

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ENGENHARIA DE COMPUTAÇÃO

HENRIQUE GONÇALVES DE PÁDUA REIS
NICOLAS MANSUR BELESKI

**DETECÇÃO DE SOMBRAS DE VEÍCULOS EM MOVIMENTO EM
VÍDEOS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2017

HENRIQUE GONÇALVES DE PÁDUA REIS
NICOLAS MANSUR BELESKI

**DETECÇÃO DE SOMBRAS DE VEÍCULOS EM MOVIMENTO EM
VÍDEOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Engenheiro de Computação.

Orientador: Prof. Dr. Bogdan Tomoyuki Nassu

CURITIBA

2017

Henrique Gonçalves de Pádua Reis
Nicolas Mansur Beleski

Detecção de Sombras de Veículos em Movimento em Vídeos

Este trabalho de conclusão de curso foi julgado e aprovado como requisito parcial para obtenção do grau de Engenheiro de Computação pela Universidade Tecnológica Federal do Paraná.

Curitiba, 30 de Junho de 2017.

Prof. Bogdan Tomoyuki Nassu

Prof. Gustavo Benvenuto Borba

Prof. Rodrigo Minetto

AGRADECIMENTOS

Agradecemos ao professor e orientador Bogdan Tomoyuki Nassu pelo apoio singular, conhecimento transmitido, paciência e assistência, sempre nos escutando e aconselhando em todas as dificuldades encontradas.

A esta universidade e seu corpo docente, em especial aos professores do DAINF e DAELN que nos acompanharam, auxiliaram e contribuíram diretamente em nosso desenvolvimento pessoal e profissional durante toda a graduação. Aos nossos pais e familiares pelo amor e apoio incondicional, pois são eles que tornaram este trabalho e muitos dos nossos sonhos possíveis.

“We can only see a short distance ahead, but we can see plenty there that needs to be done.”

— Alan Turing

RESUMO

REIS, Henrique; BELESKI, Nicolas. DETECÇÃO DE SOMBRAS DE VEÍCULOS EM MOVIMENTO EM VÍDEOS. 71 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

A identificação e rastreamento de veículos em vias é fator importante no processo de decisão e desenvolvimento urbano. Existe uma grande variedade de informações disponíveis, variando de contagem de veículos, estimação de velocidade, detecção de infrações, identificação de veículos ilegais entre outras. No entanto, técnicas relacionadas a processamento de imagens digitais ainda não estão consolidadas como soluções robustas pois no geral não oferecem resultados tão precisos quanto sensores físicos, apesar de apresentarem vantagens relacionadas a custos de instalação e equipamentos necessários.

Um dos obstáculos mais comuns encontrados durante a segmentação de veículos é a influência negativa causada pela presença de sombras projetadas no asfalto pelos mesmos quando em movimento. Com o objetivo de abordar este problema específico o trabalho proposto consiste de uma análise de técnicas presentes na literatura recente e na aplicação de uma série de algoritmos de processamento de imagem sobre um vídeo que emula uma câmera de segurança de trânsito. Diferentes abordagens são utilizadas com o objetivo de obter um melhor resultado quanto à posição dos respectivos veículos em movimento no vídeo, buscando minimizar os efeitos da sombra presente.

O projeto é estruturado de forma que os diferentes algoritmos são aplicados em ordem sequencial, adaptando os resultados parciais encontrados. A partir da determinação de um plano de fundo inicial e da segmentação dos objetos em movimento, a análise é feita em três etapas principais: classificação inicial de pixels candidatos pela luminosidade, refinamento dos mesmos através da análise de correlação de textura, e subsequentemente a aplicação de uma análise probabilística iterativa usando a distância de Mahalanobis como referência para descartar pixels aberrantes.

Entre os 135 veículos analisados, a similaridade entre a região calculada e o resultado esperado foi de 81,15%. O resultado é considerado satisfatório pois entre o conjunto de técnicas apresentadas, a análise de correlação de textura e o uso da distância de Mahalanobis são promissores e possibilitam estudos futuros.

Palavras-chave: Segmentação de Veículos, Detecção de Sombras, Análise de Correlação de Textura, Distância de Mahalanobis

ABSTRACT

REIS, Henrique; BELESKI, Nicolas. CAST SHADOW DETECTION FOR MOVING VEHICLES IN VIDEOS. 71 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

Vehicle tracking and identification on roads is an important factor in the decision-making process for urban development. There is a wide range of available information, varying from vehicle counting, speed estimation, infringement detection, illegal vehicle identification, and more. However, digital image processing techniques have not been consolidated yet as robust solutions as they usually do not provide results as accurate as those from physical sensors, although they have advantages related to the costs of installation and necessary equipment.

One of the most common obstacles encountered during vehicle segmentation is the negative influence caused by the presence of shadows cast on the road by the objects in motion. In order to address this specific problem, the proposed work consists of an analysis of techniques present in recent literature and the subsequent application of a series of image processing algorithms on a sample video that emulates a traffic safety camera. Different techniques are applied in order to obtain an improved result regarding the position of said vehicles in motion in the video, minimizing the effects of the cast shadow.

The project is structured in such manner so that the different algorithms are applied in a sequential order, improving upon each intermediate result. From the initial determination of a background frame and the segmentation of moving objects, the analysis has three main steps: initial classification of candidate pixels by luminosity, refinement through texture correlation analysis, and subsequently the application of an iterative probabilistic analysis through the use of the Mahalanobis distance in order to discard outlier pixels.

Among the 135 vehicles analyzed, the similarity between the calculated region and the expected result was 81.15%. The result is considered satisfactory for the reason that among the set of techniques presented, texture correlation analysis and the use of Mahalanobis distance are promising and support future studies.

Keywords: Vehicle Segmentation, Cast Shadow Detection, Texture Correlation Analysis, Mahalanobis Distance

LISTA DE FIGURAS

FIGURA 1	– Detecção de veículos.	9
FIGURA 2	– Espaços de representação de cores.	11
FIGURA 3	– Espaços de representação de cores.	12
FIGURA 4	– Exemplo de segmentação apresentado por Sen-Ching e Kamath (2004). ..	14
FIGURA 5	– MHI aplicado em 5 quadros consecutivos.	16
FIGURA 6	– Exemplo de detecção de sombra através da geometria por Yoneyama et al. (2003).	21
FIGURA 7	– Diagrama com as etapas de processamento.	22
FIGURA 8	– Exemplo de conversão para escala de cinza.	23
FIGURA 9	– Representação em escala de cinza dos valores dos canais individuais.	24
FIGURA 10	– Quadros utilizados na estimativa de um plano de fundo inicial.	25
FIGURA 11	– Plano de fundo estimado.	25
FIGURA 12	– Representação da comparação usando um espaço bi-dimensional.	28
FIGURA 13	– Máscara da segmentação e pós processamento em um quadro selecionado.	31
FIGURA 14	– Acompanhamento dos passos iniciais em alguns quadros selecionados. ..	35
FIGURA 15	– Pós-processamento para correção da coluna (d) da Figura 14	36
FIGURA 16	– Distância de Mahalanobis aplicada à uma distribuição de exemplo.	37
FIGURA 17	– Autovetores e autovalores para diferentes matrizes de covariância em exemplos de distribuições.	39
FIGURA 18	– Uso do limiar baseado na distância de Mahalanobis para substituir pontos aberrantes.	40
FIGURA 19	– Exemplos de áreas determinadas manualmente para validação.	43
FIGURA 20	– Interseção entre a elipse de referência e a elipse gerada pelo algoritmo. ..	44
FIGURA 21	– Histograma com resultado dos testes e linha da média.	47
FIGURA 22	– Exemplos de resultados com similaridade de área variadas. A elipse vermelha é a referência, enquanto a elipse verde é gerada pelo algoritmo.	48
FIGURA 23	– Mascara do exemplo (c) da Figura 22 com detecção de <i>pixels</i> candidatos.	48

LISTA DE TABELAS

TABELA 1	– Teste com iterações e limiar para distância de Mahalanobis fixados	45
TABELA 2	– Tamanho da janela e limiar para distância de Mahalanobis fixados	45
TABELA 3	– Iterações e tamanho da janela fixados	46
TABELA 4	– Parâmetros escolhidos	46
TABELA 5	– Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.	55
TABELA 5	– Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.	56
TABELA 5	– Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.	57
TABELA 5	– Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.	58
TABELA 5	– Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.	59
TABELA 6	– Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10. Janela de textura = 11; Mahalanobis = 2,2.	59
TABELA 6	– Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10. Janela de textura = 11; Mahalanobis = 2,2.	60
TABELA 6	– Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10. Janela de textura = 11; Mahalanobis = 2,2.	61
TABELA 6	– Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10. Janela de textura = 11; Mahalanobis = 2,2.	62
TABELA 6	– Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10. Janela de textura = 11; Mahalanobis = 2,2.	63
TABELA 6	– Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10. Janela de textura = 11; Mahalanobis = 2,2.	64
TABELA 7	– Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.	64
TABELA 7	– Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.	65
TABELA 7	– Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.	66
TABELA 7	– Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.	67
TABELA 7	– Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.	68
TABELA 7	– Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.	69
TABELA 8	– Resultados obtidos com os parâmetros fixos escolhidos. Janela de textura = 11; Elipses = 8; Mahalanobis = 2,2.	69
TABELA 8	– Resultados obtidos com os parâmetros fixos escolhidos. Janela de textura = 11; Elipses = 8; Mahalanobis = 2,2.	70

TABELA 8 – Resultados obtidos com os parâmetros fixos escolhidos. Janela de textura = 11; Elipses = 8; Mahalanobis = 2,2. 71

SUMÁRIO

1	INTRODUÇÃO	8
1.1	OBJETIVOS DO TRABALHO	9
1.2	ORGANIZAÇÃO DO DOCUMENTO	9
2	DEFINIÇÕES	10
2.1	IMAGEM DIGITAL	10
2.2	ESPAÇO DE REPRESENTAÇÃO DE CORES	11
3	REFERENCIAL TEÓRICO	13
3.1	DETECÇÃO DE VEÍCULOS EM MOVIMENTO	13
3.1.1	Subtração de Fundo	13
3.1.2	Métodos Baseados em Características	14
3.1.3	Diferenciação de Quadros e Métodos Baseados no Movimento	15
3.2	SEGMENTAÇÃO DE OBJETOS USANDO SUBTRAÇÃO DE FUNDO	16
3.2.1	Classificação de Métodos de Subtração de Fundo	16
3.2.2	Técnicas de Modelagem de Plano de Fundo	17
3.2.2.1	Mediana	17
3.2.2.2	Filtro de Kalman	18
3.2.2.3	Mistura de Gaussianas (MoG)	18
3.2.2.4	K-Means	18
3.3	DETECÇÃO DE SOMBRAS	19
3.3.1	Luminância	19
3.3.2	Crominância	20
3.3.3	Propriedades Físicas	20
3.3.4	Geometria	20
3.3.5	Textura	21
4	DESENVOLVIMENTO	22
4.1	DIAGRAMA GERAL DA SOLUÇÃO	22
4.2	PROCESSAMENTO INICIAL	22
4.3	DETERMINAÇÃO DE UM PLANO DE FUNDO INICIAL	24
4.4	SEGMENTAÇÃO	26
4.4.1	Classificação dos <i>Pixels</i>	27
4.4.2	Inicialização do Plano de Fundo	29
4.4.3	Atualização do Modelo de Plano de Fundo	30
4.5	PÓS-PROCESSAMENTO	31
4.6	TESTE DE LUMINÂNCIA	31
4.7	TESTE DE TEXTURA	32
4.8	APROXIMAÇÃO POR ELIPSES	36
4.8.1	Cálculo da Distância de Mahalanobis	37
4.8.2	Determinação das Dimensões da Elipse	38
4.8.3	Implementação do Método Usando Distância de Mahalanobis	39
5	EXPERIMENTOS E RESULTADOS	42
5.1	RECURSOS UTILIZADOS	42

5.2 DADOS DE REFERÊNCIA	43
5.3 METODOLOGIA DE TESTE	43
5.3.1 Parametrização	44
5.4 ANÁLISE DOS RESULTADOS FINAIS	46
6 CONSIDERAÇÕES FINAIS	49
6.1 TRABALHOS FUTUROS	49
REFERÊNCIAS	51
Apêndice A – RESULTADOS DOS TESTES DO ALGORITMO	55

1 INTRODUÇÃO

Na área de processamento digital de imagens, um problema comumente abordado é a detecção e rastreamento de veículos em movimento. Um fator constante nestas aplicações é a presença da sombra na imagem, que pode ser classificada como estática e dinâmica. A sombra estática gerada a partir de objetos estacionários faz parte integral do plano de fundo e pode ser interpretada como tal. No entanto, sombras dinâmicas são aquelas projetadas pelos próprios objetos em movimento e podem ter influência direta sobre resultados e análises que dependem da forma e posição do veículo.

A literatura relacionada à detecção e remoção de sombras apresenta variadas abordagens, normalmente aplicadas a um caso de estudo específico e não como uma resposta generalizada em razão da dinâmica do problema. As abordagens existentes se baseiam em propriedades como a geometria da sombra projetada em relação à forma do objeto analisado (YONEYAMA et al., 2003), variações de luminância e cor (HUANG; CHEN, 2009), e características de textura aliadas a um modelo de mistura de Gaussianas (TIAN et al., 2005).

Trabalhos que relacionam a detecção de sombras em vídeos de trânsito e/ou pessoas em movimento são diversos (YONEYAMA et al., 2003; JIANYONG et al., 2012; SANIN et al., 2012; HUANG; CHEN, 2009; SUN; LI, 2010). Entretanto, as técnicas apresentadas são no geral adaptadas para casos específicos e podem provar-se ineficientes para diferentes variações do mesmo problema da presença de sombras dinâmicas. As técnicas específicas possuem características distintas, com vantagens e desvantagens. Estas diferentes abordagens com resultados variados induzem a pesquisa e desenvolvimento de uma abordagem própria, buscando resultados mais adequados para um caso específico.

Este trabalho propõe analisar e aplicar um conjunto de técnicas de processamento digital de imagens para fim de identificar sombras de veículos em um vídeo que emula aquele de uma câmera de segurança de trânsito. Aplica-se uma sequência de algoritmos para analisar características como cor, textura e distribuição de pixels de forma a estimar a posição dos veículos em movimento no vídeo excluindo as suas sombras projetadas.

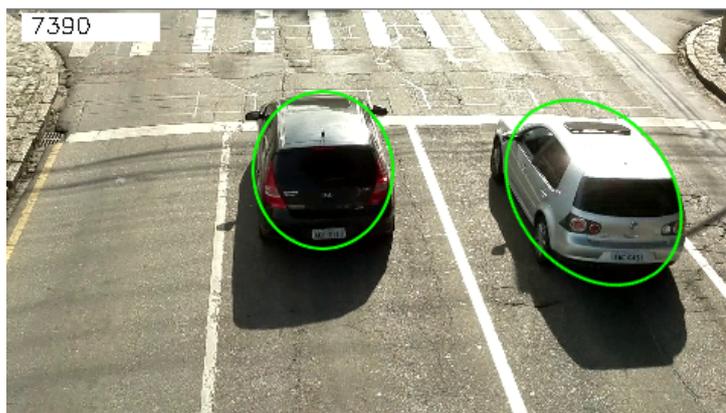


Figura 1: Detecção de veículos.

A Figura 1 ilustra a detecção de veículos e suas áreas desconsiderando a presença da sombra. Os resultados adquiridos usando a combinação de técnicas proposta foi de 81,15% para a similaridade com a área esperada.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é aplicar e analisar um conjunto de técnicas de processamento de imagem para identificar regiões de sombra dinâmica em um vídeo. Os algoritmos aplicados devem se complementar de forma a obter uma melhor estimativa da área que abrange um veículo rastreado, excluindo sua sombra.

1.2 ORGANIZAÇÃO DO DOCUMENTO

Este documento é estruturado em cinco capítulos. O segundo capítulo consiste na apresentação das definições e representações utilizadas no desenvolvimento do trabalho. O terceiro capítulo apresenta o referencial teórico, onde são apresentados os princípios básicos relevantes na área de processamento digital de imagens, trabalhos relacionados, e algoritmos utilizados neste trabalho. O quarto capítulo apresenta a metodologia utilizada no desenvolvimento do projeto, com uma visão do sistema como um todo e uma descrição detalhada dos algoritmos e técnicas implementadas para a obtenção dos resultados. O quinto capítulo apresenta os resultados obtidos através da implementação das técnicas descritas e uma análise comparativa com técnicas já existentes.

2 DEFINIÇÕES

Neste capítulo são apresentadas as definições e representações que são utilizadas durante o desenvolvimento, no capítulo subsequente.

2.1 IMAGEM DIGITAL

Uma imagem digital I de duas dimensões $I_{m,n}$ representa um conjunto de posições ($m = 1, 2, \dots, M; n = 1, 2, \dots, N$) em um espaço cartesiano discreto. Os índices m e n representam as linhas e colunas da imagem respectivamente, ou seja, $I(m, n)$ representa o valor do *pixel* localizado na m -ésima linha e n -ésima coluna. Os valores N e M são limitadas pelo número de colunas (C) e linhas (R) da imagem.

Uma imagem tem um ou mais canais que definem a intensidade e coloração para cada *pixel* $I(m, n)$. Representações mais simples como escala de cinza costumam usar 8 bits para representar todos os valores do preto (zero) ao branco (255), por outro lado é possível que os valores de cada canal de uma imagem colorida sejam representados usando números de ponto flutuante, caso a precisão seja relevante.

O valor de um *pixel* é dado por $I(m, n) = c_1, c_2, \dots, c_i$, onde i é o número de canais na imagem em questão. Por exemplo, um *pixel* branco em uma imagem RGB com 3 canais de 8 bits é dado por $I(m, n) = 255, 255, 255$.

A representação comumente utilizada para imagens coloridas, por exemplo, consiste de três canais de 8 bits por *pixel*, representando as cores vermelho, verde e azul (RGB) e totalizando 24 bits por *pixel*, proporcionando aproximadamente 16 milhões de possibilidades de cores (GONZALEZ; WOODS, 2002).

2.2 ESPAÇO DE REPRESENTAÇÃO DE CORES

Um dos parâmetros fundamentais para uma aplicação de visão computacional é a escolha de um espaço de cor compatível com a necessidade. Entre as diversas opções disponíveis quando tratam-se de imagens coloridas, duas das opções mais comuns são os espaços HSV e Lab (BORA et al., 2015). Apesar do espaço Lab melhor representar a percepção humana de diferença entre duas cores, diferentes espaços podem ter resultados melhores em determinados casos.

Além do espaço RGB, existem outras formas de representação de uma imagem, com respectivas propriedades. Estas permitem análises de diferentes características da imagem. Enquanto no espaço RGB as cores são representadas através de um sistema aditivo de três tonalidades distintas, o espaço HSV por exemplo separa uma cor entre o valor, saturação e matiz. A Figura 2 ilustra os dois espaços.

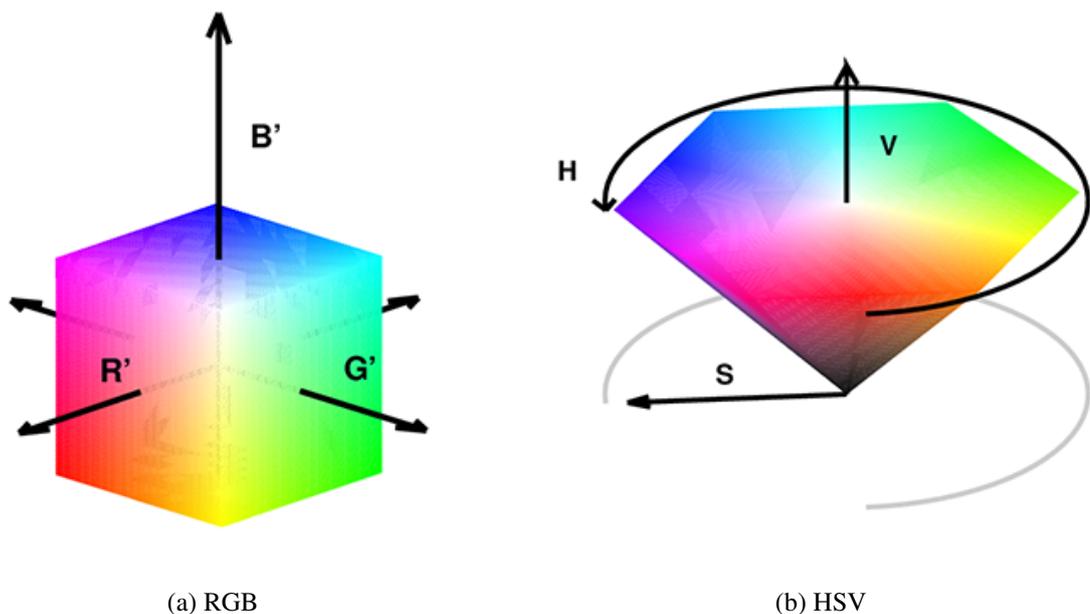


Figura 2: Espaços de representação de cores.

Uma propriedade que pode ser muito relevante quanto a caracterização de um *pixel* é a luminância. Por esta razão existem outras representações que descrevem uma cor em relação a intensidade de luminosidade entre suas características. Exemplos de espaços assim são o HSL (*Hue, Saturation, Luminance*) e o Lab (Luminância, canal A e canal B).

O espaço HSL é análogo ao espaço HSV, porém o canal da intensidade (V) é substituído pela luminância. A diferença tem caráter de representação e não apresenta vantagens

técnicas sobre o sistema HSV. Uma forma alternativa de representação usando a luminância é o espaço Lab, que tenta criar um espaço no qual a distância entre duas cores é proporcional à nossa percepção da diferença entre elas, permitindo correções de equilíbrio de cor mais precisas através de seus três canais: Luminância, a para a representação do intervalo de verde à vermelho e b para a representação do intervalo de azul à amarelo. Na Figura 3 é possível observar as representações dos espaços HSL e Lab.

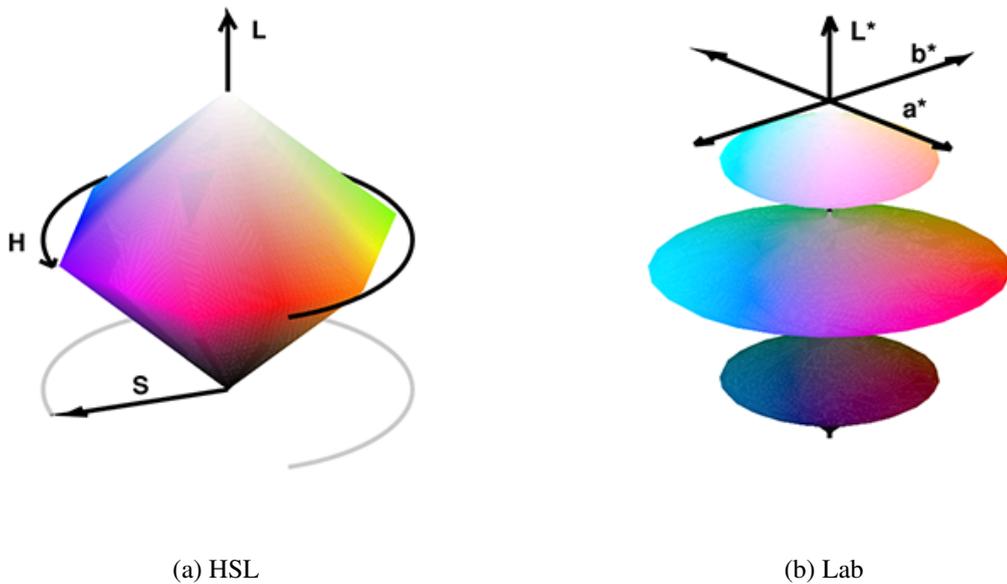


Figura 3: Espaços de representação de cores.

3 REFERENCIAL TEÓRICO

Como base para fundamentar os algoritmos usados no desenvolvimento deste trabalho, este capítulo apresenta o estado da arte das áreas correlatas. A seção 3.1 apresenta uma visão geral dos conceitos de detecção de veículos em movimento, a seção 3.2 descreve um método de segmentação de objetos utilizando subtração de fundo e a seção 3.3 discorre sobre técnicas de detecção de sombras.

3.1 DETECÇÃO DE VEÍCULOS EM MOVIMENTO

Um problema bastante explorado no processamento digital de imagens é a identificação de veículos em movimento em vias urbanas. As características dinâmicas do problema permitem variadas abordagens distintas, cada qual com suas vantagens e desvantagens. Abordagens que têm os objetos a serem rastreados inicializados manualmente não são consideradas neste trabalho, pois não permitem o seu uso em sistemas automáticos. Técnicas de detecção de veículos podem ser classificadas pelos seguintes métodos gerais:

- Métodos de subtração de fundo.
- Métodos baseados em características.
- Diferenciação de quadros e métodos baseados no movimento.

3.1.1 SUBTRAÇÃO DE FUNDO

O processo de segmentação entre objetos em movimento no primeiro plano e um plano de fundo, estático ou dinâmico gerado a partir de um vídeo, é chamado de subtração de fundo, no qual a informação extraída é resultado da diferenciação de imagens. Uma das dificuldades mais notáveis quando o método é aplicado na detecção de veículos em movimento em vias são as variações de iluminação e do clima, que prejudicam a adaptação do plano de fundo

de referência (HADI et al., 2014). A Figura 4 ilustra um exemplo de resultado comum em segmentação através de subtração de fundo.



Figura 4: Exemplo de segmentação apresentado por Sen-Ching e Kamath (2004).

Diversos trabalhos na área sugerem o uso de técnicas estatísticas e paramétricas para uma melhor adaptação em condições irregulares de iluminação. Um dos métodos recorrentes, por exemplo, faz uso de distribuições gaussianas de probabilidade para descrever cada *pixel* da imagem, onde o modelo de distribuição é atualizado conforme novos quadros são analisados, e cada *pixel* é então caracterizado como parte do plano de fundo ou parte de um objeto em primeiro plano (SEN-CHING; KAMATH, 2004). Diferentes trabalhos apresentam refinamentos da ideia básica da subtração de fundo, buscando melhorar a robustez em relação a mudanças na luminosidade (MANDELLOS et al., 2011; HAINES; XIANG, 2014), efeito de sombras (HSIEH et al., 2006; KATO et al., 2004) e o problema de oclusão de veículos (MELO et al., 2006).

3.1.2 MÉTODOS BASEADOS EM CARACTERÍSTICAS

A forma, cantos, bordas e outros detalhes compreendem características que podem ser coletadas e investigadas como forma de identificar e detectar veículos em uma sequência de quadros, uma vez que é esperado que na análise em uma via urbana, as características de um veículo se mantenham as mesmas. Além disso, comparado com técnicas de subtração de fundo, estes métodos tratam de maneira mais eficiente casos onde há oclusão e sobreposição de veículos na imagem (HADI et al., 2014).

Assim como a subtração de fundo, as abordagens dos métodos baseados em

características são variadas. Kazemi et al. (2007) faz uso de *Support Vector Machine*(SVM) como modelo de aprendizado supervisionado, usando imagens da parte traseira dos veículos e transformadas discretas de Curvelet para extrair características para uso no conjunto de treinamento. Este método permite através de um treinamento refinado, detectar veículos em uma via e em certos casos identificar alguns modelos específicos. Sivaraman e Trivedi (2010) expandem neste conceito, propondo uma abordagem de aprendizado ativa, com treinamento realizado dirigindo em vias movimentadas.

Outra abordagem possível é o uso de características chamadas *Haar-like*, originalmente introduzidas para a detecção de faces e assim nomeadas por serem semelhante às *wavelets Haar*. O método foi adaptado para a detecção de veículos por Ponsa et al. (2005), por exemplo. Haselhoff e Kummert (2009) faz a detecção de veículos com sucesso usando uma combinação de características triangulares e retangulares, também baseado em características *Haar-like*.

3.1.3 DIFERENCIAÇÃO DE QUADROS E MÉTODOS BASEADOS NO MOVIMENTO

A diferenciação de quadros é o processo de subtração entre dois quadros subsequentes em uma série de imagens de forma a segmentar objetos em movimento. O processo de segmentação pelo movimento é especialmente em casos onde é necessário diferenciar veículos de outros objetos em movimento através da orientação e velocidade do movimento (HAN et al., 2007; ZHANG et al., 2010).

Um método que trata unicamente com a diferenciação de quadros no tempo é o *Motion history image* (MHI) (AHAD et al., 2012). Nesta técnica, a máscara de segmentação gerada é construída a partir das diferenças dos N últimos quadros, de forma a filtrar parte do ruído gerado na diferenciação direta. A Figura 5 apresenta um exemplo de resultado da aplicação da técnica, nota-se que detalhes como os retrovisores do veículo se repetem 5 vezes, segmentadas a partir da diferença de quadros subsequentes.

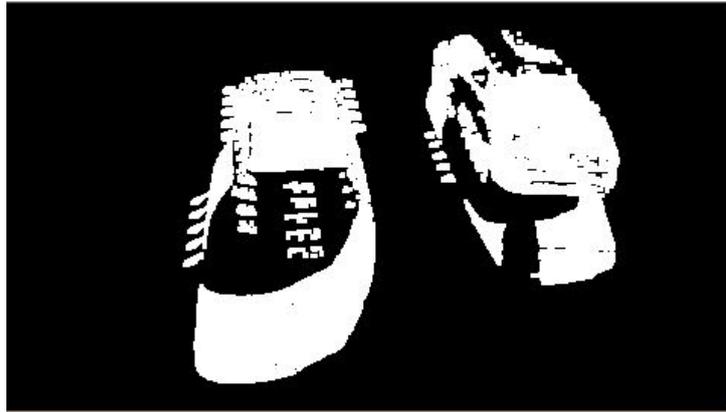


Figura 5: MHI aplicado em 5 quadros consecutivos.

3.2 SEGMENTAÇÃO DE OBJETOS USANDO SUBTRAÇÃO DE FUNDO

Entre os trabalhos de detecção de veículos em movimento, o método mais utilizado é a subtração de fundo como forma de segmentação.

Na análise digital de imagens, a identificação de objetos de interesse e a segmentação dos mesmos é parte fundamental em muitos sistemas. A segmentação de imagens é a separação entre o fundo, onde ficam as regiões estáticas ou de pouca relevância, e o primeiro plano, onde encontram-se os objetos de interesse. Quando trabalha-se com vídeos, o fundo normalmente refere-se à parte estática da imagem, enquanto no primeiro plano encontram-se os objetos mutáveis e/ou em movimento.

Existem diversos trabalhos voltados para a modelagem do plano de fundo. Diferentes abordagens que buscam um equilíbrio entre a robustez em relação a mudanças rápidas e lentas no ambiente, e a sensibilidade para identificar todos os objetos de interesse. A análise de vídeos especificamente permite que diferenças sejam encontradas comparando quadros subsequentes, mesmo sem um conhecimento prévio da posição ou presença dos objetos (SOBRAL; VACAVANT, 2014).

3.2.1 CLASSIFICAÇÃO DE MÉTODOS DE SUBTRAÇÃO DE FUNDO

Uma forma de adquirir um modelo do fundo é conseguir um quadro ou imagem que não inclua objetos mutáveis, porém isto não é possível em muitos casos, como em lugares movimentados, por exemplo. O plano de fundo a ser modelado está sujeito a constantes mudanças, como variações na iluminação e objetos sendo colocados e removidos do plano da imagem. Para lidar com estas mudanças as técnicas de modelagem de plano de fundo são

variadas e distintas. Bouwmans (2011) apresenta uma forma de classificação para técnicas de modelagem de plano de fundo:

- Modelagem de Fundo Básica: o fundo é estimado usando a média, mediana ou análise do histograma em relação ao tempo.
- Modelagem de Fundo Estatístico: o fundo é modelado através de uma distribuição Gaussiana, uma mistura de Gaussianas ou um outro *kernel*. Variáveis estatísticas são usadas para classificar um *pixel* como parte do fundo.
- Modelagem por Agrupamento: Cada *pixel* em um quadro é representado temporariamente por um *cluster*. Variações nos *pixels* são comparadas com os agrupamentos existentes para determinar se eles fazem parte do plano de fundo. Um método de agrupamento clássico é o *K-means*.
- Modelagem usando Filtro: o fundo é modelado utilizando o princípio de que qualquer *pixel* significativamente diferente de seu valor previsto é considerado como parte do plano da imagem.

As seções a seguir apresentam um resumo de algumas técnicas relevantes.

3.2.2 TÉCNICAS DE MODELAGEM DE PLANO DE FUNDO

Modelagem de plano de fundo é uma tarefa crítica em imagens externas, que podem sofrer alterações significativas causadas por eventos naturais ou artificiais. Um modelo para atualização de plano de fundo deve ser capaz de detectar *pixels* estáticos no plano de fundo, bem como detectar *pixels* dinâmicos e variações de iluminação. Aqui são apresentadas algumas das principais técnicas de modelagem de plano de fundo, com uma breve descrição e exemplos ilustrando seu funcionamento.

3.2.2.1 MEDIANA

O método da mediana é amplamente utilizado para modelagem de plano de fundo (PICCARDI, 2004; LO; VELASTIN, 2001; CUCCHIARA et al., 2003). O método supõe que o plano de fundo é o que sempre permanece na imagem, e então o modelo usado como plano de fundo é a mediana dos últimos n quadros da imagem. Essa é uma técnica de fácil implementação e uso, tem pouco custo computacional, porém utiliza uma quantidade relativamente alta de memória.

Em comparação com a média, a mediana apresenta uma melhor rejeição à valores atípicos. Com isso, um objeto se movendo na imagem não alteraria seus valores de *pixel* tão rapidamente quanto se alteraria com a média. Por se tratar de um método amplamente utilizado, existem inúmeras variações de implementação, como em (LO; VELASTIN, 2001; CUCCHIARA et al., 2003).

3.2.2.2 FILTRO DE KALMAN

O filtro de Kalman é característico pela velocidade de processamento e adaptabilidade a mudanças bruscas no ambiente, visto que trabalha apenas com memória a curto prazo. Estas características do filtro de Kalman o fazem recorrente em diversos trabalhos (MESSELODI et al., 2005; SCOTT et al., 2009; VARGAS et al., 2010; BALCILAR; SÖNMEZ, 2008).

Esse método é baseado em um sólido modelo estatístico, com um bom nível de robustez. Esta técnica proporciona uma ótima estimativa, minimizando a variância do erro de estimativa, e limitando a média das saídas estimadas (MESSELODI et al., 2005).

3.2.2.3 MISTURA DE GAUSSIANAS (MOG)

A segmentação através da Mistura de Gaussianas é outro método recorrente na literatura (STAUFFER; GRIMSON, 2000; SEN-CHING; KAMATH, 2004; BOUWMANS, 2011). O método monitora um conjunto de diferentes distribuições Gaussianas ao mesmo tempo. Sendo uma evolução direta de outros métodos, a *MoG* estima uma função de densidade para cada *pixel*, sendo capaz de lidar com múltiplas distribuições diferentes representando um plano de fundo. O método é classificado como paramétrico e seus parâmetros são adaptados continuamente durante a execução.

Dada a popularidade do método, existem diversas variações na implementação que buscam resultados melhores alterando uma ou várias partes do algoritmo, em especial: na inicialização do plano de fundo, no método de atualização do plano de fundo e na detecção do primeiro plano. Bouwmans (2011) em seu trabalho apresenta um compilado destas modificações.

3.2.2.4 K-MEANS

O K-Means é um algoritmo iterativo para agrupamento de dados. Este método é utilizado para particionar n amostras em k agrupamentos, sendo que cada amostra pertence ao agrupamento com a média mais próxima. Este método é amplamente utilizado para

segmentação de plano de fundo (CHAROENPONG et al., 2010; CHEN et al., 2007; BUTLER et al., 2003; KUMAR; SURESHKUMAR, 2013).

3.3 DETECÇÃO DE SOMBRAS

Como mencionado anteriormente, a separação entre o plano de fundo e o primeiro plano através de técnicas variadas compõem a maior parte das abordagens mais comuns de detecção e rastreamento de objetos. Entretanto, estas técnicas apresentam uma desvantagem notável: sombras tendem a ser classificadas como parte do primeiro plano por se movimentarem em conjunto com os objetos de interesse.

Como a sombra projetada pode ter um tamanho significativo comparado ao objeto, isto implica em um grau de inexatidão na detecção e possivelmente problemas no rastreamento. A presença da sombra afeta diretamente a determinação da forma do objeto e prejudica o tratamento da sobreposição de formas. Por este motivo a remoção de sombras se tornou um passo fundamental na implementação de sistemas robustos de detecção.

Sanin et al. (2012) apresentam uma análise comparativa de diferentes técnicas e algoritmos de detecção de sombras. O principal ponto evidenciado é que apenas métodos mais simples são generalizáveis, e cada cenário específico permite aperfeiçoamentos pela adição de pressupostos, ou seja, conclui-se que não há solução perfeita para a detecção de sombras e que determinadas aplicações requerem implementações próprias de acordo com a natureza do problema.

Sanin et al. (2012) apresentam uma classificação baseada em características que podem ser utilizadas nas técnicas de detecção de sombras. Nesta seção descrevemos um conjunto de características relevantes baseada em trabalhos na área (NADIMI; BHANU, 2004; SANIN et al., 2012), e como cada uma pode ser usada na detecção de sombras.

3.3.1 LUMINÂNCIA

A suposição mais direta em relação a detecção de sombras é a de que regiões do plano de fundo sob a sombra encontram-se mais escuras, dado o bloqueio direto da iluminação. A aplicação deste conceito em ambientes com iluminação controlada permite um grau de previsão a respeito de quanto a intensidade ou luminância deve variar. Por ser uma característica simples, é usada regularmente como um estágio inicial na determinação de regiões sombreadas (HUANG; CHEN, 2009; LEONE; DISTANTE, 2007; CUCCHIARA et al., 2003).

3.3.2 CROMINÂNCIA

A característica análoga à luminância é a informação referente à cor. Diferente da luminância, uma região sob sombra não varia significativamente a croma, ou seja, as cores se mantêm próximas uma da outra. Modelos de representação de cores como HSV e Lab são capazes de reproduzir variações na luminância em canais independentes das cores, relevando sua utilidade na análise de sombras em trabalhos (CUCCHIARA et al., 2003; SUN; LI, 2010). Não obstante, a característica da croma não é invariante a mudanças de iluminação e sofre em situações onde a iluminação é muito intensa e as sombras muito escuras (SANIN et al., 2012).

3.3.3 PROPRIEDADES FÍSICAS

Em ambientes externos, as fontes de luz mais predominantes são a luz branca do Sol e a luz azul refletida do céu. A luz do Sol é de maior intensidade e comumente é a fonte de iluminação dominante. Quando a iluminação do Sol é bloqueada, o efeito da luz refletida do céu torna-se mais notável, alterando a croma da região afetada. Trabalhos foram então propostos baseados em um modelo de atenuação não-linear que trata de diferentes casos de iluminação no exterior e interior (MARTEL-BRISSON; ZACCARIN, 2008; HUANG; CHEN, 2009). Outros trabalhos evitam o uso de um modelo de atenuação buscando prever a aparência que cada *pixel* tem quando sob uma sombra. Estes métodos, que tentam aprender o comportamento de *pixels* de sombra, são chamados de abordagens físicas e tendem a ser mais precisos que análises diretas da croma por apresentarem a modelagem de casos específicos (SANIN et al., 2012). Entretanto, sendo ainda dependente das propriedades espectrais, os métodos ainda tem dificuldade quando a croma do objeto é muito similar ao plano de fundo (HUANG; CHEN, 2009).

3.3.4 GEOMETRIA

Usando-se de algumas simplificações e suposições, é possível prever o tamanho e a forma de sombras com o conhecimento prévio das fontes de iluminação, do plano da superfície e do objeto a ser rastreado. Diversos trabalhos buscam usar esta informação para separar a sombra do objeto rastreado (YONEYAMA et al., 2003; CHEN; AGGARWAL, 2010; FANG et al., 2008). Uma das principais vantagens do uso da geometria dos objetos é a sua aplicabilidade diretamente sobre os quadros de entrada sem a necessidade de uma estimativa precisa do plano de fundo (SANIN et al., 2012). Em contrapartida, a análise por geometria envolve

limitações quanto a natureza do problema, normalmente trabalhando com objetos específicos, como pessoas em pé (CHEN; AGGARWAL, 2010) ou veículos como na Figura 6, com fonte de iluminação única. Um desafio para os métodos baseados na geometria são os casos onde mais de uma fonte de luz significativa está presente, fazendo com que cada objeto tenha múltiplas sombras.

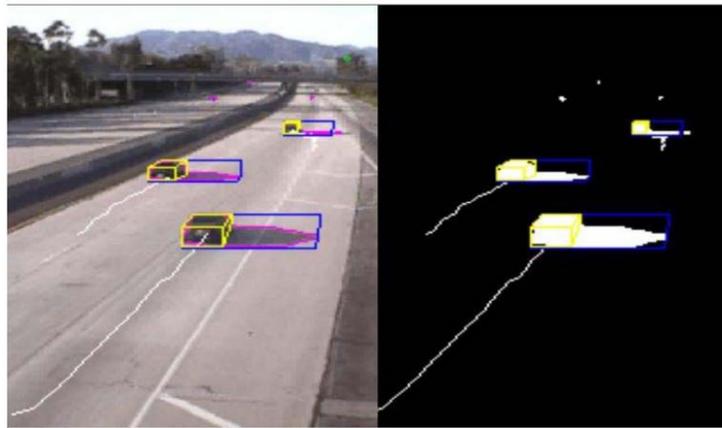


Figura 6: Exemplo de detecção de sombra através da geometria por Yoneyama et al. (2003).

3.3.5 TEXTURA

Outra característica que pode ser explorada é a textura de uma superfície, uma vez que sob uma sombra esta não se altera. Os métodos que fazem uso desta característica trabalham partindo de duas referências: regiões candidatas pré-determinadas através de outro método e um plano de fundo para comparação. As regiões candidatas são comparadas com o plano de fundo livre de sombras usando uma correlação de texturas. Como a sombra altera principalmente as características de luminância de uma região, valores próximos para os descritores de textura indicam que uma região está coberta por uma sombra e não um objeto. Mantendo esta ideia fundamental, diversos métodos foram propostos para fazer a correlação de textura: usando correlação cruzada (TIAN et al., 2005), gradientes (LEONE; DISTANTE, 2007; XU et al., 2005; LIU; GLEICHER, 2008), transformações ortogonais (ZHANG et al., 2006), e campos aleatórios condicionais (QIN et al., 2010). A análise da textura é eficiente em situações onde as mesmas são altamente distinguíveis, é independente de cores e é robusta em relação a variações nas condições de luminosidade. Em contrapartida, os métodos normalmente são lentos computacionalmente, pois envolvem a análise e comparações na vizinhanças de cada *pixel* (SANIN et al., 2012).

4 DESENVOLVIMENTO

Este capítulo descreve o processo de desenvolvimento do projeto, com o detalhamento dos algoritmos e vídeo de referência utilizado.

4.1 DIAGRAMA GERAL DA SOLUÇÃO

A solução proposta neste trabalho consiste em uma combinação de diferentes técnicas de processamento de imagens em forma de cascata, onde cada uma das etapas tem um função diferente, variando entre preparar a imagem, tratamento de falhas, remoção de ruído, segmentação, refinamento dos resultados e apresentação dos resultados.

A Figura 7 apresenta um diagrama geral da solução, apresentando cada uma das etapas de processamento até a solução final. As seções a seguir apresentam em detalhes cada uma das etapas compreendidas na solução.

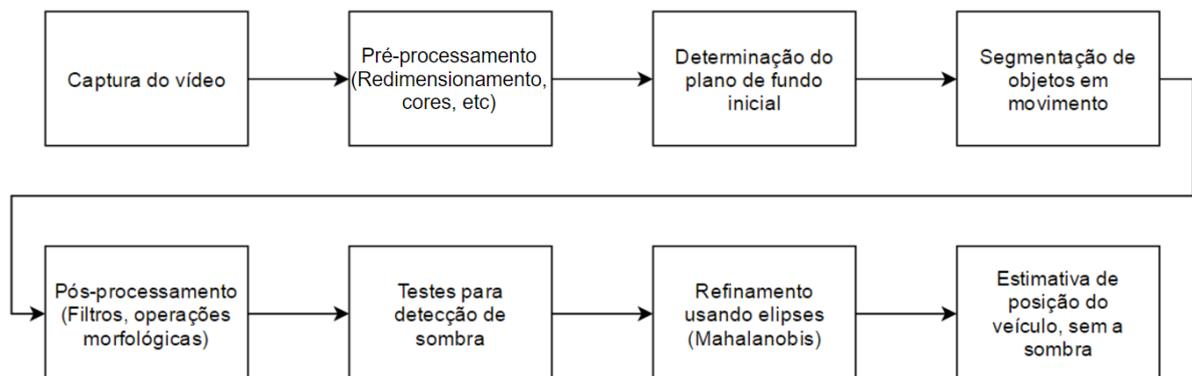


Figura 7: Diagrama com as etapas de processamento.

4.2 PROCESSAMENTO INICIAL

O denominado processamento inicial neste trabalho abrange toda preparação e tratamento dos quadros do vídeo de referência anteriores à aplicação dos algoritmos de

segmentação. O objetivo desta etapa é aprimorar as imagens dentro do contexto de segmentação de sombras e obter resultados satisfatórios nos passos subsequentes.

Para este caso de estudo específico, o primeiro passo do pré-processamento é a redução da resolução do vídeo de entrada. O vídeo obtido tem resolução nativa de 1920x1080, que para este trabalho não oferece benefícios em relação aos resultados finais, enquanto simultaneamente aumenta consideravelmente o custo computacional do processamento. O redimensionamento realizado é uma divisão de fator 2 na resolução do vídeo, obtendo por fim quadros de tamanho 960x540.

Nesta etapa do processo são criadas variáveis que vão receber os quadros de entrada convertidos para os respectivos formatos necessários. As três conversões necessárias são: para a escala de cinza, para o espaço de cores Lab e a conversão para ponto flutuante. A Figura 8 apresenta o quadro original e o quadro convertido para escala de cinza. Nota-se que existe uma similaridade entres os quadros dado as cores predominantes do asfalto. A Figura 9 exemplifica a diferença criada na conversão para o espaço Lab apresentando representações unidimensionais (com valores de 1 a 255) dos canais individuais do espaço RBG e Lab, respectivamente.

A conversão para ponto flutuante não apresenta diferenças visuais e é obtida dividindo-se os valores por 255, colocando-os em um intervalo entre 0 e 1. A conversão de 8 para 32 bits faz com que operações envolvendo os valores dos pixels sejam feitas com muito mais algoritmos significativos, além de possibilitar uma gama de operações que não seriam possíveis se mantivéssemos a informação em um único *byte*.

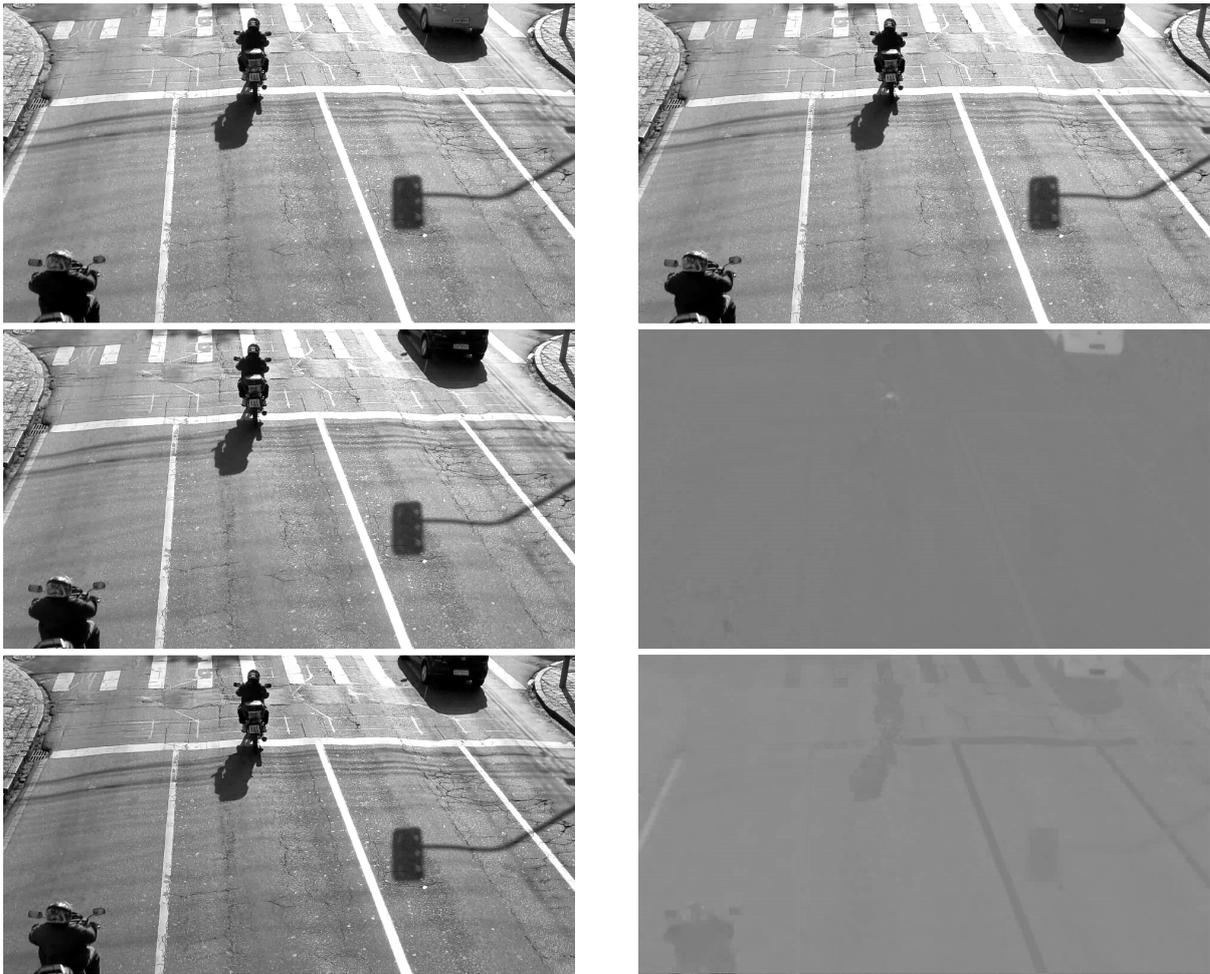


(a) Quadro original



(b) Escala de cinza

Figura 8: Exemplo de conversão para escala de cinza.



(a) Separação dos canais RGB

(b) Separação dos canais Lab

Figura 9: Representação em escala de cinza dos valores dos canais individuais.

4.3 DETERMINAÇÃO DE UM PLANO DE FUNDO INICIAL

Uma das etapas fundamentais para o funcionamento eficaz da segmentação utilizada é a determinação de um plano de fundo inicial. Idealmente, o plano de fundo inicial escolhido consiste de um quadro do vídeo de entrada sem nenhum objeto dinâmico presente, o que na prática nem sempre é possível em vias movimentadas. Para evitar que seja necessário buscar um possível quadro ideal, este trabalho usa uma técnica de amostragem simples para determinar um plano de fundo satisfatório.

O processo para a estimativa de um plano de fundo inicial é feita usando um número N de quadros interpolados, evitando imagens similares que prejudicariam o resultado final. Os resultados apresentados neste trabalho usam 11 quadros de referência selecionados com aproximadamente meio segundo de intervalo entre eles. A Figura 10 apresenta os quadros selecionados em uma execução padrão usando o vídeo de referência. Nota-se que nenhum dos

quadros está completamente livre de interferência de objetos móveis.

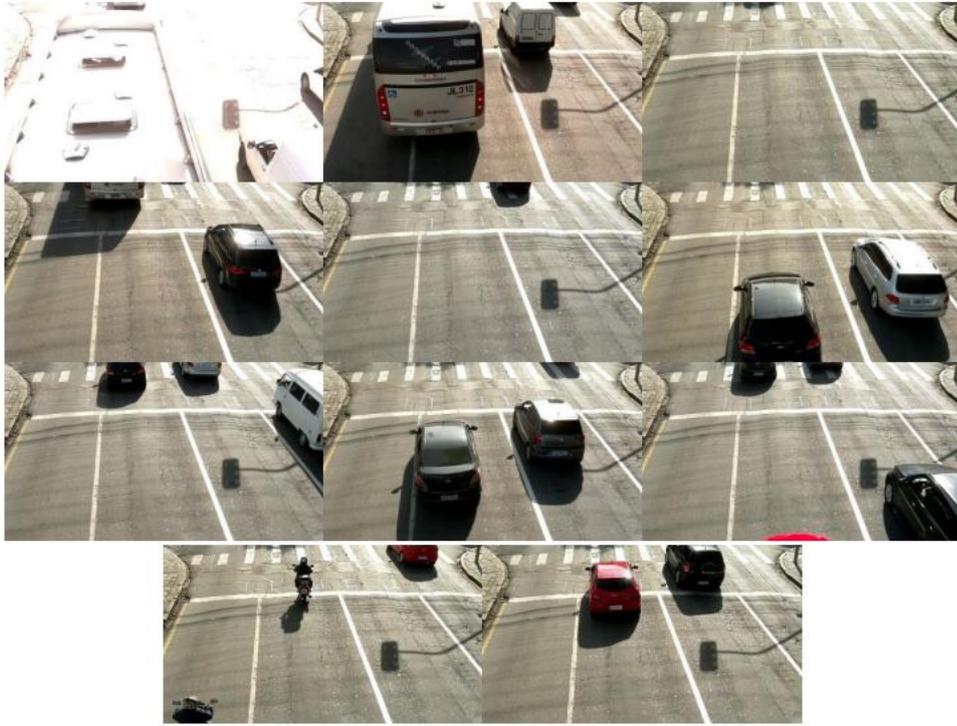


Figura 10: Quadros utilizados na estimativa de um plano de fundo inicial.

Usando os 11 quadros selecionados como referência, extrai-se o valor mediano para cada pixel individual. A técnica é idêntica à subtração de fundo usando a mediana, porém aqui é usada apenas como referência inicial para o conjunto de amostras que são atualizados usando o algoritmo de segmentação escolhido. A Figura 11 mostra o resultado obtido usando os quadros da Figura 10. Nota-se que o resultado obtido aproxima-se muito de um plano de fundo ideal.



Figura 11: Plano de fundo estimado.

O Algoritmo 1 é referente à etapa da estimação do plano de fundo baseada nas amostras, recebendo como entrada um vetor *amostras* com N amostras, tal que *amostras*[i] é a imagem da posição i do vetor e onde todas as amostras possuem o mesmo número de linhas e colunas, enquanto *novo_frame* é o quadro de entrada atual do vídeo. A entrada *intervalo* corresponde ao intervalo que deve ser dado para a amostragem dos quadros de entrada e *pixel_list* é o vetor de N *pixels* em uma mesma posição, enquanto a saída é o plano de fundo estimado *Bg* formado pelos pixels medianos entre as amostras selecionadas encontrados usando o algoritmo *Quick Select*.

Algoritmo 1: Plano de fundo inicial

```

1  entrada:  $I_{m,n}$  amostras[n_amostras], novo_frame
2          int intervalo
3  saida:  $I_{m,n}$  Bg
4  inicio
5      cont_t  $\leftarrow$  0
6      cont_n  $\leftarrow$  0
7  enquanto cont_n < n_amostras:
8      frame  $\leftarrow$  novo_frame
9      se cont_t % intervalo = 0
10         amostras[cont_n]  $\leftarrow$  frame
11         cont_n = cont_n + 1
12         cont_t = cont_t + 1
13
14     // para cada pixel
15     para j  $\leftarrow$  0 : ROWS
16         para i  $\leftarrow$  0 : COLUMNS
17             Bg(j, i)  $\leftarrow$  quickselect(amostras, n_amostras / 2 + 1)
```

4.4 SEGMENTAÇÃO

Com o plano de fundo estimado, a etapa seguinte é a segmentação dos objetos em movimento. Para esta etapa buscou-se uma técnica que primariamente apresentasse uma segmentação precisa dos objetos em movimento, filtrando ao máximo falsos positivos e ruídos devido a variações no plano de fundo. Com este fator em questão e buscando uma solução com custo computacional eficiente a técnica escolhida é a ViBe (*Visual Background Extractor*) (BARNICH; DROOGENBROECK, 2011; DROOGENBROECK; PAQUOT, 2012), descrita em mais detalhes nas próximas subseções.

4.4.1 CLASSIFICAÇÃO DOS PIXELS

Após determinado o plano de fundo inicial a etapa subsequente é a implementação do processo de classificação dos pixels como primeiro plano (objetos em movimento) ou parte do plano de fundo. A técnica *ViBe* utilizada modela cada *pixel* a ser classificado do plano de fundo usando um conjunto de amostras ao invés de uma função de probabilidade ou um modelo estático. Neste caso, o valor do *pixel* a ser classificado é comparado com um subconjunto das amostras dentro do conjunto total. Segundo Barnich e Droogenbroeck (2011) esta é uma distinção importante com relação a outras técnicas clássicas de segmentação.

Sendo $I(m, n)$ o valor em determinado espaço de cores de um *pixel* localizado em (m, n) da imagem I e v_i uma amostra do plano de fundo de índice i , cada *pixel* $I(m, n)$ do plano de fundo é modelado a partir de um conjunto M de amostras inicializadas na forma da equação 1.

$$M = \{v_1, v_2, \dots, v_N\} \quad (1)$$

Cada *pixel* $I(m, n)$ do quadro de entrada é comparado com um subconjunto de amostras de M . A comparação realizada é a distância entre o valor dos *pixels* em seu respectivo espaço de cores. Para imagens em escala de cinza a distância é a diferença absoluta entre os valores dos *pixels* comparados, enquanto em espaços de cores de três dimensões a diferença calculada é a distância Euclidiana. O valor do *pixel* $I(m, n)$ é então classificado como parte do plano de fundo se a cardinalidade \aleph da intersecção entre a esfera de raio R S_R e a coleção de amostras $M(x)$ é maior ou igual ao parâmetro \aleph_{min} , ou seja,

$$\aleph \geq \aleph \{S_R(v(x)) \cap \{v_1, v_2, \dots, v_N\}\}. \quad (2)$$

A classificação de um *pixel* pode ser descrita como a comparação de $I(m, n)$ com o subconjunto de N amostras de M usando um raio R como parâmetro limitante. A Figura 12 ilustra este processo em um espaço bi-dimensional para representar o valor de cada pixel. Os pontos P_i são os pixels do subconjunto de amostras selecionadas de M e o círculo de raio R serve para calcular a cardinalidade de $I_{m, n}$.

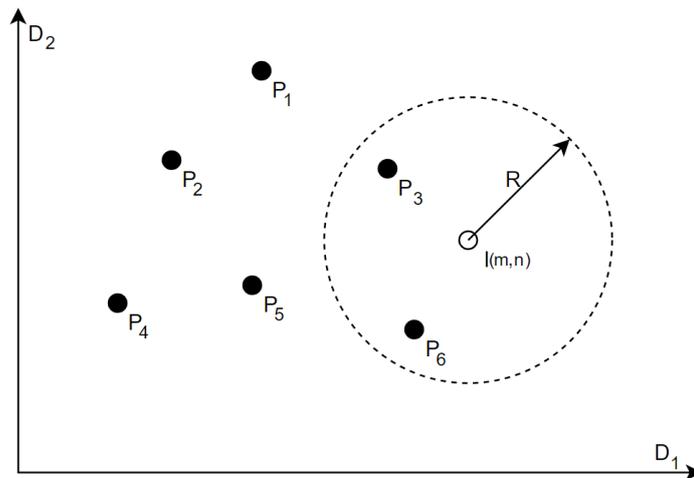


Figura 12: Representação da comparação usando um espaço bi-dimensional.

Nota-se que não é necessário que todas as amostras se encontrem dentro do limite estabelecido, o processo de segmentação trabalha até encontrar o valor mínimo de cardinalidade \aleph_{min} .

Em seu trabalho, Barnich e Droogenbroeck (2011) determinam que os valores de 20 para a distância do raio R em um espaço de cores de representado por 8 bits (0-255) e 2 para a cardinalidade mínima são satisfatórios para uma correta segmentação da imagem. Estes valores não são alterados ou adaptados durante a execução do programa.

O Algoritmo 2 apresenta o pseudo-código do *loop* principal do ViBe, para o qual nota-se que pixels do quadro de entrada são comparados com o raio R e usados para substituir amostras do plano de fundo selecionadas de forma aleatória. O algoritmo recebe os parâmetros raio R , número de amostras a serem comparadas N e cardinalidade mínima \aleph_{min} . A saída é a máscara com um único canal de 8 bits $mask_{m,n}$ com os pixels classificados como parte do plano de fundo (0) ou parte da frente correspondente a um objeto em movimento (255).

Algoritmo 2: ViBe

```

1  entrada: int R, int N, int  $\mathfrak{K}_{min}$ 
2  saida: I_{m,n} mask
3  inicio
4
5  background  $\leftarrow$  0
6  foreground  $\leftarrow$  255
7
8  // Para cada pixel
9  para j  $\leftarrow$  0 : ROWS
10   para i  $\leftarrow$  0 : COLUMNS
11     cont  $\leftarrow$  0
12
13     index  $\leftarrow$  0
14
15     enquanto ((cont <  $\mathfrak{K}_{min}$ ) && (index < N)):
16       dist  $\leftarrow$  calcDistance(frame(j, i), amostra(j, i)[index])
17
18       se dist < R:
19         count  $\leftarrow$  count + 1
20
21         index  $\leftarrow$  index + 1
22
23         se count  $\geq$   $\mathfrak{K}_{min}$ :
24           mask(j, i)  $\mathfrak{K}_{min}$  background
25
26           rand  $\mathfrak{K}_{min}$  randomNumber(0, N)
27           amostra(j, i)[rand]  $\mathfrak{K}_{min}$  frame(j, i)
28
29           rand  $\mathfrak{K}_{min}$  randomNumber(0, N)
30            $x_v \leftarrow$  randomNumber(-R, R)
31            $y_v \leftarrow$  randomNumber(-R, R)
32           amostra(j +  $x_v$ , i +  $y_v$ )[rand]  $\mathfrak{K}_{min}$  frame(j, i)
33
34       senao:
35         mask(j, i)  $\leftarrow$  foreground

```

4.4.2 INICIALIZAÇÃO DO PLANO DE FUNDO

As amostras de plano de fundo utilizados pelo algoritmo de segmentação são atualizados durante a execução. Para uma execução consideravelmente mais eficiente é recomendado inicializar os modelos de referência de alguma forma, para tal este projeto faz uso do plano de fundo obtido na seção 4.3 como o conjunto inicial de amostras.

Como a seleção das amostras para comparação e atualização é feita de forma aleatória, durante a execução do programa as amostras são adaptadas de forma diferente, mesmo que inicializadas iguais. O único ponto negativo desta abordagem é que são necessários alguns

segundos de execução do vídeo para a geração do plano de fundo inicial, segundos estes onde a segmentação não é realizada.

4.4.3 ATUALIZAÇÃO DO MODELO DE PLANO DE FUNDO

O próximo passo fundamental do algoritmo ViBe é a atualização das amostras de plano de fundo para que estas se adaptem de acordo com as mudanças no ambiente, como iluminação e novos objetos estacionários.

A abordagem clássica em um sistema de múltiplas amostras de plano de fundo é a substituição integral dos quadros de referência mais antigos por novos quadros de entrada do programa. Usando esta lógica, várias vezes serão substituídos valores de *pixel* individuais desnecessariamente.

O algoritmo ViBe em contrapartida apresenta as seguintes características:

- Atualização das amostras de plano de fundo sem memória: diferente de remover a amostra mais antiga do modelo, escolhe-se uma amostra de forma aleatória seguindo uma função de probabilidade uniforme. Ao contrário da abordagem clássica, neste caso nem sempre a amostra mais antiga é substituída. A atualização é chamada de *memoryless* pois a probabilidade de uma amostra ser preservada em um intervalo de tempo é independente deste tempo em questão.
- Subamostragem no tempo: quando um *pixel* é classificado como parte do plano de fundo ele não substitui uma das amostras automaticamente, é calculada uma chance usando uma função de número aleatório (aproximadamente 1 em 16 vezes). Esta chance de substituição tem como resultado a extensão do tempo de vida das amostras. O produto final é um sistema mais robusto em relação a mudanças instantâneas e curtas no ambiente.
- Propagação de *pixels* nas amostras de plano de fundo: Aqueles *pixels* do plano de fundo cobertos por objetos segmentados também precisam ser atualizados. O algoritmo ViBe faz isso sorteando *pixels* vizinhos para propagar a atualização da amostra de plano de fundo para estas áreas cobertas por objetos segmentados. Isto faz com que alguns *pixels* sejam alterados erroneamente, contudo o número de amostras diferentes garante que valores incorretos sejam diluídos e não afetem o resultado final de maneira significativa.

4.5 PÓS-PROCESSAMENTO

Além da segmentação direta usando a técnica proposta, são necessários alguns ajustes para obter-se o resultado esperado final, ou seja, máscaras uniformes e com o mínimo de ruído possível nas áreas de plano de fundo.

Neste etapa são aplicadas duas técnicas: fechamento morfológico que consiste de uma erosão seguida de uma dilatação da imagem em questão, seguido de um filtro da mediana binário.

A Figura 13a e 13b apresentam respectivamente máscaras antes e após a aplicação dos filtros mencionados. Nota-se que a máscara final gerada aproxima-se consideravelmente do esperado, envolvendo em sua totalidade o objeto em movimento. Inconsistências no interior da máscara acontecem devido à proximidade entre a área detectada e o plano de fundo (como por exemplo veículo com cor similar ao asfalto) e estão ligadas ao limiar usado na segmentação.

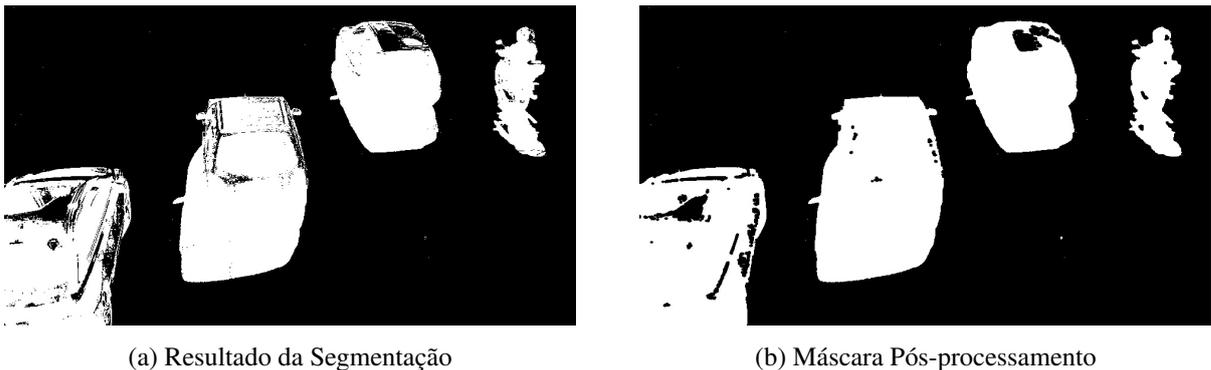


Figura 13: Máscara da segmentação e pós processamento em um quadro selecionado.

4.6 TESTE DE LUMINÂNCIA

Para o processo de segmentação das sombras dos veículos em movimento, o teste inicial e de caráter mais trivial é em relação à característica básica das regiões sombreadas: estas são mais escuras em relação às mesmas regiões quando não sombreadas.

A um nível objetivo de análise de imagens digitais, usa-se o espaço de cores Lab para determinar regiões candidatas. O canal L do espaço de cores serve de referência para a comparação de luminosidade, enquanto os canais a e b são usados para detectar diferenças drásticas de cor, o que também faz do pixel analisado um possível candidato.

O Algoritmo 3 a seguir ilustra este primeiro teste. O algoritmo recebe dois *pixels* I_m e B_g provenientes do quadro a ser examinado e do modelo de plano de fundo respectivamente,

ambos convertidos para o espaço Lab. As comparações do primeiro canal são relativas à luminosidade e são feitas na seguinte ordem: testa se o pixel do quadro é mais escuro que o plano de fundo, testa se o pixel não é muito claro e finalmente testa se existe uma diferença significativa entre os dois *pixels* quando aquele proveniente do quadro é muito escuro. Com a análise de luminosidade feita, ainda é realizado um teste extra relativo à similaridade de cor entre os *pixels*. Os parâmetros utilizados nos testes (20, 80) foram decididos após testes. A saída é a classificação do *pixel* como candidato (verdadeiro) ou não (falso).

Algoritmo 3: Teste no Espaço Lab

```

1  entrada: Pixel Im, Bg
2  saída: Booleana is_candidate
3
4  inicio
5    se Im[0] > Bg[0]:
6      retorna falso
7
8    se Im[0] > 80:
9      retorna falso
10
11   se Im[0] < 20 || Bg[0] > 80:
12     retorna verdadeiro
13
14   diff_a ← Im[1] - Bg[1]
15   diff_b ← Im[2] - Bg[2]
16   diff ←  $\sqrt{\text{diff}_a^2 + \text{diff}_b^2}$ 
17   se diff > 20:
18     retorna falso
19
20   retorna verdadeiro

```

A coluna (c) da Figura 14 apresenta o resultado parcial aplicado a algumas das máscaras geradas durante a segmentação. Nota-se que nesta etapa já existe um resultado positivo em formação, com uma separação aproximada do veículo e sua sombra.

4.7 TESTE DE TEXTURA

Com os primeiros *pixels* candidatos definidos, é necessária a aplicação de uma técnica subsequente para aprimorar os resultados parciais. Neste caso aplica-se um teste de correlação de textura. Entende-se que regiões sombreadas mantêm uma textura similar àquela do plano de fundo e conseqüentemente apresentam um valor alto de correlação de textura, enquanto objetos detectados no geral tem um valor de correlação inferior. Assim é necessária uma forma de estimar um valor que represente a diferença de textura entre uma região do plano de fundo e aquela a ser analisada.

A classificação proposta é feita baseada no método proposto por Sanin et al. (2010) no qual a diferença nos valores de textura é calculada a partir dos gradientes da área em questão no quadro de entrada e no plano de fundo. No trabalho citado o autor propõe determinar duas regiões distintas do maior tamanho possível e sugere o uso de detecção de bordas para separar estas duas regiões. Durante experimentações com a técnica, locais de alta intensidade de sombra provaram-se de difícil determinação de uma borda enquanto a variação na forma das sombras não contribuiu para o estabelecimento de uma regra geral para a separação entre sombra e veículo. Estes obstáculos induziram o desenvolvimento de uma outra abordagem congênere.

Neste trabalho, propõe-se fazer a análise de correlação de textura usando janelas quadrangulares, fazendo a análise para cada *pixel* através da sua vizinhança imediata. A análise desta forma torna-se mais localizada e o parâmetro determinante para o desempenho torna-se o tamanho da janela utilizada.

Neste ponto do processo têm-se disponíveis as máscaras obtidas através da segmentação e aplicação do teste de luminosidade e cor para classificação de *pixels* candidatos iniciais. O primeiro passo para a estimativa de uma correlação de textura é o cálculo da magnitude $|P|$ e orientação θ_p dos gradientes através das fórmulas:

$$|P| = \sqrt{\nabla_x^2 + \nabla_y^2} \quad (3)$$

$$\theta_p = \arctan(\nabla_y/\nabla_x) \quad (4)$$

Nas equações acima, ∇_x e ∇_y são o gradiente horizontal e o gradiente vertical respectivamente, calculados através da diferença em intensidade entre *pixels* vizinhos. O Algoritmo 4 representa a função utilizada na aquisição dos gradientes necessários para esta etapa. A função calcula para cada *pixel* da imagem os valores dos gradientes com base nos vizinhos imediatos e então calcula a orientação e magnitude baseada nestes valores. As saídas são dadas na forma de matrizes de tamanho igual ao das imagens e onde os valores são armazenados nas posições correspondentes as coordenadas de dados *pixels*.

Algoritmo 4: Cálculo dos Gradientes

```

1  entrada:  $I_{m,n}$  imagem
2  saida:  $\nabla_x[][]$ ,  $\nabla_y[][]$ ,  $\theta[][]$ ,  $|P|[][]$ 
3  inicio
4    para  $j \leftarrow 1 : \text{ROWS} - 1$ 
5      para  $i \leftarrow 1 : \text{COLUMNS} - 1$ 
6         $\nabla_x[j][i] \leftarrow |\text{image}[j][i-1] - \text{image}[j][i+1]|$ 
7         $\nabla_y[j][i] \leftarrow |\text{image}[j-1][i] - \text{image}[j+1][i]|$ 
8         $\theta[j][i] \leftarrow \arctan(\nabla_x[j][i], \nabla_y[j][i])$ 
9         $|P|[j][i] = \sqrt{\nabla_x^2 + \nabla_y^2}$ 
10 fim

```

A correlação é dada pela média da diferença de orientação entre os *pixels* do quadro e os *pixels* do plano de fundo. Esta diferença de orientação é calculada através da Equação 5, que corresponde ao cálculo da similaridade de cossenos e é ocasionalmente utilizada em testes de similaridade de imagens e reconhecimento facial (CHEN; CHU, 2005; NGUYEN; BAI, 2010). Na equação F_x e F_y correspondem aos gradientes x e y do quadro de entrada enquanto B_x e B_y correspondem aos gradientes x e y do plano de fundo.

$$\Delta\theta_p = \arccos \left(\frac{\nabla F_x \nabla B_x + \nabla F_y \nabla B_y}{\sqrt{(\nabla F_x^2 + \nabla F_y^2)(\nabla B_x^2 + \nabla B_y^2)}} \right) \quad (5)$$

O resultado desta etapa pode ser observado na coluna (d) da Figura 14. Nota-se que a análise de correlação de textura refinou os candidatos do veículo, no entanto criou um conjunto de pixels erroneamente classificados ao redor da borda da sombra. Isto ocorre pois a correlação de textura é calculada a partir dos gradientes, e uma de suas propriedades é a maior magnitude nas bordas da imagem (DING; GOSHTASBY, 2001). A Figura 15 mostra os quadros após um pós-processamento destas bordas usando operações morfológicas para suavizar o erro induzido.

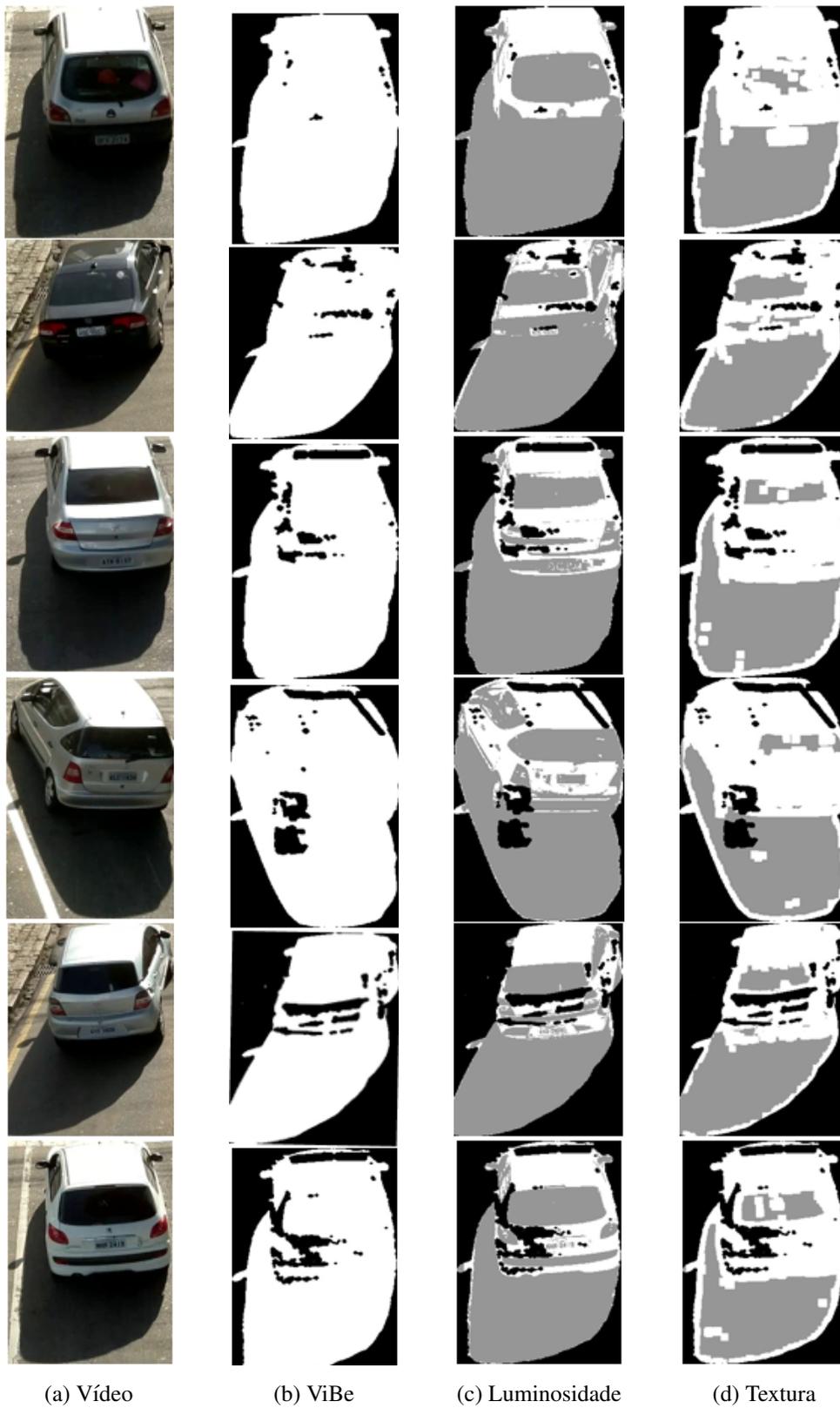


Figura 14: Acompanhamento dos passos iniciais em alguns quadros seleccionados.

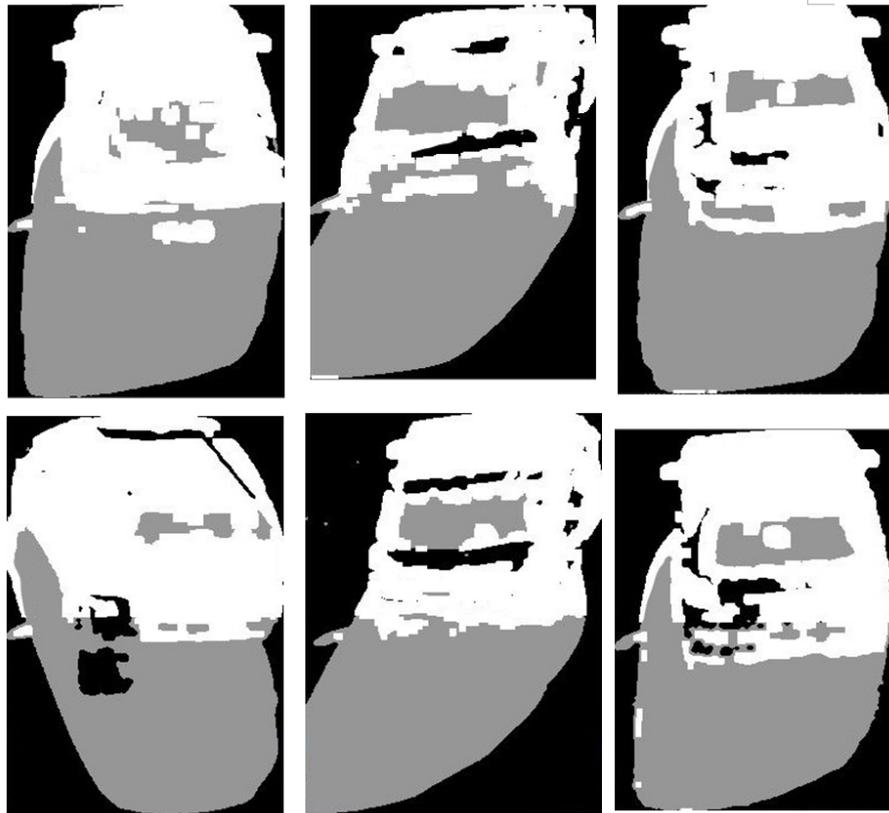


Figura 15: Pós-processamento para correção da coluna (d) da Figura 14

4.8 APROXIMAÇÃO POR ELIPSES

Os resultados obtidos até esta etapa são significativos e a maior parte dos *pixels* identificados como parte do objeto encontram-se sobre os veículos em questão, porém ainda existe um número de falsos positivos espalhados sobre a região sombreada da via, induzindo portanto a aplicação de um próximo nível de refinamento da máscara gerada. Dada a característica do problema, opta-se por uma abordagem probabilística para a eliminação parcial de *pixels* aberrantes. Como os veículos não apresentam uma forma uniforme, uma abordagem de aplicação direta do desvio padrão para excluir pontos mais afastados da centroides do *blob* se torna inadequada.

A distância de Mahalanobis destaca-se como uma solução viável uma vez que descreve uma abordagem multidimensional da distribuição Gaussiana usada para avaliar amostras em relação a um conjunto. A distância de Mahalanobis é dada em função da média da distribuição analisada e leva em consideração nos cálculos os desvios-padrão e a matriz de covariância de todas as variáveis envolvidas. A região da distância de Mahalanobis que descreve um determinado valor de variância ao redor da média é descrito na forma de uma elipse no

espaço 2D (ou seja, quando duas variáveis são envolvidas). A Figura 16 ilustra a distância de Mahalanobis aplicada a uma distribuição exemplo, onde as elipses definem regiões limitantes para diferentes percentis de variância.

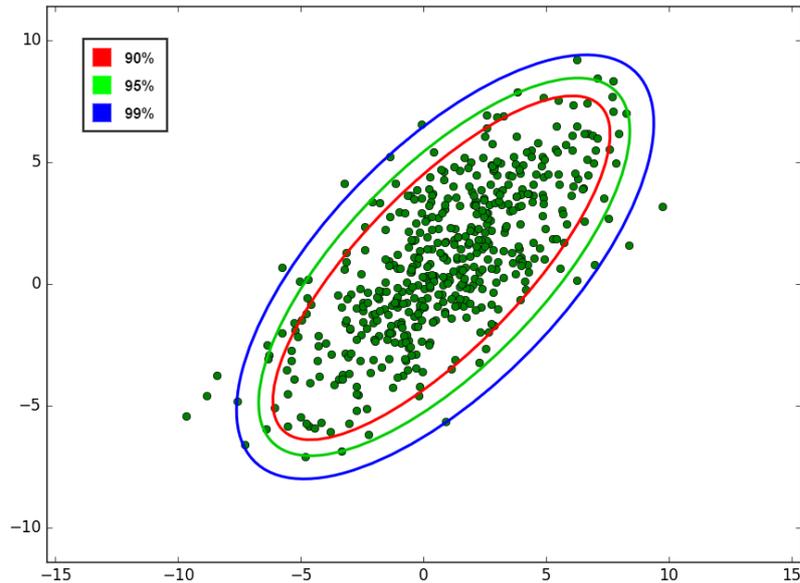


Figura 16: Distância de Mahalanobis aplicada à uma distribuição de exemplo.

Para a aplicação proposta não tratamos de uma distribuição de amostras propriamente dita, mas de *pixels* em uma imagem, e usa-se da forma da elipse como delimitador para o veículo. Os *pixels* candidatos servem como amostras da distribuição com suas coordenadas x e y como sua valoração.

4.8.1 CÁLCULO DA DISTÂNCIA DE MAHALANOBIS

Para um espaço de amostras bi-dimensional como o da Figura 16, a correlação entre as variáveis x e y pode ser descrita através de uma matriz 2×2 chamada matriz de covariância na forma:

$$S = \begin{bmatrix} \sigma(x,x) & \sigma(x,y) \\ \sigma(y,x) & \sigma(y,y) \end{bmatrix} \quad (6)$$

Onde $\sigma(x,x)$, $\sigma(y,y)$, $\sigma(x,y)$ e $\sigma(y,x)$ descrevem as covariâncias entre as variáveis e são calculadas conforme a equação Figura 7 onde $\mathbb{E}(x)$ é o valor esperado de x , neste caso a média de x :

$$\sigma(x, y) = \mathbb{E}[(x - \mathbb{E}(x))(y - \mathbb{E}(y))] \quad (7)$$

A matriz de covariância pode então ser utilizada no cálculo da distância de Mahalanobis $d_M(x, y)$ através da Equação 8.

$$d_M(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)} \quad (8)$$

4.8.2 DETERMINAÇÃO DAS DIMENSÕES DA ELIPSE

A partir da matriz de covariância é possível determinar a forma elipsoidal da distribuição. Isto é possível pois os dados relativos à variância correspondem à extensão da distribuição dos dados, enquanto as covariâncias podem ser interpretadas como a orientação dos mesmos. Mais precisamente, para um espaço bi-dimensional, os dois autovetores de maior intensidade da matriz de covariância sempre apontam na direção das duas maiores variações dos dados enquanto a magnitude destes vetores é igual aos autovalores correspondentes.

A Figura 17 ilustra diferentes distribuições de exemplo com suas respectivas matrizes de covariâncias, assim como seus respectivos autovetores. Destaca-se neste momento os vetores que servirão como referência para a orientação dos eixos da elipse.

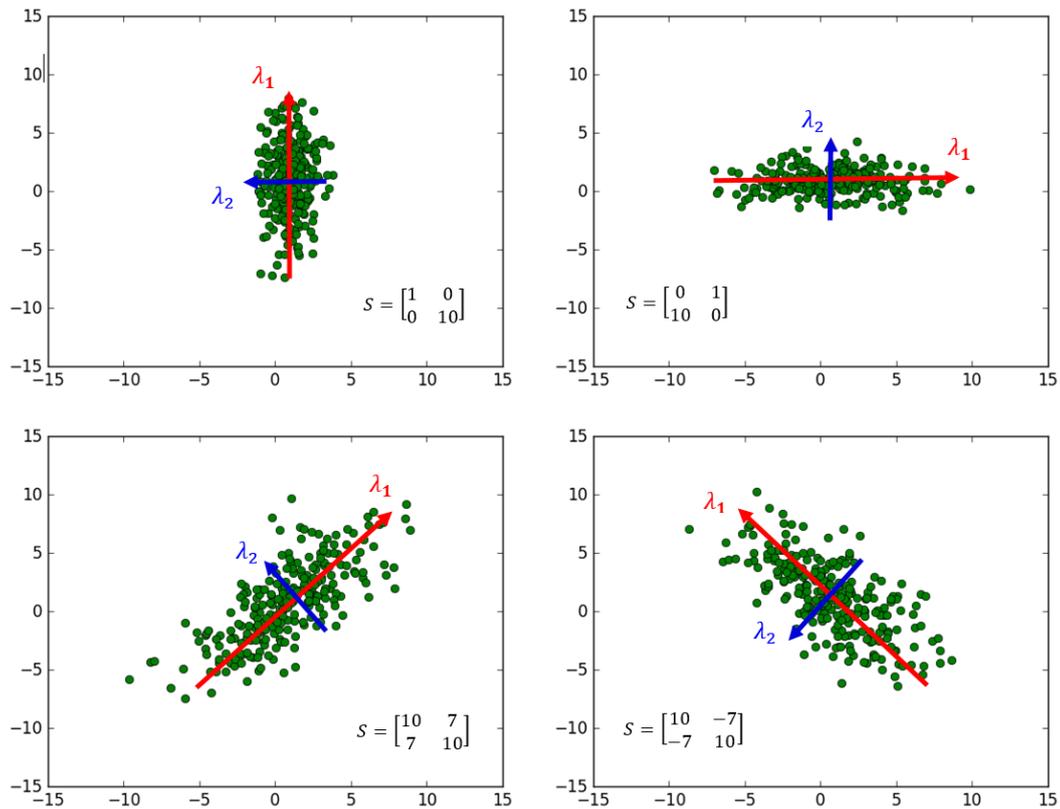


Figura 17: Autovetores e autovalores para diferentes matrizes de covariância em exemplos de distribuições.

4.8.3 IMPLEMENTAÇÃO DO MÉTODO USANDO DISTÂNCIA DE MAHALANOBIS

Determinado os *pixels* candidatos que servem como amostra, calcula-se a matriz de covariância e os respectivos autovetores e autovalores. Com estes valores é possível aplicar a distância de Mahalanobis para desenhar a elipse envolvendo os dados, e a partir desta distância exclui-se aqueles *pixels* mais distantes da centroide do *blob* (que coincide com o centro da distribuição). A distância de Mahalanobis, levando em consideração a correlação entre os pontos, faz com que os *pixels* mais distantes, na maior parte dos casos falsos-positivos, sejam identificados.

Estes primeiros *pixels* aberrantes identificados são então re-classificados como parte da sombra e um dos *pixels* internos da elipse é marcado como positivo em seu lugar. É realizada basicamente uma operação de substituição, onde *pixels* menos prováveis de fazerem parte do objeto são substituídos por candidatos mais próximos da centroide do *blob*. A Figura 18 ilustra de maneira simplificada como a elipse é usada para substituir pontos externos por internos durante este processo.

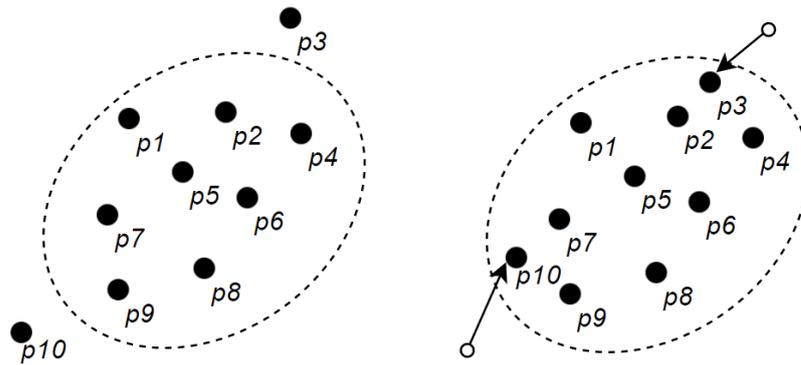


Figura 18: Uso do limiar baseado na distância de Mahalanobis para substituir pontos aberrantes.

Este processo de cálculo da distância de Mahalanobis e da re-classificação de *pixels* é repetida de forma iterativa, com o objetivo de cada execução aproximar de forma mais correta a posição da elipse sobre o veículo rastreado. O Algoritmo 5 apresenta um resumo da lógica implementada. O algoritmo recebe como entrada um vetor com todos os *pixels* candidatos, o limiar que define o percentil para a exclusão dos pontos e o número de iterações para qual os valores são re-classificados. A saída são os parâmetros adaptados dos autovalores e autovetores usados no desenho da elipse sobre a máscara do veículo após a última iteração do processo.

Algoritmo 5: Refinamento usando elipses

```

1  entrada: pixel* pontos_mascara ,
2          float limiar
3          int iter
4  saida: Elipse el
5  inicio
6
7      background ← 0
8      foreground ← 255
9
10     para iteracao ← 0 : iter:
11         covarMatriz ← calcCovarMatrix(pontos_candidatos)
12         invertCovarMatriz ← invert(covarMatriz)
13         para i ← 0 : p_candidatos.size()
14             se Mahalanobis(p_candidatos[i]) > limiar: // fora da elipse
15                 p_candidatos[i] ← background
16
17                  $x_n \leftarrow \text{random\_pixel\_x\_position}()$ 
18                  $y_n \leftarrow \text{random\_pixel\_y\_position}()$ 
19                 enquanto Mahalanobis(p_substituto( $x_n, y_n$ )) > limiar)
20                      $x_n \leftarrow \text{random\_pixel\_x\_position}()$ 
21                      $y_n \leftarrow \text{random\_pixel\_y\_position}()$ 
22
23                 \\ substitui o pixel e marca-o como objeto
24                 p_candidatos[i] ← p_substituto( $x_n, y_n$ )
25                 p_candidatos[i] ← foreground
26
27
28     el ← findEllipse(p_candidatos)
29     retorna el

```

5 EXPERIMENTOS E RESULTADOS

Neste capítulo apresentamos os recursos utilizados, experimentos realizados para avaliação de nosso sistema, e uma explicação dos resultados obtidos. Está descrita a métrica utilizada para a obtenção do conjunto de dados utilizados para a avaliação do sistema, bem como os parâmetros utilizados. Como conclusão detalhamos os resultados obtidos com a avaliação do sistema.

5.1 RECURSOS UTILIZADOS

Nesta seção serão descritas a linguagem de programação, a biblioteca, e o ambiente de desenvolvimento utilizado para a implementação do sistema. Dentre os recursos utilizados podemos citar:

- C++ é uma linguagem de uso geral e multi-paradigma, que combina características de linguagens alto nível e de baixo nível. A linguagem é amplamente utilizada devido ao seu desempenho e é considerada uma das linguagens mais populares.
- OpenCV é uma biblioteca para desenvolvimento na área de visão computacional. Foi desenvolvida pela Intel, e é uma biblioteca livre para uso acadêmico e comercial. A biblioteca foi desenvolvida nas linguagens C/C++, e atualmente conta com suporte para Java, Visual Basic e Python.
- Microsoft Visual Studio é um ambiente de desenvolvimento com amplo suporte para diversas linguagens de programação, entre elas podemos citar: C, C++, C#, Visual Basic, J# e atualmente conta com suporte para Python. Faremos uso do Microsoft Visual Studio Community, versão que é disponibilizada gratuitamente.

Todos os testes foram realizados utilizando linguagem C++, em uma máquina equipada com processador Intel Core i5 (2,0 GHz), com 8 GB de memória RAM e sistema operacional Windows 10 Home 64 bits.

5.2 DADOS DE REFERÊNCIA

Os dados de referência usados foram gerados manualmente. No processo de marcação manual as elipses foram desenhadas várias vezes, até que o resultado obtido fosse o mais próximo possível de um resultado ideal esperado, ou seja, envolvendo o carro analisado e evitando o máximo da região sombreada possível. Para os testes foram escolhidos 150 valores de referência, que obedecem o critério de marcação. A Figura 19 apresenta alguns exemplos de elipses usadas na validação do algoritmo.



Figura 19: Exemplos de áreas determinadas manualmente para validação.

5.3 METODOLOGIA DE TESTE

O método de avaliação das regiões determinadas é baseado na métrica aplicada no *Pascal VOC Challenge* onde é utilizado um conjunto de dados e as respectivas áreas correspondentes dos objetos segmentados para comparação (EVERINGHAM et al., 2010). Enquanto no *Pascal VOC Challenge* a detecção é dada na forma de 4 resultados objetivos (*true positive*, *true negative*, *false positive*, *false negative*), neste trabalho a comparação realizada ocorre entre duas elipses, onde o resultado de interesse é a similaridade entre as respectivas áreas, calculada através da Equação 9, onde A indica a área da máscara.

$$IoU = \frac{A_{calculada} \cap A_{referencia}}{A_{calculada} \cup A_{referencia}} \quad (9)$$

A razão entre a área coincidente dada pela intersecção e a área total dada pela união retorna um valor percentual que representa o quanto as duas elipses coincidem, nomeada aqui *IoU* (*intersection over union*). Nota-se que a área mencionada indica a posição e disposição da elipse no quadro e não apenas do tamanho da mesma. A Figura 20 apresenta a intersecção entre as elipses, ilustrando como a área válida é encontrada. Na implementação, a intersecção e a união são calculadas *pixel a pixel*.

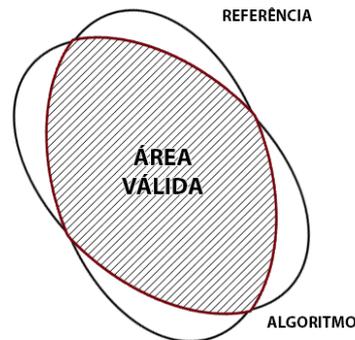


Figura 20: Intersecção entre a elipse de referência e a elipse gerada pelo algoritmo.

5.3.1 PARAMETRIZAÇÃO

O sistema como um todo contém diferentes etapas onde os resultados parciais tem clara influência no resultado final. Tendo a segmentação como satisfatória usando *ViBe*, ainda existem alguns parâmetros que influenciam a eficácia e o desempenho do algoritmo. Os parâmetros principais são:

- **Tamanho da janela usada no cálculo da correlação de textura:** este parâmetro é relacionado ao processo de análise de texturas. O cálculo é feito em relação a vizinhança do pixel em questão. Janelas menores apresentam resultados mais localizados, enquanto áreas maiores tem uma descrição melhor da correlação de textura porém podem classificar *pixels* erroneamente por causa da vizinhança.
- **Número de iterações na aplicação de Mahalanobis:** a distância de Mahalanobis é utilizada para a exclusão de *pixels aberrantes* não identificados através das etapas anteriores. Este parâmetro determina quantas vezes este processo é realizado. Nota-se que na forma como está aplicada, o aumento no número de iterações apresenta um retorno decrescente, ou seja, por exemplo a diferença entre uma e 4 iterações é significativa, enquanto a diferença entre 8 e 10 iterações aproxima-se de zero.

- **Limiar da distância de Mahalanobis usada no descarte de *pixels*:** diferente do número de vezes que a técnica aplicada, a distância comparada a um limiar serve como parâmetro na forma de quão distantes *pixels* precisam estar para serem considerados aberrantes. Um limiar maior pode ser contrabalanceado com mais iterações, porém um limiar menor corre o risco de descartar *pixels* que fazem parte do veículo rastreado.

Antes da produção de um resultado final, o algoritmo é primeiramente colocado em teste variando estes parâmetros incrementalmente de forma a identificar os valores mais adequados, tais que estes são usados durante a produção e análise dos resultados finais. Nota-se que estes parâmetros são situacionais de acordo com o vídeo de entrada que é utilizado, características como a resolução do vídeo tem um efeito significativo na escolha dos valores.

As Tabelas 1, 2 e 3 contêm a média e desvio padrão do resultado de similaridade *IoU* das áreas variando os parâmetros apresentados. As tabelas completas encontram-se no Apêndice A. Resultados notáveis são a variação mais significativa em relação ao limiar para a distância de Mahalanobis usada, e a convergência dos resultados no incremento no número de iterações do método da elipse.

Tabela 1: Teste com iterações e limiar para distância de Mahalanobis fixados

Tamanho da janela	Média	Desvio Padrão
7	0.8036	0.08947
9	0.8074	0.08768
11	0.8081	0.08611
13	0.8053	0.08609
15	0.8007	0.08690

Fonte: Autoria própria.

Tabela 2: Tamanho da janela e limiar para distância de Mahalanobis fixados

Nº iterações	Média	Desvio Padrão
2	0.7385	0.1147
4	0.8081	0.08611
6	0.8072	0.08283
8	0.8086	0.08412
10	0.8077	0.08419

Fonte: Autoria própria.

Com estes dados como orientação, determinam-se os parâmetros para o teste final seguindo a Tabela 4.

Tabela 3: Iterações e tamanho da janela fixados

Limiar Mahalanobis	Média	Desvio Padrão
1.8	0.6131	0.08282
2	0.7813	0.08609
2.2	0.8081	0.08611
2.4	0.7801	0.1130
2.6	0.7150	0.1382

Fonte: Autoria própria.

Tabela 4: Parâmetros escolhidos

Parâmetro	Valor
Tamanho da janela na correlação de textura	11
Número de iterações usando elipse	8
Distância de Mahalanobis	2.2

Fonte: Autoria própria.

5.4 ANÁLISE DOS RESULTADOS FINAIS

Aplicando os parâmetros da Tabela 4 produz-se a análise final do conjunto de técnicas propostas para detectar a sombra de veículos em movimento e assim rastreá-los sem a influência da mesma. Para as amostras de testes propostas o percentual de similaridade *IoU* entre as áreas detectadas e ideais é de 0.8115 (81,15%) na média, com desvio padrão de 0.0729 (7,29%). A Figura 21 apresenta um histograma do resultado dos testes, e a Tabela A no Apêndice A apresenta os resultados individuais para as amostras.

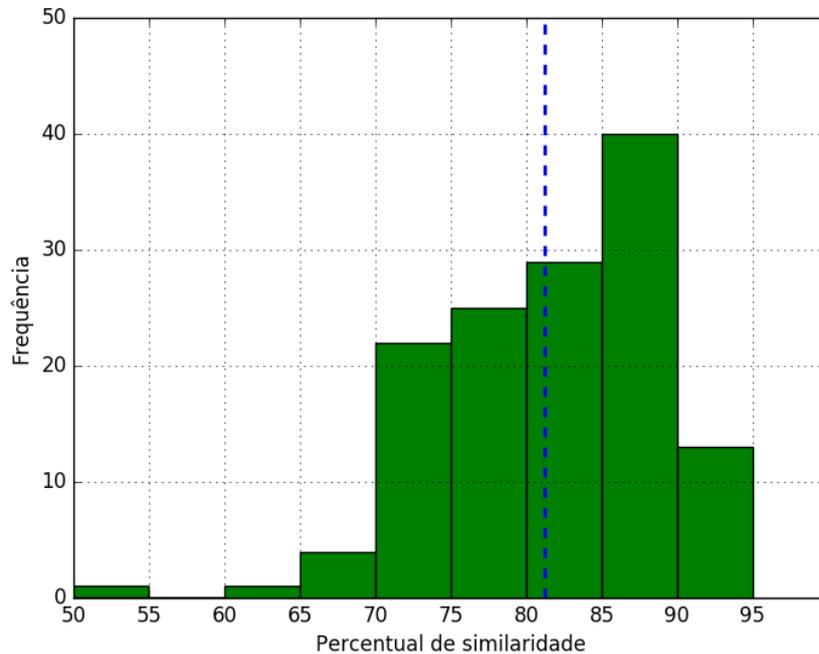


Figura 21: Histograma com resultado dos testes e linha da média.

Os valores máximo e mínimo encontrados de *IoU* foram 0.9417 (94.17%) e 0.5182 (51.82%) respectivamente. As duas situações que provocaram os erros mais drásticos foram: veículos com formas variadas, por exemplo uma caminhonete carregando uma escada na caçamba e uma parcela das motos (outras motos tiveram resultados satisfatórios); e veículos pretos com filmes nas janelas, onde torna-se mais difícil a separação entre objeto e sombra.

A Figura 22 apresenta alguns exemplos de resultados obtidos usando os parâmetros escolhidos. O algoritmo marca a região dos veículos de forma considerável, excluindo a maior parte da sombra na maioria dos casos. Entretanto, nota-se que o desvio padrão é consideravelmente elevado, devido a casos como a Figura (c), onde a elipse encontra-se distorcida e a métrica usada que é rigorosa.

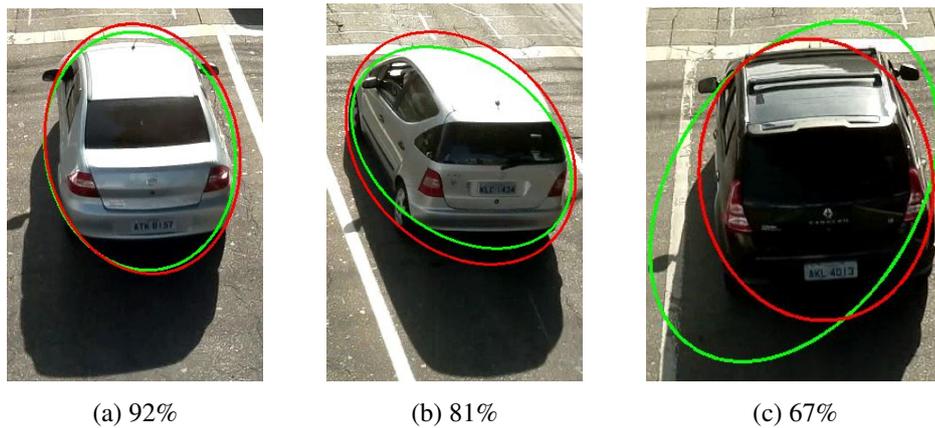


Figura 22: Exemplos de resultados com similaridade de área variadas. A elipse vermelha é a referência, enquanto a elipse verde é gerada pelo algoritmo.

O problema visualizado no exemplo (c) da Figura 22 acontece em casos onde a detecção de *pixels* candidatos à sombra não é ideal, seja por causa da etapa de segmentação ou nos testes relacionados à luminosidade e textura. A Figura 23 ilustra três fatores que prejudicam o resultado. Primeiramente nota-se que o vidro traseiro do veículo é detectado como parte da sombra por encontrar-se mais escuro que o resto da imagem, possivelmente por causa de algum tipo de filme. Em segundo lugar é visível também uma falha na segmentação do teto do veículo, causada pela reflexão parcial da luz solar sobre a lataria do veículo. Por último, existe também um erro em uma das faixas do asfalto, que é classificada como similar ao plano de fundo em alguns momentos da segmentação.



Figura 23: Mascara do exemplo (c) da Figura 22 com detecção de *pixels* candidatos.

6 CONSIDERAÇÕES FINAIS

Este projeto foi iniciado com o intuito de estudar diferentes possibilidades de detecção de sombras em vídeos de veículos em vias. Apresenta-se um resumo das diferentes abordagens da literatura ao problema e propõe-se uma combinação de técnicas que se complementem para atingir o objetivo proposto de rastrear os veículos sem suas respectivas sombras.

Usando um conjunto de técnicas que variam entre algoritmos provados academicamente e conceitos originais, obteve-se resultados significativos com margens de similaridade de áreas calculadas aceitáveis, alcançando um valor de 81,15% na melhor média. Este valor ainda não é completamente satisfatório e estabelece novas questões quanto a possíveis melhoramentos.

6.1 TRABALHOS FUTUROS

Com o processamento sequencial dividido em etapas distintas, é possível trabalhar em desenvolvimentos para cada uma das partes individualmente. A estruturação do algoritmo faz com que melhorias sejam propagadas no processamento e permite que novas etapas sejam adicionadas.

Na parte da segmentação fez-se o uso do algoritmo ViBe apresentado por Barnich e Droogenbroeck (2011). A técnica é aplicada neste trabalho pois apresenta uma conformidade satisfatória entre robustez e velocidade de adaptação a mudanças leves no plano de fundo. O algoritmo em contrapartida não resolve bem uma das situações práticas na aplicação proposta, o caso de veículos parados durante a luz vermelha do semáforo. Seria viável estudar alternativas para este caso específico.

A análise através da luminosidade é logicamente compreensível, contudo apresenta nuances que podem ser exploradas. O número de espaços de cor que podem ser trabalhados não é trivial e cada um apresenta uma forma diferente de estimar o brilho da luz, seja através da luminosidade, luminância ou luma. Neste trabalho a escolha do espaço Lab apresentou

resultados satisfatórios.

A abordagem de correlação de textura usa das ideias apresentadas por Tian et al. (2005) e Sanin et al. (2012). O conceito fundamental é o uso das características de magnitude e orientação de gradientes da imagem para determinar um valor abstrato que representa a textura em uma região. Levando em conta a forma física dos veículos e a dificuldade em determinar limites claros entre veículo e sombra, optou-se por aplicar a técnica em janelas, fazendo análise da vizinhança e classificando candidatos baseado nos *pixels* próximos. A aplicação nesta forma não está otimizada onde pontos muito escuros de sombra distorcem demasiadamente os valores de textura, entretanto os resultados ainda foram positivos, classificando corretamente pixels que não foram detectados no teste de luminosidade e melhorando a máscara para a etapa final.

A metodologia proposta mais inovadora consiste do uso da distribuição de probabilidade para o descarte de *pixels* aberrantes através da aplicação da distância de Mahalanobis diretamente sobre a máscara de *pixels* candidatos. Acredita-se que a abordagem probabilística apresente potencial na detecção de objetos com formas uniformes e é um assunto que requer mais estudos.

REFERÊNCIAS

- AHAD, M. A. R. et al. Motion history image: its variants and applications. **Machine Vision and Applications**, Springer, v. 23, n. 2, p. 255–281, 2012.
- BALCILAR, M.; SÖNMEZ, A. C. Extracting vehicle density from background estimation using kalman filter. In: IEEE. **Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on**. [S.l.], 2008. p. 1–5.
- BARNICH, O.; DROOGENBROECK, M. V. Vibe: A universal background subtraction algorithm for video sequences. **Image Processing, IEEE Transactions on**, IEEE, v. 20, n. 6, p. 1709–1724, 2011.
- BORA, D. J.; GUPTA, A. K.; KHAN, F. A. Comparing the performance of l^* a^* b^* and hsv color spaces with respect to color image segmentation. **arXiv preprint arXiv:1506.01472**, 2015.
- BOUWMANS, T. Recent advanced statistical background modeling for foreground detection—a systematic survey. **Recent Patents on Computer Science**, Bentham Science Publishers, v. 4, n. 3, p. 147–176, 2011.
- BUTLER, D.; SRIDHARAN, S.; JR, V. M. B. Real-time adaptive background segmentation. In: IEEE. **Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on**. [S.l.], 2003. v. 3, p. III–349.
- CHAROENPONG, T.; SUPASUTEEKUL, A.; NUTHONG, C. Adaptive background modeling from an image sequence by using k-means clustering. In: IEEE. **Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on**. [S.l.], 2010. p. 880–883.
- CHEN, C.-C.; AGGARWAL, J. K. Human shadow removal with unknown light source. In: IEEE. **Pattern Recognition (ICPR), 2010 20th International Conference on**. [S.l.], 2010. p. 2407–2410.
- CHEN, C.-C.; CHU, H.-T. Similarity measurement between images. In: IEEE. **Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International**. [S.l.], 2005. v. 2, p. 41–42.
- CHEN, T.-W.; HSU, S.-C.; CHIEN, S.-Y. Robust video object segmentation based on k-means background clustering and watershed in ill-conditioned surveillance systems. In: IEEE. **Multimedia and Expo, 2007 IEEE International Conference on**. [S.l.], 2007. p. 787–790.
- CUCCHIARA, R. et al. Detecting moving objects, ghosts, and shadows in video streams. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 25, n. 10, p. 1337–1342, 2003.
- DING, L.; GOSHTASBY, A. On the canny edge detector. **Pattern Recognition**, Elsevier, v. 34, n. 3, p. 721–725, 2001.

DROOGENBROECK, M. V.; PAQUOT, O. Background subtraction: Experiments and improvements for vbe. In: IEEE. **Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on**. [S.l.], 2012. p. 32–37.

EVERINGHAM, M. et al. The pascal visual object classes (voc) challenge. **International journal of computer vision**, Springer, v. 88, n. 2, p. 303–338, 2010.

FANG, L. Z.; QIONG, W. Y.; SHENG, Y. Z. A method to segment moving vehicle cast shadow based on wavelet transform. **Pattern Recognition Letters**, Elsevier, v. 29, n. 16, p. 2182–2188, 2008.

GONZALEZ, R. C.; WOODS, R. E. Digital image processing. Prentice hall, 2002.

HADI, R. A.; SULONG, G.; GEORGE, L. E. Vehicle detection and tracking techniques: a concise review. **arXiv preprint arXiv:1410.5894**, 2014.

HAINES, T. S.; XIANG, T. Background subtraction with dirichletprocess mixture models. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 36, n. 4, p. 670–683, 2014.

HAN, B. et al. Motion-segmentation-based change detection. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Defense and Security Symposium**. [S.l.], 2007. p. 65680Q–65680Q.

HASELHOFF, A.; KUMMERT, A. An evolutionary optimized vehicle tracker in collaboration with a detection system. In: IEEE. **Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on**. [S.l.], 2009. p. 1–6.

HSIEH, J.-W. et al. Automatic traffic surveillance system for vehicle tracking and classification. **Intelligent Transportation Systems, IEEE Transactions on**, IEEE, v. 7, n. 2, p. 175–187, 2006.

HUANG, J.-B.; CHEN, C.-S. Moving cast shadow detection using physics-based features. In: IEEE. **Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on**. [S.l.], 2009. p. 2310–2317.

JIANYONG, B.; RUNFENG, Y.; YANG, Y. A novel vehicle's shadow detection and removal algorithm. In: IEEE. **Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on**. [S.l.], 2012. p. 822–826.

KATO, J. et al. An hmm/mrf-based stochastic framework for robust vehicle tracking. **Intelligent Transportation Systems, IEEE Transactions on**, IEEE, v. 5, n. 3, p. 142–154, 2004.

KAZEMI, F. M. et al. Vehicle recognition using curvelet transform and svm. In: IEEE. **Information Technology, 2007. ITNG'07. Fourth International Conference on**. [S.l.], 2007. p. 516–521.

KUMAR, A. N.; SURESHKUMAR, C. Background subtraction based on threshold detection using modified k-means algorithm. In: IEEE. **Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on**. [S.l.], 2013. p. 378–382.

LEONE, A.; DISTANTE, C. Shadow detection for moving objects based on texture analysis. **Pattern Recognition**, Elsevier, v. 40, n. 4, p. 1222–1233, 2007.

- LIU, F.; GLEICHER, M. Texture-consistent shadow removal. In: **Computer Vision–ECCV 2008**. [S.l.]: Springer, 2008. p. 437–450.
- LO, B.; VELASTIN, S. Automatic congestion detection system for underground platforms. In: IEEE. **Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on**. [S.l.], 2001. p. 158–161.
- MANDELLOS, N. A.; KERAMITSOGLU, I.; KIRANOUDIS, C. T. A background subtraction algorithm for detecting and tracking vehicles. **Expert Systems with Applications**, Elsevier, v. 38, n. 3, p. 1619–1631, 2011.
- MARTEL-BRISSON, N.; ZACCARIN, A. Kernel-based learning of cast shadows from a physical model of light sources and surfaces for low-level segmentation. In: IEEE. **Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on**. [S.l.], 2008. p. 1–8.
- MELO, J. et al. Detection and classification of highway lanes using vehicle motion trajectories. **Intelligent Transportation Systems, IEEE Transactions on**, IEEE, v. 7, n. 2, p. 188–200, 2006.
- MESSELODI, S. et al. A kalman filter based background updating algorithm robust to sharp illumination changes. In: **Image Analysis and Processing–ICIAP 2005**. [S.l.]: Springer, 2005. p. 163–170.
- NADIMI, S.; BHANU, B. Physical models for moving shadow and object detection in video. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 26, n. 8, p. 1079–1087, 2004.
- NGUYEN, H. V.; BAI, L. Cosine similarity metric learning for face verification. In: SPRINGER. **Asian Conference on Computer Vision**. [S.l.], 2010. p. 709–720.
- PICCARDI, M. Background subtraction techniques: a review. In: IEEE. **Systems, man and cybernetics, 2004 IEEE international conference on**. [S.l.], 2004. v. 4, p. 3099–3104.
- PONSA, D. et al. 3d vehicle sensor based on monocular vision. In: IEEE. **Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE**. [S.l.], 2005. p. 1096–1101.
- QIN, R. et al. Moving cast shadow removal based on local descriptors. In: IEEE. **Pattern Recognition (ICPR), 2010 20th International Conference on**. [S.l.], 2010. p. 1377–1380.
- SANIN, A.; SANDERSON, C.; LOVELL, B. C. Improved shadow removal for robust person tracking in surveillance scenarios. In: IEEE. **Pattern Recognition (ICPR), 2010 20th International Conference on**. [S.l.], 2010. p. 141–144.
- SANIN, A.; SANDERSON, C.; LOVELL, B. C. Shadow detection: A survey and comparative evaluation of recent methods. **Pattern recognition**, Elsevier, v. 45, n. 4, p. 1684–1695, 2012.
- SCOTT, J.; PUSATERI, M. A.; CORNISH, D. Kalman filter based video background estimation. In: IEEE. **Applied Imagery Pattern Recognition Workshop (AIPRW), 2009 IEEE**. [S.l.], 2009. p. 1–7.

SEN-CHING, S. C.; KAMATH, C. Robust techniques for background subtraction in urban traffic video. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Electronic Imaging 2004**. [S.l.], 2004. p. 881–892.

SIVARAMAN, S.; TRIVEDI, M. M. A general active-learning framework for on-road vehicle recognition and tracking. **Intelligent Transportation Systems, IEEE Transactions on**, IEEE, v. 11, n. 2, p. 267–276, 2010.

SOBRAL, A.; VACAVANT, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. **Computer Vision and Image Understanding**, Elsevier, v. 122, p. 4–21, 2014.

STAUFFER, C.; GRIMSON, W. E. L. Learning patterns of activity using real-time tracking. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 22, n. 8, p. 747–757, 2000.

SUN, B.; LI, S. Moving cast shadow detection of vehicle using combined color models. In: IEEE. **Pattern Recognition (CCPR), 2010 Chinese Conference on**. [S.l.], 2010. p. 1–5.

TIAN, Y.-L.; LU, M.; HAMPAPUR, A. Robust and efficient foreground analysis for real-time video surveillance. In: IEEE. **Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on**. [S.l.], 2005. v. 1, p. 1182–1187.

VARGAS, M. et al. An enhanced background estimation algorithm for vehicle detection in urban traffic scenes. **Vehicular Technology, IEEE Transactions on**, IEEE, v. 59, n. 8, p. 3694–3709, 2010.

XU, D. et al. Cast shadow detection in video segmentation. **Pattern Recognition Letters**, Elsevier, v. 26, n. 1, p. 91–99, 2005.

YONEYAMA, A.; YEH, C. H.; KUO, C. J. Moving cast shadow elimination for robust vehicle extraction based on 2d joint vehicle/shadow models. In: IEEE. **Advanced Video and Signal Based Surveillance, 2003. Proceedings. IEEE Conference on**. [S.l.], 2003. p. 229–236.

ZHANG, W.; FANG, X. Z.; XU, Y. Detection of moving cast shadows using image orthogonal transform. In: IEEE. **Pattern Recognition, 2006. ICPR 2006. 18th International Conference on**. [S.l.], 2006. v. 1, p. 626–629.

ZHANG, W.; WU, Q. J.; YIN, H. Bing. Moving vehicles detection based on adaptive motion histogram. **Digital Signal Processing**, Elsevier, v. 20, n. 3, p. 793–805, 2010.

APÊNDICE A – RESULTADOS DOS TESTES DO ALGORITMO

Tabela 5: Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.

Amostra	7	9	11	13	15
1	0,7769	0,7654	0,7327	0,7092	0,6941
2	0,7448	0,7725	0,7866	0,8001	0,8098
3	0,7996	0,8340	0,8458	0,8521	0,8573
4	0,7577	0,7462	0,7496	0,7376	0,7208
5	0,8051	0,8017	0,8002	0,7941	0,7800
6	0,7316	0,7245	0,7320	0,7317	0,7331
7	0,7847	0,7978	0,8015	0,8152	0,8245
8	0,7300	0,7414	0,7529	0,7665	0,7757
9	0,7363	0,7521	0,7587	0,7634	0,7661
10	0,9084	0,8539	0,8192	0,7740	0,7305
11	0,7831	0,8135	0,8272	0,8334	0,8244
12	0,7807	0,7895	0,8005	0,7970	0,7989
13	0,6552	0,6759	0,6993	0,7047	0,7176
14	0,7443	0,7768	0,7964	0,8074	0,8254
15	0,8328	0,8540	0,8660	0,8756	0,8798
16	0,8889	0,8867	0,8718	0,8593	0,8451
17	0,8018	0,8109	0,8119	0,8143	0,8114
18	0,7455	0,7629	0,7835	0,8067	0,8238
19	0,8236	0,8302	0,8336	0,8349	0,8316
20	0,8731	0,8746	0,8726	0,8768	0,8687
21	0,9125	0,9178	0,9267	0,9195	0,9028
22	0,9337	0,9018	0,8798	0,8558	0,8409
23	0,8837	0,8581	0,8401	0,8303	0,8148

Tabela 5: Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.

Amostra	7	9	11	13	15
24	0,7635	0,7856	0,8049	0,8198	0,8314
25	0,7411	0,7234	0,7110	0,6999	0,6897
26	0,8046	0,7870	0,7714	0,7579	0,7474
27	0,8729	0,8766	0,8767	0,8751	0,8543
28	0,7073	0,7153	0,7163	0,7133	0,7168
29	0,6237	0,6121	0,5993	0,5910	0,5826
30	0,7737	0,7796	0,7864	0,7951	0,8070
31	0,8651	0,8793	0,8865	0,8847	0,8798
32	0,9036	0,9033	0,9006	0,8991	0,8944
33	0,8769	0,8802	0,8809	0,8775	0,8647
34	0,7883	0,7879	0,7822	0,7767	0,7726
35	0,7307	0,7220	0,7130	0,6952	0,6760
36	0,7313	0,7187	0,7073	0,6944	0,6751
37	0,7924	0,7984	0,8162	0,8312	0,8469
38	0,8372	0,8442	0,8456	0,8419	0,8329
39	0,8954	0,8823	0,8686	0,8539	0,8349
40	0,7922	0,7915	0,7806	0,7565	0,7383
41	0,7620	0,7817	0,8081	0,8521	0,8779
42	0,7315	0,7593	0,7832	0,7958	0,8018
43	0,9639	0,9714	0,9530	0,9129	0,9156
44	0,8940	0,8680	0,8453	0,8344	0,8163
45	0,9706	0,9648	0,9451	0,9144	0,8848
46	0,9210	0,9471	0,9649	0,9492	0,9308
47	0,8178	0,8299	0,8367	0,8398	0,8396
48	0,8242	0,8276	0,8381	0,8497	0,8538
49	0,8616	0,8515	0,8468	0,8390	0,8311
50	0,8350	0,8368	0,8388	0,8192	0,8019
51	0,6680	0,6664	0,6735	0,6664	0,6631
52	0,8431	0,8666	0,8769	0,8797	0,8762
53	0,6839	0,6900	0,7002	0,7006	0,6913

Tabela 5: Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.

Amostra	7	9	11	13	15
54	0,8335	0,8470	0,8602	0,8701	0,8717
55	0,7606	0,7876	0,8137	0,8362	0,8525
56	0,8305	0,8402	0,8516	0,8562	0,8544
57	0,9214	0,9521	0,9612	0,9390	0,9076
58	0,8846	0,8519	0,8275	0,8031	0,7847
59	0,8539	0,8619	0,8639	0,8660	0,8612
60	0,8481	0,8558	0,8568	0,8604	0,8607
61	0,9537	0,9249	0,9002	0,8785	0,8552
62	0,7611	0,7644	0,7671	0,7715	0,7635
63	0,7118	0,6945	0,6847	0,6658	0,6548
64	0,7128	0,7338	0,7611	0,7871	0,8085
65	0,7524	0,7901	0,8118	0,8318	0,8386
66	0,8599	0,8480	0,8312	0,8128	0,7947
67	0,8460	0,8424	0,8389	0,8398	0,8181
68	0,8716	0,8842	0,8916	0,8953	0,8922
69	0,8956	0,9019	0,9128	0,9245	0,9348
70	0,8199	0,8281	0,8264	0,8212	0,8164
71	0,8941	0,9056	0,8847	0,8479	0,8458
72	0,8520	0,8302	0,8071	0,7828	0,7642
73	0,7548	0,7766	0,7996	0,8184	0,8276
74	0,7524	0,7767	0,8013	0,8246	0,8418
75	0,9532	0,9324	0,9118	0,8934	0,8777
76	0,8131	0,7977	0,7802	0,7612	0,7481
77	0,7833	0,8020	0,8197	0,8349	0,8467
78	0,7786	0,7904	0,8100	0,8241	0,8390
79	0,9115	0,9112	0,9100	0,8986	0,8829
80	0,8808	0,8623	0,8438	0,8318	0,8155
81	0,8497	0,8477	0,8429	0,8390	0,8391
82	0,8076	0,7952	0,7875	0,7771	0,7705
83	0,6834	0,7294	0,7473	0,7710	0,7892

Tabela 5: Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.

Amostra	7	9	11	13	15
84	0,7359	0,7554	0,7745	0,7911	0,8043
85	0,8272	0,8218	0,7958	0,7878	0,7728
86	0,8042	0,8638	0,8956	0,9167	0,9345
87	0,4258	0,4178	0,4091	0,4134	0,4123
88	0,8594	0,8876	0,9102	0,9175	0,9159
89	0,9358	0,9340	0,9227	0,9109	0,9005
90	0,6842	0,7069	0,7343	0,7484	0,7771
91	0,6699	0,6944	0,7200	0,7337	0,7461
92	0,8034	0,8001	0,7937	0,7786	0,7564
93	0,7338	0,7128	0,7054	0,6795	0,6696
94	0,7097	0,7457	0,7706	0,7958	0,8063
95	0,7237	0,7534	0,7737	0,7843	0,8005
96	0,8206	0,8301	0,8280	0,8350	0,8409
97	0,8355	0,8494	0,8672	0,8805	0,8889
98	0,6420	0,6301	0,6150	0,5990	0,5788
99	0,6814	0,6863	0,7035	0,7013	0,6961
100	0,9199	0,9084	0,9001	0,8834	0,8721
101	0,8679	0,8528	0,8375	0,8220	0,8115
102	0,6299	0,6257	0,6290	0,6297	0,6325
103	0,6012	0,6092	0,6207	0,6217	0,6162
104	0,8398	0,8594	0,8824	0,8995	0,9122
105	0,8998	0,9116	0,9062	0,8978	0,8872
106	0,7682	0,7630	0,7706	0,7584	0,7356
107	0,7583	0,7616	0,7617	0,7631	0,7649
108	0,7926	0,8105	0,8239	0,8363	0,8433
109	0,8451	0,8569	0,8615	0,8621	0,8586
110	0,8694	0,8972	0,8933	0,8831	0,8739
111	0,9108	0,9182	0,9109	0,9037	0,8965
112	0,7616	0,7489	0,7386	0,7278	0,7155
113	0,5899	0,5864	0,5751	0,5653	0,5614

Tabela 5: Resultados obtidos com o tamanho da janela de textura em 7, 9, 11, 13 e 15. Elipses = 4; Mahalanobis = 2,2.

Amostra	7	9	11	13	15
114	0,6349	0,6334	0,6323	0,6299	0,6267
115	0,8479	0,8254	0,8101	0,7934	0,7742
116	0,7200	0,7279	0,7305	0,7340	0,7462
117	0,9508	0,9300	0,9042	0,8884	0,8623
118	0,8949	0,8736	0,8508	0,8344	0,8208
119	0,7379	0,7586	0,7720	0,7850	0,7914
120	0,7220	0,7455	0,7596	0,7576	0,7549
121	0,9131	0,9146	0,9049	0,9010	0,8867
122	0,8988	0,9201	0,9151	0,8992	0,8738
123	0,8892	0,9108	0,9118	0,9033	0,8887
124	0,8351	0,8172	0,7987	0,7825	0,7674
125	0,8212	0,8031	0,7909	0,7793	0,7733
126	0,7347	0,7374	0,7389	0,7417	0,7340
127	0,7176	0,7185	0,7188	0,6901	0,6789
128	0,7774	0,8082	0,8323	0,8544	0,8773
129	0,8246	0,8411	0,8595	0,8787	0,8895
130	0,7430	0,6943	0,6633	0,6466	0,6335
131	0,8028	0,7672	0,7407	0,7221	0,7047
132	0,9269	0,9210	0,9026	0,8808	0,8570
133	0,8636	0,8499	0,8351	0,8224	0,8078
134	0,8725	0,8920	0,9095	0,9236	0,9232
135	0,8687	0,8483	0,8270	0,8037	0,7714

Fonte: Autoria própria.

Tabela 6: Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10. Janela de textura = 11; Mahalanobis = 2,2.

Amostras	2	4	6	8	10
1	0,5102	0,7327	0,7248	0,7338	0,7317
2	0,8317	0,7866	0,8032	0,7780	0,7773

Tabela 6: Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10.
Janela de textura = 11; Mahalanobis = 2,2.

Amostras	2	4	6	8	10
3	0,8195	0,8458	0,8600	0,8349	0,8214
4	0,5855	0,7496	0,7581	0,7945	0,7941
5	0,6644	0,8002	0,7952	0,8184	0,8161
6	0,7247	0,7320	0,7191	0,7176	0,7147
7	0,8240	0,8015	0,8034	0,7387	0,7248
8	0,7793	0,7529	0,7670	0,7416	0,7379
9	0,7432	0,7587	0,7587	0,7434	0,7436
10	0,4980	0,8192	0,8048	0,7216	0,6955
11	0,7416	0,8272	0,8346	0,8275	0,8290
12	0,6120	0,8005	0,8117	0,8079	0,8052
13	0,8328	0,6993	0,6727	0,6448	0,6361
14	0,8573	0,7964	0,8056	0,7725	0,7693
15	0,8619	0,8660	0,8803	0,8655	0,8658
16	0,7653	0,8718	0,8508	0,8820	0,8827
17	0,7998	0,8119	0,8083	0,8118	0,8146
18	0,8765	0,7835	0,7912	0,7552	0,7550
19	0,8086	0,8336	0,8290	0,8288	0,8275
20	0,8510	0,8726	0,8740	0,8746	0,8741
21	0,8544	0,9267	0,9137	0,9045	0,9023
22	0,7503	0,8798	0,8530	0,9003	0,9015
23	0,7332	0,8401	0,8308	0,8633	0,8665
24	0,8501	0,8049	0,8191	0,7899	0,7796
25	0,6110	0,7110	0,7057	0,7489	0,7614
26	0,6883	0,7714	0,7648	0,8117	0,8152
27	0,7787	0,8767	0,8704	0,8759	0,8748
28	0,6316	0,7163	0,7542	0,7698	0,7726
29	0,5788	0,5993	0,5774	0,5847	0,5844
30	0,8122	0,7864	0,7875	0,7520	0,7491
31	0,8555	0,8865	0,8820	0,8812	0,8811
32	0,8422	0,9006	0,8959	0,9006	0,9001

Tabela 6: Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10.
Janela de textura = 11; Mahalanobis = 2,2.

Amostras	2	4	6	8	10
33	0,7226	0,8809	0,8798	0,8781	0,8775
34	0,7050	0,7822	0,7907	0,7995	0,7993
35	0,6272	0,7130	0,6985	0,7567	0,7611
36	0,6032	0,7073	0,7027	0,7419	0,7516
37	0,9002	0,8162	0,8228	0,7852	0,7850
38	0,8058	0,8456	0,8328	0,8414	0,8407
39	0,7802	0,8686	0,8409	0,8737	0,8749
40	0,7037	0,7806	0,7314	0,7468	0,7406
41	0,7512	0,8081	0,7787	0,7085	0,7005
42	0,7801	0,7832	0,7973	0,7585	0,7560
43	0,7719	0,9530	0,9519	0,9793	0,9737
44	0,7141	0,8453	0,8282	0,8658	0,8665
45	0,7726	0,9451	0,9239	0,9407	0,9347
46	0,8057	0,9649	0,9576	0,9315	0,9289
47	0,9461	0,8367	0,8250	0,8220	0,8201
48	0,8426	0,8381	0,8523	0,8285	0,8304
49	0,7576	0,8468	0,8454	0,8559	0,8560
50	0,6058	0,8388	0,8377	0,8738	0,8764
51	0,5206	0,6735	0,7168	0,7414	0,7647
52	0,7870	0,8769	0,8875	0,8660	0,8615
53	0,5289	0,7002	0,7457	0,7645	0,7780
54	0,8348	0,8602	0,8769	0,8619	0,8609
55	0,8561	0,8137	0,8044	0,7409	0,7305
56	0,8396	0,8516	0,8439	0,8333	0,8275
57	0,7502	0,9612	0,9413	0,8813	0,8627
58	0,6618	0,8275	0,8171	0,8796	0,8893
59	0,7862	0,8639	0,8662	0,8516	0,8495
60	0,8805	0,8568	0,8631	0,8555	0,8562
61	0,7162	0,9002	0,8863	0,9441	0,9524
62	0,7103	0,7671	0,7744	0,7873	0,7860

Tabela 6: Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10.
Janela de textura = 11; Mahalanobis = 2,2.

Amostras	2	4	6	8	10
63	0,5633	0,6847	0,6822	0,7115	0,7137
64	0,8859	0,7611	0,7974	0,7376	0,7365
65	0,8542	0,8118	0,8336	0,7934	0,7929
66	0,5852	0,8312	0,8348	0,8610	0,8647
67	0,6254	0,8389	0,8519	0,8789	0,8862
68	0,8585	0,8916	0,8885	0,8810	0,8791
69	0,8701	0,9128	0,9198	0,8910	0,8873
70	0,6754	0,8264	0,8460	0,8562	0,8596
71	0,6786	0,8847	0,8982	0,8655	0,8273
72	0,5968	0,8071	0,8074	0,8655	0,8454
73	0,8949	0,7996	0,7896	0,7496	0,7449
74	0,8938	0,8013	0,8210	0,7832	0,7830
75	0,7990	0,9118	0,8906	0,9276	0,9268
76	0,6747	0,7802	0,7700	0,7926	0,7924
77	0,8766	0,8197	0,8172	0,7896	0,7853
78	0,8579	0,8100	0,8337	0,7904	0,7911
79	0,7379	0,9100	0,8991	0,9284	0,9286
80	0,6505	0,8438	0,8410	0,8650	0,8672
81	0,7690	0,8429	0,8478	0,8285	0,8221
82	0,6953	0,7875	0,7940	0,8178	0,8227
83	0,8878	0,7473	0,7135	0,6828	0,6805
84	0,8025	0,7745	0,7920	0,7565	0,7534
85	0,6803	0,7958	0,7862	0,8087	0,8098
86	0,7057	0,8956	0,8758	0,7972	0,7860
87	0,4158	0,4091	0,4039	0,3920	0,3924
88	0,8450	0,9102	0,9261	0,9026	0,9019
89	0,8470	0,9227	0,9034	0,9186	0,9199
90	0,8156	0,7343	0,7402	0,6535	0,6238
91	0,8134	0,7200	0,7010	0,6411	0,6315
92	0,6519	0,7937	0,7917	0,8541	0,8693

Tabela 6: Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10.
 Janela de textura = 11; Mahalanobis = 2,2.

Amostras	2	4	6	8	10
93	0,5555	0,7054	0,7021	0,7564	0,7601
94	0,8875	0,7706	0,7600	0,7137	0,7041
95	0,8253	0,7737	0,7602	0,7322	0,7262
96	0,9152	0,8280	0,8183	0,8013	0,8015
97	0,8672	0,8672	0,8780	0,8382	0,8343
98	0,5149	0,6150	0,6314	0,6782	0,6845
99	0,4908	0,7035	0,7423	0,7563	0,7653
100	0,8050	0,9001	0,8871	0,9186	0,9193
101	0,7534	0,8375	0,8227	0,8600	0,8627
102	0,5475	0,6290	0,6765	0,6772	0,6835
103	0,5278	0,6207	0,6502	0,6876	0,7002
104	0,8987	0,8824	0,8892	0,8486	0,8458
105	0,8308	0,9062	0,8954	0,9096	0,9089
106	0,5466	0,7706	0,7839	0,8152	0,8189
107	0,5444	0,7617	0,8041	0,7837	0,7881
108	0,8598	0,8239	0,8182	0,7891	0,7778
109	0,8457	0,8615	0,8505	0,8462	0,8418
110	0,7880	0,8933	0,8841	0,9073	0,9070
111	0,8101	0,9109	0,8977	0,9084	0,9095
112	0,6320	0,7386	0,7535	0,8064	0,8225
113	0,5469	0,5751	0,5762	0,6392	0,6545
114	0,5461	0,6323	0,6680	0,6911	0,7003
115	0,6592	0,8101	0,8011	0,8544	0,8651
116	0,6829	0,7305	0,7859	0,7793	0,7810
117	0,7532	0,9042	0,8816	0,9273	0,9276
118	0,7585	0,8508	0,8259	0,8543	0,8574
119	0,7367	0,7720	0,7776	0,7505	0,7446
120	0,7516	0,7596	0,7405	0,7348	0,7343
121	0,7501	0,9049	0,9142	0,9202	0,9036
122	0,7792	0,9151	0,9044	0,8994	0,8952

Tabela 6: Resultados obtidos com o número de elipses em 2, 4, 6, 8 e 10.
Janela de textura = 11; Mahalanobis = 2,2.

Amostras	2	4	6	8	10
123	0,7956	0,9118	0,9053	0,8999	0,8942
124	0,6780	0,7987	0,7877	0,8183	0,8194
125	0,7018	0,7909	0,7825	0,8063	0,8099
126	0,6671	0,7389	0,7463	0,7570	0,7606
127	0,6025	0,7188	0,7003	0,7491	0,7506
128	0,8172	0,8323	0,7864	0,7398	0,7279
129	0,8343	0,8595	0,8858	0,8335	0,8300
130	0,6190	0,6633	0,6454	0,7093	0,7222
131	0,6710	0,7407	0,7152	0,7660	0,7756
132	0,7097	0,9026	0,8823	0,9209	0,9169
133	0,7407	0,8351	0,8140	0,8406	0,8372
134	0,8851	0,9095	0,8895	0,8737	0,8672
135	0,6168	0,8270	0,8257	0,8738	0,8763

Fonte: Autoria própria.

Tabela 7: Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6.
Janela de textura = 11; Elipses = 4.

Amostra	1,8	2	2,2	2,4	2,6
1	0,5800	0,7064	0,7327	0,7225	0,6397
2	0,5441	0,7042	0,7866	0,8006	0,8018
3	0,5600	0,7635	0,8458	0,8457	0,8355
4	0,7032	0,7999	0,7496	0,6978	0,4741
5	0,5742	0,7706	0,8002	0,7407	0,5156
6	0,5675	0,6802	0,7320	0,7493	0,7396
7	0,6267	0,7540	0,8015	0,8291	0,8360
8	0,5717	0,7146	0,7529	0,7752	0,7776
9	0,5760	0,7118	0,7587	0,7646	0,7606
10	0,5765	0,8180	0,8192	0,7162	0,5179
11	0,6726	0,8119	0,8272	0,8149	0,7011

Tabela 7: Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.

Amostra	1,8	2	2,2	2,4	2,6
12	0,6456	0,7791	0,8005	0,7365	0,4677
13	0,4872	0,6352	0,6993	0,7331	0,7585
14	0,5060	0,7089	0,7964	0,8216	0,8338
15	0,6039	0,8273	0,8660	0,8691	0,8612
16	0,6569	0,8651	0,8718	0,8673	0,6928
17	0,5804	0,7597	0,8119	0,8174	0,8231
18	0,5027	0,6993	0,7835	0,8108	0,8354
19	0,5444	0,7656	0,8336	0,8347	0,7486
20	0,6224	0,8510	0,8726	0,8627	0,8579
21	0,5898	0,8315	0,9267	0,9280	0,9137
22	0,6854	0,8699	0,8798	0,8486	0,7657
23	0,7569	0,9316	0,8401	0,8114	0,7450
24	0,5909	0,7522	0,8049	0,8142	0,8343
25	0,6448	0,7233	0,7110	0,6786	0,5262
26	0,5972	0,7431	0,7714	0,7534	0,6677
27	0,5515	0,7898	0,8767	0,8497	0,7857
28	0,6000	0,7338	0,7163	0,6471	0,5924
29	0,4745	0,5669	0,5993	0,6025	0,5931
30	0,5222	0,6942	0,7864	0,8098	0,8301
31	0,5416	0,7968	0,8865	0,8853	0,8842
32	0,6151	0,8539	0,9006	0,8940	0,8857
33	0,6803	0,8511	0,8809	0,8550	0,8121
34	0,7037	0,7914	0,7822	0,7537	0,7308
35	0,6135	0,7100	0,7130	0,6656	0,6083
36	0,6642	0,7267	0,7073	0,6318	0,5744
37	0,5259	0,7281	0,8162	0,8528	0,8878
38	0,5241	0,7735	0,8456	0,8447	0,8420
39	0,6712	0,8346	0,8686	0,8547	0,8238
40	0,6381	0,7271	0,7806	0,7690	0,6711
41	0,5096	0,6994	0,8081	0,8934	0,6596

Tabela 7: Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.

Amostra	1,8	2	2,2	2,4	2,6
42	0,4996	0,6981	0,7832	0,7980	0,6820
43	0,6795	0,9316	0,9530	0,8935	0,7627
44	0,7191	0,8528	0,8453	0,8016	0,6109
45	0,6296	0,8971	0,9451	0,8857	0,6853
46	0,5572	0,8733	0,9649	0,9155	0,7061
47	0,5280	0,7651	0,8367	0,8676	0,9312
48	0,5263	0,7583	0,8381	0,8423	0,8555
49	0,6929	0,8394	0,8468	0,8239	0,6844
50	0,7557	0,8977	0,8388	0,7727	0,5319
51	0,7086	0,7389	0,6735	0,4739	0,4709
52	0,5877	0,8179	0,8769	0,8656	0,8274
53	0,6876	0,7558	0,7002	0,4866	0,4866
54	0,5817	0,8140	0,8602	0,8562	0,8513
55	0,5530	0,7205	0,8137	0,8472	0,8595
56	0,6092	0,7982	0,8516	0,8543	0,8487
57	0,6524	0,8952	0,9612	0,9116	0,7444
58	0,7397	0,8650	0,8275	0,7672	0,6169
59	0,5657	0,7940	0,8639	0,8649	0,8077
60	0,5398	0,7861	0,8568	0,8607	0,8646
61	0,6757	0,9421	0,9002	0,8536	0,7623
62	0,5813	0,7394	0,7671	0,7294	0,6227
63	0,6289	0,7299	0,6847	0,5578	0,5234
64	0,4869	0,6868	0,7611	0,7834	0,8000
65	0,5265	0,7312	0,8118	0,8235	0,8292
66	0,6743	0,8244	0,8312	0,7076	0,5221
67	0,7244	0,8549	0,8389	0,7555	0,5164
68	0,5842	0,8061	0,8916	0,9068	0,9074
69	0,5734	0,8409	0,9128	0,9188	0,9135
70	0,7547	0,8486	0,8264	0,7767	0,5335
71	0,6941	0,8966	0,8847	0,8200	0,6865

Tabela 7: Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.

Amostra	1,8	2	2,2	2,4	2,6
72	0,7759	0,8941	0,8071	0,7349	0,6109
73	0,5068	0,6976	0,7996	0,8290	0,8522
74	0,5637	0,7591	0,8013	0,8207	0,7698
75	0,6791	0,9143	0,9118	0,8679	0,8298
76	0,6896	0,8200	0,7802	0,7314	0,6653
77	0,5392	0,7204	0,8197	0,8454	0,8739
78	0,5617	0,7566	0,8100	0,8180	0,8257
79	0,7287	0,9325	0,9100	0,8742	0,7440
80	0,7873	0,9080	0,8438	0,7738	0,6009
81	0,6167	0,8211	0,8429	0,8116	0,6650
82	0,5986	0,7923	0,7875	0,7261	0,6239
83	0,5099	0,6789	0,7473	0,8038	0,8836
84	0,5553	0,7379	0,7745	0,7999	0,7573
85	0,6694	0,7805	0,7958	0,7625	0,5556
86	0,5638	0,7726	0,8956	0,8836	0,5922
87	0,3078	0,3721	0,4091	0,4044	0,4241
88	0,6145	0,8618	0,9102	0,9047	0,8846
89	0,6590	0,9033	0,9227	0,9102	0,8788
90	0,4723	0,6166	0,7343	0,7764	0,7870
91	0,4651	0,6172	0,7200	0,7670	0,8038
92	0,6861	0,8023	0,7937	0,7290	0,5982
93	0,7840	0,7727	0,7054	0,6362	0,4972
94	0,4859	0,6538	0,7706	0,8205	0,8523
95	0,5232	0,6800	0,7737	0,8098	0,8319
96	0,5285	0,7530	0,8280	0,8677	0,9101
97	0,5431	0,7737	0,8672	0,8813	0,8826
98	0,6638	0,6750	0,6150	0,5270	0,4690
99	0,6908	0,7377	0,7035	0,4448	0,4448
100	0,7252	0,9222	0,9001	0,8684	0,7200
101	0,7121	0,8431	0,8375	0,8112	0,7741

Tabela 7: Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.

Amostra	1,8	2	2,2	2,4	2,6
102	0,6380	0,6857	0,6290	0,5155	0,4887
103	0,6127	0,6905	0,6207	0,5196	0,4807
104	0,5406	0,7720	0,8824	0,9101	0,9221
105	0,6082	0,8718	0,9062	0,8964	0,8696
106	0,7764	0,8103	0,7706	0,5808	0,4919
107	0,7238	0,8040	0,7617	0,5855	0,4760
108	0,5203	0,6882	0,8239	0,8497	0,8649
109	0,5407	0,7395	0,8615	0,8632	0,8612
110	0,6157	0,8539	0,8933	0,8832	0,8151
111	0,6253	0,8661	0,9109	0,9003	0,8539
112	0,6558	0,7584	0,7386	0,6257	0,5533
113	0,5973	0,6304	0,5751	0,5370	0,5164
114	0,6105	0,6745	0,6323	0,5483	0,4842
115	0,7341	0,8612	0,8101	0,7333	0,5637
116	0,6430	0,7654	0,7305	0,6842	0,6504
117	0,7161	0,9302	0,9042	0,8644	0,7963
118	0,6707	0,8465	0,8508	0,8332	0,7557
119	0,6085	0,7424	0,7720	0,7745	0,7129
120	0,5833	0,7177	0,7596	0,7749	0,7383
121	0,7014	0,8956	0,9049	0,8589	0,8040
122	0,5613	0,8646	0,9151	0,8678	0,8242
123	0,5620	0,8498	0,9118	0,8830	0,8382
124	0,7099	0,8066	0,7987	0,7379	0,5957
125	0,6762	0,8164	0,7909	0,7416	0,5982
126	0,6171	0,7457	0,7389	0,6813	0,5854
127	0,6095	0,7265	0,7188	0,6272	0,5247
128	0,4957	0,6934	0,8323	0,9073	0,7185
129	0,5424	0,7849	0,8595	0,8752	0,7733
130	0,6137	0,7210	0,6633	0,6141	0,6138
131	0,6296	0,7692	0,7407	0,6911	0,6521

Tabela 7: Resultados obtidos com a distância de Mahalanobis em 1,8, 2, 2,2, 2,4 e 2,6. Janela de textura = 11; Elipses = 4.

Amostra	1,8	2	2,2	2,4	2,6
132	0,7147	0,9226	0,9026	0,8609	0,7186
133	0,6610	0,8486	0,8351	0,8085	0,7446
134	0,5502	0,8072	0,9095	0,9476	0,8993
135	0,7721	0,8935	0,8270	0,6168	0,5671

Fonte: Autoria própria.

Tabela 8: Resultados obtidos com os parâmetros fixos escolhidos. Janela de textura = 11; Elipses = 8; Mahalanobis = 2,2.

Amostra	Resultado	Amostra	Resultado
1	0,8054	69	0,8576
2	0,7246	70	0,8155
3	0,7734	71	0,9069
4	0,7709	72	0,8697
5	0,7955	73	0,7481
6	0,7097	74	0,7396
7	0,7372	75	0,9417
8	0,7286	76	0,8142
9	0,7358	77	0,7884
10	0,7782	78	0,7583
11	0,7760	79	0,8856
12	0,7433	80	0,8710
13	0,6270	81	0,8269
14	0,7422	82	0,8325
15	0,8235	83	0,6854
16	0,8540	84	0,7177
17	0,7668	85	0,8434
18	0,7647	86	0,7617
19	0,8232	87	0,5182
20	0,8492	88	0,8169

Tabela 8: Resultados obtidos com os parâmetros fixos escolhidos.
 Janela de textura = 11; Elipses = 8; Mahalanobis = 2,2.

Amostra	Resultado	Amostra	Resultado
21	0,8921	89	0,9045
22	0,9269	90	0,7097
23	0,8648	91	0,6919
24	0,7393	92	0,8970
25	0,8046	93	0,7628
26	0,8479	94	0,7117
27	0,9129	95	0,7090
28	0,7607	96	0,8269
29	0,6823	97	0,8412
30	0,7809	98	0,6818
31	0,8795	99	0,7226
32	0,8900	100	0,8939
33	0,8525	101	0,8518
34	0,7861	102	0,7158
35	0,8103	103	0,7014
36	0,7901	104	0,8588
37	0,8254	105	0,8826
38	0,8521	106	0,7680
39	0,8685	107	0,7583
40	0,7960	108	0,7992
41	0,7316	109	0,8355
42	0,7389	110	0,8772
43	0,9329	111	0,8795
44	0,8847	112	0,8548
45	0,9214	113	0,7587
46	0,8981	114	0,7651
47	0,8296	115	0,8751
48	0,8138	116	0,7050
49	0,8603	117	0,9303
50	0,8360	118	0,8971

Tabela 8: Resultados obtidos com os parâmetros fixos escolhidos.
 Janela de textura = 11; Elipses = 8; Mahalanobis = 2,2.

Amostra	Resultado	Amostra	Resultado
51	0,7831	119	0,7071
52	0,8419	120	0,7121
53	0,7693	121	0,8929
54	0,8277	122	0,8891
55	0,7503	123	0,8818
56	0,8091	124	0,8895
57	0,9071	125	0,8643
58	0,8783	126	0,8307
59	0,8633	127	0,8059
60	0,8379	128	0,7754
61	0,9150	129	0,8002
62	0,8183	130	0,8708
63	0,7701	131	0,8728
64	0,7169	132	0,8918
65	0,7558	133	0,8675
66	0,8483	134	0,8707
67	0,8633	135	0,8859
68	0,8805		

Fonte: Autoria própria.