mais claras em comparação a imagem de H_1 . Quanto aos outros métodos, mostram a mesma tendência ao indicar em suas magnitudes a pouca proximidade destes histogramas.

Porém a informação mais contundente é obtida com a comparação entre o histograma H_1 gerado pela metade superior da imagem da palma em local fechado, com o histograma H_2 referente a imagem da palma ao ar livre e sob o sol. Pelo método da correlação, a magnitude obtida foi 0,00 - indicativo de correlação inexistente. Foi gerada pelas diferenças dos fundos das imagens e principalmente da cor da palma da mão. A intensa luminância sobre a palma da mão alterou os valores dos pixels para cores bem mais claras da imagem referente a H_2 . Os outros métodos mostraram a mesma tendência, ao indicar em suas magnitudes, a correlação inexistente destes histogramas.

2.5 Filtragem digital

2.5.1 Conceito

No contexto de processamento de imagens, Marques Filho e Vieira Neto (1999) definem a filtragem digital como sendo a aplicação de certas técnicas, conhecidas como operações de convolução com máscaras, objetivando acentuar, suavizar ou corrigir determinadas características da imagem em estudo. As máscaras agem sobre cada pixel, conforme ilustrado na figura 5, e o novo valor assumido leva em consideração o valor original e o de sua vizinhança.

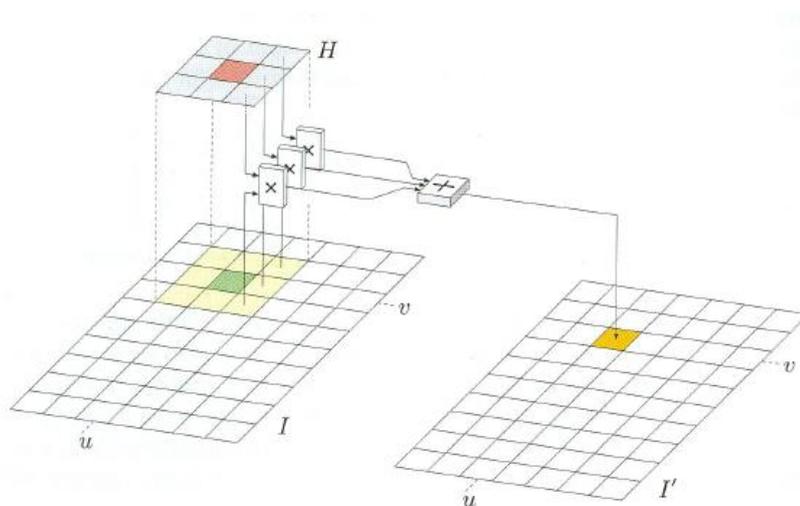


Figura 5: Aplicação da máscara H sobre o pixel da imagem I . Novo valor assumido representado na imagem suavizada I' (Burger e Burge (2008) p. 90).

No presente trabalho, os filtros de média, de mediana, de Gauss e filtro bilateral foram testados e utilizados, cujas as características são descritas a seguir:

2.5.2 Filtro de média

Conhecido também como filtro-caixa (do inglês – *box filter*). Por ser considerado um filtro do tipo passa-baixa, atua na suavização da imagem e redução de ruído. Porém, a técnica contribui para o embaçamento da imagem, removendo alguns detalhes finos da imagem.

Na figura 6, são apresentadas algumas máscaras utilizadas pelo filtro de média. A constante de multiplicação diante de cada máscara é igual a 1 dividido pela quantidade total dos pixels da máscara, caracterizando o cálculo da média.

$$\begin{array}{ccc}
 \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} & \frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\
 \text{(a)} & \text{(b)} & \text{(c)}
 \end{array}$$

Figura 6: Máscaras para cálculo do filtro da média. (a) 3 x 3; (b) 5 x 5; (c) 7 x 7 (Marques Filho e Vieira Neto (1999) p. 86).

Na figura 7, são mostrados os resultados da aplicação destas máscaras com dimensões $n \times n$, com $n = 3, 5, 7, 17$ e 31 . Nota-se que a medida que se aumenta a dimensão das máscaras, contribui-se para o embaçamento da imagem. Porém, dependendo da aplicação, o embaçamento da imagem pode ser irrelevante em relação a diminuição do ruído, por exemplo.

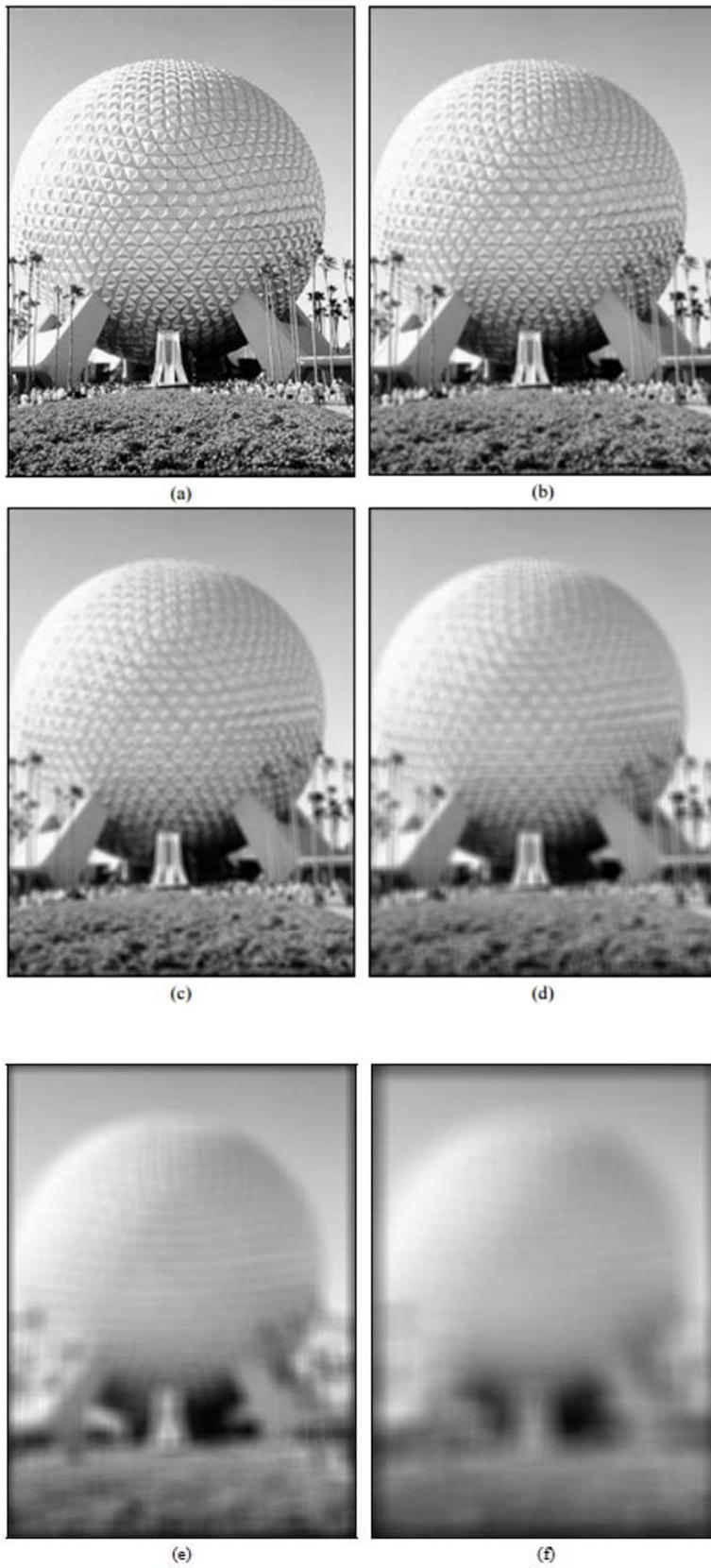


Figura 7: (a) Imagem original. (b) a (f) Resultados da suavização com filtro de média utilizando máscaras $n \times n$, com $n = 3, 5, 7, 17$ e 31 (Marques Filho e Vieira Neto (1999) p. 87 e 88).

Na figura 8, é exemplificada uma aplicação industrial de filtros. A imagem de raios X de uma placa de circuitos eletrônico é corrompida pelo ruído conhecido como sal e pimenta (figura 8(a)). Após a aplicação do filtro de média (figura 8(b)), a imagem apresenta uma sensível redução do ruído.

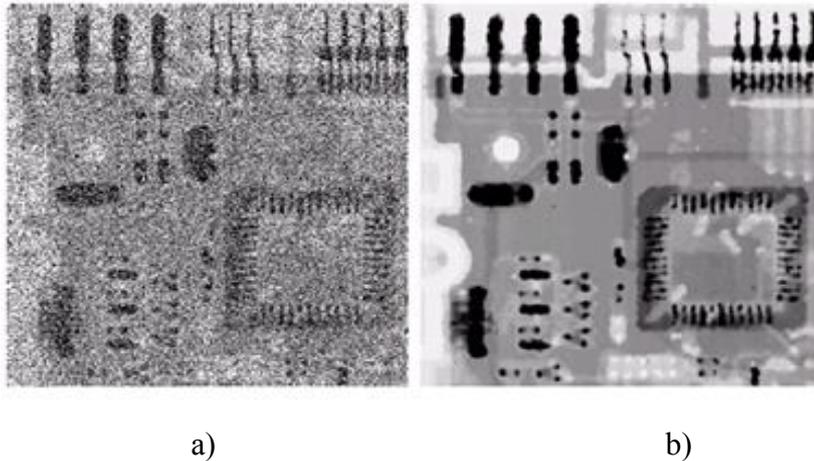


Figura 8: Aplicação de filtros em imagem com ruído (Gonzalez (2010) p. 103)

2.5.3 Filtro de mediana

Segundo Marques Filho e Vieira Neto (1999), uma das principais limitações do filtro de média está na sua deficiência de preservar bordas e detalhes finos da imagem. Para contorná-la, uma técnica alternativa é o filtro de mediana. Nesta técnica, o pixel central da janela é substituído pela mediana dos pixels situados em sua vizinhança. Para se obter a mediana de uma máscara de n pixels, é necessário primeiramente ordenar seus valores. Se n for ímpar, a mediana assumirá o valor do elemento central do conjunto ordenado, conforme ilustra a figura 9. Se n for par, a mediana é calculada pela média aritmética dos dois elementos mais próximos do centro.

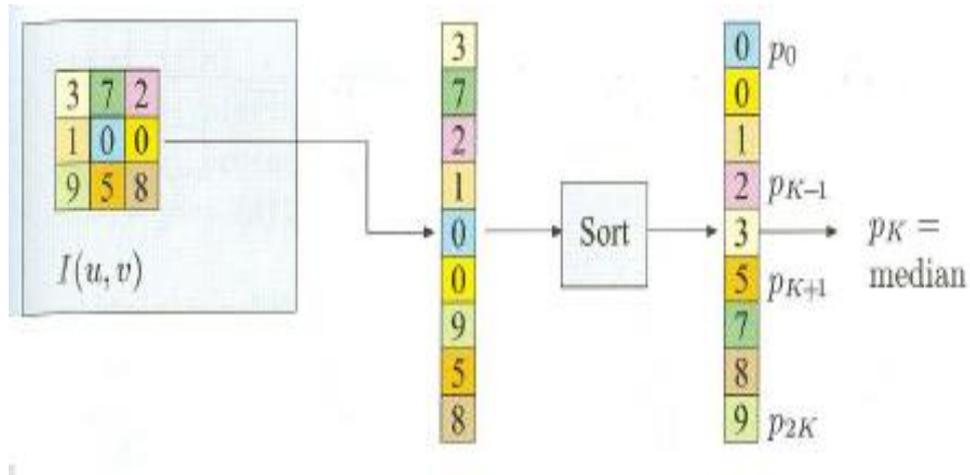


Figura 9: Determinação da mediana de uma máscara 3×3 (Burger e Burge (2008) p. 107).

Este método apresenta desempenho particularmente bom em situações nas quais a imagem é contaminada por ruído impulsivo (sal-e-pimenta). Na figura 10, é apresentado um comparativo entre os filtros de média e de mediana aplicados a uma imagem com este ruído.

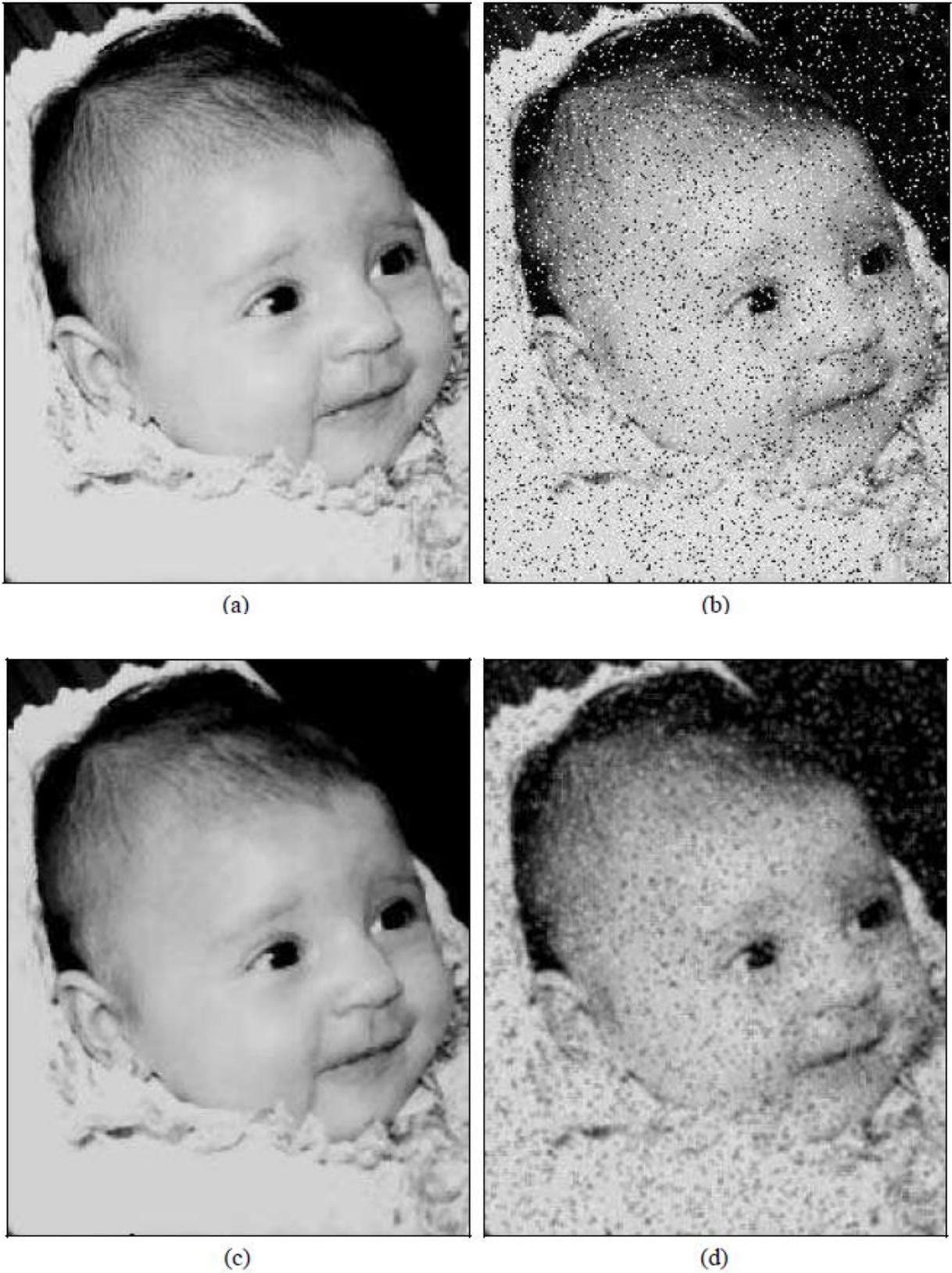


Figura 10: (a) Imagem original; (b) imagem contaminada por ruído impulsivo (sal e pimenta); (c) resultado da filtragem pelo filtro de mediana com máscara 3x3; (d) resultado da filtragem pelo filtro de média com máscara 3 x 3 (Marques Filho e Vieira Neto (1999) p. 91 e 92).

A imagem original (figura 10(a)) é contaminada por um ruído impulsivo conhecido como sal e pimenta, resultando na imagem da figura 10(b). O filtro de mediana com máscara 3×3 é aplicado, resultando na imagem da figura 10(c), praticamente preservando os contornos e retirando o ruído. A eficiência deste filtro está no fato do ruído ser impulsivo, ou seja, contamina os pixels da imagem com cores próximo ao branco (efeito sal) ou preto (efeito pimenta). Considerando uma escala de zero a 255, o efeito sal pontua a imagem com valores próximos de 255 e o efeito pimenta com valores próximos a zero. Favorecido pelo fato deste ruído não atingir grandes áreas da imagem, no momento da ordenação dos pixels, estes valores são substituídos pela mediana da máscara.

Na figura 10(d) apresenta o resultado do filtro de média com máscara 3×3 aplicado à imagem original. Nota-se que visualmente o resultado está aquém daquele obtido pelo filtro de mediana. Não há a eliminação significativa de ruído. Neste caso, somente a aplicação de máscaras maiores poderá melhorar o desempenho; em contrapartida os contornos ficarão comprometidos, contribuindo para o efeito de embaçamento da imagem.

2.5.4 Filtro de Gauss

Segundo Burger e Burge (2008), o filtro de Gauss é regido pela função discreta e bidimensional de Gauss, calculada conforme a equação 12:

$$G_{\sigma} = e^{-\frac{r^2}{2\sigma^2}} \quad \text{ou} \quad G_{\sigma}(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (12)$$

Onde: σ é o desvio padrão,

r é a distância do centro.

Na figura 11, é ilustrada uma representação em 3D e a sua máscara. Nota-se que o pixel central recebe o valor máximo, diminuindo de forma isotrópica à medida que se afasta do centro. Para diminuir os custos computacionais, o filtro bidimensional de Gauss é efetuado a partir de um par de filtros unidimensionais.

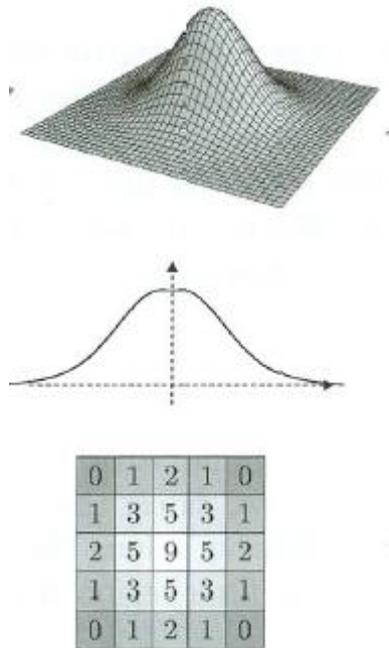


Figura 11: Representação em 3D do filtro de Gauss e sua respectiva máscara (Burger e Burge (2008) p. 97).

2.5.5 Filtro bilateral

Ao se suavizar uma imagem com o filtro de Gauss para se dissolver um ruído, os contornos desta imagem também serão suavizados. Tomasi e Manduchi (1998) propuseram o modelo de filtro bilateral para preservar satisfatoriamente as bordas das imagens. A abordagem teórica é apresentada a seguir:

Considerando-se uma imagem $f(x)$, à qual são aplicados filtros passa-baixas, o resultado $h(x)$ da filtragem será conforme a equação 13:

$$h(x) = k_d^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi - x) d\xi \quad (13)$$

Onde $c(\xi - x)$ mensura a proximidade geométrica entre o centro x e um certo ponto ξ .

Para se preservar a componente DC, $k_d(x)$ deverá assumir valores conforme a equação 14:

$$k_d(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi) d\xi \quad (14)$$

Se o filtro for invariante ao deslocamento, $c(\xi - x)$ é somente uma função do vetor da diferença $(\xi - x)$ e k_d é constante.

O intervalo de filtragem é semelhantemente definido conforme equação 15:

$$h(x) = k_r^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) s(f(\xi), f(x)) d\xi \quad (15)$$

Neste caso, o núcleo $(f(\xi), f(x))$ mensura a semelhança fotométrica entre os pixels. A função de similaridade s opera no intervalo da imagem da função f , enquanto que a função de proximidade c opera no domínio de f . A constante de normalização k_r é representada pela equação 16:

$$k_r(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(f(\xi), f(x)) d\xi \quad (16)$$

A distribuição espacial das intensidades, por si só, não desempenha nenhum papel importante no intervalo de filtragem. Para obter-se uma solução apropriada, é necessário combinar o filtro de domínio e de intervalo, reforçando assim tanto a localidade geométrica quanto a fotométrica. É descrita pela equação 17:

$$h(x) = k^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d\xi \quad (17)$$

Com a normalização (equação 18):

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, x) s(f(\xi), f(x)) d\xi \quad (18)$$

Esta combinação é denominada filtragem bilateral. O valor do pixel na posição x é substituído por uma média de valores de pixels semelhantes e próximas.

Um caso particular e importante de filtragem bilateral é utilizando a filtragem de Gauss invariante ao deslocamento, em que tanto a funções de proximidade c e de similaridade s são funções de Gauss da distância euclidiana entre os seus argumentos.

A função de proximidade c é radialmente simétrica e está representada pela equação 19:

$$c(\xi - x) = e^{-\frac{1}{2}\left(\frac{d(\xi-x)}{\sigma_d}\right)^2} \quad (19)$$

Onde:

$d(\xi - x)$ é a distância euclidiana.

σ_d representa a difusão geométrica no domínio, sendo definida conforme a quantidade desejada de filtragem passa baixa. Altos valores tendem a embaçar mais a imagem, pois combinam pixels mais distantes.

De forma análoga, a função similaridade s é calculada conforme a equação 20:

$$s(\xi - x) = e^{-\frac{1}{2}\left(\frac{\delta(f(\xi), f(x))}{\sigma_r}\right)^2} \quad (20)$$

Onde:

$\delta(f(\xi), f(x))$ é uma medida adequada da distância entre os valores de $f(\xi)$ e $f(x)$. No caso escalar, representa a diferença absoluta da diferença de pixels.

σ_r representa a propagação fotométrica, definida como a quantidade desejada de combinação de valores de pixel dentro do intervalo de uma imagem. Normalmente, pixels com valores próximos de σ_r são combinados.

Na figura 12, é apresentado um exemplo de filtragem bilateral com parâmetros $\sigma_d = 3$ pixels e $\sigma_r = 50$ valores de intensidade aplicados à imagem da figura 12 (a), para se obter a imagem ilustrada na figura 12 (b). A maior parte da textura fina foi eliminada por filtragem, porém os contornos foram mantidos tais quais eram na imagem original. A figura 12 (c) mostra um detalhe da figura 12 (a), e a figura 12 (d) mostra a correspondente versão filtrada. As duas cebolas assumiram uma aparência artificial e nota-se que a textura fina foi eliminada. No entanto, o sombreamento global é preservado, uma vez que está situado dentro da banda do filtro de domínio, sendo também praticamente afetado pelo filtro de intervalo. Além disso, os limites das cebolas são mantidos.

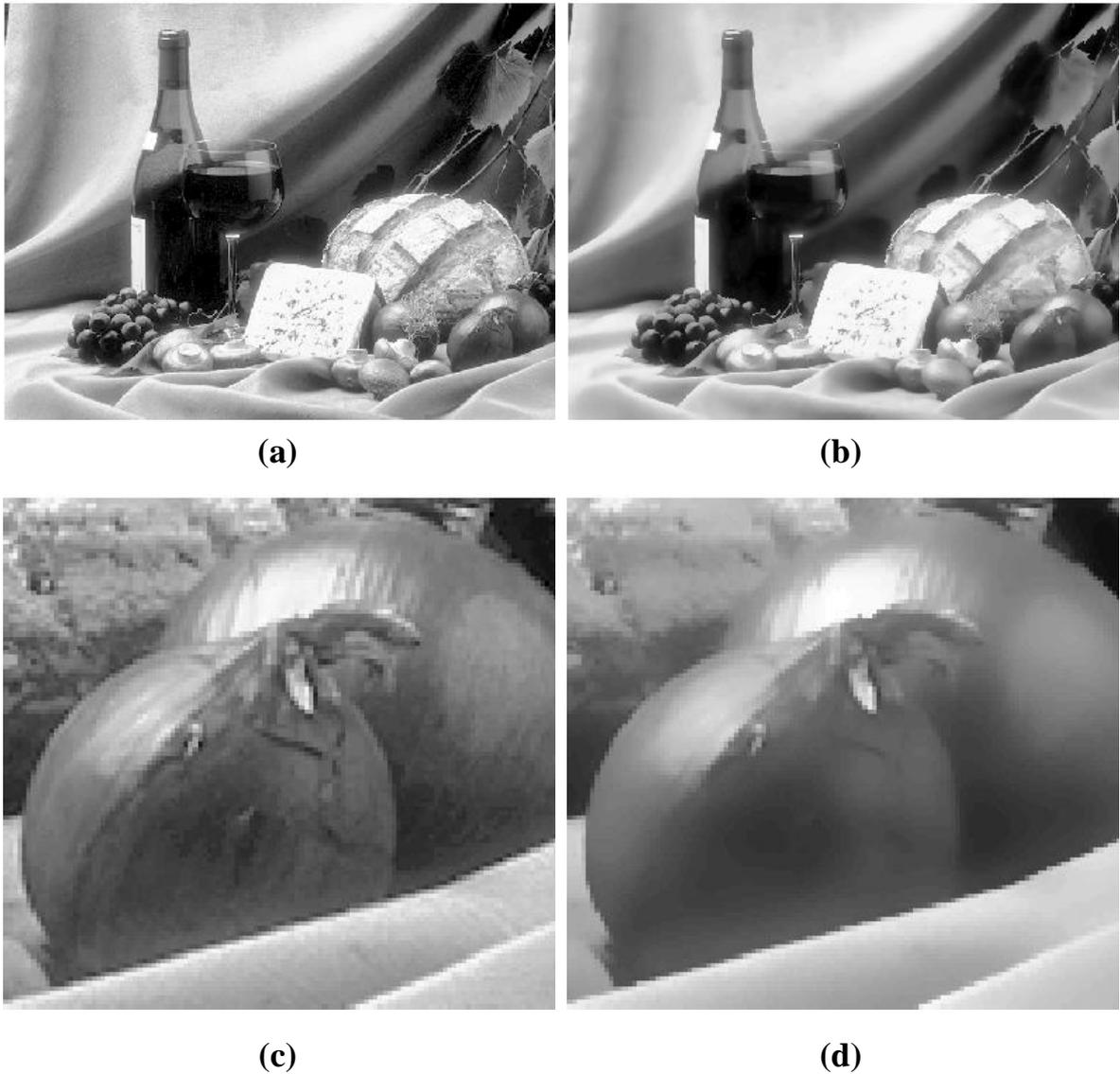


Figura 12: (a) Imagem original; (b) Resultado da aplicação com filtro bilateral; (c) e (d) Detalhes de (a). (Tomasi e Manduchi, 1998).

Na figura 13, são mostrados os efeitos da variação dos parâmetros σ_d e σ_r . No caso apresentado, a preservação de bordas principais começa a ficar comprometida a partir da combinação $\sigma_d = 3$ e $\sigma_r = 100$. Nota-se também que existe pouca influência de σ_r na imagem, quando $\sigma_d = 1$.

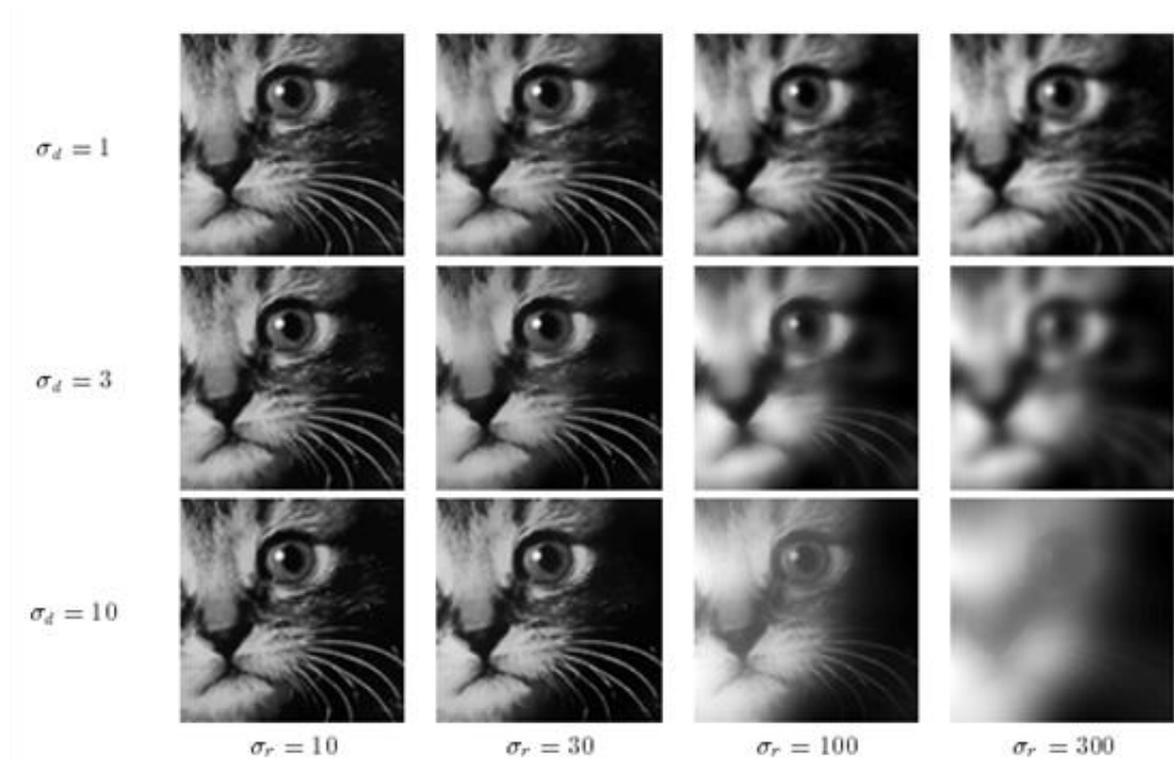


Figura 13: Efeitos da variação dos parâmetros σ_d e σ_r . (Tomasi e Manduchi, 1998).

2.6 Segmentação

A segmentação consiste em seccionar a imagem em regiões, objetos ou cores para posteriormente ser analisada. Esta divisão deve ocorrer até que se consiga isolar o objeto de interesse. Por exemplo, no caso de software de reconhecimento facial, o mais recomendado é efetuar uma segmentação da imagem de forma a separar tudo que se assemelha a formatos de rostos para posteriormente serem analisados, o que implica em um menor custo computacional.

A segmentação é um dos passos mais relevantes do processamento de imagens, pois pode determinar o sucesso ou fracasso na etapa seguinte.

Existem inúmeras técnicas que podem ser utilizadas para a segmentação, dentre as quais podem ser citadas:

- Por detecção de borda.
- Limiarização.
- Por crescimento de região.
- Movimento

Devido às características da presente pesquisa, foi utilizada somente a segmentação por uso do movimento, a qual será comentada a seguir:

2.6.1 Segmentação por movimento

A utilização de movimento como forma de segmentar uma imagem é uma das técnicas mais convenientes para o objeto de interesse desta dissertação, principalmente pelo custo computacional, pois faz a extração do objeto de um fundo de detalhes irrelevantes. Devido às características deste trabalho, será abordado somente no domínio espacial. No domínio da frequência, seriam determinados parâmetros que não seriam úteis para o projeto, como a velocidade máxima do objeto, por exemplo.

2.6.1.1 Métodos de segmentação por movimento

Segundo Gonzalez e Woods (2000), a técnica mais simples se dá por comparação de dois quadros em sequência da imagem: $f(x, y, t_i)$ e $f(x, y, t_j)$, tomados nos instantes t_i e t_j , respectivamente. Uma das formas de realizar a comparação é através da diferença entre as duas imagens, o que resulta em cancelamento dos componentes estacionários e conseqüentemente deixando posições não nulas nos componentes não estacionários. É definida pela equação 21:

$$d_{ij}(x, y) = \begin{cases} 1 & \text{se } |f(x, y, t_i) - f(x, y, t_j)| > \theta \\ 0 & \text{caso contrário} \end{cases} \quad (21)$$

Em que θ é um limiar. A equação sugere que todos os pixels com valor igual a 1 são resultados de um objeto em movimento. Muitos pixels isolados podem expressar este valor, muitas vezes em decorrência de ruídos. A sua remoção pode ser efetuada considerando somente regiões com determinado número de elementos conectados, por exemplo pixels conectados 4 a 4 ou 8 a 8. O método não é robusto, pois dependendo das condições do ambiente, como variação da iluminação e geração de sombras, pode gerar falsas detecções. Apresenta, também, problemas quando algum objeto é inserido de forma definitiva na cena, integrando o fundo a partir de um certo momento, necessitando de uma inicialização manual do programa.

As propostas para solucionar o problema do fundo evoluíram no decorrer do tempo. Ridder et al (1995) propuseram a filtragem de Kalman para estimativa de fundo. Seu sistema apresentou robustez quanto a mudança de iluminação, pois a considerou na estimativa de fundo, ou seja, aquela mudança não foi detectada como primeiro plano.

O algoritmo Pfinder, citado em artigo de Wren (1997), teve como objetivo específico detectar pessoas, utilizando um modelo estatístico de subtração de fundo considerando-o como uma superfície de textura. Cada pixel do fundo, então, foi associado a um valor de cor e uma distribuição sobre a média conforme uma única Gaussiana.

Friedman e Russel (1997), em sua pesquisa voltada para detecção de veículos, separaram a imagem em classes: cores correspondentes a estrada, sombra e veículos; e através de três distribuições separadas de gaussianas, classificaram cada pixel. A atualização dos quadros ocorreu mediante o algoritmo EM (*Expectation-Maximization*) para que os objetos lentos fossem tratados eficazmente. Seu trabalho também identificou e eliminou sombras adequadamente.

Com uma proposta parecida, Stauffer e Grinsom (1999) modelaram cada pixel como uma mistura de gaussianas. Com base na variância, conseguiram determinar quais gaussianas correspondiam às cores do fundo. Os valores dos pixels que não se encaixavam nas distribuições de fundo foram considerados de primeiro plano até que fosse constituída uma

gaussiana que os incluisse com consistência. Seu sistema mostrou-se eficaz com mudanças na iluminação, com movimentos repetitivos dos elementos da cena, com a introdução e a remoção dos objetos da cena e lidou convenientemente com objetos lentos. Seu método *backgrounding* constitui-se de dois significativos parâmetros: α – constante de aprendizagem e T – proporção dos dados que devem ser considerados para o fundo.

No entanto, conforme artigo de KaewTraKuPong e Bowden (2001), este método apresenta aprendizagem lenta no início, especialmente em ambientes ruidosos. Além disso, não faz distinção entre o movimento de sombras e os próprios objetos em movimento. O artigo apresenta um método que melhora este modelo de mistura de fundo adaptativo, mantendo o conceito de mistura de gaussianas, especificamente entre $K = 3$ a 5 distribuições. Utiliza equações diferentes de atualização em diferentes fases. Isso permite que o sistema proposto aprenda mais rapidamente e com maior precisão, adaptando-se efetivamente a ambientes em mudança. Uma nova proposta de detecção de sombra também é introduzida neste trabalho. Os resultados apresentados atestam a evolução do modelo. Este modelo, denominado MOG na biblioteca OpenCV, foi um dos algoritmos testados para a concepção da presente dissertação.

Anos mais tarde, em dois artigos publicados por Z.Zivkovic em 2004 e 2006, é proposta uma nova melhoria no modelo. A evolução está exatamente na adaptabilidade do algoritmo, pois não só os parâmetros, mas também o número apropriado de distribuições de Gauss para cada pixel são avaliados constantemente. Vale ressaltar que no trabalho de KaewTraKuPong e Bowden (2001), é definido um número fixo de K distribuições gaussianas para todo o algoritmo. Também foi testado e avaliado na presente dissertação. Na biblioteca OpenCV, é denominado MOG2.

2.7 Etapas de um sistema de visão artificial

Conforme Marques Filho e Vieira Neto (1999), um sistema de visão artificial tradicional é composto pelos seguintes blocos:

- Aquisição da imagem: é a primeira etapa do sistema e sua principal função é adquirir imagens do ambiente e convertê-las para imagens digitais, sendo executadas por uma câmera adequada.

- **Pré-processamento:** a imagem resultante do passo anterior pode apresentar diversas imperfeições, tais como: presença de pixels ruidosos, contraste e/ou brilho inadequado, etc. A função da etapa de pré-processamento é adequar a imagem para as etapas subsequentes. As operações efetuadas nesta etapa são normalmente efetuadas por filtros conforme mencionado anteriormente no tópico 2.5.
- **Segmentação:** A tarefa básica desta etapa é a de dividir uma imagem em suas unidades significativas, ou seja, nos objetos que a compõem, até a obtenção do objeto de interesse, conforme mencionado no tópico 2.6.
- **Extração de características:** Esta etapa procura extrair características da imagem resultante da segmentação através de descritores que permitam caracterizar com precisão cada objeto e que apresentem bom poder de discriminação entre objetos parecidos, como a figura humana com e sem capacete. Estes descritores devem ser representados por uma estrutura de dados adequada ao algoritmo de reconhecimento. Neste projeto, esta função foi atribuída aos histogramas conforme teoria apresentada no tópico 2.4.
- **Reconhecimento e interpretação:** Denomina-se reconhecimento o processo de atribuição de um rótulo a um objeto baseado em suas características, traduzidas por seus descritores. A tarefa de interpretação, também conhecida por classificação, consiste em se atribuir significado a este conjunto de objetos reconhecidos. Nesta pesquisa, estas funções foram executadas pelos comparadores de histogramas mencionados no tópico 2.4.

2.8 Biblioteca OpenCV

OpenCV foi a ferramenta escolhida para a pesquisa. É uma biblioteca de código aberto (*open source*) liberada sob a licença BSD e, gratuita tanto para uso acadêmico e comercial. Ele tem interfaces de C++, C, Python e Java e suporta o Windows, Linux, Mac OS, iOS e Android. Foi projetada para a eficiência computacional e com um forte apelo em aplicações de tempo real. Escrita e otimizada em C / C++, a biblioteca pode tirar proveito do processamento multicore. Um de seus objetivos é simplificar a estrutura da visão computacional. Ela possui mais de 500 funções que podem ser utilizadas nas mais variadas áreas do conhecimento humano como: imagens médicas, controle de qualidade nas indústrias, segurança, robótica e outras.

3. Materiais e métodos

3.1 Materiais.

Os materiais utilizados no desenvolvimento da pesquisa foram: um computador com processador dual core; sistema operacional Ubuntu Linux; biblioteca computacional *opensource* OpenCV; linguagem de programação Python; filmadora com tecnologia CCD (dispositivo de carga acoplada), e filmagem cedida pela Igreja Metodista Livre em conjunto com a empresa Ale Art's.

3.2 Métodos.

A atividade central neste trabalho foi o desenvolvimento do algoritmo de visão de máquina, composto de várias etapas e obedecendo a um fluxo, conforme apresentado no diagrama da figura 14.



Figura 14: fluxo das etapas do algoritmo de visão de máquina. Fonte: autor

3.2.1 Aquisição da imagem.

O primeiro passo no processo foi a aquisição de imagens dos ambientes. A condição ideal mas não imprescindível é a filmagem inicial, por alguns segundos, do ambiente sem as pessoas. Caso não seja possível, o objetivo do algoritmo não será prejudicado, pois o segmentador (citado na etapa seguinte) consegue se adaptar às novas situações. Os comandos do programa que estabelecem a aquisição, são descritos a seguir:

Para vídeos gravados:

```
import cv2
cap = cv2.VideoCapture("A1.avi")
while(True):
    ret, frame = cap.read()
```

Para filmagens em tempo real:

```
import cv2
cap = cv2.VideoCapture(0)
while(True):
    ret, frame = cap.read()
```

Devido ao fato de a luminância interferir nas etapas posteriores, conforme já exposto em teoria no item 2.4.5, optou-se em colher modelos do próprio ambiente, da filmagem corrente e de pessoas com e sem capacetes. No programa, ao se teclar a letra “a”, “b” ou “c” são estabelecidos modelos de cabeças com capacete. Ao se teclar a letra “d” são estabelecidos modelos de cabeças sem capacete. Os comandos do programa para se estabelecer um modelo de cabeça com capacete, teclando-se a letra “a”, são colocados a seguir:

```
key = cv2.waitKey(5)
if key == ord('a'):
    hista = cv2.calcHist( [blur], [0], None, [10], [1,180] )
    cv2.normalize(hista,hista,0,100,cv2.NORM_MINMAX)
```

Onde:

- *key* – variável a espera de um comando pelo teclado.

- *cv2.waitKey(5)* – comando de teclado com tempo determinado de 5 milisegundos.
- *if key == ord('a')* - condição de execução de programa ao se teclar a letra 'a'.
- *hista* - nome de variável atribuída a um determinado histograma (detalhes mais adiante) *cv2.normalize* – função para normalizar histogramas.

Porém existe a flexibilidade de se definir previamente os modelos, bastando para isto, estabelecer os modelos antes da captura dos quadros do vídeo ou da filmagem. Os comandos são descritos a seguir:

```
capacete1 = cv2.imread("14.png")
capacete4 = cv2.imread("18.png")
hista = cv2.calcHist( [capacete1], [0], None, [180], [1,180] )
hisd = cv2.calcHist( [capacete1], [0], None, [180], [1,180] )
hista4 = cv2.calcHist( [capacete4], [0], None, [45], [1,180] )
hisd4 = cv2.calcHist( [capacete4], [0], None, [45], [1,180] )
```

Onde:

- *capacete1, capacete4* - nomes de variáveis atribuídas a imagens.
- *cv2.imread("14.png")* – leitura de imagem, neste caso da imagem 14.png.
- *hista, hisd, hista4 e hisd4* - nomes de variáveis atribuídas a histogramas (detalhes mais adiante).

Em termos físicos, utilizou-se uma câmera e filmadora com tecnologia CCD, cujo principal atributo requerido foi a conversão de imagem analógica em imagem digital.

3.2.2 Segmentação.

A tarefa básica da etapa de segmentação é dividir uma imagem em objetos de interesse. Para melhor entendimento, a segmentação foi dividida em quatro fases. Na primeira fase, utilizou-se a biblioteca OpenCV em ambiente Ubuntu/Linux, mais especificamente os algoritmos MOG (Mistura de Gaussianas – versão KaewTraKulPong e Bowden, 2001) ou MOG2 (Mistura de Gaussianas – versão Zivkovic and van der Heijden, 2004) para obter a segmentação das figuras humanas em movimento. As linhas do programa que estabelecem esta

fase são:

```

    backsub = cv2.BackgroundSubtractorMOG2()
    .
    .
    .
    if ret:
        fgmask = backsub.apply(frame, None, 0.01)
        contours, hierarchy = cv2.findContours(fgmask.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
        try: hierarchy = hierarchy[0]
        except: hierarchy = []

```

Onde:

- *cv2.BackgroundSubtractorMOG2()* - função de subtração de fundo de imagem.
- *contours, hierarchy = cv2.findContours(fgmask.copy(), cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)* - comando para estabelecer o contorno externo da imagem em movimento.

Na segunda fase, através de recursos de programação Python, definiu-se o tamanho mínimo (50 (L) x 80 (h) pixels) e máximo (300 (L) x 400 (h) pixels) das janelas de detecção, no intuito de trabalhar somente com imagens a curta e a média distância das câmeras. Na terceira fase, utilizando novamente os recursos de programação Python, gerou-se uma sub imagem referente a região da cabeça das pessoas, consistindo em 1/10 da altura das imagens em movimento. Na quarta fase, para melhorar a precisão, foi efetuada também a remoção do fundo da sub imagem com o uso de máscaras, resultando com isto somente a imagem da cabeça para ser trabalhada nas próximas etapas. Os comandos de programação que estabelecem a segunda, terceira e quarta fase, estão descritos a seguir:

```

for contour, hier in zip(contours, hierarchy):
    (x,y,w,h) = cv2.boundingRect(contour)
    if (w > 50 and h > 80) and (w < 300 and h < 400 ):
        fatia = frame[y:y+h//10, x:x+w]

```

```

fatia1 = fgmask[y:y+h//10, x:x+w]
res = cv2.bitwise_and(fatia,fatia, mask= fatia1)

```

Onde:

- (x,y,w,h) – coordenadas (x,y) e dimensões do retângulo (w,h) .
- *cv2.boundingRect(contour)* – função para delimitar com retângulo os limites do contorno de uma imagem.
- *cv2.bitwise_and(fatia,fatia, mask = fatia1)* - remoção do fundo da imagem através de máscara.

3.2.3 Conversão do sistema de cores RGB para HSV

Com os recursos do OpenCV, a representação da imagem foi convertida do sistema de cores RGB (do inglês: red, green e blue), original no processo de aquisição, para o sistema HSV (do inglês: hue, saturation e value), mais conveniente para o algoritmo de visão de máquina a partir desta etapa. A principal justificativa para a conversão está no fato de o sistema RGB requerer um vetor de três posições para efetuar operações, ou seja, um alto custo computacional. Nesta dissertação, utilizando o sistema HSV, houve a possibilidade de trabalhar com vetores de uma posição, mais especificamente na dimensão H ou S, conforme será mencionado em tópicos adiante. É necessário comentar que a biblioteca OpenCV adota limites próprios e diferentes daqueles apresentados na teoria contida no item 2.3.2. A tabela 2 mostra um comparativo paralelo entre estas diferenças.

Tabela 2: comparativo entre os limites HSV da biblioteca OpenCV e a teoria apresentada tópico 2.3.2

	Biblioteca OpenCV	Teoria 2.3.2
Limites de H	0 a 180°	0 a 360°
Limites de S	0 a 255	0 a 100%
Limites de V	0 a 255	0 a 100%

A diferença também está na nomenclatura RGB, que é denominada BGR na biblioteca OpenCV; somente uma troca de posição no vetor, entre os valores de R (vermelho) e B (azul).

Apesar destas diferenças, a conversão de sistemas não é prejudicada. É dada pela seguinte linha do programa:

```
hsv = cv2.cvtColor(res, cv2.COLOR_BGR2HSV)
```

Onde:

- *hsv* – nome dado a imagem convertida para HSV.
- *cv2.cvtColor* – função de conversão de sistemas de cores.
- *res* – imagem de entrada.
- *cv2.COLOR_BGR2HSV* – conversão RGB (BGR no OpenCV) para HSV.

3.2.4 Filtragem

A função da etapa de filtragem é aprimorar a qualidade da imagem para a etapa subsequente da descrição. As operações efetuadas nesta etapa são de baixo nível porque trabalham diretamente com os valores de intensidade dos pixels. A imagem resultante desta etapa é uma imagem digitalizada de melhor qualidade que a original. A qualidade mencionada neste tópico, não se trata de uma percepção visual humana como a nitidez, suavização ou contraste; mas a uma percepção artificial que poderá ser mensurada somente na última etapa do algoritmo de visão de máquina. Nesta dissertação, através dos recursos do OpenCV foram implementados os filtros bilateral, gaussiano, de média e de mediana. Normalmente a filtragem é realizada após a aquisição, sobre toda a imagem, porém para reduzir o custo computacional foi realizada somente na sub imagem referente a cabeça das pessoas. Os comandos utilizados nesta etapa foram:

```
blur = cv2.GaussianBlur(res1,(15,15),0)
```

```
bilateral = cv2.bilateralFilter(res1,15,75,75)
```

```
smoothed = cv2.filter2D(res1,-1,kernel)
```

```
median = cv2.medianBlur(res1,15)
```

Onde:

- *blur*, *bilateral*, *smoothed* e *median*: referem-se a nomes aleatórios das imagens filtradas.
- *cv2.GaussianBlur*: função filtro gaussiano.
- *cv2.bilateralFilter*: função filtro bilateral.
- *cv2.filter2D*: função filtro de média.
- *cv2.medianBlur*: função filtro de mediana.
- *res1*: corresponde a a imagem de entrada.

3.2.5 Descrição – cálculo de histogramas.

A descrição consiste em extrair as características da imagem resultante das etapas anteriores - através do cálculo de histogramas na dimensão H para capacetes coloridos e cabeças humanas; e na dimensão S para capacetes brancos e cabeças humanas – dimensões estas referentes ao sistema de cores HSV. Na biblioteca OpenCV, a cor azul ocupa uma região entre 105° e 130° e a amarela entre 15° e 40° da dimensão H, enquanto a cor branca entre 0 e 20 da dimensão S. É importante ressaltar que, sendo a cor branca considerada a junção de todas as cores, ela não ocupa uma região específica na região H. Em termos práticos: para valores de S entre 0 e 20, preferencialmente mais próximos de zero, combinados com qualquer valor de H, resultará sempre a cor branca. À medida que os valores de S vão aumentando, os matizes vão se sobressaindo. Por exemplo: aumentando-se o valor de S gradativamente de 0 a 255, mantendo-se o valor de H em 115° e fixando um valor fixo de luminância em 248; resultaria nas cores branco (S = 5), azul claro (S = 53), azul celeste (S = 107), azul puro (S = 252). A variação é a mesma para qualquer matiz.

Quanto aos experimentos efetuados, os descritores permitiram caracterizar com boa precisão as cabeças humanas com e sem os capacetes. Nesta etapa, há a conversão de imagens (entrada) para um conjunto de dados correspondentes (saída). No tópico - resultados obtidos – estes dados foram utilizados para gerar um histograma gráfico para melhor visualização e entendimento do processo. Os comandos de programação para esta etapa estão descritos a seguir:

Para capacetes coloridos e cabeças humanas:

```
hista2 = cv2.calcHist( [bilateral], [0], None, [45], [1,180] )
cv2.normalize(hista2,hista2,0,100,cv2.NORM_MINMAX)
```

Onde:

- *hista2* - nome dado ao conjunto de dados que representam o histograma da imagem.
- *cv2.calcHist* – função para calcular o histograma.
- *[bilateral]* – refere-se a imagem de entrada; neste caso já com filtragem bilateral.
- *[0]* – corresponde ao campo hue do sistema HSV.
- *None* – refere-se a máscara; neste caso inexistente.
- *[45]* – número de intervalos de classe.
- *[1,180]* – extensão, neste caso do campo hue.
- *cv2.normalize* – função para normalizar histogramas.

Para capacetes brancos e cabeças humanas:

```
hisd2 = cv2.calcHist( [bilateral], [1], None, [64], [1,255] )
cv2.normalize(hisd2,hisd2,0,100,cv2.NORM_MINMAX)
```

Onde:

- *[1]* – corresponde ao campo saturation do sistema HSV.

A definição é a mesma para os demais campos, alterando somente os valores.

3.2.6 Classificação.

Nesta etapa, foram utilizados os comparadores de histogramas para o reconhecimento de pessoas que não estão utilizando capacetes. Com os recursos disponíveis do OpenCV, foram implantados os comparadores: distância Bhattacharyya, correlação, qui quadrado e intersecção.

O processo inicia quando os histogramas das sub imagens correntes são comparados com os histogramas dos modelos de cabeças de pessoas com e sem os capacetes. Esta comparação gera uma magnitude. No caso do comparador Bhattacharyya, uma magnitude menor implica em maior semelhança dos histogramas. Por exemplo: se a comparações dos histogramas gerarem magnitudes de 0,7254 (imagem corrente x modelo capacete) e 0,2358 (imagem corrente x modelo cabeça), implica em maior proximidade da imagem corrente com o modelo de cabeças de pessoas, efetuando a detecção através da imagem da pessoa sendo envolvida por um retângulo vermelho no frame corrente. Em contrapartida, se a comparação implicar em maior proximidade com histogramas de modelos com capacetes, então o algoritmo não executará nenhuma tarefa.

No programa, as linhas de comando que definem esta etapa da classificação são colocadas a seguir:

```

Correlation_a = cv2.compareHist(hista, histfatia,
cv2.cv.CV_COMP_CORREL)
Chi_Square_a = cv2.compareHist(hista, histfatia,
cv2.cv.CV_COMP_CHISQR)
Intersection_a = cv2.compareHist(hista,histfatia,
cv2.cv.CV_COMP_INTERSECT)
Bhattacharyya_a = cv2.compareHist(hista, histfatia,
cv2.cv.CV_COMP_BHATTACHARYYA)

Correlation_d = cv2.compareHist(histd, histfatia,
cv2.cv.CV_COMP_CORREL)
Chi_Square_d = cv2.compareHist(histd, histfatia,
cv2.cv.CV_COMP_CHISQR)
Intersection_d = cv2.compareHist(histd,histfatia,
cv2.cv.CV_COMP_INTERSECT)
Bhattacharyya_d = cv2.compareHist(histd, histfatia,
cv2.cv.CV_COMP_BHATTACHARYYA)

if Bhattacharyya_d < Bhattacharyya_a:

```

```

    best_id = best_id + 1
    cv2.rectangle(frame, (x,y), (x+w, y+h), (255, 0, 0), 2)
    cv2.putText(frame, str(best_id), (x,y-5), cv2.FONT_HERSHEY_SIMPLEX,0.5,
(255, 0, 0), 2)

```

Onde:

- *Correlation_a* - variável que conservará provisoriamente a magnitude de comparação.
- *cv2.compareHist* – função para comparar histogramas.
- *hista* – refere-se ao histograma de um dos modelos que foi gerado na etapa de descrição.
- *histfatia* – corresponde ao histograma da subimagem corrente.
- *cv2.cv.CV_COMP_CORREL* – aplicação da correlação, um dos comparadores.

A definição é a mesma para as linhas que seguem o mesmo formato, exceto para:

- *cv2.cv.CV_COMP_CHISQR* – aplicação do comparador qui quadrado.
- *cv2.cv.CV_COMP_INTERSECT* – aplicação do comparador intersecção.
- *cv2.cv.CV_COMP_BHATTACHARYYA* – aplicação do comparador Bhattacharyya.

Ao efetuar a detecção dado por:

if Bhattacharyya_d < Bhattacharyya_a:

Os seguintes comandos serão executados:

- *best_id = best_id + 1* : em várias detecções no mesmo frame, refere-se ao número da pessoa que foi detectada.
- *cv2.rectangle(frame, (x,y), (x+w, y+h), (255, 0, 0), 2)*: função para contornar com retângulo, localização (*frame*), coordenadas e dimensões do retângulo (*(x,y), (x+w, y+h)*) e cor da linha (*255, 0, 0*) dado em BGR.
- *cv2.putText(frame, str(best_id), (x,y-5), cv2.FONT_HERSHEY_SIMPLEX,0.5, (255, 0, 0), 2)*: inclusão de texto no retângulo, sua localização, coordenadas e cor.

3.2.7 Integração filtragem-descrição-classificação.

Neste projeto, a filtragem, a descrição e a classificação tiveram que ser efetuadas concomitantemente para a descoberta da melhor combinação para atingir o objetivo traçado. No item – resultados e discussão – esta abordagem está mais detalhada.

A figura 15 apresenta um fluxo ilustrativo com as etapas do algoritmo para melhor entendimento.

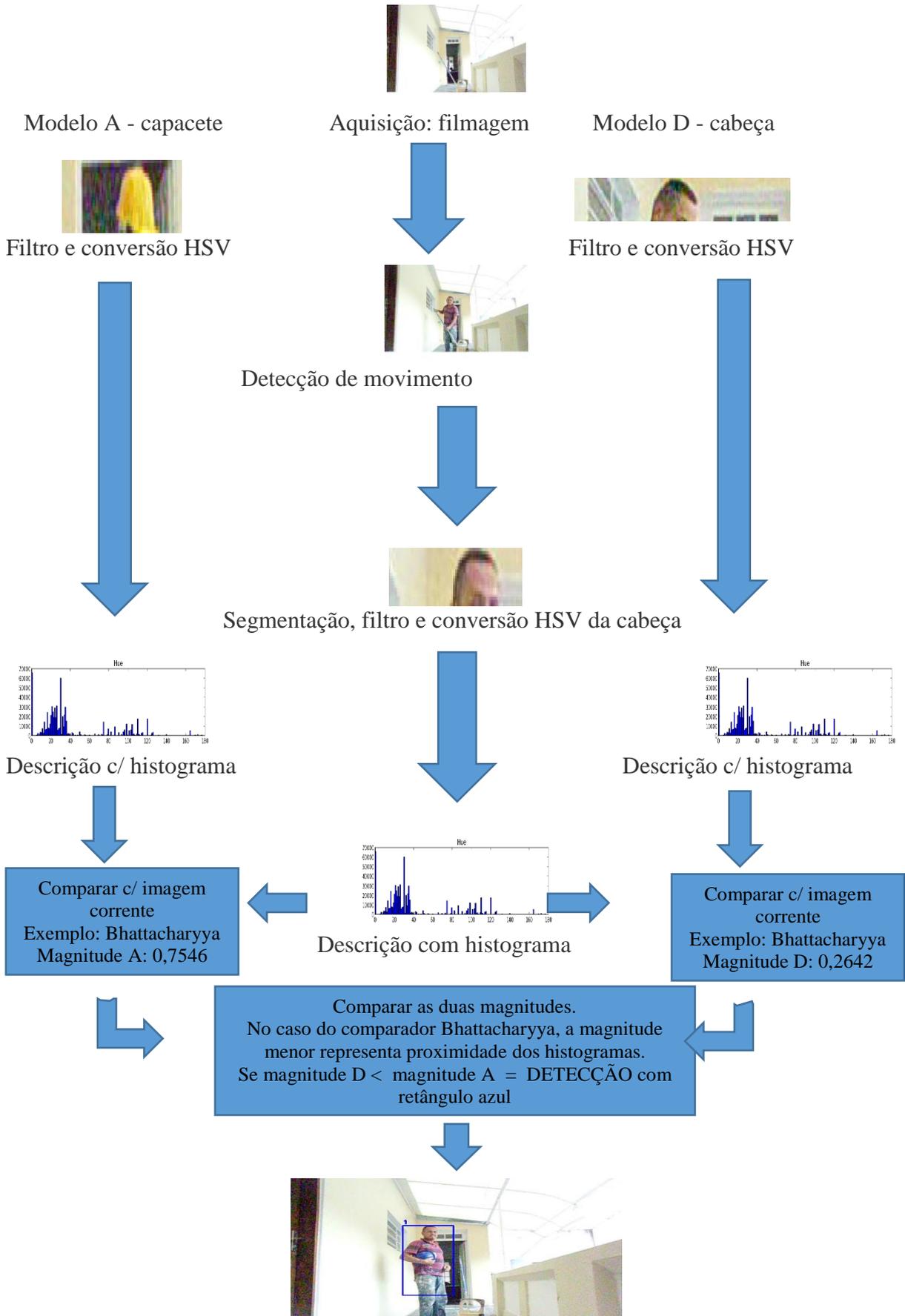


Figura 15: resumo ilustrativo do algoritmo de visão de máquina. Fonte: autor

4. Resultados e discussão

4.1 Etapa da segmentação:

Os algoritmos foram selecionados conforme observação dos ambientes de pesquisa. No caso de um canteiro de obras relacionados com a construção civil, foi observado que os trabalhadores normalmente estão atuando com intenso movimento corporal. Devido a estas características, foram testadas as segmentações por movimento MOG e MOG2. Ambos os algoritmos apresentaram 100% de detecção de movimento dos vídeos analisados. Porém, o algoritmo MOG, no caso específico, teve dificuldades em diferenciar o capacete branco com o fundo branco, conforme mostra a figura 16.

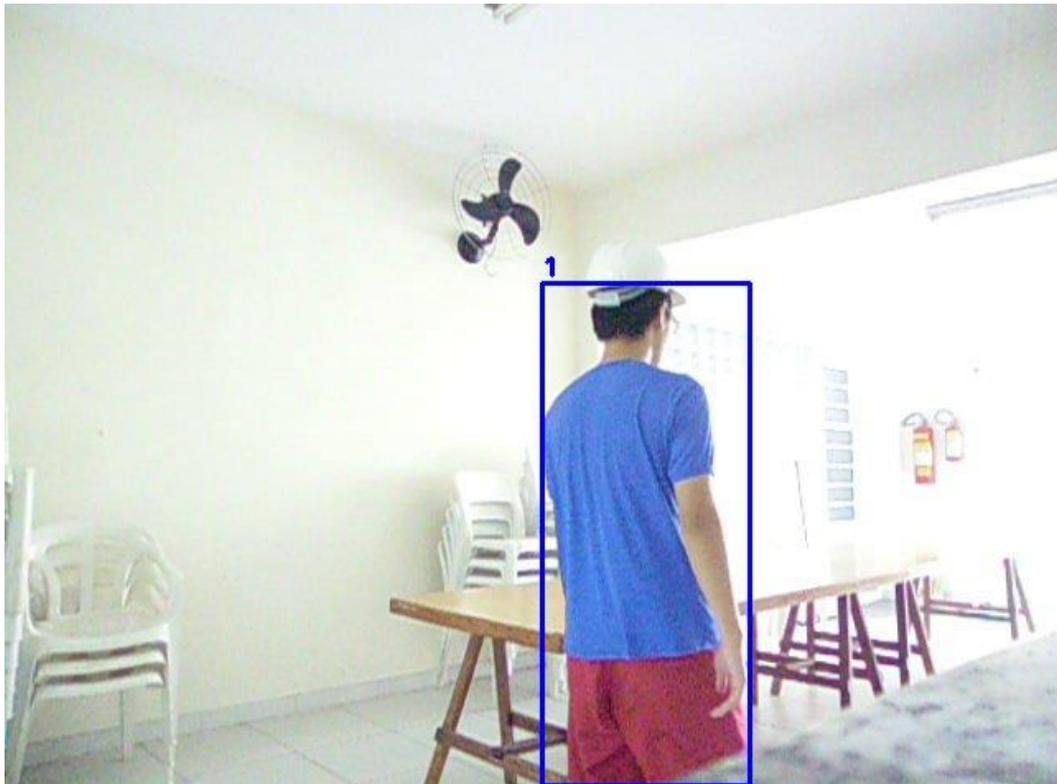


Figura 16: segmentação por movimento MOG. Fonte: autor.

O segmentador MOG2 apresentou melhor desempenho, pois conseguiu distinguir o capacete branco do fundo branco. O resultado obtido conforme figura 17 comprova sua eficácia.



Figura 17: segmentação por movimento MOG2. Fonte: autor

As imagens de interesse, referente a metade superior da cabeça das pessoas, são então obtidas a partir das imagens detectadas pela segmentação MOG2. Normalmente, em boa parte de seu expediente, foi observado que o trabalhador atua na posição de pé. Considerando estas circunstâncias, a metade superior das cabeças corresponde a aproximadamente a 1/10 superior da altura das pessoas. Através dos recursos da biblioteca OpenCV, foi possível conseguir estas sub imagens. Um exemplo destes resultados está expresso na figura 18.



Figura 18: Geração de imagem de interesse a partir de sub imagem do segmentador MOG2. Fonte: autor

No entanto, é necessário extrair o fundo destas sub imagens para se comparar com exatidão somente os dados referentes à cabeça dos trabalhadores. Isto é obtido utilizando-se mais uma vez, os recursos de programação Python e também do OpenCV, através de máscaras conforme ilustrado na figura 19.



Figura 19: exemplo de uma das máscaras utilizadas na pesquisa. Fonte: autor

No decorrer das aplicações dos algoritmos, houve a geração de algumas contingências como por exemplo a detecção de movimento do cabeça separada do corpo. Foi mitigada uma parte destas contingências com o aumento da janela de detecção. Porém é um recurso limitado, principalmente quando se deseja detectar imagens um pouco mais distantes da câmera. Até o final desta pesquisa, não foi encontrada uma solução viável. A figura 20 exemplifica esta contingência.

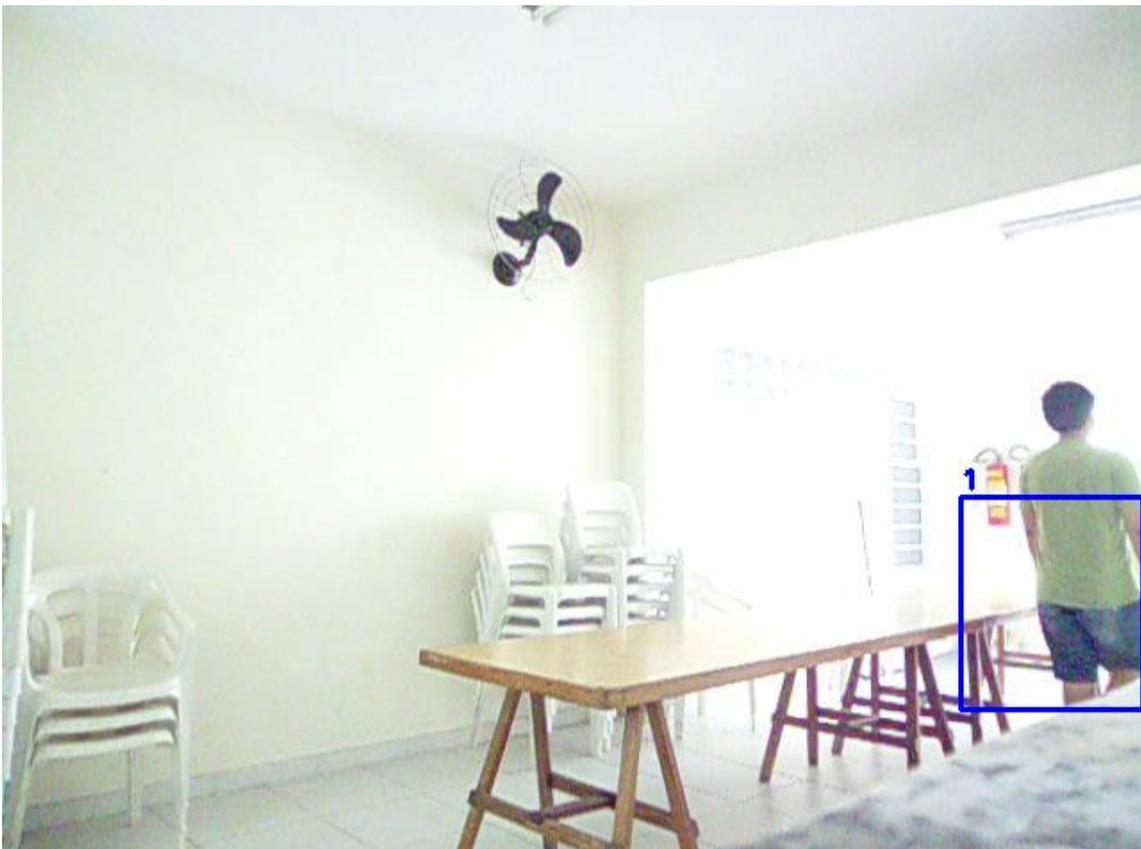


Figura 20: contingência no algoritmo MOG2 – detecção de movimento do tronco separado da cabeça. Fonte: autor

4.2 Etapa de definição do número de intervalos de classes de histogramas, filtros e classificadores-comparadores:

A obtenção do número mais conveniente de intervalos de classe de histogramas não pode ser feita desvinculando-se filtros e classificadores. Foram efetuadas algumas simulações acerca da influência do número de intervalos de classe de histogramas nos resultados de classificação, cujos resultados são apresentados e comentados a seguir.

A figura 21 refere-se a imagem segmentada do topo de uma cabeça humana, enquanto, na figura 22, a imagem é da mesma cabeça, porém utilizando capacete amarelo. O gráfico da figura 23 mostra os seus histogramas respectivamente nas cores azul e vermelho, considerando 45 intervalos de classe e a utilização de filtro bilateral. Já na figura 24, a mesma representação é efetuada para 180 intervalos de classe.



Figura 21: imagem segmentada do topo de uma cabeça humana. Fonte: autor



Figura 22: imagem segmentada do topo de uma cabeça humana utilizando capacete amarelo. Fonte: autor

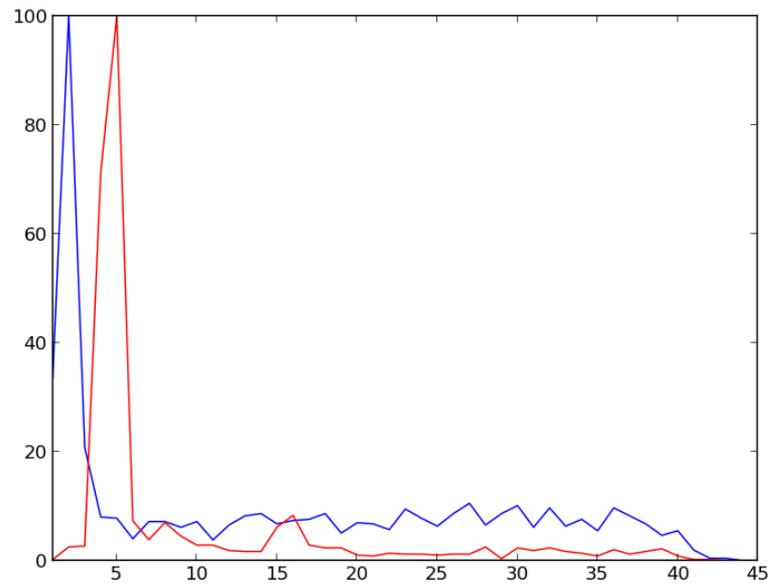


Figura 23: gráficos representando os histogramas de 45 intervalos de classe com a utilização de filtro bilateral, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).

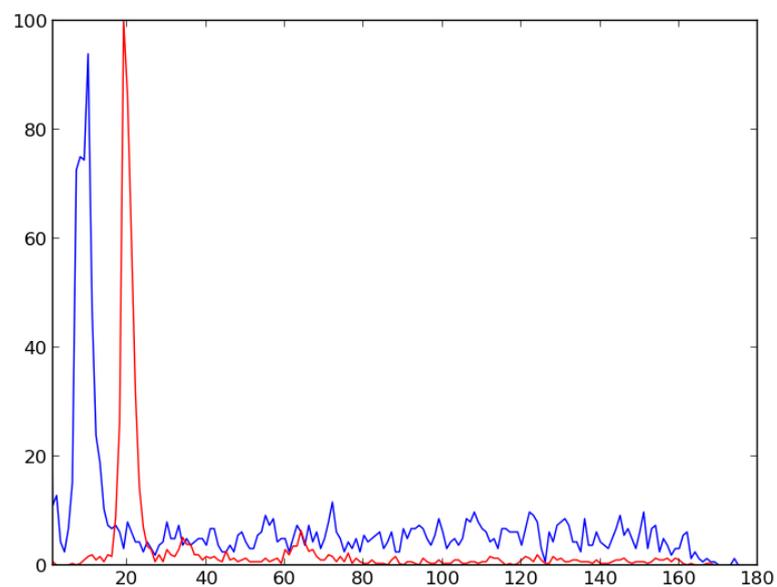


Figura 24: gráficos representando os histogramas de 180 intervalos de classe com a utilização de filtro bilateral, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).

A tabela 3 mostra a influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro bilateral.

Tabela 3: influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro bilateral.

	45 intervalos de classe	180 intervalos de classe
Bhattacharyya	0,6174	0,6593
Correlação	-0,0387	-0,0281
Qui quadrado	17452	36092
Intersecção	102,39	208,11

No caso do Bhattacharyya é desejável que o valor esteja mais próximo de 1,0000 (incompatibilidade máxima) e distante de 0,000 (histogramas idênticos). No caso específico deste classificador-comparador, mantido o filtro bilateral, mostra um melhor resultado para 180 intervalos de classe, conforme a tabela1.

Na correlação, é desejável que o valor esteja entre -1,0000 (incompatibilidade máxima) e zero (nível mínimo de similaridade). O valor conseguido de -0,0387 aponta uma melhor sensibilidade para 45 intervalos de classe.

Para o qui quadrado existe valor definido somente para similaridade máxima, de magnitude zero, referente a comparação de dois histogramas idênticos. Portanto, a incompatibilidade pode gerar valores elevados, se comparados a outros. A magnitude de 36.092 indica superioridade para 180 intervalos de classe.

Já na intersecção, o valor desejado é a magnitude mínima, conseguida através de 45 intervalos de classe, no valor de 102,39.

Portanto não há predominância de um intervalo de classe específico, no caso do filtro bilateral.

As figuras 25 e 26 e tabela 4 seguem a mesma linha de raciocínio, porém com a utilização de filtro de Gauss.

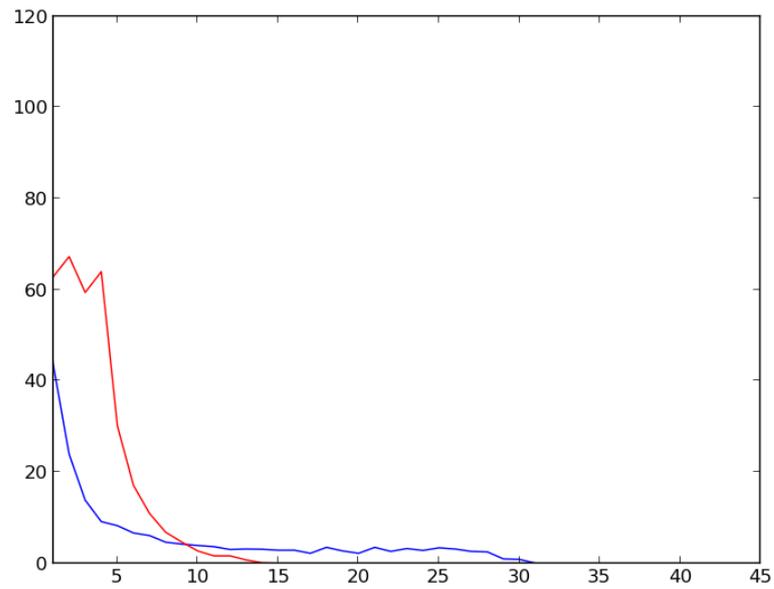


Figura 25: gráficos representando os histogramas de 45 intervalos de classe com a utilização de filtro gaussiano, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).

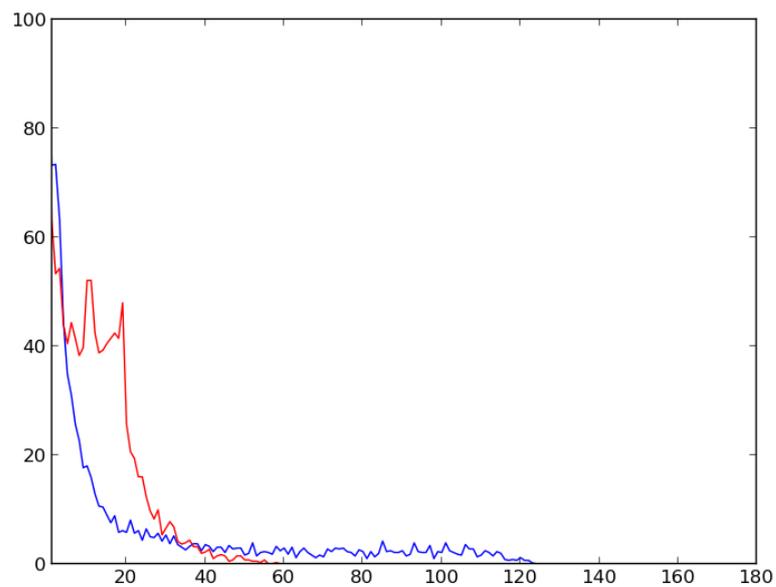


Figura 26: gráficos representando os histogramas de 180 intervalos de classe com a utilização de filtro gaussiano, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).

Tabela 4: influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro gaussiano.

	45 intervalos de classe	180 intervalos de classe
Bhattacharyya	0,3657	0,3719
Correlação	0,8333	0,8292
Qui quadrado	251,48	468,97
Intersecção	227,11	660,02

A tabela 4, com os dados da utilização do filtro gaussiano, mostra que houve uma sensível piora nas magnitudes de todos os classificadores-comparadores, independentemente dos intervalos de classe, se comparados com a tabela 3.

A propriedade do filtro gaussiano de suavização da imagem comprometeu a interpretação dos classificadores-comparadores.

O pior caso é o da correlação, que saiu de uma condição de incompatibilidade, expressa pelas magnitudes negativas conforme tabela 3; para uma condição próxima de similaridade com valores de 0,8333 (45 intervalos de classe) e 0,8292 (180 intervalos de classe) conforme tabela 4. Lembrando que a similaridade máxima é expressa entre histogramas idênticos e com magnitude de 1,0000.

Quanto aos intervalos de classe, a utilização do filtro gaussiano fez os classificadores-comparadores Bhattacharyya, correlação e qui quadrado apresentarem melhores resultados em 45 intervalos de classe, enquanto a intersecção teve um melhor desempenho em 180 intervalos de classe.

A seguir as figuras 27 e 28 e tabela 5, mostram os dados obtidos com a utilização de filtro de média.

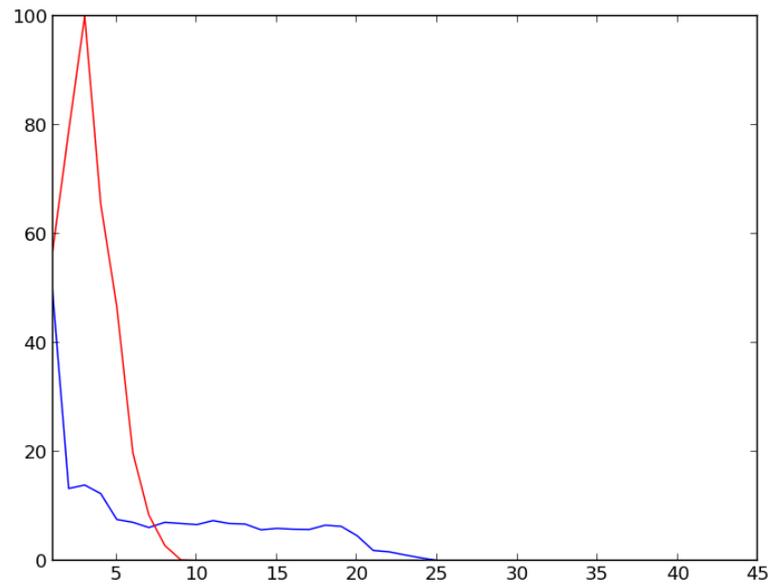


Figura 27: gráficos representando os histogramas de 45 intervalos de classe com a utilização de filtro de média, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).

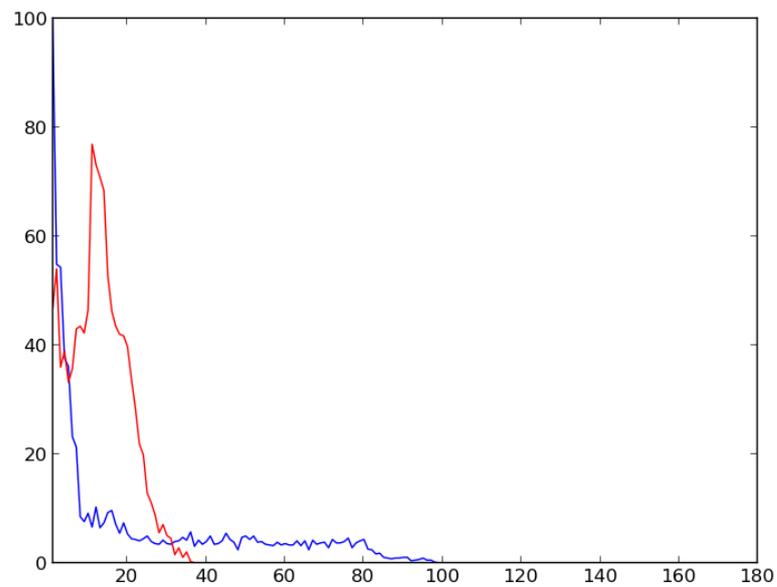


Figura 28: gráficos representando os histogramas de 180 intervalos de classe com a utilização de filtro de média, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha). (Fonte: autor).

Tabela 5: influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro de média.

	45 intervalos de classe	180 intervalos de classe
Bhattacharyya	0,4748	0,4978
Correlação	0,6711	0,5524
Qui quadrado	696,45	862,13
Intersecção	201,25	434,60

A tabela 5, com os dados da utilização do filtro de média, mostra que houve uma melhora nas magnitudes de todos os classificadores-comparadores, independentemente dos intervalos de classe, se comparados com a tabela 4, porém distante ainda dos índices apresentados na tabela 3.

A propriedade do filtro de média, assim como o filtro gaussiano, é de suavização. Também comprometeu a interpretação dos classificadores-comparadores mas em menor proporção.

Quanto aos intervalos de classe, a utilização do filtro de média fez os classificadores-comparadores Bhattacharyya, correlação e qui quadrado apresentarem melhores resultados em 45 intervalos de classe, enquanto a intersecção teve um melhor desempenho em 180 intervalos de classe.

A seguir, as figuras 29 e 30 e tabela 6, mostram os dados com a utilização de filtro de mediana.

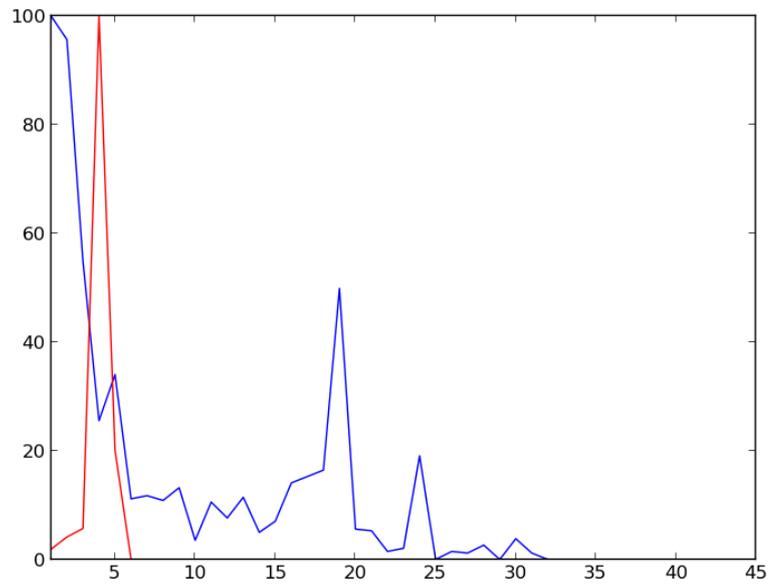


Figura 29: gráficos representando os histogramas de 45 intervalos de classe com a utilização de filtro de mediana, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha).
(Fonte: autor).

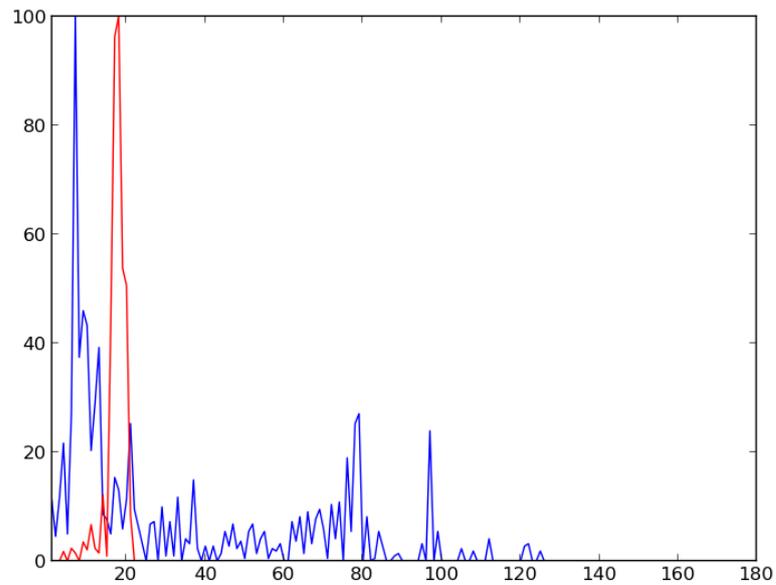


Figura 30: gráficos representando os histogramas de 180 intervalos de classe com a utilização de filtro de mediana, referente ao topo da cabeça humana sem capacete (cor azul) e com capacete amarelo (cor vermelha).
(Fonte: autor).

Tabela 6: influência dos intervalos de classe nas magnitudes dos classificadores-comparadores, considerando a utilização de filtro de mediana.

	45 intervalos de classe	180 intervalos de classe
Bhattacharyya	0,7244	0,7710
Correlação	0,1673	0,1320
Qui quadrado	11556	10251
Intersecção	57,372	89,934

A tabela 6, com os dados da utilização do filtro de mediana, mostra que as magnitudes de todos os classificadores-comparadores, independentemente dos intervalos de classe, são melhores que os dos filtros de suavização. Se comparado com o filtro bilateral, da tabela 3, mostra melhores resultados com a utilização do classificador-comparador intersecção.

Quanto aos intervalos de classe, a utilização do filtro de mediana fez os classificadores-comparadores Bhattacharyya e correlação apresentarem melhores resultados em 45 intervalos de classe, enquanto qui quadrado e intersecção tiveram melhores desempenhos em 180 intervalos de classe.

A tabela 7 expõem de forma mais ampla e direcionada, a influência dos filtros nas magnitudes dos classificadores-comparadores. Para gerar esta tabela, foram utilizados os dados contidos nas tabelas 3 a 6.

Tabela 7: influência dos filtros nas magnitudes dos classificadores-comparadores.

	Bilateral		Gauss		Média		Mediana	
	45	180	45	180	45	180	45	180
Intervalos de classes								
Bhattacharyya	0,6174	0,6593	0,3657	0,3719	0,4748	0,4978	0,7244	0,771
Correlação	-0,0387	-0,0281	0,8333	0,8292	0,6711	0,5524	0,1673	0,132
Qui quadrado	17452	36092	251,48	468,97	696,45	862,13	11556	10251
Intersecção	102,39	208,11	227,11	660,02	201,25	434,6	57,372	89,934

A detecção de pessoas que não estão utilizando capacetes depende exclusivamente dos valores gerados pelos classificadores-comparadores. Para comparar dois histogramas distintos, um referente a cabeça humana com capacete e outro sem capacete, é desejável que o classificador-comparador Bhattacharyya esteja mais próximo de 1,0000 (incompatibilidade máxima). Para comparar dois histogramas próximos, referentes a duas cabeças humanas com capacetes ou duas sem capacetes, é desejável que o classificador-comparador Bhattacharyya esteja mais próximo de 0,0000 (similaridade perfeita). Para comparar dois histogramas

distintos, os dados da tabela 7 indicam que o valor máximo conseguido foi de 0,7244 referente a combinação 45 intervalos de classe com filtro de mediana.

Na correlação, é desejável que o valor esteja entre -1,0000 (incompatibilidade máxima) e zero (nível mínimo de similaridade) para comparar dois histogramas distintos. O valor mínimo conseguido na tabela 7, foi de -0,0387 referente a combinação 45 intervalos de classe com filtro bilateral.

Para o qui quadrado, existe valor definido somente para similaridade máxima, de magnitude zero, referente a comparação de dois histogramas idênticos. Portanto, a incompatibilidade pode gerar valores elevados, se comparados a outros. Conforme a tabela 7, a maior magnitude foi de 36.092, observada na combinação de 180 intervalos de classe com filtro bilateral.

Já na intersecção, o valor desejado é a magnitude mínima, conseguida através da combinação de 45 intervalos de classe com filtro de mediana, no valor de 57,372.

Portanto, neste contexto, não há uma combinação predominante, se forem considerados somente estes dados.

Foram realizados então, testes de campo, mensurando o nível de detecção de pessoas que não estão utilizando capacetes e os respectivos falsos positivos. Foram utilizados capacetes amarelos, brancos e azuis.

Desta forma, foram efetuadas todas as combinações possíveis entre: 45 e 180 intervalos de classe de histogramas; com os filtros bilateral, de Gauss, de média e de mediana; e também com os classificadores-comparadores correlação, intersecção, qui quadrado e distância Bhattacharyya. As tabelas de 8 a 10 indicam os resultados obtidos. Vale ressaltar que para mensurar a detecção, foram analisadas filmagens em que pessoas sem capacete estavam em 100% dos frames. O mesmo procedimento foi efetuado para mensurar os falsos positivos, porém utilizando filmagens em que pessoas com capacete estavam em 100% dos frames.

Tabela 8: resultado da detecção de pessoas que não estão utilizando capacetes e os respectivos falsos positivos, comparando-se os modelos de cabeças com os capacetes amarelos. Foram utilizadas todas combinações possíveis entre intervalos de classe, filtros e classificadores-comparadores.

Intervalos de classes de histogramas. Modelos de cabeças comparados com modelos de capacetes amarelos.	Filtros	Classificador-comparador	Detecção de cabeças sem capacete	Falsos positivos	Acurácia
180	Bilateral	Bhattacharyya	88,80%	17,32%	73,42%
		Correlação	80,94%	14,20%	69,44%
		Qui quadrado	42,28%	5,20%	40,08%
		Intersecção	97,02%	40,18%	58,03%
180	Gauss	Bhattacharyya	77,01%	9,92%	69,37%
		Correlação	99,55%	29,99%	69,69%
		Qui quadrado	82,81%	25,03%	62,08%
		Intersecção	100,00%	68,63%	31,37%
180	Média	Bhattacharyya	63,74%	4,96%	60,58%
		Correlação	78,21%	14,88%	66,57%
		Qui quadrado	0,90%	0,12%	0,90%
		Intersecção	96,29%	44,52%	53,42%
180	Mediana	Bhattacharyya	85,17%	10,73%	76,04%
		Correlação	79,84%	13,03%	69,43%
		Qui quadrado	40,24%	9,11%	36,57%
		Intersecção	90,51%	38,99%	55,22%
45	Bilateral	Bhattacharyya	92,95%	17,76%	76,44%
		Correlação	75,68%	7,73%	69,83%
		Qui quadrado	0,00%	0,00%	0,00%
		Intersecção	93,67%	20,30%	74,66%
45	Gauss	Bhattacharyya	67,09%	5,07%	63,68%
		Correlação	98,28%	26,64%	72,10%
		Qui quadrado	57,50%	7,61%	53,13%
		Intersecção	99,46%	31,72%	67,91%
45	Média	Bhattacharyya	68,44%	6,81%	63,79%
		Correlação	89,87%	22,72%	69,45%
		Qui quadrado	16,64%	0,12%	16,62%
		Intersecção	94,12%	27,91%	67,85%
45	Mediana	Bhattacharyya	86,98%	11,30%	77,15%
		Correlação	83,00%	29,99%	58,11%
		Qui quadrado	51,63%	4,61%	49,25%
		Intersecção	91,77%	61,71%	35,14%

Tabela 9: resultado da detecção de pessoas que não estão utilizando capacetes e os respectivos falsos positivos, comparando-se os modelos de cabeças com os capacetes azuis. Foram utilizadas todas combinações possíveis entre intervalos de classe, filtros e classificadores-comparadores.

Intervalos de classes de histogramas. Modelos de cabeças comparados com modelos de capacetes azuis.	Filtros	Classificador-comparador	Deteção de cabeças sem capacete	Falsos positivos	Acurácia
180	Bilateral	Bhattacharyya	84,10%	16,39%	70,31%
		Correlação	88,62%	25,06%	66,41%
		Qui quadrado	70,64%	26,81%	51,70%
		Intersecção	74,07%	9,37%	67,14%
180	Gauss	Bhattacharyya	39,84%	18,85%	32,33%
		Correlação	13,28%	1,64%	13,06%
		Qui quadrado	90,97%	88,29%	10,65%
		Intersecção	4,61%	0,23%	4,60%
180	Média	Bhattacharyya	78,41%	29,98%	54,91%
		Correlação	68,47%	62,76%	25,50%
		Qui quadrado	46,25%	48,24%	23,94%
		Intersecção	3,34%	3,04%	3,24%
180	Mediana	Bhattacharyya	78,23%	15,22%	66,32%
		Correlação	83,83%	21,08%	66,16%
		Qui quadrado	96,57%	97,42%	2,49%
		Intersecção	79,67%	15,69%	67,17%
45	Bilateral	Bhattacharyya	82,75%	15,11%	70,25%
		Correlação	74,53%	7,49%	68,94%
		Qui quadrado	95,12%	86,53%	12,81%
		Intersecção	90,70%	42,51%	52,14%
45	Gauss	Bhattacharyya	79,67%	29,86%	55,88%
		Correlação	94,40%	45,67%	51,29%
		Qui quadrado	88,62%	73,77%	23,24%
		Intersecção	56,55%	19,09%	45,76%
45	Média	Bhattacharyya	57,90%	24,12%	43,94%
		Correlação	63,96%	27,63%	46,28%
		Qui quadrado	97,92%	93,56%	6,31%
		Intersecção	7,77%	4,68%	7,40%
45	Mediana	Bhattacharyya	78,86%	16,74%	65,66%
		Correlação	80,40%	17,92%	65,99%
		Qui quadrado	95,75%	93,56%	6,17%
		Intersecção	77,06%	14,64%	65,78%

Tabela 10: resultado da detecção de pessoas que não estão utilizando capacetes e os respectivos falsos positivos, comparando-se os modelos de cabeças com os capacetes brancos. Foram utilizadas todas combinações possíveis entre intervalos de classe, filtros e classificadores-comparadores.

Intervalos de classes de histogramas. Modelos de cabeças comparados com modelos de capacetes brancos.	Filtros	Classificador-comparador	Deteção de cabeças sem capacete	Falsos positivos	Acurácia
256	Bilateral	Bhattacharyya	80,11%	14,83%	68,23%
		Correlação	64,95%	20,72%	51,49%
		Qui quadrado	70,53%	47,98%	36,69%
		Intersecção	99,71%	62,70%	37,19%
256	Gauss	Bhattacharyya	85,84%	8,29%	78,72%
		Correlação	80,69%	2,29%	78,84%
		Qui quadrado	0,00%	0,00%	0,00%
		Intersecção	92,13%	19,30%	74,35%
256	Média	Bhattacharyya	85,55%	13,96%	73,61%
		Correlação	78,40%	1,31%	77,37%
		Qui quadrado	11,44%	0,33%	11,41%
		Intersecção	94,56%	26,72%	69,30%
256	Mediana	Bhattacharyya	91,13%	9,16%	82,78%
		Correlação	58,37%	3,16%	56,52%
		Qui quadrado	30,76%	55,18%	13,79%
		Intersecção	93,99%	15,05%	79,85%
64	Bilateral	Bhattacharyya	96,42%	10,91%	85,91%
		Correlação	89,13%	6,32%	83,49%
		Qui quadrado	50,36%	18,76%	40,91%
		Intersecção	97,85%	26,83%	71,60%
64	Gauss	Bhattacharyya	86,70%	11,01%	77,15%
		Correlação	74,11%	1,31%	73,14%
		Qui quadrado	0,00%	0,00%	0,00%
		Intersecção	95,57%	30,32%	66,59%
64	Média	Bhattacharyya	87,27%	11,34%	77,37%
		Correlação	81,26%	1,64%	79,93%
		Qui quadrado	50,50%	2,73%	49,12%
		Intersecção	98,57%	81,57%	18,17%
64	Mediana	Bhattacharyya	91,56%	12,43%	80,18%
		Correlação	63,52%	4,14%	60,89%
		Qui quadrado	76,68%	39,80%	46,16%
		Intersecção	94,56%	42,97%	53,93%

5. Conclusões e sugestões para trabalhos futuros

Analisando os gráficos e tabelas do item resultados e discussão, observou-se:

- Quanto aos filtros:

A suavização efetuada pelos filtros de média e de Gauss, prejudicou o desempenho dos classificadores-comparadores, no contexto dos capacetes amarelos e azuis. Apresentaram uma melhor performance, os filtros que tem como principal característica a preservação de bordas (bilateral e mediana), sendo o melhor deles o filtro bilateral. Conforme fundamentos apresentados no item 2.5.5, foram utilizados os parâmetros σ_r (propagação fotométrica) correspondente a 75 valores de intensidade e σ_d (difusão geométrica no domínio) equivalente a 75 pixels. Uma sugestão para trabalho futuro é obter de forma experimental os valores destes parâmetros que maximizem os resultados.

- Quanto aos intervalos de classe:

De uma maneira global, os resultados apresentados pela utilização do número máximo de intervalos de classe – 180 para a dimensão H e 256 para a dimensão S – tiveram um desempenho inferior aos apresentados pelos intervalos – 45 para dimensão H e 64 para a dimensão S. Semelhantemente ao que foi sugerido nos filtros, a proposta para trabalhos futuros é encontrar empiricamente o intervalo de classe ideal para a otimização do algoritmo.

- Quanto aos comparadores-classificadores:

Considerando somente a coluna – detecção de pessoas sem capacete – das tabelas 8 a 10, o melhor desempenho foi alcançado pelo comparador-classificador intersecção. Seus maiores índices chegaram a 100% (tabela 8), 90,7% (tabela 9) e 99,71% (tabela 10). Porém o nível de falsos positivos associados a estes índices foram respectivamente 68,63%, 42,51% e 62,70%; valores altos se comparados a outros classificadores.

O comparador-classificador Qui quadrado apresentou incoerências como detecção e falsos positivos com valor zero (tabelas 8 e 10) e nível de detecção alto de 96,57% associados a falsos positivos igualmente altos de 97,46% (tabela 9).

O comparador-classificador correlação teve sua melhor performance de detecção de pessoas sem capacete com a combinação de 180 intervalos de classe com filtro de Gauss. O índice atingido foi de 99,55 % (tabela 8).

O comparador-classificador distância Bhattacharyya, de uma maneira global, teve seus melhores desempenhos utilizando os filtros de preservação de borda (bilateral e mediana).

Diante das observações expostas e para tentar estabelecer um parâmetro que considere tanto as detecções quanto os falsos positivos, foi criado pelo autor desta dissertação, o parâmetro acurácia (detecção% x (1 – falso positivo%)), que mensura a precisão do processo. Foi inserido nas tabelas de 8 a 10 em todas as combinações testadas. Utilizando este parâmetro, chega-se à conclusão que a combinação de filtro bilateral – 45 ou 64 intervalos de classe – distância Bhattacharyya apresentou o melhor desempenho.

Quanto a mais sugestões para trabalhos futuros, o desafio é pesquisar as formas de se mitigar as incidências de falsos positivos, que prejudicaram muitas combinações com alto nível de detecção. Estampas em camisas e mudanças de cores nas roupas que cobrem o tronco e as pernas, geraram muitos falsos positivos no sistema de detecção proposto. Excetuando estas falhas do segmentador e efetuando a contagem de forma manual, chegou-se a resultados, cujo melhor deles atingiu um nível de detecção de 99,7% e falsos positivos na ordem de 1%.

6. Referências bibliográficas

Anuário Estatístico da Previdência Social AEPS 2014. Ministério do Trabalho e Previdência Social – MTPS, Brasília, 31 mar.2016. Disponível em <<http://www.previdencia.gov.br/wp-content/uploads/2016/07/AEPS-2014.pdf>>. Acesso em 20 jun.2016.

BRADSKI, Gary; KAEHLER, Adrian. “Learning OpenCV Computer Vision with the OpenCV Library”, Sebastopol, USA: O Reilly Media, Inc., 2008, 555 p.

BURGER, Wilhelm; BURGE, Mark James. “Digital image processing : an algorithmic introduction using Java “, New York, USA : Springer, 2008, 564 p.

FRIEDMAN, Nir; RUSSELL, Stuart. “Image segmentation in video sequences: A probabilistic approach”, Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence, pp. 175-181, Jan. 1997.

GONZALEZ, Rafael C; WOODS, Richard E. Processamento de imagens digitais, São Paulo, SP: Edgard Blucher, 2000, 509 p.

GONZALEZ, Rafael C; WOODS, Richard E. Processamento de imagens digitais, São Paulo, SP: Pearson Education, 2010, 624 p.

KaewTraKulPong, Pakorn; BOWDEN, Richard. “An improved adaptive background mixture model for real-time tracking with shadow detection”, Proceedings of 2nd European Workshop on Advanced Video Based Surveillance Systems, 2001.

KONWAR, A. S., BORAH, D. S. & TUITUNG, C., 2014. An American Sign Language detection system using HSV color model and edge detection. Communications and Signal Processing (ICCSP), 2014 International Conference on, pp. 743 - 747.

MARQUES FILHO , Ogê; VIEIRA NETO, Hugo. Processamento Digital de Imagens, Rio de Janeiro, RJ: Brasport, 1999, 307 p .

Números comprovam eficácia de integração no monitoramento. Diário Regional, São Bernardo do Campo, 19 jul.2016. Minha Cidade. Disponível em <http://www.diarioregional.com.br/edicoes_anteriores/2016/julho/edicao_190716.pdf>. Acesso em 01 set.2016.

OpenCV, About. Disponível em <<http://opencv.org/about.html>> Acessado em 20 out.2015.

RIDDER, Christof; MUNKELT, Olaf; KIRCHNER, Harald. “Adaptive Background Estimation and Foreground Detection using Kalman-Filtering”, Proceedings of International Conference on recent Advances in Mechatronics, ICRAM’95, UNESCO Chair on Mechatronics, pp. 193-199, 1995.

SCHIELE, Bernt; CROWLEY, James L. “Object recognition using multidimensional receptive field histograms”, European Conference on Computer Vision, pp. 610-619, Jan. 1996.

SCHIELE, Bernt; CROWLEY, James L. “Comprehensive colour image normalization”, European Conference on Computer Vision, pp. 475-490, Jan. 1998.

SMITH, Alvy Ray. “Color Gamut Transform Pairs”, Siggraph Computer Graphics, Vol 12, Ed 3, pp. 12-19, Ago.1978.

SP ganha nova etapa do Detecta, sistema de monitoramento criminal. Secretaria de Estado da Segurança Pública, São Paulo, 16 abr. 2014. Últimas Notícias. Disponível em <<http://www.ssp.sp.gov.br/noticia/lenoticia.aspx?id=33930>>. Acesso em 01 set.2016.

STAUFFER, Chris; GRIMSON, W Eric L.; “Adaptive background mixture models for real-time tracking”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 2, 1999.

SWAIN, Michael J; BALLARD, Dana H. “Color indexing”, International journal of computer vision, Vol 7, Ed 1, pp. 11-32, Nov. 1991.

TOMASI, Carlo; MANDUCHI, Roberto. “Bilateral filtering for gray and color images”, IEEE Sixth International Conference on Computer Vision, pp. 839-846, Jan. 1998.

WREN, Christopher R.; AZARBAYEJANI, Ali; DARREL, Trevor; PENTLAND, Alex. "Pfinder: Real-Time Tracking of the Human Body", IEEE Transactions on pattern analysis and machine intelligence, Vol 19, Ed 7, pp. 780-785, Jul. 1997.

ZIVKOVIC, Zoran. "Improved Adaptive Gaussian Mixture Model for Background Subtraction", IEEE Proceedings of the 17th International Conference on Pattern Recognition, Vol 2, pp. 28-31, Ago.2004.

ZIVKOVIC, Zoran; HEIJDEN, Ferdinand van der. Der.Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction", Pattern Recognition Letters, Vol 27, Ed 7, pp. 773-780, Mai. 2006.

7. APÊNDICE A - código fonte

Observações:

- Os comentários estão grafados em **vermelho**.
- Python não admite caracteres especiais, como cedilha e acentos, mesmo para os comentários

```
#importacao das bibliotecas
```

```
import time
```

```
import numpy as np
```

```
import cv2
```

```
from matplotlib import pyplot as plt
```

```
import shutil
```

```
# Segmentador de movimentos MOG2
```

```
backsub = cv2.BackgroundSubtractorMOG2()
```

```
#Leitura das imagens (pode ser predeterminada ou nao) e respectivo calculo dos histogramas
```

```
capacete1 = cv2.imread("14.png")
```

```
hista1 = cv2.calcHist( [capacete1], [0], None, [45], [1,180] )
```

```
hista2 = cv2.calcHist( [capacete1], [0], None, [45], [1,180] )
```

```
hista3 = cv2.calcHist( [capacete1], [0], None, [45], [1,180] )
```

```
hista4 = cv2.calcHist( [capacete1], [0], None, [45], [1,180] )
```

```
hista5 = cv2.calcHist( [capacete1], [1], None, [64], [1,255] )
```

```

hista6 = cv2.calcHist( [capacete1], [1], None, [64], [1,255] )
hista7 = cv2.calcHist( [capacete1], [1], None, [45], [1,180] )
hista8 = cv2.calcHist( [capacete1], [0], None, [64], [1,255] )
hista9 = cv2.calcHist( [capacete1], [0], None, [45], [1,180] )
hista10 = cv2.calcHist( [capacete1], [1], None, [64], [1,255] )

```

```
# captura de video
```

```
cap = cv2.VideoCapture("A1.avi")
```

```
# contador de frames
```

```
frame_counter = 0
```

```
while(True):
```

```
# Captura de imagens quadro a quadro
```

```
    ret, frame = cap.read()
```

```
    frame_counter += 1
```

```
# Comando para repetir o video
```

```
    if frame_counter == cap.get(cv2.cv.CV_CAP_PROP_FRAME_COUNT):
```

```
        frame_counter = 0
```

```
        cap.set(cv2.cv.CV_CAP_PROP_POS_FRAMES, 0)
```

```
# Inicio das operacoes com os quadros
```

```
    if ret:
```

```
# Aplicacao do segmentador na imagem em movimento do quadro corrente
```

```
        fgmask = backsub.apply(frame, None, 0.01)
```

```
# Aplicacao da funcao contorno na imagem em movimento
```

```
        contours, hierarchy = cv2.findContours(fgmask.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
```

```
        try: hierarchy = hierarchy[0]
```

```
        except: hierarchy = []
```

```
        best_id = 0
```

```
        for contour, hier in zip(contours, hierarchy):
```

```
# Definicao do retangulo como contorno da imagem em movimento
```

```
# Comentario: x: valor movel do eixo das abcissas no momento que detectou o movimento.
```

```

#     y: idem, idem em relacao as ordenadas
#     w: valor da largura absoluta
#     h: valor da altura absoluta

(x,y,w,h) = cv2.boundingRect(contour)

if (w > 50 and h > 80) and ( w < 300 and h < 400 ):

    kernel = np.ones((15,15),np.float32)/225
    fatia = frame[y:y+h//10, x:x+w]
    fatia1 = fgmask[y:y+h//10, x:x+w]
    res = cv2.bitwise_and(fatia,fatia, mask= fatia1)

# Conversao BGR para HSV
    hsv = cv2.cvtColor(res, cv2.COLOR_BGR2HSV)

# Chamada para mostrar a imagem fatia e fatia 1.
    cv2.imshow("fatia", fatia)
    cv2.imshow("fatia1",fatia1)
    branco_inferior = np.array([1,1,1])
    branco_superior = np.array([180,255,255])

#definicao de mascara
    mask = cv2.inRange(hsv, branco_inferior, branco_superior)

#retirar o fundo da subimagem
    res1 = cv2.bitwise_and(hsv,hsv, mask= mask)

#definicao do filtro bilateral
    bilateral = cv2.bilateralFilter(res1,15,75,75)

#comando do teclado para inserir modelos
    key = cv2.waitKey(5)
    if key == ord('a'):

# teclas a, b, c, d inserem modelos da dimensao Hue

```

```

hista1 = cv2.calcHist( [bilateral], [0], None, [45], [1,180] )
    cv2.normalize(hista1,hista1,0,100,cv2.NORM_MINMAX)
    cv2.imshow("amarelo1", fatia)
elif key == ord('b'):
    hista2 = cv2.calcHist( [bilateral], [0], None, [45], [1,180] )
    cv2.normalize(hista2,hista2,0,100,cv2.NORM_MINMAX)
    cv2.imshow("amarelo2", fatia)
elif key == ord('c'):
    hista3 = cv2.calcHist( [bilateral], [0], None, [45], [1,180] )
    cv2.normalize(hista3,hista3,0,100,cv2.NORM_MINMAX)
    cv2.imshow("azul1", fatia)
elif key == ord('d'):
    hista4 = cv2.calcHist( [bilateral], [0], None, [45], [1,180] )
    cv2.normalize(hista4,hista4,0,100,cv2.NORM_MINMAX)
    cv2.imshow("azul2", fatia)

```

#teclas e, f inserem modelos da dimensao saturation

```

elif key == ord('e'):
    hista5 = cv2.calcHist( [bilateral], [1], None, [64], [1,255] )
    cv2.normalize(hista5,hista5,0,100,cv2.NORM_MINMAX)
    cv2.imshow("branco1", fatia)
elif key == ord('f'):
    hista6 = cv2.calcHist( [bilateral], [1], None, [64], [1,255] )
    cv2.normalize(hista6,hista6,0,100,cv2.NORM_MINMAX)
    cv2.imshow("branco2", fatia)

```

#teclas g, h inserem modelos das dimensoes hue e saturation

```

elif key == ord('g'):
    hista7 = cv2.calcHist( [bilateral], [0], None, [45], [1,180] )
    cv2.normalize(hista7,hista7,0,100,cv2.NORM_MINMAX)
    hista8 = cv2.calcHist( [bilateral], [1], None, [64], [1,255] )
    cv2.normalize(hista8,hista8,0,100,cv2.NORM_MINMAX)

```

```

cv2.imshow("cabeca1", fatia)
elif key == ord('h'):
    hista9 = cv2.calcHist( [bilateral], [0], None, [45], [1,180] )
    cv2.normalize(hista9,hista9,0,100,cv2.NORM_MINMAX)
    hista10 = cv2.calcHist( [bilateral], [1], None, [64], [1,255] )
    cv2.normalize(hista10,hista10,0,100,cv2.NORM_MINMAX)
    cv2.imshow("cabeca2", fatia)

```

#calculo do histograma da subimagem corrente na dimensao hue

```

histfatia = cv2.calcHist( [bilateral], [0], None, [45], [1,180] )
cv2.normalize(histfatia,histfatia,0,100,cv2.NORM_MINMAX)

```

#calculo do histograma da subimagem corrente na dimensao saturation

```

histfatiabranco = cv2.calcHist( [bilateral], [1], None, [64], [1,255] )
cv2.normalize(histfatiabranco,histfatiabranco,0,100,cv2.NORM_MINMAX)

```

#comparacao do histograma da subimagem com os modelos

```

Bhattacharyya_A = cv2.compareHist(hista1, histfatia,
cv2.cv.CV_COMP_BHATTACHARYYA)

Bhattacharyya_B = cv2.compareHist(hista2, histfatia,
cv2.cv.CV_COMP_BHATTACHARYYA)

Bhattacharyya_C = cv2.compareHist(hista3, histfatia,
cv2.cv.CV_COMP_BHATTACHARYYA)

Bhattacharyya_D = cv2.compareHist(hista4, histfatia,
cv2.cv.CV_COMP_BHATTACHARYYA)

Bhattacharyya_E = cv2.compareHist(hista5, histfatiabranco,
cv2.cv.CV_COMP_BHATTACHARYYA)

Bhattacharyya_F = cv2.compareHist(hista6, histfatiabranco,
cv2.cv.CV_COMP_BHATTACHARYYA)

Bhattacharyya_GA = cv2.compareHist(hista7, histfatia,
cv2.cv.CV_COMP_BHATTACHARYYA)

Bhattacharyya_GB= cv2.compareHist(hista8, histfatiabranco,

```

```
cv2.cv.CV_COMP_BHATTACHARYYA)
```

```
    Bhattacharyya_HA = cv2.compareHist(hista9, histfatia,
cv2.cv.CV_COMP_BHATTACHARYYA)
```

```
    Bhattacharyya_HB= cv2.compareHist(hista10, histfatiabranco,
cv2.cv.CV_COMP_BHATTACHARYYA)
```

```
#condicao de classificacao para detectar somente cabeças humanas sem capacete
```

```
#manda imprimir os valores Bhattacharyya no terminal Ubuntu
```

```
    if (Bhattacharyya_GA < Bhattacharyya_A) and (Bhattacharyya_GA <
Bhattacharyya_C) and (Bhattacharyya_HA < Bhattacharyya_A) and (Bhattacharyya_HA <
Bhattacharyya_C) and (Bhattacharyya_GB < Bhattacharyya_E) and (Bhattacharyya_GB <
Bhattacharyya_F) and (Bhattacharyya_HB < Bhattacharyya_E) and (Bhattacharyya_HB <
Bhattacharyya_F) :
```

```
        print ('Bhattacharyya distance cabeça hue1= ', Bhattacharyya_GA)
        print ('Bhattacharyya distance cabeça hue2 = ', Bhattacharyya_HA)
        print ('Bhattacharyya distance cap amarelo1 = ', Bhattacharyya_A)
        print ('Bhattacharyya distance cap amarelo2 = ', Bhattacharyya_B)
        print ('Bhattacharyya distance cap azul1 = ', Bhattacharyya_C)
        print ('Bhattacharyya distance cap azul2 = ', Bhattacharyya_D)
        print ('
        ')
        print ('Bhattacharyya distance cabeça sat1= ', Bhattacharyya_GB)
        print ('Bhattacharyya distance cabeça sat2= ', Bhattacharyya_HB)
        print ('Bhattacharyya distance cap branco1 = ', Bhattacharyya_E)

        print ('Bhattacharyya distance cap branco2 = ', Bhattacharyya_F)
```

```
# contorna com retangulo azul a imagem de pessoa sem capacete
```

```
    best_id = best_id + 1
```

```
    cv2.rectangle(frame, (x,y), (x+w, y+h), (255, 0, 0), 2)
```

```
    cv2.putText(frame, str(best_id), (x,y-5), cv2.FONT_HERSHEY_SIMPLEX,0.5,
(255, 0, 0), 2)
```

```
# Mostra o quadro resultante
cv2.imshow('frame',frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
# recapturar o video
```