

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

PABLO ROBERLAN MANKE BARCELLOS

**Detecção e Contagem de Veículos em  
Vídeos de Tráfego Urbano**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de  
Mestre em Ciência da Computação

Prof. Dr. Jacob Scharcanski  
Orientador

Porto Alegre, abril de 2014

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Barcellos, Pablo Roberlan Manke

Detecção e Contagem de Veículos em Vídeos de Tráfego Urbano / Pablo Roberlan Manke Barcellos. – Porto Alegre: PPGC da UFRGS, 2014.

115 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2014. Orientador: Jacob Scharcanski.

1. Contagem de veículos. 2. Detecção de veículos. 3. Rastreamento de veículos. 4. Agrupamento de partículas. 5. Processamento de vídeos. 6. Visão computacional. I. Scharcanski, Jacob. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Pró-Reitor de Coordenação Acadêmica: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do PPGC: Prof. Luigi Carro

Bibliotecário-chefe do Instituto de Informática: Alexander Borges Ribeiro

## **AGRADECIMENTOS**

Primeiramente, agradeço a toda minha família e especialmente aos meus pais por todo o apoio, incentivo e motivação, que foram fundamentais para que eu chegasse até aqui.

Ao meu orientador, Prof. Jacob Scharcanski, pelo apoio, dedicação, ensinamentos e todo o auxílio recebido durante o curso, essenciais para a realização deste trabalho.

Aos professores do curso de Ciência da Computação da UFPel, que contribuíram significativamente para minha formação. Em especial ao meu orientador da graduação, Prof. Lucas Ferrari de Oliveira, sou grato pela motivação e ensinamentos que recebi.

Agradeço aos colegas do PPGC e do PPGEE pela convivência e por todos os momentos passados durante esse período, e que sempre estiveram presentes ao meu lado, nos bons e maus momentos, me incentivando e apoiando.

Agradeço também ao PPGC, pela oportunidade de realizar este trabalho, aos professores da UFRGS pelo esforço, dedicação e conhecimento partilhado ao longo do curso, e a todos que, de uma forma ou de outra, sempre ajudaram e contribuíram para a minha formação.

# SUMÁRIO

<b>LISTA DE FIGURAS</b> . . . . .	6
<b>LISTA DE TABELAS</b> . . . . .	8
<b>RESUMO</b> . . . . .	9
<b>ABSTRACT</b> . . . . .	10
<b>1 INTRODUÇÃO</b> . . . . .	11
1.1 <b>Motivação</b> . . . . .	13
1.2 <b>Definição do Problema e Objetivos</b> . . . . .	13
1.3 <b>Apresentação do Método Proposto e Contribuições</b> . . . . .	14
<b>2 REVISÃO BIBLIOGRÁFICA SOBRE MONITORAMENTO DE TRÁFEGO DE VEÍCULOS</b> . . . . .	16
2.1 <b>Visão Geral</b> . . . . .	16
2.2 <b>Segmentação dos Objetos no Primeiro Plano do Vídeo</b> . . . . .	17
2.2.1 <b>Diferença entre Quadros</b> . . . . .	18
2.2.2 <b>Diferença do Quadro para o Fundo da Cena</b> . . . . .	18
2.2.3 <b>Subtração do Fundo Usando a Média e Mediana</b> . . . . .	19
2.3 <b>Rastreamento e Contagem Utilizando Câmeras de Vídeo</b> . . . . .	19
<b>3 FUNDAMENTOS DE PROCESSAMENTO DE IMAGENS E VISÃO COMPUTACIONAL</b> . . . . .	32
3.1 <b>Representação de Imagens Digitais</b> . . . . .	32
3.2 <b>Modelo de Cores</b> . . . . .	34
3.3 <b>Relações entre Pixels</b> . . . . .	35
3.3.1 <b>Vizinhança de um Pixel</b> . . . . .	35
3.3.2 <b>Conectividade e Adjacência de Pixels</b> . . . . .	36
3.4 <b>Métricas de Imagens</b> . . . . .	36
3.5 <b>Operações Lógicas e Aritméticas</b> . . . . .	37
3.6 <b>Operações Morfológicas</b> . . . . .	37
3.6.1 <b>Dilatação</b> . . . . .	38
3.6.2 <b>Erosão</b> . . . . .	38
3.6.3 <b>Abertura e Fechamento</b> . . . . .	38
3.7 <b>Envoltória Convexa</b> . . . . .	39
3.8 <b>Comparação de Histogramas</b> . . . . .	40
3.9 <b>Detector de Cantos</b> . . . . .	41
3.10 <b>Estimativa de Movimento</b> . . . . .	43

<b>3.11 Ruído em Imagens</b> . . . . .	43
3.11.1 Ruído Gaussiano . . . . .	44
3.11.2 Ruído Impulsivo . . . . .	44
<b>3.12 Algoritmos de Agrupamento de Dados</b> . . . . .	45
3.12.1 Algoritmo K-means . . . . .	45
3.12.2 Outros Métodos de Agrupamento . . . . .	46
<b>4 MÉTODO PROPOSTO PARA RASTREAMENTO E CONTAGEM DE VEÍCULOS EM VÍDEOS DE TRÁFEGO URBANO</b> . . . . .	51
4.1 Visão Geral do Método Proposto . . . . .	51
4.2 Inicialização . . . . .	56
4.3 Obtenção da Máscara com Objetos no Primeiro Plano do Vídeo . . . . .	58
4.4 Detectando os Agrupamentos de Partículas . . . . .	61
4.4.1 Agrupamento Inicial de Partículas . . . . .	62
4.5 Divisão de Clusters . . . . .	65
4.6 Fusão de Clusters . . . . .	66
4.7 Detecção de Veículos . . . . .	67
4.8 Rastreamento de Veículos . . . . .	70
4.9 Contagem de Veículos . . . . .	75
<b>5 RESULTADOS EXPERIMENTAIS</b> . . . . .	78
5.1 Resultados e Discussão . . . . .	79
<b>6 CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	108
6.1 Trabalhos Futuros . . . . .	109
6.2 Publicações . . . . .	109
<b>REFERÊNCIAS</b> . . . . .	111

## LISTA DE FIGURAS

2.1	Exemplo diferença entre quadros . . . . .	18
2.2	Exemplo diferença do quadro para o fundo . . . . .	19
2.3	Exemplo de subtração de fundo usando a mediana . . . . .	20
2.4	Exemplo do método proposto por Kanhere e Birchfield . . . . .	20
2.5	Exemplo do método proposto por Purnama et al . . . . .	21
2.6	Exemplo método proposto por Chen et al. . . . .	22
2.7	Exemplo método proposto por Kim . . . . .	23
2.8	Exemplo de rastreamento utilizando <i>eigenbasis</i> . . . . .	24
2.9	Exemplo método de detecção baseado em modelos deformáveis . . . . .	24
2.10	Exemplo de rastreamento do método proposto por Mei e Ling . . . . .	25
2.11	Exemplos do método proposto por Li et al. . . . .	26
2.12	Exemplo do método proposto por Feris et al. . . . .	26
2.13	Exemplo do método proposto por Sanchez et al. . . . .	28
2.14	Exemplo método proposto por Mossi et al. . . . .	29
2.15	Exemplo de rastreamento noturno . . . . .	29
2.16	Detecção em imagens diurnas . . . . .	30
2.17	Exemplo do método proposto por Chan et al. . . . .	31
3.1	Processo de amostragem e quantização . . . . .	33
3.2	Representação de uma imagem digital . . . . .	34
3.3	Cubo de cores RGB . . . . .	35
3.4	Exemplos de vizinhanças de pixels . . . . .	35
3.5	Exemplos de distancias entre pixels . . . . .	37
3.6	Exemplo de operações morfológicas . . . . .	39
3.7	Exemplo de região convexa e não convexa . . . . .	39
3.8	Exemplo envoltória convexa . . . . .	40
3.9	Exemplo de histograma de uma imagem . . . . .	41
3.10	Detecção de cantos em uma imagem . . . . .	42
3.11	Exemplo comparação de blocos dentro de uma área de busca. . . . .	43
3.12	Função densidade de probabilidade do ruído impulsivo. . . . .	44
3.13	Função densidade de probabilidade do ruído impulsivo. . . . .	45
3.14	Exemplo de dendrograma . . . . .	48
3.15	Exemplo de agrupamento de dados utilizando árvores . . . . .	49
3.16	Ilustração de um agrupamento fuzzy . . . . .	50
4.1	Exemplos de vídeos utilizados . . . . .	52
4.2	Exemplos de regiões de interesse . . . . .	53
4.3	Exemplos de laços virtuais . . . . .	53

4.4	Exemplo de detecção dos objetos no primeiro plano do vídeo . . . . .	53
4.5	Visão geral do método proposto . . . . .	56
4.6	Exemplo do processo de obtenção da região de interesse . . . . .	57
4.7	Exemplo de máscara binária obtida através do método GMM . . . . .	60
4.8	Exemplo de máscara binária obtida através do método MEI . . . . .	60
4.9	Máscara binária resultante da combinação dos métodos GMM e MEI	61
4.10	Exemplo de partículas detectadas no <i>foreground</i> . . . . .	62
4.11	Ilustração do processo de agrupamento inicial . . . . .	64
4.12	Exemplo de agrupamento inadequado . . . . .	64
4.13	Exemplo de divisão de clusters. . . . .	66
4.14	Exemplo de fusão de clusters. . . . .	67
4.15	Exemplo de análise da forma do cluster . . . . .	68
4.16	Exemplo de análise do <i>foreground</i> do cluster . . . . .	70
4.17	Exemplo veículo detectado. . . . .	70
4.18	Exemplos de histogramas de cor não similares . . . . .	73
4.19	Exemplos de histogramas de cor similares . . . . .	74
4.20	Exemplo de laços virtuais . . . . .	75
4.21	Ilustração do processo de incremento do contador de veículos . . . . .	76
4.22	Ilustração de veículo sobre mais de um laço . . . . .	77
5.1	Vídeo Cam06 . . . . .	80
5.2	Exemplo de rastreamento na Cam06 . . . . .	81
5.3	Exemplo do processo de contagem de veículos na Cam06 . . . . .	82
5.4	Vídeo Cam25 . . . . .	85
5.5	Exemplo de rastreamento na Cam25 . . . . .	86
5.6	Exemplo de erros de detecção na Cam25 . . . . .	87
5.7	Vídeo Cam41 . . . . .	89
5.8	Defeito na obtenção dos vídeos da Cam41. . . . .	89
5.9	Exemplo de rastreamento na Cam41 . . . . .	90
5.10	Exemplo de falha na contagem na Cam41 . . . . .	91
5.11	Vídeo Cam64 . . . . .	93
5.12	Exemplo de falha de detecção na Cam64 . . . . .	94
5.13	Exemplo oclusões na Cam64 . . . . .	95
5.14	Vídeo Cam68 . . . . .	97
5.15	Exemplo de rastreamento na Cam68 . . . . .	98
5.16	Exemplo de falha na contagem na Cam68 . . . . .	99
5.17	Vídeo Cam73 . . . . .	102
5.18	Rastreamento de veículo maior que a ROI . . . . .	103
5.19	Exemplo de detecção múltipla de veículos na Cam73 . . . . .	104

## LISTA DE TABELAS

3.1	Exemplo de histograma de uma imagem . . . . .	41
5.1	Contagem de veículos para a Cam06. . . . .	81
5.2	Veículos não detectados na Cam06. . . . .	83
5.3	Veículos detectados múltiplas vezes na Cam06. . . . .	83
5.4	Contagem de veículos para a Cam25. . . . .	85
5.5	Veículos não detectados na Cam25. . . . .	86
5.6	Veículos detectados múltiplas vezes na Cam25. . . . .	87
5.7	Contagem de veículos para a Cam41. . . . .	90
5.8	Veículos não detectados na Cam41. . . . .	91
5.9	Veículos detectados múltiplas vezes na Cam41. . . . .	92
5.10	Contagem de veículos para a Cam64. . . . .	95
5.11	Veículos não detectados na Cam64. . . . .	96
5.12	Veículos detectados múltiplas vezes na Cam64. . . . .	96
5.13	Contagem de veículos para a Cam68. . . . .	100
5.14	Veículos não detectados na Cam68. . . . .	100
5.15	Veículos detectados múltiplas vezes na Cam68. . . . .	101
5.16	Contagem de veículos para a Cam73. . . . .	102
5.17	Veículos não detectados na Cam73. . . . .	103
5.18	Veículos detectados múltiplas vezes na Cam73. . . . .	104
5.19	Resultado geral da contagem de veículos. . . . .	106
5.20	Resultado geral dos veículos não detectados. . . . .	106
5.21	Resultado geral dos veículos detectados múltiplas vezes. . . . .	107



## RESUMO

Este trabalho apresenta um novo método para o rastreamento e contagem de veículos em vídeos de tráfego urbano. Usando técnicas de processamento de imagens e de agrupamentos de partículas, o método proposto usa coerência de movimento e coerência espacial para agrupar partículas, de modo que cada grupo represente veículos nas sequências de vídeo. Uma máscara contendo os objetos do primeiro plano é criada usando os métodos *Gaussian Mixture Model* e *Motion Energy Images* para determinar os locais onde as partículas devem ser geradas, e as regiões convexas dos agrupamentos são então analisadas para verificar se correspondem a um veículo. Esta análise leva em consideração a forma convexa dos grupos de partículas (objetos) e a máscara de *foreground* para realizar a fusão ou divisão dos agrupamentos obtidos. Depois que um veículo é identificado, ele é rastreado utilizando similaridade de histogramas de cor em janelas centradas nas partículas dos agrupamentos. A contagem de veículos acontece em laços virtuais definidos pelo usuário, através da interseção dos veículos rastreados com os laços virtuais. Testes foram realizados utilizando seis diferentes vídeos de tráfego, em um total de 80000 quadros. Os resultados foram comparados com métodos semelhantes disponíveis na literatura, fornecendo resultados equivalentes ou superiores.

**Palavras-chave:** Contagem de veículos, detecção de veículos, rastreamento de veículos, agrupamento de partículas, processamento de vídeos, visão computacional.

## Detecting and Counting Vehicles in Urban Traffic Video

### ABSTRACT

This work presents a new method for tracking and counting vehicles in traffic videos. Using techniques of image processing and particle clustering, the proposed method uses motion coherence and spatial adjacency to group particles so that each group represents vehicles in the video sequences. A foreground mask is created using Gaussian Mixture Model and Motion Energy Images to determine the locations where the particles must be generated, and the convex shapes of detecting groups are then analyzed for the potential detection of vehicles. This analysis takes into consideration the convex shape of the particle groups (objects) and the foreground mask to merge or split the obtained groupings. After a vehicle is identified, it is tracked using the similarity of color histograms on windows centered at the particle locations. The vehicle count takes place on user-defined virtual loops, through the intersections of tracked vehicles with the virtual loops. Tests were conducted using six different traffic videos, on a total of 80.000 frames. The results were compared with similar methods available in the literature, providing results equivalent or superior.

**Keywords:** vehicle counting, vehicle detection, vehicle tracking, Particle Clustering, video processing, computer vision.

# 1 INTRODUÇÃO

Com o desenvolvimento da urbanização, aumentou também o número de veículos nas cidades, fazendo com que seja necessária a utilização de algum sistema de monitoramento do tráfego de veículos. Através do monitoramento do tráfego de veículos podemos extrair diversas informações sobre, por exemplo, congestionamentos do tráfego, incidentes e violações de trânsito, e as informações coletadas a partir desse monitoramento ajudam as autoridades a identificar e solucionar esses problemas, permitindo uma melhor utilização das vias urbanas. Por isso, extrair essas informações de maneira confiável e precisa torna-se um grande desafio nos sistemas de gestão de tráfego.

Um dado importante no monitoramento de tráfego é a contagem dos veículos que estão passando por determinada via, o que possibilita detectar quais vias estão com tráfego lento, congestionamentos, pontos com interrupção do tráfego, entre outras situações.

Para realizar a contagem de veículos é possível utilizar sistemas automatizados de contagem, como, por exemplo, detectores por laço indutivo, cabos piezelétricos, ultrassom, infravermelho, micro-ondas, laser e vídeo. Uma maneira comum de realizar essa contagem de maneira automática é instalando sensores indutivos fixados ao pavimento. Esses sensores realizam a contagem dos veículos que cruzam a seção da via, podendo ainda classificar os veículos através da análise dos perfis magnéticos dos veículos, como em Magalhães (2008).

A análise automática de tráfego através de vídeos é um campo de pesquisa emergente, com grande relevância para sistemas de monitoramento de tráfego urbano. Com o avanço da visão computacional, a câmera de vídeo torna-se um meio promissor e de baixo custo para o monitoramento do fluxo de tráfego (ROBERT, 2009). A análise do tráfego utilizando câmeras possui muitas vantagens em relação aos outros meios de monitoramento. As câmeras possuem fácil instalação, não causam danos às estradas e possuem custo relativamente baixo de instalação e manutenção. Além disso, elas fornecem uma ampla área de monitoramento, podendo monitorar diversas pistas ao mesmo tempo, permitindo uma melhor análise do fluxo do tráfego, medição de velocidade, contagem de veículos, classificação de veículos e avaliação de rodovias. O monitoramento automático do tráfego é uma fonte rica de informações, e os dados coletados através das câmeras podem ser utilizados para as mais diversas finalidades.

Visão computacional é o processo de utilizar um computador para extrair informações de alto nível de uma imagem digital. Uma câmera fornece o vídeo que é digitalizado e inserido no computador, onde algoritmos de visão computacional realizam tarefas como a detecção, o rastreamento e a classificação dos veículos. Sensores de vídeo oferecem um custo de instalação relativamente baixo, com pouca interrupção do tráfego para sua manutenção, e permitem extrair diversas informações da via, onde poderiam ser necessários a utilização de diversos tipos de sensores para extrair a mesma informação.

Visão é potencialmente mais eficaz do que qualquer outro sensor disponível atualmente. A instalação de câmeras de vídeo para monitorar redes de estradas é barato e menos prejudicial do que a instalação de outros sensores, e, geralmente, um grande número de câmeras já estão instaladas nas vias para fins de fiscalização. Uma única câmera é capaz de monitorar mais de uma faixa de tráfego ao longo de várias centenas de metros da estrada. Sistemas baseados em visão têm o potencial para extrair uma variedade muito mais rica de informações, tais como o caminho percorrido pelo veículo, a forma do veículo, as dimensões e cor. Um sistema de visão poderia, teoricamente, ter os mesmos poderes de observação que um observador humano, mas sem os efeitos prejudiciais de cansaço e tédio causados pela tarefa repetitiva (SETCHELL, 1997).

Apesar de existirem vantagens na utilização de câmeras de vídeo, também existem algumas desvantagens. As câmeras estão sujeitas a vibrações devido ao vento e passagem de veículos pesados. A posição da câmera também é um fator importante, pois por estar posicionada acima da via, dependendo da posição e ângulo que a câmera captura, podem ocorrer oclusões na presença de veículos grandes. Muitas vezes, são aproveitadas câmeras de vigilância que já estavam previamente instaladas, que geralmente não foram originalmente especificadas ou instaladas com a finalidade de utilizar algoritmos de visão computacional (MAGALHAES, 2008), aumentando a complexidade dos métodos de monitoramento por câmeras, pois será necessário contornar também essas dificuldades não diretamente relacionadas ao monitoramento do tráfego.

As aplicações de tráfego devem levar em consideração diversos aspectos, que permitem diferentes interpretações, como clima, iluminação, objetos parados na via, etc. Assim, é muito difícil um algoritmo de visão computacional levar em conta todas essas variações e definir parâmetros que funcionem em todas as diferentes situações. Por isso, muitas vezes é necessário diferentes algoritmos ou sistemas de monitoramento apresentam diferentes resultados para uma mesma cena (KASTRINAKI; ZERVAKIS; KALAITZAKIS, 2003).

O monitoramento de tráfego envolve a coleta de dados que descrevem as características dos veículos e seus movimentos através das redes rodoviárias. De acordo com Setchell (1997) esses dados podem ser utilizados para as mais diversas finalidades, como por exemplo:

**Cumprimento da Lei:** detectar veículos em alta velocidade, condução perigosa, uso ilegal de corredores de ônibus, detecção de veículos roubados ou procurados.

**Pedágios Automáticos:** pedágios manuais exigem que o veículo pare e o motorista pague uma tarifa adequada. Em um sistema automático o veículo já não precisa parar. À medida que passa o pedágio, os veículos seriam automaticamente classificados, a fim de calcular a tarifa correta. O número da placa poderia ser automaticamente decifrado e uma fatura mensal enviada ao proprietário.

**Detecção de Congestionamentos e Incidentes:** filas de trânsito, acidentes e veículos lentos são potencialmente perigosos para os veículos que se aproximam. Se tais incidentes forem detectados, painéis poderiam exibir mensagens informativas, a fim de alertar os motoristas que se aproximam.

**Aumento da Capacidade da Estrada:** Possuindo informações suficientes sobre o estado de uma rede de estradas é possível automaticamente encaminhar o tráfego ao longo das estradas menos congestionadas, a fim de otimizar a capacidade global da rede.

## 1.1 Motivação

No Brasil, os sistemas de monitoramento que estão instalados permanentemente nas vias utilizam, geralmente, cabos piezoelétricos e laços indutivos (ILD - Inductive Loop Detector), que capta mudanças na indutância de uma bobina enterrada no pavimento quando um objeto metálico (veículo) passa sobre ela (MAGALHAES, 2008). A instalação desses sensores na pista podem apresentar algumas desvantagens, como custo, restrição do monitoramento a apenas algumas áreas de maior interesse, entre outros. Por precisar ser embutido no pavimento, também causam a interrupção das vias para a sua instalação e eventuais reparos em caso de defeitos.

A aplicação de técnicas de processamento de imagens e visão computacional em sequências de vídeos oferece uma considerável melhoria sobre os métodos existentes de coleta de dados de tráfego e monitoramento de estradas (KASTRINAKI; ZERVAKIS; KALAITZAKIS, 2003). As câmeras de vídeo podem monitorar uma área bastante ampla e não requerem a interrupção do tráfego para instalação e manutenção.

Como mencionado anteriormente, o monitoramento automático do tráfego é uma fonte rica de informações, e os dados coletados através das câmeras podem ser empregados para os mais diversos fins. Uma diferença entre o monitoramento do tráfego e, por exemplo, reconhecimento de objetos genéricos, é que a tarefa de reconhecer um objeto geralmente se concentra em imagens de alta resolução, com poucas restrições sobre o ângulo de visão. Já no monitoramento de tráfego urbano é preciso lidar com um ambiente externo, mudanças climáticas e de iluminação, câmeras com baixa resolução e uma quantidade limitada de detalhes visuais, e o fato de os veículos andarem muito próximos, que dependendo do posicionamento da câmera, pode favorecer a ocorrência de oclusões (BUCH; VELASTIN; ORWELL, 2011).

Portanto, os principais desafios deste trabalho são lidar com câmeras que operam em ambientes externos e precisam lidar com situações como o tráfego lento e denso de veículos, e com veículos dos mais variados tipos e dimensões, trafegando com diferentes velocidades e muito próximos uns dos outros. Esses fatores fazem com que lidar com detecção e rastreamento de veículos por vídeos seja realmente uma tarefa desafiadora. Dessa forma, a motivação para a realização deste trabalho foi desenvolver um método automático e de baixo custo para realizar a detecção e o rastreamento para possibilitar a contagem de veículos em vídeos de tráfego urbano através de sequências de vídeo.

## 1.2 Definição do Problema e Objetivos

A primeira tarefa a ser resolvida é a definição de um método que analise as sequências de imagens e consiga diferenciar o que é veículo ou não. A separação dos veículos que estão em movimento no primeiro plano do vídeo (*foreground*) da parte estática da cena, ou seja, a região que está ao fundo da cena (*background*) de maneira confiável e robusta é geralmente o primeiro passo utilizado para o monitoramento de fluxo de tráfego. Com essa abordagem, os veículos são detectados levando em consideração o movimento do veículo através do vídeo, uma vez que, geralmente, os veículos os quais se deseja detectar apresentam algum tipo de movimento, enquanto a via e os objetos ao redor, como veículos estacionados, árvores e prédios, estão fixos, não apresentando uma mudança de posição ao longo do vídeo.

Uma vez que os veículos foram detectados, a próxima tarefa a ser definida é como fazer o rastreamento desses veículos ao longo do vídeo, ou seja, determinar se um veículo

detectado corresponde a algum veículo que já foi detectado previamente, nas sequências de imagens anteriores. Por fim, é preciso definir a melhor maneira de realizar a tarefa de contagem dos veículos, que nos permite ter a informação de quantos veículos estão passando por cada uma das faixas das vias em determinado momento.

Portanto, este trabalho tem como objetivo o desenvolvimento de um sistema automático para realizar a detecção, o rastreamento, e a contagem de veículos em tráfego urbano por câmeras de vídeos utilizando técnicas de processamento de imagens e visão computacional.

Para a segmentação dos objetos de interesse na cena (veículos) propõe-se o uso de um método para criar um modelo de fundo, pois o posicionamento da câmera é fixo em relação ao cenário de fundo, e, portanto, os objetos do fundo estão estáticos na cena, estando apenas os veículos em movimento sobre o fundo estático.

### 1.3 Apresentação do Método Proposto e Contribuições

O método proposto utiliza um modelo de Mistura de Gaussianas (*Gaussian Mixture Models* - GMM) (STAUFFER; GRIMSON, 1999), e a Energia de Movimento (*Motion Energy Images* - MEI) (DAVIS; BOBICK, 1997) para gerar uma máscara binária contendo os objetos móveis detectados no primeiro plano do vídeo. Após, são detectadas partículas presentes apenas nessas regiões onde foram detectados objetos em movimento. As partículas correspondem a cantos detectados na imagem, através do uso de um método de detecção de cantos. Essas partículas são então agrupadas usando o algoritmo de agrupamento de dados K-médias (*K-means clustering*), de acordo com a coerência de movimento dessas partículas ao longo do vídeo. As partículas são rastreadas através da similaridade de histogramas de cor, e os centroides de cada um desses agrupamentos são armazenados para que possa ocorrer a comparação com os agrupamentos formados nas sequências de imagens seguintes.

A contagem é realizada através da identificação de interseções entre as regiões espaciais associadas aos grupos de partículas rastreadas, que representam os veículos, e as regiões formadas por laços virtuais, que são marcações no vídeo definidas pelo usuário, especificando regiões onde os veículos deverão ser contados caso passem sobre elas.

Dentre as contribuições deste trabalho podemos citar:

- A introdução de uma aplicação para um sistema de contagem de veículos através de câmeras de vídeo estáticas (Capítulo 4);
- Uma melhor segmentação dos objetos que estão em movimento no primeiro plano do vídeo utilizando GMM e MEI. Dessa forma é criada uma máscara binária que representa os objetos em movimento no vídeo de maneira mais confiável (Seção 4.3);
- A realização da detecção de partículas apenas em regiões onde algum movimento foi detectado, o que aumenta a acurácia do método, pois as partículas pertencem às regiões mais prováveis de corresponder a um veículo, além de reduzir o custo computacional através da diminuição do número de partículas que precisam ser processadas (Seção 4.4);
- A proposta de um método para detectar os veículos através do agrupamento das partículas, de forma que cada um desses agrupamentos corresponda aos veículos do vídeo. Também são propostos métodos que utilizam a máscara com os objetos

móveis no primeiro plano para melhor identificar a qual agrupamento pertence cada uma das partículas e verificar se realmente os agrupamentos pertencem aos veículos, fazendo operações de fusão e divisão desses agrupamentos quando necessário (Seção 4.5, Seção 4.6 e Seção 4.7);

- A proposta de utilização de um método de rastreamento baseado em histogramas de cor para rastrear cada uma das partículas, permitindo determinar se um agrupamento que corresponde a um veículo já foi detectado previamente (Seção 4.8).

Durante o desenvolvimento deste trabalho, foram elaborados artigos científicos para eventos e periódicos da área, tendo como tema detecção e contagem de veículos. Nesses artigos são relatados os resultados obtidos e as contribuições do trabalho desenvolvido durante o mestrado.

A partir dos resultados obtidos com a pesquisa foram elaboradas duas publicações. Uma já aceita e publicada e outra em processo de submissão, listadas abaixo:

- **Tracking and Counting Vehicles in Traffic Video Sequences Using Particle Filtering:** Aceita e publicada na IEEE Instrumentation and Measurement Technology Conference (I2MTC 2013).
- **Counting and Tracking Vehicles in Urban Traffic Videos:** Em processo de submissão para a revista IEEE Transactions on Instrumentation and Measurement.

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta uma revisão bibliográfica sobre os principais métodos utilizados no monitoramento de tráfego de veículos, divididos em métodos de segmentação dos objetos em movimento na cena, métodos de detecção de veículos e métodos de rastreamento de veículos. Em seguida, o Capítulo 3 apresenta os fundamentos teóricos das técnicas e métodos da área de processamento de imagens e visão computacional. O Capítulo 4 descreve de forma detalhada cada uma das etapas do método proposto. No Capítulo 5 são expostos os resultados experimentais e discussões sobre o método proposto. Por fim, no Capítulo 6 são descritas as conclusões e os trabalhos futuros.

## 2 REVISÃO BIBLIOGRÁFICA SOBRE MONITORAMENTO DE TRÁFEGO DE VEÍCULOS

Esta seção apresenta um resumo das principais técnicas relacionadas com o rastreamento e contagem de veículos utilizando sequências de imagens de vídeo. Primeiramente será apresentada uma visão geral sobre o monitoramento de tráfego e métodos utilizados atualmente. Em seguida, uma breve descrição dos métodos mais utilizados e de novas abordagens propostas recentemente na área de monitoramento de tráfego utilizando câmeras de vídeo.

### 2.1 Visão Geral

O monitoramento de tráfego é uma importante ferramenta para o desenvolvimento de sistemas de transporte inteligentes, e a classificação de veículos tem surgido como um relevante campo de estudo na área devido a sua importância para sistemas de segurança, vigilância, prevenção de congestionamento de tráfego e de acidentes, sendo essencial para algumas tarefas, como, por exemplo, monitorar o volume e a taxa de fluxo de diferentes categorias de veículos (CHEN; ELLIS; VELASTIN, 2011).

A contagem dos veículos geralmente é realizada com o uso de sensores indutivos, que apesar de possuírem um bom desempenho, são muito intrusivos, necessitando cortes no pavimento para sua instalação, fazendo com que tenham um alto custo de instalação e manutenção. De acordo com Tian et al. (2011), comparada com as tecnologias atuais de monitoramento de fluxo de trânsito, a visão computacional usando câmeras de vídeo apresenta vantagens significantes, como produzir uma quantidade rica de informação sem afetar a integridade da via. Além disso, com a utilização de câmeras de vídeo para o monitoramento de tráfego de veículos é possível cobrir uma área mais ampla e também obter outros parâmetros, como, por exemplo, a classificação dos veículos (BUCH; VELASTIN; ORWELL, 2011).

Segundo Tian et al. (2011), a detecção dos objetos no primeiro plano do vídeo de maneira robusta e confiável é a primeira etapa para o monitoramento do tráfego de veículos, e com o resultado da detecção, o rastreamento e a classificação são geralmente realizados para obter as trajetórias e as categorias dos objetos em primeiro plano.

A detecção de objetos pode ser baseada na modelagem do fundo, que consiste em criar um modelo do fundo estático da cena, chamado *background*, subtrair o quadro atual do modelo de *background* e obter o quadro de diferença. Uma abordagem muito empregada é utilizar a média ou mediana dos  $n$  frames anteriores para o cálculo do modelo de *background* (BUCH; VELASTIN; ORWELL, 2011) (CUCCHIARA et al., 2003). Métodos baseados em modelos estatísticos e cálculos de probabilidade também são muito



utilizados para modelar o *background* em uma imagem, como, por exemplo, calculando a probabilidade de pixel pertencer ao fundo de acordo com o tempo que permanece com a mesma cor ou ainda utilizando Modelo de Misturas de Gaussianas (GMM) para modelar cada pixel do fundo, como introduzido por Stauffer e Grimson (1999).

Em Kastrinaki et al. (2003) são considerados a estimativa de geometria da pista e a detecção de veículos e obstáculos como as principais tarefas para os sistemas para monitoramento de tráfego por vídeo. O trabalho em Wang et al. (2008) considera a seleção das Regiões de Interesse (*Region of Interest- ROI*), através do uso de diversos métodos, como métodos de subtração de quadros e métodos de atualização de um modelo de fundo, como a etapa inicial no processamento de vídeos, em seguida, diversas técnicas podem ser usadas para detectar se existe um veículo na área de interesse, e sugerem que métodos de remoção de sombras devem ser explorados para melhorar a acurácia dos métodos de detecção de veículos. A remoção de sombras também é considerada por Zhong e Junping (2008) para melhorar a identificação de veículos e evitar a fusão indesejada de objetos próximos das sombras.

Outra etapa fundamental é a etapa de rastreamento. Rastrear um objeto no vídeo significa detectar as regiões correspondentes a este objeto em movimento ao longo de uma sequência de imagens. Os métodos de rastreamento podem ser classificados em métodos baseados em modelos, baseados em regiões, baseados em padrões deformáveis e baseados em características, segundo Liang et al. (2002).

Antes de iniciar o rastreamento é necessário identificar quais objetos devemos rastrear. A identificação desses objetos pode ser realizada de diversas maneiras (TIAN et al., 2011), entre elas a detecção baseada em modelos que utilizam conhecimento prévio do alvo desejado, como cor, tamanho, proporções ou formas pré definidas (SHEN, 2008) (LAI; HUANG; TSENG, 2010), a utilização de modelos deformáveis, ou seja, baseados em banco de imagens com modelos dos veículos que são deformadas para encaixarem-se nos objetos detectados (TAKEUCHI; MITA; MCALLESTER, 2010), e a utilização de características como cantos e bordas (TU; XU; ZHOU, 2008)

Os alvos identificados (veículos) podem ser rastreados através de diferentes abordagens, tais como *mean-shift* (BOUTTEFROY et al., 2008), filtros de Kalman (XIE et al., 2005), ou ainda filtragem de partículas (SCHARCANSKI et al., 2011).

Em Takeuchi et al. (2010) foram utilizados *Latente Support Vector Machine* (LSVM) e Histograma de Gradientes Orientados (*Histogram of Oriented Gradients - HOG*) para aprender características como a textura e a forma de um modelo deformável. Outro método baseado em contorno ativo integra informações de cor, forma e movimento para realizar o rastreamento (HU et al., 2013).

A contagem de veículos pode ser realizada diretamente com base nos veículos rastreados, onde cada novo veículo detectado incrementa um contador de veículos (SANCHEZ et al., 2011), ou os alvos rastreados (objetos em movimento) podem ser contados em locais com marcadores específicos na pista (laços virtuais) (TSENG; LIN; SMITH, 2002), ou ainda, a contagem pode ser efetuada diretamente nestes marcadores, sem o uso de um método de rastreamento de veículos, onde a simples passagem de um veículo sobre os laços virtuais incrementam os seus contadores (PURNAMA et al., 2009).

## 2.2 Segmentação dos Objetos no Primeiro Plano do Vídeo

Estimar e segmentar o primeiro plano fazem parte do primeiro estágio de vários sistemas de vigilância visuais, e consiste de realizar o processo de extração dos objetos em

primeiro plano no vídeo, também chamado de *foreground*. As regiões do primeiro plano são extraídas e utilizadas para formar uma máscara de *foreground*, uma máscara binária indicando as regiões onde estão os objetos do primeiro plano da cena, que é utilizada posteriormente para o processamento dos objetos detectados. Fazem parte do primeiro plano todos os objetos que não estão fixos em uma cena. A seguir serão conhecidos alguns dos métodos de segmentação de *foreground* mais utilizados.

### 2.2.1 Diferença entre Quadros

Possivelmente, o método mais simples para a segmentação do *foreground* é a diferenciação de quadros. A diferença pixel a pixel é calculada entre dois quadros consecutivos. A partir desta diferença, é verificado o valor de cada pixel, e se valor for maior que um limiar o pixel é considerado como pertencente ao primeiro plano, criando assim uma máscara com os objetos em primeiro plano no vídeo. Este algoritmo é muito rápido, no entanto, não consegue lidar com o ruído, mudanças bruscas de iluminação, ou movimentos periódicos no fundo, como árvores. Pode ser definido como:

$$|\text{quadro}_i - \text{quadro}_{i-1}| > \text{Limiar}. \quad (2.1)$$

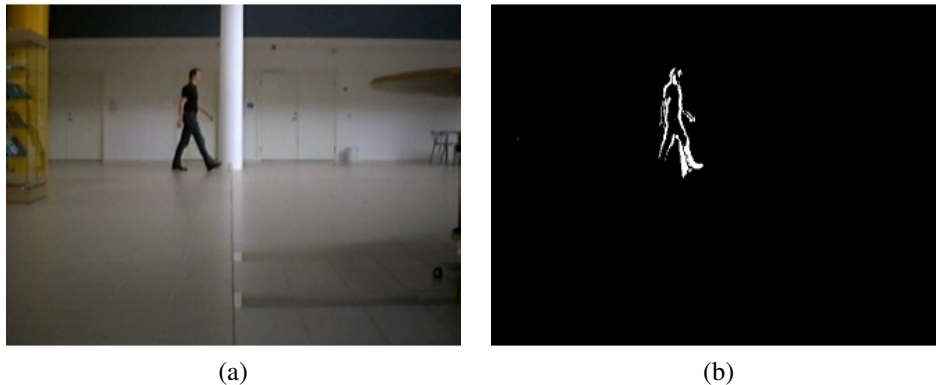


Figura 2.1: Exemplo diferença entre quadros. (a) um quadro do vídeo; (b) Máscara binária com o objeto do primeiro plano detectado (JOUBERT, 2009).

### 2.2.2 Diferença do Quadro para o Fundo da Cena

Este método baseia-se no uso de uma imagem de *background* para a separação dos objetos em movimento do fundo estático. A imagem de fundo pode ser especificada manualmente, através da adoção de uma imagem sem veículos. A detecção é então conseguida através da subtração da imagem de referência a partir da imagem atual.

É aplicado um limiar para determinar a presença ou ausência de informação de um objeto em movimento. O fundo pode mudar significativamente com sombras de edifícios e nuvens, ou simplesmente devido a mudanças nas condições de iluminação (CHEN; LIN; CHEN, 2007).

Inicialmente, uma imagem estática do fundo é obtida para ser o quadro de referência e então a técnica de diferença entre frames é utilizada para detectar as diferenças. Para melhorar o resultado final, operações morfológicas podem ser utilizadas (GONZALEZ; WOODS, 2002). A função de detecção pode ser escrita como a seguir:

$$|\text{quadro}_i - \text{fundo}| > \text{Limiar}. \quad (2.2)$$

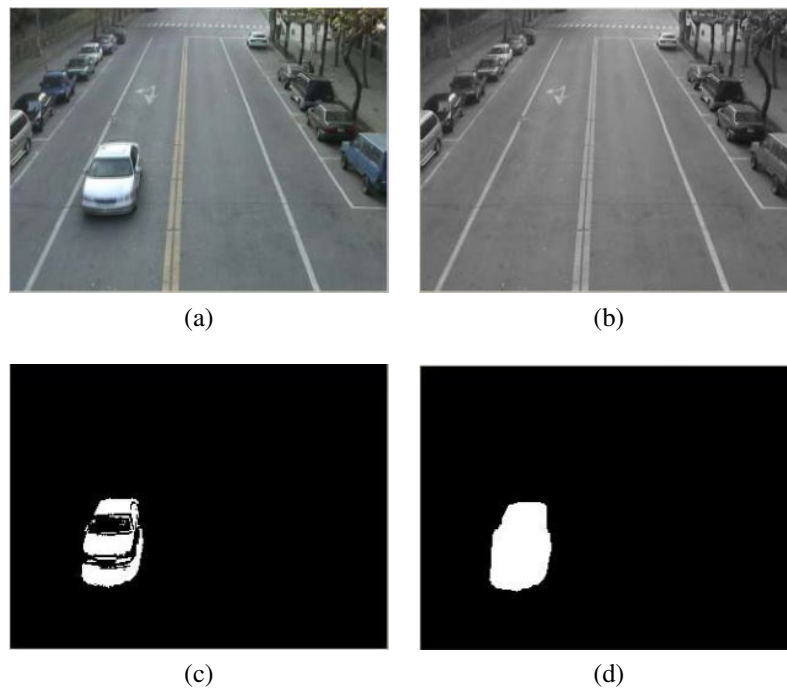


Figura 2.2: Exemplo diferença do quadro para o *background*. (a) imagem de Entrada. (b) imagem de fundo. (c) imagem diferença entre (a) e (b). (d) resultado após operações morfológicas (CHEN; LIN; CHEN, 2007).

### 2.2.3 Subtração do Fundo Usando a Média e Mediana

Nos métodos da média/mediana para o cálculo do fundo, são usadas a média ou mediana dos  $n$  quadros anteriores do vídeo como imagem de fundo (LO; VELASTIN, 2001) (CUCCHIARA et al., 2003). Os quadros do vídeo são então subtraídos dessa imagem de fundo para detectar os objetos em movimento no vídeo.

Neste método busca-se a melhor forma de estimar uma imagem de fundo da cena para fazer a subtração com o quadro atual do vídeo. Um limiar é, então, aplicado à imagem de diferença resultante, gerando a máscara de primeiro plano. Esses métodos diferem na maneira como a imagem de *background* é obtida, resultando em diferentes níveis de qualidade de imagem para diferentes níveis de complexidade computacional (BUCH; VELASTIN; ORWELL, 2011).

## 2.3 Rastreamento e Contagem Utilizando Câmeras de Vídeo

O método proposto por Kanhere e Birchfield (2008) utiliza cantos (os vértices ou cruzamentos entre duas bordas) detectados na imagem para realizar o rastreamento de veículos. O usuário faz a entrada manual de duas linhas ao longo das bordas da via e uma perpendicular à direção do tráfego para calibrar o sistema e o sistema obtém automaticamente uma projeção tridimensional da zona de detecção. Para eliminar regiões de sombra o método elimina as partículas muito próximas das bordas da pista conforme podemos observar na Figura 2.4a, onde os círculos brancos pertencem ao *background*, os quadros pertencem às bordas das sombras e os círculos maiores pertencem aos veículos que estão se movendo. O método de Shi e Tomasi (1994) é utilizado para fazer o rastreamento das partículas, que são divididas em duas categorias: estáveis e instáveis.



Figura 2.3: Exemplo de subtração de fundo usando a mediana. (a) um quadro do vídeo original; (b) objeto do primeiro plano detectado usando a mediana para o cálculo do fundo (JOUBERT, 2009).

As partículas consideradas estáveis são agrupadas no plano da via conforme as suas posições na direção do movimento para fornecer a segmentação dos veículos. As partículas consideradas instáveis são então atribuídas a esses agrupamentos utilizando uma projeção de alinhamento e coerência de movimento. A etapa final envolve eliminar os grupos que não parecem corresponder aos veículos. Os veículos são identificados quando os agrupamentos possuem um formato de caixa, considerando o espaço de dimensões tridimensional, resultando na Figura 2.4b. Os autores realizaram testes em 11 sequências de vídeo em tons de cinza com resolução de  $320 \times 240$ . Esse método apresentou entre 91% e 97% de acerto para as cenas testadas, onde o resultado é calculado como número de veículos detectados e rastreados corretamente em relação ao total de veículos na cena..

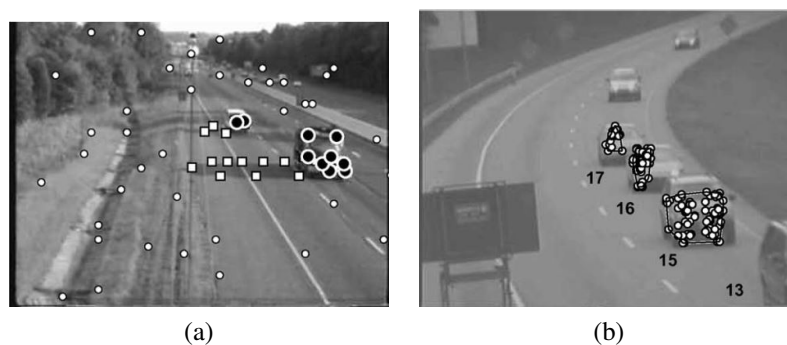


Figura 2.4: Exemplo do método proposto por Kanhere e Birchfield (2008). (a) Diferentes tipos de partículas; (b) agrupamentos representando os veículos.

Em Purnama et al. (2009) foi proposto um método bastante simples, que apenas faz a contagem de veículos, sem realizar nenhum tipo de rastreamento. O método pode ser resumido nos seguintes passos: uma etapa de inicialização, onde será obtido o fundo e a ROI, região onde os veículos serão contados; uma etapa de obtenção da máscara com os objetos móveis, através de um simples método de subtração entre o quadro processado e o fundo; aplicação de um limiar para segmentar os veículos e a utilização de operações morfológicas (erosão e dilatação) para garantir que as áreas correspondentes a cada um dos veículos estejam separadas; obtenção dos objetos conectados e classificação dos veículos, baseando-se no tamanho de suas áreas;

A Figura 2.5a apresenta a máscara com os objetos conectados e a Figura 2.5b apresenta o objeto detectado com esse método.

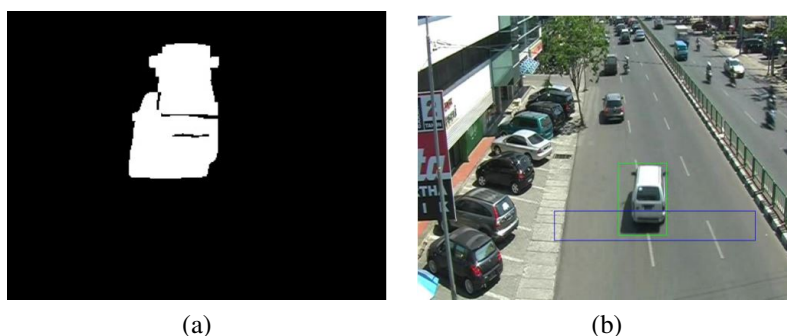


Figura 2.5: Exemplo do método proposto por Purnama et al. (2009). (a) máscara com objetos conectados; (b) veículo detectado sobre a região de interesse.

Este método apresentou uma taxa de acerto entre 80% e 95% em condições ideais de iluminação, já com a presença de sombras a taxa de acerto cai bastante, ficando entre 27% e 50%, pois o método é bastante simples, e não utiliza detecção de sombras e depende muito da qualidade do modelo de fundo que é utilizado. Os autores também testaram a contagem durante a noite, obtendo resultados variando entre 60% e 90%. As taxas foram calculadas de acordo com o número de contagem obtido pelo sistema em relação ao número real de veículos.

Outro método baseado nas características dos objetos conectados foi apresentado por Chen et al. (2007). O método proposto primeiro segmenta os objetos da sequência de imagem usando a detecção de mudança entre quadros sucessivos e uma atualização do modelo de fundo. A combinação da máscara de diferença entre quadros e a máscara de subtração do quadro pelo modelo de fundo é usada para adquirir a máscara inicial do objeto e resolver problemas não cobertos pelo modelo de fundo. Operações morfológicas também são utilizadas para melhorar a máscara binária. Em seguida, cada objeto conectado, representando um veículo, é delimitado por um retângulo e a altura, largura e área desse retângulo são consideradas como características desse veículo.

Os autores utilizaram essas características para classificar cada um dos objetos conectados em carros ou motocicletas. O rastreamento é feito baseado na proximidade dos objetos conectados em quadros adjacentes, usando a distância euclidiana para medir a distância entre os centroides de cada objeto no quadro atual e o quadro adjacente para verificar se eles possuem uma distância mínima e um tamanho similar para serem considerados como o mesmo objeto. A Figura 2.6 mostra um exemplo resultante desse método. Os autores utilizaram sequências de vídeos coloridos com resolução de  $320 \times 240$  pixels com taxa de 30 quadros por segundo. O método apresentou uma acurácia de 91.7% na classificação dos veículos para os vídeos testados pelos autores, calculada como o número de veículos classificados incorretamente sobre o número real de veículos.

Um método proposto por Kim (2008) também utiliza partículas (cantos) como feições para rastrear os veículos. Este método é usado como método de comparação com o método proposto neste trabalho, e os resultados podem ser vistos no capítulo de resultados experimentais (Capítulo 5). O método combina subtração de fundo, rastreamento de feições e algoritmos de agrupamento.

Os autores combinam filtro de kalman com uma mediana temporal e um procedimento para a correção de iluminação para melhorar o método de subtração de fundo,

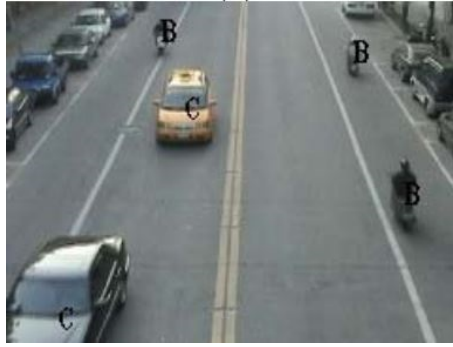


Figura 2.6: Exemplo método proposto por Chen et al.. Carros são identificados com a letra C e motocicletas com a letra B (CHEN; LIN; CHEN, 2007).

obtendo, então, uma máscara de *foreground* mais robusta. Nessa máscara são aplicados alguns procedimentos padrões, como operadores morfológicos e análise de componentes conectados, para preencher buracos, remover pequenas regiões e encontrar os *blobs* que representam os objetos. Após os *blobs* serem encontrados, são realizadas validações: os autores assumem que dentro de cada *blob* válido deve existir pelo menos um canto (feição) válido, ou seja, um canto que não está presente na imagem de background.

As feições são detectadas através dos autovalores das somas locais das derivadas, (que representam cantos na imagem) apenas nas regiões de *foreground*. Primeiramente, os cantos são agrupados em pequenos clusters, e, posteriormente, são agrupados em nível de objetos. Os cantos são rastreados utilizando correlação cruzada em janelas de  $9 \times 9$  pixels centradas em cada uma das feições no quadro atual e em uma região de busca no quadro adjacente. As feições são validadas com uma comparação com a imagem de fundo, utilizando também o método de correlação cruzada para encontrar um casamento de padrões. Se um canto tiver uma correspondência com a imagem de background, ele é considerado inválido e é removido.

A seguir, essas feições são agrupadas em pequenos clusters, cada um representado por um círculo, com a utilização de uma variação do algoritmo EM (*Expectation Maximization*). Para cada cluster, a sua posição esperada e tamanho no novo quadro são determinadas a partir das atuais feições membros. Então, para cada feição, a sua pertinência no quadro atual é redeterminada através da comparação da sua posição, histórico de movimento, trajetória passada, e o histórico indicando a qual *blob* a feição pertencia, como pode ser visto na Figura 2.7a. A posição e o tamanho do cluster são redeterminados através da atualização dos membros do cluster.

Por fim, os clusters são agrupados em objetos, utilizando o mesmo algoritmo EM utilizado para o agrupamento de feições, com algumas diferenças: para agrupar os clusters, a forma do um objeto deve ser uma elipse em vez de um círculo; a posição e formato final do objeto são determinados não pela posição dos membros do cluster, mas sim pela posição das feições dos membros do cluster; um critério 3D é usado para determinar o tamanho mínimo e máximo da elipse, como mostrado na Figura 2.7b.

Em Scharcanski et al. (2011) foi apresentado um método que utiliza filtro de partículas adaptativo para fazer o rastreamento de veículos. O método possui dois modos de operação, um para quando o veículo rastreado não está ocluso, que usa a função densidade de probabilidade (pdf) normal bivariada para gerar novos conjunto de partículas e outro quando o veículo está ocluso, que utiliza uma função densidade de probabilidade bivariada e Rayleigh conjuntamente, que ira criar novas partículas ao longo da direção

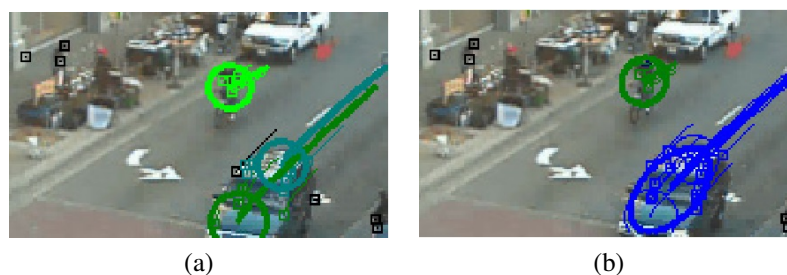


Figura 2.7: Exemplo método proposto por Kim. (a) agrupando os cantos detectados na imagem; (b) agrupamento em nível de objeto (KIM, 2008).

que o veículo está se movendo. O veículo rastreado pode ser detectado manualmente ou por algum outro método de detecção. Histogramas locais de cores são utilizados para rastrear os veículos, usando a distância de Hellinger para medir a similaridade entre os histogramas de um veículo no quadro atual e no quadro adjacente.

De acordo com Ross et al. (2007), a maioria dos algoritmos de rastreamentos falham na presença de variações significativas da aparência do objeto ou da iluminação ao redor desse objeto. Por isso, os autores propuseram um rastreador baseado em aparência que incrementalmente aprende uma representação de um subespaço de baixa dimensionalidade utilizando *eigenbasis* (conjunto de autovetores com o mesmo autovalor, juntamente com o vetor nulo) para o rastreamento de objetos, conseguindo se adaptar às mudanças de aparência dos objetos rastreados de maneira eficiente. A atualização do modelo é baseada em algoritmos incrementais para a análise de componentes principais. Um método incremental atualiza o modelo do subespaço continuamente para refletir as mudanças de aparência do objeto. Enquanto a maioria dos métodos assume que a posição da câmera é fixa, esse método é capaz de rastrear mesmo com a câmera em movimento.

Ainda assim falhas acontecem quando há uma combinação de mudanças rápidas de posição e alterações drásticas de iluminação. A Figura 2.8 exibe um exemplo de rastreamento em uma sequência de quadros, onde a primeira linha exibe o objeto rastreado, a segunda linha exibe o centro do subespaço, a imagem rastreada, o resíduo e a imagem reconstruída usando as *eigenbasis*, respectivamente. A terceira coluna exibe as imagens da *eigenbasis* do subespaço atual.

Outro método, proposto por Takeuchi et al. (2010) utiliza Máquinas de Vetores de Suporte Latentes (*Latente Support Vector Machine - LSVM*), conjunto de métodos de aprendizado supervisionado que analisam os dados e reconhecem padrões, e Histograma de Gradientes Orientados (*Histogram of Oriented Gradients - HOG*) para aprender características como a textura e a forma de um modelo de deformável. O método consiste em quatro processos; aprendizado, detecção, rastreamento e confirmação.

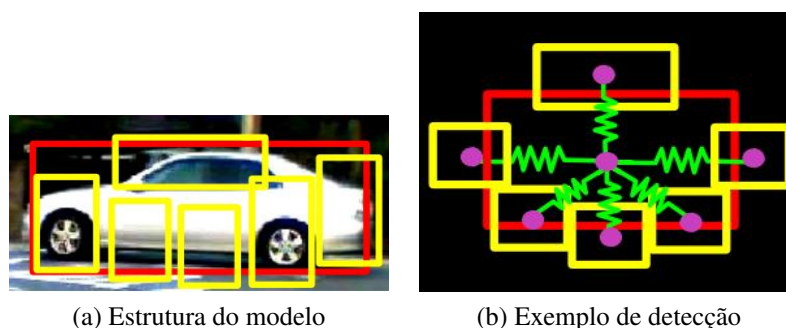
No processo de aprendizado, é utilizado máquinas de vetores de suporte latentes como um detector de veículos. A partir de um conjunto de treinamento, são aprendidas informações de textura e forma dos veículos. Os dados de treinamento são divididos pela taxa de proporção entre largura e altura, e filtros para a dianteira e traseira e também para as laterais dos veículos são aprendidos. As regiões dos veículos são escolhidas de acordo com uma pontuação calculada pela convolução de cada filtro do detector de veículos e as características HOG da imagem de entrada, considerando uma função de custo de deformação. Cada veículo é detectado aplicando-se um limiar de acordo com a pontuação calculada e é rastreado usando filtro de partículas com probabilidade integrada. Se um veículo é detectado pela primeira vez,  $N$  partículas são inicializadas de acordo com uma



Figura 2.8: Exemplo de rastreamento do método proposto por Ross et al. em uma sequência de quadros, onde a primeira linha exibe o objeto rastreado, a segunda linha exibe o centro do subespaço, a imagem rastreada, o resíduo e a imagem reconstruída usando as *eigenbasis*, respectivamente. A terceira linha exibe as imagens da *eigenbasis* do subespaço atual (ROSS et al., 2007).

distribuição gaussiana. Se o veículo já foi detectado no quadro anterior, a probabilidade de cada partícula é atualizada integrando as pontuações da detecção com base em modelos deformáveis e correlações de intensidade. Por fim, os resultados do rastreamento são confirmados utilizando informação de distância de um scanner laser.

A Figura 2.9 exibe um exemplo de detecção usando modelos deformáveis. Os autores criaram um *ground truth* para cada um dos quadros dos vídeos, indicando os pixels que pertencentes às regiões de veículos, e definiram a taxa de detecção como a razão entre o número de quadros detectados corretamente e o número total de quadros do vídeo, e consideram um veículo como rastreado corretamente quando todos os quadros em que o veículo aparece são detectados corretamente. O método faz a detecção de veículos em mais de 96% dos quadros e o rastreamento em mais de 83% dos veículos corretamente para os cenários testados pelos autores.



(a) Estrutura do modelo

(b) Exemplo de detecção

Figura 2.9: Estrutura do modelo de detector de veículos baseado em modelos deformáveis e um exemplo de detecção do método proposto por Takeuchi et al.. (a) O retângulo vermelho central é um filtro raiz, e os seis retângulos amarelos são filtros de partes. As seis molas em verde representam funções de custo de deformação. Cada filtro de partes corresponde às partes de um veículo, tal como uma roda ou para-brisas (TAKEUCHI; MITA; MCALLESTER, 2010).



O trabalho apresentado por Mei e Ling (2011) propõe um método que trata o rastreamento como um problema de aproximação esparsa em um *framework* de filtro de partículas. Para encontrar um alvo rastreado em um novo quadro do vídeo, cada candidato a alvo é representado esparsamente no espaço gerado por modelos de alvo e modelos triviais. A esparsidade é alcançada através da solução de um problema de mínimos quadrados regularizado em  $l_1$ . O candidato com o menor erro de projeção é adotado como alvo rastreado. Após, o rastreamento é prosseguido usando um *framework* de inferência Bayesiano, que utiliza a informação de deslocamento das partículas obtidas no quadro anterior. Uma transformação afim na imagem é aplicada para modelar o deslocamento do objeto em dois quadros consecutivos. Duas estratégias são usadas para melhorar o desempenho no rastreamento. Primeiro, os candidatos são atualizados dinamicamente para capturar mudanças de aparência. Depois, restrições são aplicadas para reduzir erros provocados por ruído e variações de iluminações e sombras.

A Figura 2.10 mostra um exemplo de rastreamento utilizando esse método em um vídeo com mudanças drásticas de iluminação. Como podemos observar na Figura 2.10, o método também funciona mesmo com a câmera em movimento.



Figura 2.10: Exemplo de rastreamento utilizando o método proposto por Mei e Ling (2011).

Em vídeos de tráfego urbano é comum ocorrer oclusões de veículos devido ao congestionamento. Para tentar resolver esse problema de oclusão, em Li et al. (2013) foi proposto uma abordagem baseada em grafos AND-OR (AOG). No grafo AND-OR, um objeto é decomposto hierarquicamente em diversas partes. Os autores utilizaram a visão frontal e traseira do veículo por ser a posição mais comum de veículos em vídeos de monitoramento de tráfego.

O método é composto de três etapas: construir o grafo AND-OR para representar os veículos no tráfego congestionado; treinar os parâmetros no AOG; e detectar os veículos usando inferência ascendente. Na construção do grafo AND-OR é definido um nodo raiz que é então hierarquicamente decomposto até nodos terminais, que devem ser modelados de maneira que os nodos terminais possam ser relacionados com a imagem do vídeo. Os autores utilizaram para isso um modelo deformável que pode esboçar as bordas de objeto e adaptar-se a forma do objeto. Em seguida, é realizada uma etapa para o aprendizado dos parâmetros e as probabilidades de cada nodo. Na etapa de inferência é utilizada uma abordagem multiescalar para fazer a correspondência do modelo com os veículos.

A Figura 2.11a exibe o grafo AND-OR e sua decomposição até os nodos terminais, e Figura 2.11b exemplos de veículos detectados utilizando esse método.

Em Feris et al. (2012), foi proposto um método de detecção de veículos que utiliza a técnica de co-treinamento, onde os dados são capturados em condições simples baseados no movimento e formas e então são usados para treinar detectores baseados em aparência que funcionam bem em condições complexas. Esses detectores baseados em aparência são treinados usando seleção de características em larga escala. Um banco de dados com imagens de veículos é gerado de forma semiautomática. Uma região de interesse (ROI) é

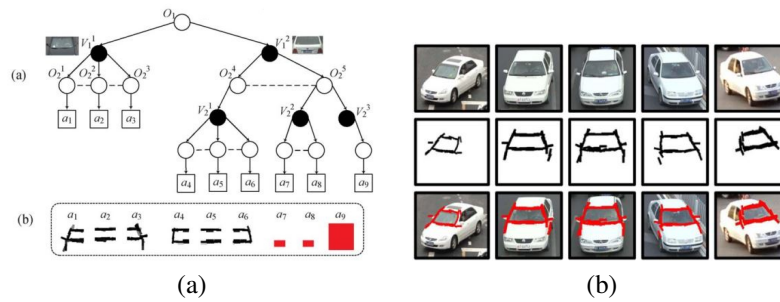


Figura 2.11: Exemplos do método proposto por Li et al.. (a) Grafo AND-OR para o objeto veículo e os modelos de nodos terminais; (b) resultados da detecção de veículos em diferentes posições (LI et al., 2013).

definida manualmente, e o método de subtração de fundo é utilizado para obter os objetos em movimento em cada quadro do vídeo. Um simples classificador baseado em regras é utilizado para analisar a forma, comprimento, largura e direção do movimento de cada um dos objetos detectados no primeiro plano e se estiverem dentro de um intervalo de valores predefinidos são considerados como veículos. Foram coletadas mais de um milhão de imagens de veículos para o banco de dados, e os veículos foram classificados em 12 categorias diferentes, de acordo com a direção de movimento de cada um deles. A Figura 2.12 mostra alguns exemplos de 12 categorias, de acordo com a direção de movimento (de 0 a 360 graus), mostrando a diversidade de posições de veículos no banco de dados. Também foram adicionadas imagens com oclusões sintéticas parciais de veículos ao conjunto de treinamento para tornar o método mais robusto. A base para o algoritmo de aprendizado é o *framework* proposto por Viola e Jones (2001), que consiste em classificadores *Adaboost* em cascata, onde os classificadores mais fracos são simples limiares sobre as características *Haar-like*.



Figura 2.12: Exemplo do método proposto por Feris et al. mostrando as 12 categorias de veículos, de acordo com a direção de movimento dos veículos (de 0 a 360 graus), mostrando a diversidade de posições no banco de dados (FERIS et al., 2012).

O método proposto em Sanchez et al. (2011) utiliza áreas predefinidas de entrada e saída para realizar a contagem de veículos. As regiões da pista também são definidas para cada uma das faixas, ajudando a localizar e rastrear os veículos presentes em cada uma das faixas e também a resolver situações de conflito na identificação. A Figura 2.13a apresenta a imagem utilizada como modelo de fundo, com as regiões de entrada (linha verde), saída (linha azul) e as quatro linhas delimitando as faixas da pista (em branco).

As regiões de entrada e saída são predefinidas (e fixas), e representam as regiões onde os veículos entram e saem da cena analisada, respectivamente. Uma imagem binária contendo os veículos em movimento é obtida através da subtração do quadro atual de uma imagem de fundo, que pode ser um modelo fixo do background, ou obtida adaptativamente, usando uma mediana aproximada, que verifica se o valor do pixel do quadro atual

é maior que o valor do pixel correspondente na imagem de fundo. Se valor do pixel do quadro atual for maior que o valor do pixel na imagem de fundo, o pixel da imagem de fundo é incrementado, caso contrário, o pixel da imagem de fundo é decrementado.

O método verifica se o número de pixels da máscara binária que fazem interseção com as regiões de entrada e saída é maior que um determinado limiar, e nesse caso a correspondente região de entrada ou saída é ativada e o componente conectado é detectado com um alvo ativo. As regiões de entrada não podem ser simultaneamente regiões de saída, e vice-versa. Se um alvo ativo que foi rastreado ao longo do tempo atinge a região de saída, representa que esse alvo deixou a cena. Quando não existe nenhum alvo ativo e um número de pixels for maior que o limiar predeterminado aparece numa região de entrada, um novo *blob* é produzido por um veículo detectado, e seu estado é incluído na lista de veículos detectados.

Cada veículo na região analisada é caracterizado por um número único de identificação (*id*), que é atribuído a ele quando passar pela primeira vez por uma região de entrada, um valor booleano ( $f_t$ ) indicando se a posição atual do veículo é conhecida, no quadro  $t$ , as coordenadas espaciais correspondentes ao centroide do alvo ( $x_t$  e  $y_t$ ), dois valores representando os lados do retângulo (*bounding Box*) contendo o veículo ( $Lx_t$  e  $Ly_t$ ) e as correspondentes coordenadas de velocidade ( $v_{x,t}$  e  $v_{y,t}$ ).

A cada quadro, a posição e o tamanho da *bounding Box* são ajustados apropriadamente. O rastreamento é baseado em verificar se existe uma sobreposição do retângulo contendo o veículo no quadro atual e no quadro anterior. Para cada veículo detectado no quadro anterior, é verificado se um *blob* no quadro atual pode ser atribuído a ele. Se existir uma sobreposição entre as *bounding Boxes* do veículo no quadro atual e no anterior, a posição do veículo é verificada e atualizada de acordo com o movimento do veículo. Caso contrário, a posição do veículo é considerada como temporariamente perdida.

As linhas delimitando as faixas da pista são utilizadas em alguns cenários para ajudar a localizar e rastrear o conjunto de veículos ao longo do tempo, e também resolver situações difíceis de analisar, como diversos veículos compartilhando o mesmo *blob*. Um raciocínio baseado em regras é aplicado sobre esses cenários:

- Regra 1: Alguns veículos estão compartilhando o mesmo componente conectado (*blob*).

Condição prévia: o número de veículos envolvido e as posições de seus respectivos centroides são conhecidas no quadro atual.

- Particionar o conjunto de veículos em subconjuntos de acordo com a proximidade das linhas delimitando as faixas da pista;
- Para cada subconjunto, considerar as posições dos veículos no quadro anterior e determinar a ordem de posicionamento das identificações dos veículos, atribuindo a mesma identificação do quadro anterior para o veículo no quadro atual, preservando a ordem de posicionamento.

- Regra 2: um veículo é considerado como perdido durante o rastreamento.

Condição prévia: as posições dos veículos são conhecidas no quadro anterior.

- Fazer a uma previsão da posição atual do veículo considerando a posição anterior e os vetores de velocidade.
- Realizar uma busca local por *blobs* não atribuídos a veículos ao redor da posição predita.

A Figura 2.13b apresenta um exemplo do resultado do rastreamento. O método atingiu uma taxa de acerto médio de 91.7% utilizando um modelo de fundo estático e 92.1% utilizando o modelo de fundo adaptativo. Para calcular os valores das porcentagens, os autores compararam o número de veículos detectados automaticamente com o número de veículos detectados por inspeção visual. Este é um dos métodos utilizados como comparação com o método proposto neste trabalho, e o resultado pode ser visto no Capítulo 5.



Figura 2.13: Exemplo do método proposto por Sanchez et al.. (a) imagem utilizada como modelo de fundo, com as regiões de entrada (linha verde), saída (linha azul) e as quatro linhas delimitando as faixas da pista (em branco); (b) exemplo de rastreamento (SANCHEZ et al., 2011).

Alguns trabalhos estudam soluções para o rastreamento em outros cenários, como em vídeos noturnos. Os métodos que trabalham com imagens noturnas baseiam-se principalmente na detecção dos faróis dos veículos. Um método que utiliza essa abordagem é o proposto por Mossi et al. (2011). A contagem é realizada quando os faróis identificados passam sobre laços virtuais na pista. Muitas vezes os faróis causam reflexos na pista, gerando áreas similares aos faróis dentro da região de interesse, criando falsas detecções. Para resolver isso, os autores propuseram um operador morfológico semelhante a uma operação *TopHat* (diferença entre a imagem original e a abertura da imagem original), mas realizando quatro aberturas na imagem utilizando elementos estruturantes elípticos, um para cada orientação (0, 45, 90 e 135 graus). Como os reflexos dos faróis na pista geralmente possuem uma forma alongada na direção do movimento do veículo, essa operação elimina esse reflexos, mantendo apenas os pontos que realmente correspondem aos faróis dos veículos. Também são adicionadas algumas restrições baseadas na coerência temporal, como considerar que o veículo deve passar dentro do laço virtual por um número mínimo de quadros.

A Figura 2.14a mostra um veículo passando com os faróis sobre uma área de interesse, e a Figura 2.14b mostra o resultado da detecção dos faróis. Os autores realizaram testes com sequências de vídeo totalizando 165.000 quadros e 653 veículos, apresentando um total de 61 erros (falsos positivos + falsos negativos) no cálculo da intensidade (ou densidade ? número de veículos por unidade de tempo).

Outro método, proposto por Chen et al. (2011), também identifica veículos através da detecção das luzes dianteiras e traseiras dos veículos. O método utiliza um processo automático de limiarização de histogramas multinível para extrair objetos brilhantes dos veículos em movimento nas sequências de imagens de cenas noturnas. Esses objetos brilhantes são processados através de um método de agrupamento espacial e um procedi-

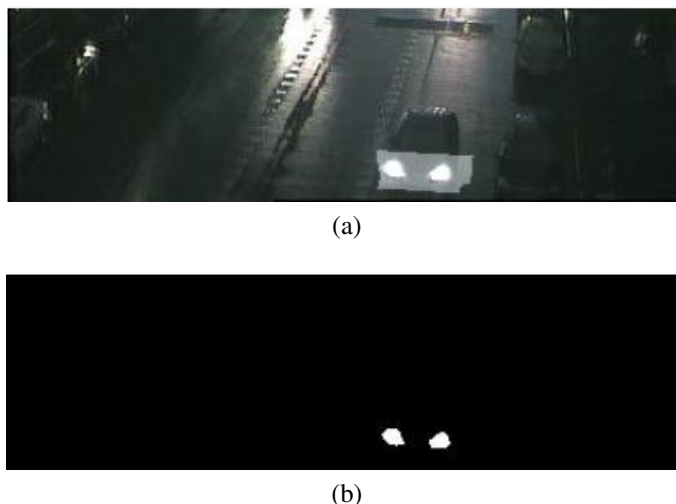


Figura 2.14: Exemplo método proposto por Mossi et al.. (a) veículo passando com os faróis sobre a área de interesse; (b) resultado da detecção dos faróis (MOSSI et al., 2011).

mento de rastreamento, que localiza e analisa as características espaciais e temporais das luzes dos veículos, como a posição em relação às linhas de fuga e as relações de distâncias horizontais e verticais entre os faróis, e a informação de movimento dos potenciais veículos são analisadas em quadros consecutivos para fazer a identificação e classificação de veículos e motocicletas. Sequências de vídeos de  $320 \times 240$  pixels com 30 quadros por segundo foram utilizadas para testes. Para a avaliação quantitativa da detecção, foi utilizado o coeficiente de Jaccard, definido como  $J = \frac{T_p}{T_p + F_p + F_n}$ , onde  $T_p$  é o número de veículos detectados corretamente,  $F_p$  é o número de veículos falsamente detectados, e  $F_n$  é o número de veículos não detectados. O coeficiente de Jaccard ( $J$ ) é determinado para cada quadro do vídeo e o coeficiente médio é obtido somando os coeficientes  $J$  de todos os quadros e dividindo pelo número total de quadro do vídeo. Assim, o método apresentou taxa de detecção média de 97% para os vídeos testados pelos autores. A Figura 2.15 mostra um exemplo de veículos sendo rastreados em uma sequência de imagens noturnas.



Figura 2.15: Exemplo de veículos sendo rastreados em uma sequência de imagens noturnas com o método proposto por Chen et al. (2011).

Um método baseado em hierarquia de características foi proposto por (ROBERT, 2009). A primeira camada da hierarquia extrai características da imagem, a segunda

camada faz a fusão dessas características para detectar características de veículos, como faróis e para-brisas, e a última camada faz a fusão das características dos veículos para detectar o veículo com mais confiança. Esse método consegue detectar veículos durante o dia e a noite.

Utilizando os parâmetros da câmera, que foi calibrada previamente, é realizada uma projeção de um modelo tridimensional representando os faróis e os para-brisas em todos os pixels do plano da câmera para extrair as características dos veículos. Um filtro de Kalman é utilizado para rastrear cada característica dos veículos na pista. As características do veículo são representadas por um ponto e um tamanho. Com imagens noturnas, o ponto rastreado é o ponto central entre os dois faróis, e o tamanho é a distancia separando os faróis, com imagens diurnas, o ponto é o centro do para-brisa, e o tamanho é a sua altura. A taxa média de detecção do método foi de 91% para os testes realizados pelos autores, e a avaliação consiste em comparar o resultado da contagem automática de veículos em vídeos contra a contagem manual (ground truth). A Figura 2.16 exibe um exemplo de detecção para uma sequência de imagens diurnas.



Figura 2.16: Exemplo de detecção para uma sequência de imagens diurnas utilizando método proposto por Robert (2009).

O método proposto por Chan et al. (2012) é baseado em um filtro de partículas, e realiza a detecção e rastreamento de veículos em câmeras móveis. O método utiliza a sombra sob o veículo, através da detecção de bordas horizontais com baixa intensidade para localizar os pixels pertencentes às sombras dos veículos, bordas verticais, obtidas com o detector de bordas Sobel, as luzes dos faróis, em cenas noturnas, e a simetria horizontal da parte traseira para detectar os veículos. Essas características são fundidas, gerando um descritor para as partículas, sendo capaz de identificar veículos em diferentes condições climáticas e de luminosidade. O algoritmo de agrupamento BSAS (THEODORIDIS; KOUTROUMBAS, 2008) é utilizado para agrupar as partículas, e um filtro de Kalman é utilizado para fazer o rastreamento. Foram realizados testes com vídeos com resolução de  $320 \times 240$  pixels, totalizando mais de 28.000 quadros. Os autores não informam como foi calculada a taxa de detecção, apenas que o método apresentou uma taxa média de acertos de 92%, sendo 5146 o número de veículos detectados corretamente e 403 o número de veículos não detectados. A Figura 2.17 exibe uma cena de exemplo desse método.



Figura 2.17: Exemplo de detecção e rastreamento utilizando o método proposto por Chan et al. (2012).

## 3 FUNDAMENTOS DE PROCESSAMENTO DE IMAGENS E VISÃO COMPUTACIONAL

Nesta seção serão apresentados os fundamentos teóricos na área de processamento de imagens e visão computacional. Serão apresentadas as definições teóricas e uma breve introdução das técnicas e métodos que foram utilizados na elaboração do método proposto, como, por exemplo, operações morfológicas, detectores de canto, medidas de distância, entre outras técnicas.

### 3.1 Representação de Imagens Digitais

Para ser processada em um computador, uma imagem precisa ser digitalizada, ou seja, representada usando uma estrutura discreta apropriada, como por exemplo, uma matriz. A imagem capturada por um sensor é expressa como uma função contínua  $f(x, y)$  de duas coordenadas no plano. A digitalização de uma imagem significa que a função  $f(x, y)$  é amostrada em uma matriz com  $M$  linhas e  $N$  colunas. Através da quantização da imagem, é atribuído um valor inteiro para cada amostra contínua, ou seja, o intervalo contínuo da função  $f(x, y)$  é dividido em  $K$  intervalos (SONKA; HLAVAC; BOYLE, 2007).

A Figura 3.1 mostra o conceito básico do processo de amostragem e quantização. A Figura 3.1a mostra uma imagem contínua,  $f(x, y)$ , que será convertida para o formato digital. A imagem é contínua em relação às coordenadas  $x$  e  $y$  e também em amplitude, e para convertê-la para a forma digital é preciso fazer a amostragem em ambas as coordenadas e amplitude. A digitalização dos valores das coordenadas é chamada amostragem e a digitalização da amplitude é chamada quantização (GONZALEZ; WOODS, 2002).

A função unidimensional na Figura 3.1b é um gráfico dos valores de amplitude (níveis de cinza) da imagem contínua ao longo do segmento de linha  $AB$  na Figura 3.1a. Para amostrar essa função, são utilizadas amostras igualmente espaçadas ao longo da linha  $AB$ , como pode ser visto na Figura 3.1c. o conjunto desses locais discretos constituem a função amostrada, contudo os valores das amostras ainda abrangem um intervalo contínuo de valores de níveis de cinza (verticalmente). Para converter a função para a forma digital, esses valores de níveis de cinza precisam ser convertidos (quantizados) em valores discretos, como podemos observar no lado direito da Figura 3.1c a escala de níveis de cinza dividida em oito níveis discretos, do preto ao branco. O resultado da imagem amostrada e quantizada é exibido na Figura 3.1d (GONZALEZ; WOODS, 2002).

Portanto, a transição entre valores contínuos de uma imagem e o seu equivalente digital é chamado quantização. O número de níveis de quantização deve ser alto o suficiente para permitir a percepção de detalhes finos de sombreamento (SONKA; HLAVAC; BOYLE, 2007).



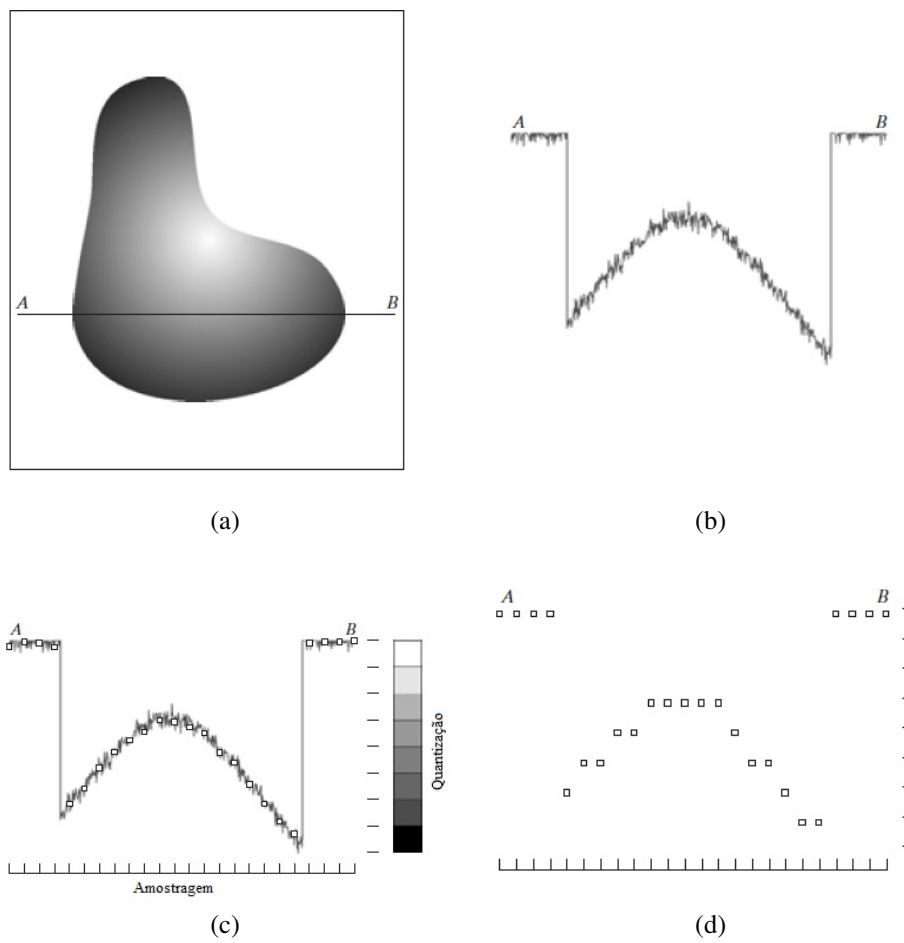


Figura 3.1: Processo de amostragem e quantização. (a) imagem contínua; (b) linha de varredura de A até B na imagem contínua; (c) amostragem e quantização; (d) linha de varredura digitalizada (GONZALEZ; WOODS, 2002).

Uma imagem contínua é digitalizada em pontos de amostragem. Estes pontos de amostragem estão ordenados no plano, em uma relação geométrica chamada grade, que geralmente possuem o formato quadrado ou hexagonal (SONKA; HLAVAC; BOYLE, 2007). Cada elemento dessa grade é chamado de pixel (abreviação de Picture Element).

Essa grade é representada como uma matriz de números reais de  $M$  linhas e  $N$  colunas, onde cada coordenada  $(x, y)$  representa um valor discreto da intensidade ou nível de cinza da imagem  $f(x, y)$ . A Figura 3.2 exibe a convenção de coordenadas mais utilizada.

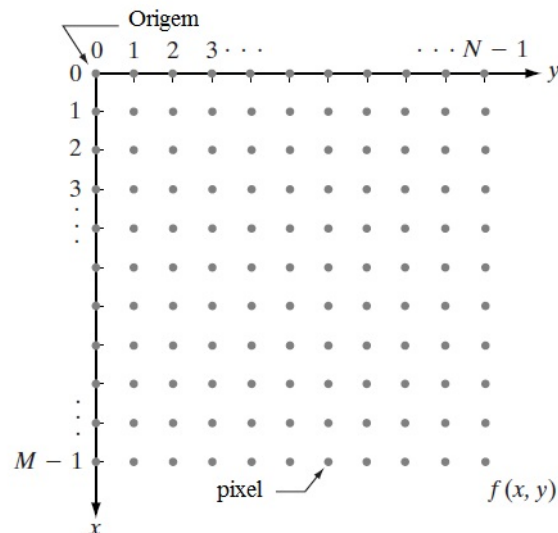


Figura 3.2: Representação de uma imagem digital  $M \times N$  (GONZALEZ; WOODS, 2002).

### 3.2 Modelo de Cores

O propósito dos modelos de cores, também chamado espaço de cores, é facilitar a especificação das cores em um formato padronizado e aceito por todos. O espaço de cores especifica um sistema de coordenadas e um subespaço dentro desse sistema onde cada cor é representada por um único ponto.

No espaço de cores RGB, cada cor aparece na sua componente espectral primária vermelha, verde e azul. O modelo RGB é baseado em um sistema de coordenadas cartesianas, que pode ser visto como um cubo, onde os valores primários (R-vermelho, G-verde e B-azul) aparecem em três vértices, e as cores secundárias, ciano, magenta e amarelo, estão em outros três vértices. O preto está na origem e o branco é o vértice mais distante da origem. A escala de cinza, que compreende os pontos com valores iguais para as componentes RGB, se estende ao longo da linha que liga esses dois vértices, conforme pode ser observado na Figura 3.3 (GONZALEZ; WOODS, 2002).

O modelo RGB é consiste na composição de três imagens, também chamadas de camadas ou canais, uma para cada cor primária, que são combinadas na tela para produzir uma imagem colorida. O número de bits usados para representar cada pixel no espaço RGB é chamado de profundidade de pixel (GONZALEZ; WOODS, 2002). O valor de uma cor em particular é expresso como um vetor de três elementos, correspondendo à intensidade das três cores primárias (SONKA; HLAVAC; BOYLE, 2007).

Considerando uma imagem RGB onde cada uma das imagens vermelha, verde e azul é uma imagem de 8 bits, cada pixel RGB, representado pela tripla (R, G, B), possui uma

profundidade de 24 bits.

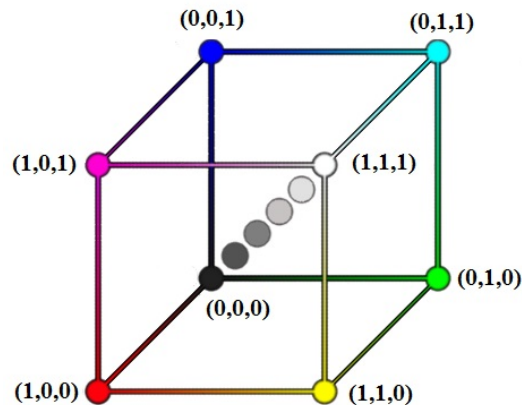


Figura 3.3: Cubo de cores RGB.

Um vídeo digital é formado por diversas imagens estáticas, chamadas quadros ou frames, que são exibidas de maneira sequencial gerando o efeito de movimento. A quantidade de quadros individuais processados e exibidos por unidade de tempo é denominado frequência de quadros ou frame rate. Nos vídeos utilizados neste trabalho, por exemplo, as frequências de quadros são de 25 ou 30 quadros por segundo.

### 3.3 Relações entre Pixels

#### 3.3.1 Vizinhança de um Pixel

Um pixel  $p$  com coordenadas  $(x, y)$  tem quatro vizinhos considerando as posições verticais e horizontais, que correspondem as coordenadas  $(x+1, y)$ ,  $(x-1, y)$ ,  $(x-1, y+1)$ ,  $(x, y-1)$ . Esses pixels formam a chamada vizinhança-4 de  $p$ , e é representada por  $N_4(p)$ . Os quatro vizinhos diagonais de  $p$  possuem as coordenadas  $(x+1, y+1)$ ,  $(x+1, y-1)$ ,  $(x-1, y+1)$ ,  $(x-1, y-1)$ , e são representados por  $N_D(p)$ . O conjunto formado pelos oito vizinhos do pixel  $p$  é chamado de vizinhança-8 de  $p$ , e é representado por  $N_8(p) = N_4(p) \cup N_D(p)$  (FILHO; NETO, 1999). A Figura 3.4 exemplifica os conceitos de vizinhança-4, vizinhança diagonal e vizinhança-8, respectivamente.

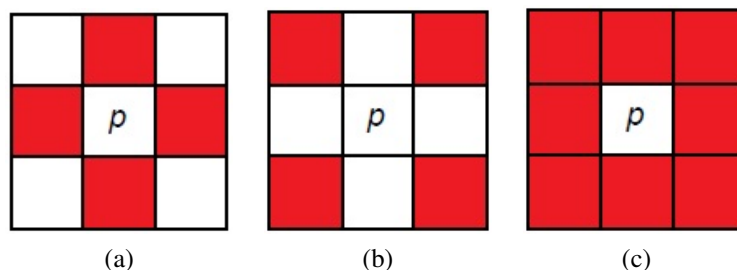


Figura 3.4: Exemplos de vizinhanças de um pixel  $p$ . (a) vizinhança-4; (b) vizinhança diagonal; (c) vizinhança-8 (GONZALEZ; WOODS, 2002).

### 3.3.2 Conectividade e Adjacência de Pixels

Conectividade entre pixels é um conceito fundamental que simplifica a definição de diversos outros conceitos de imagens digitais, como regiões e limites de objetos. Para determinar se dois pixels estão conectados, é preciso determinar se eles são vizinhos e se suas intensidades satisfazem um critério específico de similaridade, como, por exemplo, possuírem o mesmo nível de cinza (GONZALEZ; WOODS, 2002).

Dados um conjunto  $V$  de valores de tons de cinza utilizados para se definir a conectividade, podem ser definidos os seguintes critérios de conectividade (FILHO; NETO, 1999):

Conectividade-4: dois pixels  $p$  e  $q$  com valores de tom de cinza obtidos em  $V$ , são conectados-4 se  $q \in N_4(p)$ .

Conectividade-8: dois pixels  $p$  e  $q$  com valores de tom de cinza obtidos em  $V$ , são conectados-8 se  $q \in N_8(p)$ .

Conectividade- $m$  (conectividade mista): dois pixels  $p$  e  $q$  com valores de tom de cinza obtidos em  $V$ , são conectados- $m$  se:

1.  $q \in N_4(p)$  ou
2.  $q \in N_D(p)$  e  $N_4(q) \cap N_4(p) = \emptyset$

Um pixel  $p$  é adjacente a um pixel  $q$  se eles forem conectados. Existem tantos critérios de adjacência quantos são os critérios de conectividade. Dados dois subconjuntos de imagens,  $S_1$  e  $S_2$ , eles são considerados adjacentes se algum pixels em  $S_1$  é adjacente a algum pixel em  $S_2$  (FILHO; NETO, 1999).

Regiões de pixels adjacentes que tenham o mesmo valor são chamadas de componentes conectados ou componentes conexos (SZELISKI, 2010).

### 3.4 Métricas de Imagens

Algumas propriedades de imagens contínuas não têm uma analogia direta no domínio discreto, como por exemplo, as medidas de distância. Distância ou métrica é qualquer função  $D$  que satisfaça as três condições seguintes (SONKA; HLAVAC; BOYLE, 2007):

1. Identidade  
 $D(p, q) \geq 0$ , com  $(D(p, q) = 0$  se e somente se  $p = q)$
2. Simetria  
 $D(p, q) = D(q, p)$
3. Desigualdade triangular  
 $D(p, r) \leq D(p, q) + D(q, r)$

A distância entre dois pontos com coordenadas  $(i, j)$  e  $(h, k)$  podem ser definidas de diversas maneiras diferentes. A seguir serão discutidas algumas dessas diferentes métricas que podem ser utilizadas para mediar a distância entre dois pontos em uma imagem.

*Distância Euclidiana:* A distância euclidiana entre dois pontos é o comprimento do segmento de linha que os conecta, e pode ser dada por:

$$D_E((i, j), (h, k)) = \sqrt{(i - h)^2 + (j - k)^2}. \quad (3.1)$$

*Distância  $D_4$  (city block)*: Também conhecida como Manhattan, seu nome deriva da analogia com a distância entre dois locais em uma cidade com ruas divididas em quadras com blocos de casas. Apenas movimentos horizontais e verticais são permitidos. É dada pela equação:

$$D_4((i, j), (h, k)) = |i - h| + |j - k|. \quad (3.2)$$

*Distância  $D_8$  (chessboard)*: Análoga ao número mínimo de movimentos do rei no tabuleiro de xadrez, ou seja, além de movimentos horizontais e verticais, também são permitidos movimentos em diagonal, é dada por:

$$D_8((i, j), (h, k)) = \max\{|i - h| + |j - k|\}. \quad (3.3)$$

A Figura 3.5 exemplifica cada uma dessas distâncias.

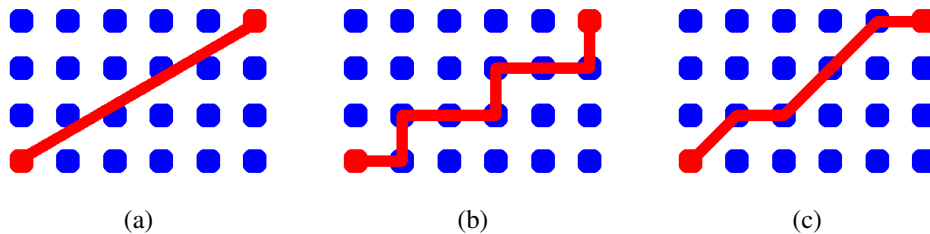


Figura 3.5: Exemplos de distancias entre pixels. (a) distância euclidiana entre dois pixels; (b) distância city block entre dois pixels; (c) distância chessboard entre dois pixels.

### 3.5 Operações Lógicas e Aritméticas

União, intersecção e complemento são operações de conjuntos que podem ser aplicadas no processamento de imagens. Quando lidamos com imagens binárias, essas operações de união, intersecção e complemento são referenciadas, geralmente, como as operações lógicas OR, AND e NOT. Considerando duas regiões (conjuntos)  $A$  e  $B$  contendo pixels no primeiro plano, ou seja, pixels com o valor lógico um, a operação lógica OR entre essas duas regiões  $A$  e  $B$  resulta em  $C = A \cup B$ , que contém elementos pertencentes a  $A$ ,  $B$  ou ambos. A operação AND entre as regiões  $A$  e  $B$  resulta em  $C = A \cap B$ , que contém elementos pertencentes a  $A$  e  $B$ . As operações  $NOT(A)$  e  $NOT(B)$  resultam em elementos que não estão presentes em  $A$  e  $B$ , respectivamente, ou seja, todos os elementos que estão em  $A$  são trocados para 0 (preto) e todos os elementos que não estão em  $A$  para 1 (branco). A operação diferença entre  $A$  e  $B$  resulta em  $C = A \cap B^C$ , que contém os elementos que estão em  $A$  mas não em  $B$ , onde  $B^C$  representa o complemento de  $B$ . Outra operação muito utilizada em processamento de imagens é a operação XOR (*exclusive OR*), que é o conjunto de pixels que pertencem a  $A$  ou  $B$ , mas não ambos (GONZALEZ; WOODS, 2002). Esses operadores são extensivamente utilizados em operações morfológicas, que serão discutidas na Seção 3.6.

### 3.6 Operações Morfológicas

Morfologia matemática usa ferramentas de álgebra não linear e opera com conjuntos de pontos, sua conectividade e forma. Operações morfológicas simplificam imagens e

quantificam e preservam as características principais da forma do objeto (SONKA; HLA-VAC; BOYLE, 2007).

Os operadores morfológicos primários são dilatação e erosão, e a partir desses dois, operações morfológicas mais complexas, como abertura e fechamento, são construídas.

Segundo Filho e Neto (1999), para compreender as operações morfológicas, inicialmente devemos conhecer as seguintes definições da teoria de conjuntos. Sejam  $A$  e  $B$  conjuntos em  $Z^2$ , cujos componentes são  $a = (a_1, a_2)$  e  $b = (b_1, b_2)$ , respectivamente. A translação de  $A$  por  $x = (x_1, x_2)$ , denotada por  $(A)_x$ , é definida como:

$$(A)_x = \{c | c = a + x, \text{ para } a \in A\}. \quad (3.4)$$

A reflexão de  $B$ , denotada  $\hat{B}$ , é definida como:

$$\hat{B} = \{x | x = -b, \text{ para } b \in B\}. \quad (3.5)$$

O complemento do conjunto  $A$  é:

$$A^C = \{x | x \notin A\}. \quad (3.6)$$

Por fim, a diferença entre dois conjuntos  $A$  e  $B$ , denotada  $A - B$ , é definida como:

$$A - B = \{x | x \in A, x \notin B\} = A \cap B^C. \quad (3.7)$$

### 3.6.1 Dilatação

Dados os conjuntos  $A$  e  $B$  no espaço  $Z^2$ , a dilatação de  $A$  por  $B$  é definida como:

$$A \oplus B = \{x | [( \hat{B} )_x \cap A] \subseteq A\}. \quad (3.8)$$

Assim, o processo de dilatação consiste em obter a reflexão de  $B$  sobre sua origem e depois deslocar esta reflexão de  $x$ , e resulta no conjunto de todos os  $x$  deslocamentos para os quais a interseção de  $\hat{B}_x$  e  $A$  inclui pelo menos um elemento diferente de zero. Dessa forma, a aplicação da operação de dilatação expande a imagem original. O conjunto  $B$  é geralmente chamado de elemento estruturante (FILHO; NETO, 1999).

### 3.6.2 Erosão

Dados os conjuntos  $A$  e  $B$  no espaço  $Z^2$ , a erosão de  $A$  por  $B$  é definida como:

$$A \ominus B = \{x | (B)_x \subseteq A\}. \quad (3.9)$$

Assim, a erosão de  $A$  por  $B$  resulta no conjunto de pontos  $x$  tais que  $B$ , transladado de  $X$ , está contido em  $A$ . Portanto, a aplicação da operação de erosão encolhe a imagem original (FILHO; NETO, 1999).

### 3.6.3 Abertura e Fechamento

A operação de abertura suaviza o contorno de uma imagem, quebra ligações finas entre objetos e elimina proeminências finas, enquanto a operação de fechamento funde pequenos objetos próximos, elimina pequenos orifícios no interior de objetos e preenche buracos no contorno. A abertura de um conjunto  $A$  por um elemento estruturante  $B$ , denotada por  $A \circ B$ , é realizada através da erosão de  $A$  por  $B$ , seguida da dilatação do resultado obtido por  $B$ , conforme a Equação 3.10. O fechamento de um conjunto  $A$

por um elemento estruturante  $B$ , denotado por  $A \bullet B$ , é obtido através da operação de dilatação de  $A$  por  $B$  seguida da erosão do resultado por  $B$ , conforme a Equação 3.11 (FILHO; NETO, 1999).

A seguir, na Figura 3.6, são exibidos alguns exemplos de operações morfológicas em imagens binárias.

$$A \circ B = (A \ominus B) \oplus B \quad (3.10)$$

$$A \bullet B = (A \oplus B) \ominus B \quad (3.11)$$

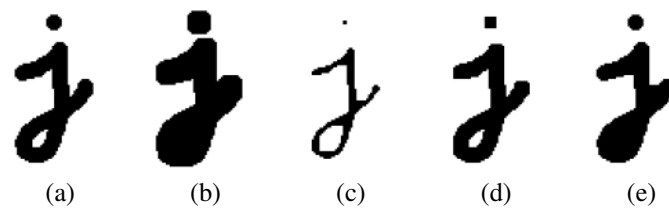


Figura 3.6: Exemplo de operações morfológicas em imagens binárias utilizando um quadrado de tamanho 5x5 como elemento estruturante. (a) imagem original; (b) dilatação; (c) erosão; (d) abertura; (e) fechamento (SZELISKI, 2010).

### 3.7 Envoltória Convexa

A envoltória convexa, ou *convex hull*, é bastante utilizada no método proposto para agrupar feições detectadas nas sequências de imagens dos vídeos. Para definirmos envoltória convexa, primeiro precisamos definir o que é uma região convexa. A região  $R$  é dita convexa se e somente se para quaisquer dois pontos  $p, q \in R$ , todo o segmento de linha  $\overline{pq}$  definido pelos seus pontos finais  $p, q$  está dentro da região  $R$  (SONKA; HLAVAC; BOYLE, 2007). A Figura 3.7 mostra dois exemplos, um de uma região considerada convexa e outra de uma região considerada não convexa.

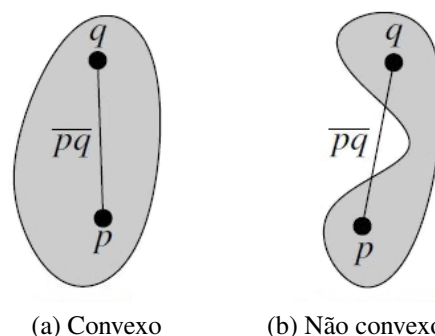


Figura 3.7: Exemplo de região convexa e não convexa. (a) região convexa; (b) região não convexa. Um subconjunto  $R$  do plano é chamado convexo se e somente se para qualquer par de pontos  $p, q \in R$  o segmento de linha  $\overline{pq}$  está completamente contido em  $R$  (BERG et al., 2008).

A envoltória convexa de uma região é a menor região convexa  $H$  que satisfaça a condição  $R \subseteq H$  (SONKA; HLAVAC; BOYLE, 2007), ou seja, é a menor região convexa

que contém  $R$ . A Figura 3.8 exibe um exemplo de envoltória convexa para um conjunto de pontos.

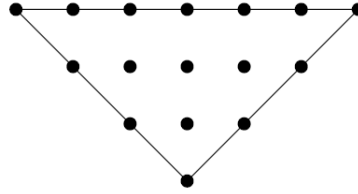


Figura 3.8: Exemplo envoltória convexa (BERG et al., 2008).

### 3.8 Comparação de Histogramas

Segundo Gonzalez e Woods (2002), dada uma imagem digital com níveis de intensidades no intervalo  $[0, L-1]$ , o histograma dessa imagem é uma função discreta  $h(r_k) = n_k$ , onde  $r_k$  é o  $k$ -ésimo valor de intensidade e  $n_k$  é o número de pixels na imagem com intensidade  $r_k$ . É comum utilizar o histograma normalizado da imagem, que é obtido dividindo cada um de seus componentes pelo número total de pixels na imagem, denotado por  $n = MN$ , onde  $M$  e  $N$  são as dimensões das linhas e colunas da imagem, respectivamente. Assim, o histograma normalizado é uma estimativa da probabilidade de ocorrência do nível de intensidade  $r_k$  na imagem, e é dado por:

$$p_r(r_k) = \frac{n_k}{n}, \quad (3.12)$$

onde:

$$0 \leq r_k \leq 1$$

$k = 0, 1, \dots, L - 1$ , onde  $L$  é o número de níveis de intensidade na imagem.

$p_r(r_k)$  = probabilidade do  $k$ -ésimo valor de intensidade.

$n_k$  = número de pixels cujo nível de intensidade corresponde a  $k$ .

A Tabela 3.1 apresenta os dados que correspondem a uma imagem de 128x128 pixels, com 8 níveis de cinza. O número de pixels que corresponde a cada um dos níveis de cinza é indicado pela segunda coluna, e as respectivas probabilidades aparecem na terceira coluna. A representação gráfica deste histograma pode ser observada na Figura 3.9 (FILHO; NETO, 1999).

Podemos comparar a semelhança entre dois histogramas utilizando os coeficientes de Bhattacharyya (COMANICIU; RAMESH; MEER, 2000), de acordo com a Equação 3.13.

$$s(\hat{h}_1, \hat{h}_2) = \sum_{i=1}^L \sqrt{\hat{h}_{1_i} \hat{h}_{2_i}} \quad (3.13)$$

onde  $s$  é a similaridade entre os histogramas normalizados  $\hat{h}_1$  e  $\hat{h}_2$ , e  $L$  é a quantidade de níveis de intensidade. Os valores de similaridade estarão no intervalo  $[0, 1]$ , onde o valor zero indica que os histogramas  $\hat{h}_1$  e  $\hat{h}_2$  são diferentes e o valor um indica que os histogramas  $\hat{h}_1$  e  $\hat{h}_2$  são iguais.

A interpretação geométrica da Equação 3.13 é o cosseno do ângulo entre os vetores unitários  $(\sqrt{\hat{h}_{1_1}}, \dots, \sqrt{\hat{h}_{1_L}})^T$  e  $(\sqrt{\hat{h}_{2_1}}, \dots, \sqrt{\hat{h}_{2_L}})^T$  (COMANICIU; RAMESH; MEER, 2000).



Tabela 3.1: Exemplo de histograma para imagem de 128x128 pixels e 8 níveis de cinza (FILHO; NETO, 1999).

Nível de cinza (rk)	nk	pr(rk)
0	1120	0.068
1/7	3214	0.196
2/7	4850	0.296
3/7	3425	0.209
4/7	1995	0.122
5/7	784	0.048
6/7	541	0.033
1	455	0.028
Total	16384	1

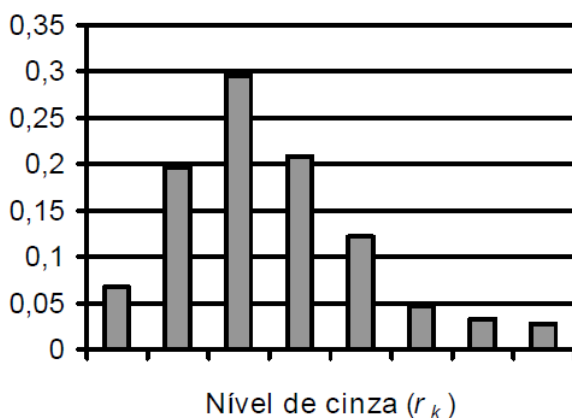


Figura 3.9: Exemplo histograma da imagem com 8 níveis de cinza (FILHO; NETO, 1999).

### 3.9 Detector de Cantos

A detecção de cantos é uma abordagem utilizada na visão computacional para extrair pontos de interesse ou feições de uma imagem. Um canto na imagem pode ser definido como um pixel no qual em sua vizinhança local existem duas bordas dominantes com direções diferentes. Os cantos podem ser obtidos usando detectores locais. Fornecendo uma imagem como entrada para o detector de cantos, a saída é uma imagem cujos valores são proporcionais à probabilidade daquele pixel ser um canto. Pontos de interesse são obtidos aplicando-se um limar sobre o resultado do detector de cantos (SONKA; HLAVAC; BOYLE, 2007). A Figura 3.10 exibe um exemplo de cantos detectados em uma imagem.

Neste trabalho foi utilizado o método dos mínimos autovalores, também conhecido como detector de cantos de Shi-Tomasi, proposto por Shi e Tomasi (1994).

Dada uma imagem em tons de cinza  $I$ , pegamos um pedaço (*patch*) dessa imagem sobre a área  $(u, v)$  e deslocamos ele por  $(x, y)$ . A soma ponderada das diferenças quadráticas (SSD) entre estas duas áreas, é denotada por:

$$s(x, y) = \sum_u \sum_v w(u, v) (I(u + x, v + y) - I(u, v))^2. \quad (3.14)$$

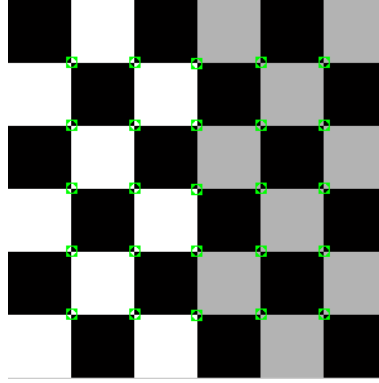


Figura 3.10: Exemplo de detecção de cantos em uma imagem. Os cantos, identificados em verde na imagem, foram detectados usando o método dos mínimos autovalores (SHI; TOMASI, 1994).

Se a imagem deslocada,  $I(u+x, v+y)$ , for aproximada por uma expansão de Taylor de primeira ordem,  $I(u+x, v+y) \approx I(u, v) + I_x(u, v)x + I_y(u, v)y$ , onde  $I_x$  e  $I_y$  são as derivadas parciais de  $I$ , isso produz a aproximação:

$$s(x, y) \approx \sum_u \sum_v w(u, v) (I_x(u, v)x + I_y(u, v)y)^2. \quad (3.15)$$

Que pode ser escrita na sua forma matricial:

$$s(x, y) \approx (x \ y) A \begin{pmatrix} x \\ y \end{pmatrix}, \quad (3.16)$$

onde  $A$  é a derivada de segunda ordem de  $S$  ao redor do ponto  $(u, v) = (0, 0)$ , e é dada por:

$$A = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}, \quad (3.17)$$

onde  $\langle \cdot \rangle$  denotam média (somatório sobre  $(u, v)$ ).

Um canto é caracterizado por uma grande variação de  $S$  em todas as direções do vetor  $(x \ y)$ , que é refletido nos autovalores  $\lambda_1$  e  $\lambda_2$  da matriz  $A$ . Essas variações podem ser encontradas usando a análise das componentes principais (PCA), e as seguintes situações podem ocorrer (SONKA; HLAVAC; BOYLE, 2007):

1. Ambos os autovalores são pequenos. Significando que é uma imagem plana ao redor do pixel examinado, sem a presença de bordas ou cantos.
2. Um autovalor é pequeno e o outro é grande. Uma borda é encontrada.
3. Ambos os autovalores possuem um valor grande. Um canto é encontrado.

Assim, o método proposto por Shi e Tomasi (1994) detecta um canto quando:

$$\min(\lambda_1, \lambda_2) > \lambda, \quad (3.18)$$

onde  $\lambda$  é um limiar predefinido.

### 3.10 Estimativa de Movimento

Neste trabalho é utilizado um algoritmo de *block matching* para estimar os vetores de movimentos de um pixel, ou seja, determinar o deslocamento em relação aos eixos  $x$  e  $y$  de um ponto de interesse entre o quadro atual do vídeo e o quadro seguinte. Para isso, foi utilizada a comparação de blocos com busca exaustiva.

O quadro atual é dividido em macro blocos, que são então comparados com o bloco correspondente e os seus vizinhos no quadro anterior para criar um vetor representando o movimento de um macro bloco de um local no quadro atual para outro local no quadro anterior. A área de busca pela melhor correspondência de um macro bloco é restringida a  $p$  pixels nas quatro direções ao redor da posição do macro bloco no quadro anterior, onde  $p$  é chamado de parâmetro de busca (BARJATYA, 2004). A Figura 3.11 exibe um bloco e a sua correspondente área de busca.

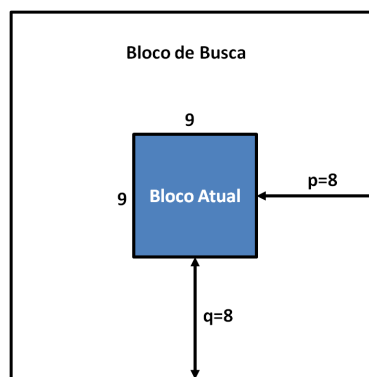


Figura 3.11: Exemplo comparação de blocos dentro de uma área de busca.

Para cada bloco, o vetor de deslocamento é avaliado através da busca de uma correspondência da informação presente em um bloco do quadro atual com a informação presente em um bloco dentro da área de busca no quadro anterior, através da minimização de uma função de custo. Encontrar o mínimo absoluto para o critério de correspondência só pode ser garantido executando uma busca exaustiva por todos os blocos candidatos dentro de um intervalo máximo de deslocamento, e essa técnica é chamada de *full-search block matching* (DUFAX; MOSCHENI, 1995). A função de custo utilizada foi a Média das Diferenças Absolutas (MDA), que é dada pela Equação 3.19, o bloco que apresentar o menor custo é escolhido como o bloco correspondente ao bloco do quadro atual.

$$MDA = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|, \quad (3.19)$$

onde  $N$  é o tamanho do macro bloco,  $C_{ij}$  e  $R_{ij}$  são os pixels que serão comparados do macro bloco atual e do macro bloco de referência, respectivamente.

### 3.11 Ruído em Imagens

Imagens são frequentemente degradadas por alguns erros aleatórios, e essa degradação é usualmente chamada de ruído. O ruído pode ocorrer durante a captura da imagem, transmissão, ou processamento, e pode ser dependente ou independente do conteúdo da imagem (SONKA; HLAVAC; BOYLE, 2007).

A seguir, serão descritos dois tipos comuns de ruído: o ruído gaussiano e o ruído impulsivo.

### 3.11.1 Ruído Gaussiano

O ruído é usualmente descrito por suas características probabilísticas. O ruído ideal, chamado de ruído branco, e tem um espectro de potência constante, significando que todas as frequências de ruído estão presentes e tem a mesma intensidade. Um caso especial de ruído branco é o ruído Gaussiano (SONKA; HLAVAC; BOYLE, 2007).

Uma variável aleatória,  $z$ , com distribuição Gaussiana (ou normal) tem sua função densidade de probabilidade dada pela curva Gaussiana, de acordo com a Equação 3.20, e pode ser visualizada na Figura 3.12.

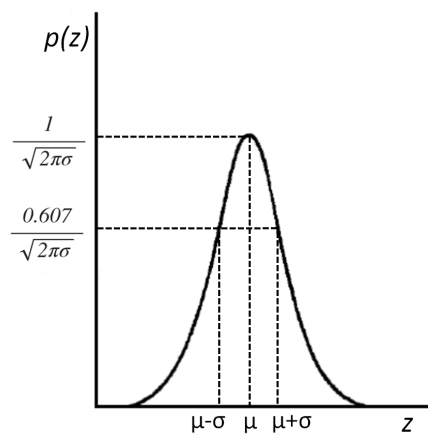


Figura 3.12: Função densidade de probabilidade do ruído impulsivo.

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}, \quad (3.20)$$

onde  $z$  representa a intensidade,  $\mu$  a média de  $z$ , e  $\sigma$  o seu desvio padrão. O desvio padrão ao quadrado,  $\sigma^2$ , é chamado de variância de  $z$ .

### 3.11.2 Ruído Impulsivo

Ruído impulsivo significa que uma imagem está corrompida com pixels individuais ruidosos que apresentam uma intensidade que difere significativamente dos pixels da sua vizinhança. O termo ruído sal e pimenta é utilizado para descrever ruídos impulsivos saturados, como por exemplo, uma imagem corrompida com pixels pretos e brancos (SONKA; HLAVAC; BOYLE, 2007).

O ruído impulsivo pode ser modelado como uma variável aleatória,  $z$ , com a função densidade de probabilidade dada pela Equação 3.21, e pode ser visualizada na Figura 3.13:

$$p(z) = \begin{cases} P_a, & \text{para } z = a \\ P_b, & \text{para } z = b \\ 0, & \text{caso contrário.} \end{cases} \quad (3.21)$$

Se  $b > a$ , pixels com intensidade  $b$  aparecerão como pontos brilhantes na imagem, enquanto com intensidade  $a$  aparecerão como pontos escuros. Se  $P_a$  ou  $P_b$  for zero, o

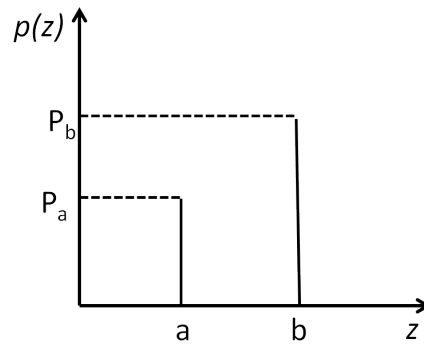


Figura 3.13: Função densidade de probabilidade do ruído impulsivo.

ruído impulsivo é chamado unipolar. Se nenhuma das probabilidades for zero e forem aproximadamente iguais, os valores do ruído impulsivo serão semelhantes a pequenos grãos de sal e pimenta espalhados aleatoriamente sobre a imagem, e o ruído impulsivo é chamado bipolar (GONZALEZ; WOODS, 2002).

### 3.12 Algoritmos de Agrupamento de Dados

O processo de criar agrupamentos de dados busca agrupar os pontos de um conjunto de dados de acordo com a similaridade entre eles, fazendo com que dados que sejam parecidos, de acordo com algum critério de similaridade, sejam considerados como pertencentes ao mesmo agrupamento, também chamado de cluster.

Um cluster é um conjunto de pontos no espaço de dados para os quais sua densidade local é alta comparada com a densidade dos pontos da região ao redor (JAIN, 1989).

Na Subseção 3.12.1, será apresentado o método k-means, que é utilizado neste trabalho, em seguida, na Subseção 3.12.2, serão apresentados outros métodos de agrupamentos de dados.

#### 3.12.1 Algoritmo K-means

O algoritmo k-means (k-médias) é um método utilizado para agrupar itens em  $k$  clusters (agrupamentos), onde cada item pertence ao cluster que tiver o centroide (média) mais próximo (JOHNSON; WICHERN, 2007).

Segundo Bishop (2006), dado um conjunto de  $N$  observações de uma variável euclidiana aleatória  $D$ -dimensional  $x = x_1, \dots, x_N$ , o objetivo é particionar o conjunto de dados em um número  $K$  de clusters. Introduzindo um conjunto de vetores  $D$ -dimensionais  $\mu_k$ , onde  $k = 1, \dots, K$ , com  $\mu_k$  representando o centro do cluster  $K$ . O objetivo é encontrar uma atribuição dos dados aos clusters e um conjunto de vetores  $\{\mu_k\}$  tal que a soma dos quadrados das distâncias de cada dado para o seu vetor  $\mu_k$  seja mínimo. Para cada dado  $x_n$ , podemos introduzir um conjunto de variáveis binárias indicadoras  $r_{nk} \in 0, 1$ , onde  $k = 1, \dots, K$  descrevendo a qual dos  $K$  clusters o dado  $x_n$  foi atribuído. Uma função objetiva, geralmente dada por:

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2, \quad (3.22)$$

que representa a soma dos quadrados das distâncias de cada dado para o seu respectivo vetor  $\mu_k$ . O objetivo é encontrar valores para  $\{r_{nk}\}$  e  $\mu_k$  de modo a minimizar  $J$ . Isso é

feito através de um procedimento iterativo, onde cada iteração envolve duas etapas sucessivas que correspondem a sucessivas otimizações com relação a  $r_{nk}$  e  $\mu_k$ . Essas etapas são listadas abaixo.

1. São escolhidos valores iniciais para  $\mu_k$
2. (a) Minimizar  $J$  em relação à  $r_{nk}$ , mantendo  $\mu_k$  fixo  
(b) Minimizar  $J$  em relação à  $\mu_k$ , mantendo  $r_{nk}$  fixo
3. Repetir a etapa 2 até a convergência.

Considerando a determinação de  $r_{nk}$ , simplesmente atribuímos o  $n$ -ésimo dado para o cluster com o centro mais próximo, que pode ser descrito como:

$$r_{nk} = \begin{cases} 1 & \text{se } k = \operatorname{argmin}_j |x_n - \mu_j|^2 \\ 0 & \text{caso contrário.} \end{cases} \quad (3.23)$$

A otimização de  $\mu_k$  pode ser dada por:

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}. \quad (3.24)$$

O denominador nessa expressão é igual ao número de dados atribuídos ao cluster  $k$ , e o resultado tem um simples interpretação: estabelecer  $\mu_k$  igual a média de todos os dados  $x_n$  atribuídos ao cluster  $k$ . Por essa razão, o procedimento é conhecido como algoritmo k-means. As duas fases de reatribuir dados para os clusters e recomputar as médias dos clusters são repetidos até não ocorrer nenhuma mudança nas atribuições ou até atingir um número máximo de iterações (BISHOP, 2006).

Outras métricas podem ser utilizadas como função objetiva  $J$  a ser minimizada. Neste trabalho foram utilizadas, por exemplo, a distância *city block*, que representa a soma das diferenças absolutas, dada pela Equação 3.25 e a distância do cosseno, dada como um menos o cosseno do ângulo formado entre os pontos (tratado como vetores), de acordo com a Equação 3.26, onde cada centroide é a média dos pontos naquele cluster, após serem normalizados para uma unidade de comprimento euclidiano.

$$D_{CB} = \sum_{n=1}^N |x_n - \mu_n| \quad (3.25)$$

$$D_{Cos} = \left( 1 - \frac{\vec{x} \cdot \vec{\mu}}{\|\vec{x}\| \cdot \|\vec{\mu}\|} \right) \quad (3.26)$$

### 3.12.2 Outros Métodos de Agrupamento

#### Método K-Medóides

Esse algoritmo é uma variação do algoritmo K-Means e ao contrário deste, o K-medóide escolhe objetos existentes como centroides. Ao final do agrupamento obtemos um objeto como o elemento central, normalmente classificado como o protótipo do agrupamento, o medóide.

Uma vantagem em relação ao K-means é sobre os ruídos ou outliers contidos no agrupamento, pois as estratégias na escolha do centroide e do medóide são diferentes. No

K-means, o centroide é dado pela média de todos os objetos do agrupamento. Desta maneira, se o agrupamento possuir um objeto muito distante dos outros, o centroide será influenciado erradamente. Já no K-medóide, o algoritmo utiliza a média do erro quadrado para validar a escolha de um medóide. Com isso o algoritmo obtém o elemento mais representativo ou central do agrupamento.

### Métodos Hierárquicos

Os métodos hierárquicos de agrupamento trabalham através de uma série de sucessivas uniões ou divisões. Métodos hierárquicos *aglomerativos* iniciam com objetos individuais, tendo assim tantos clusters quanto o número de objetos. Os objetos mais similares são agrupados, e esses agrupamentos iniciais são unidos de acordo com a suas similaridades, até que todos os subgrupos sejam unidos em um único agrupamento. Métodos hierárquicos *divisivos* trabalham de maneira oposta. Um único grupo inicial é dividido em subgrupos, e o processo continua até que cada grupo seja formado por um único objeto (JOHNSON; WICHERN, 2007).

A representação mais natural de um cluster hierárquico é um diagrama de árvore, chamado dendrograma, que mostra como cada objeto está agrupado (DUDA; HART; STORK, 2000). A Figura 3.14 exibe um exemplo de dendrograma com 8 amostras. No nível 1 cada amostra forma um grupo único, pois cada ponto no cluster é altamente similar a ele mesmo, nos próximos níveis as amostras são agrupadas de acordo com um critério de similaridade, até que todas formem um único agrupamento.

A seguir são descritos alguns dos métodos hierárquicos mais comuns. Os seguintes passos são utilizados para agrupar  $N$  objetos utilizando métodos hierárquicos aglomerativos (JOHNSON; WICHERN, 2007):

1. Iniciar com  $N$  clusters, cada um contendo um único objeto e uma matriz  $N \times N$  de distâncias (ou dissimilaridades)  $D = d_{ik}$ ;
2. Procurar pelo par de clusters mais similar na matriz de distâncias, onde a distância entre os clusters mais similares,  $U$  e  $V$  é dada por  $d_{UV}$ .
3. Unir os clusters  $U$  e  $V$ , dando um rótulo para o novo cluster formado ( $UV$ ). Atualizar a matriz de distâncias:
  - (a) deletar linhas e colunas referentes aos clusters  $U$  e  $V$ ;
  - (b) adicionar linha e coluna com as distâncias entre o cluster ( $UV$ ) e os clusters restantes.
4. Repetir os passo 2 e 3  $N - 1$  vezes, guardando a identidade dos clusters eu são unidos e em que nível (distância ou similaridade) eles ocorreram.

*Método Single Linkage* ou *Nearest Neighbour*: é o método do vizinho mais próximo, que consiste em considerar a medida de distância entre dois clusters como sendo a menor distância entre um elemento de um cluster e um elemento de outro cluster. A distância mínima entre o cluster ( $UV$ ) e outro cluster  $W$  é dada pela Equação 3.27:

$$d_{UVW} = \min\{d_{UW}, d_{VW}\}, \quad (3.27)$$

onde  $d_{UW}$  e  $d_{VW}$  são as distâncias entre os vizinhos mais próximos dos clusters  $U$  e  $W$  e clusters  $V$  e  $W$ , respectivamente.

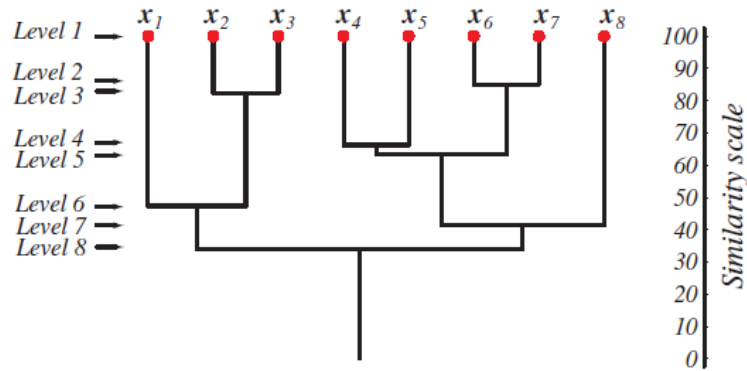


Figura 3.14: Exemplo de dendrograma. No primeiro nível cada agrupamento é composto por uma única amostra. As amostras mais similares são agrupadas, de acordo com um critério de similaridade, até que todas sejam consideradas similares, formando um único agrupamento (DUDA; HART; STORK, 2000).

*Método Complete Linkage* ou *Farthest Neighbour*: é o método do vizinho mais distante, que consiste em considerar a medida de distância entre dois clusters como sendo a maior distância entre um elemento de um cluster e um elemento de outro cluster. A distância entre o cluster ( $UV$ ) e outro cluster  $W$  é dada pela Equação 3.28:

$$d_{UVW} = \max\{d_{UW}, d_{VW}\}, \quad (3.28)$$

onde  $d_{UW}$  e  $d_{VW}$  são as distâncias entre os mais distantes membros dos clusters  $U$  e  $W$  e clusters  $V$  e  $W$ , respectivamente.

*Método Average Linkage*: é o método das distâncias médias entre clusters, que considera a distância entre dois clusters como sendo a distância média entre todos os pares de elementos, onde um elemento de cada par pertence a cada um dos clusters. A distância entre o cluster ( $UV$ ) e outro cluster  $W$  é dada pela Equação 3.29:

$$d_{(UV)W} = \frac{\sum_i \sum_k d_{ik}}{N_{(UV)}N_W}, \quad (3.29)$$

onde  $d_{ik}$  é a distância entre o objeto  $i$  no cluster ( $UV$ ) e o objeto  $k$  no cluster  $W$ , e  $N_{(UV)}$  e  $N_W$  são o número de itens nos clusters ( $UV$ ) e  $W$ , respectivamente.

*Método de Ward*: método proposto por Ward (1963) que é baseado em minimizar a perda de informação. Geralmente é implementado utilizando o erro das somas ao quadrado como critério função objetiva para determinar os dois clusters que devem ser unidos. Para um dado cluster  $k$ , seja  $ESS_k$  a soma dos quadrados dos desvios de todos os itens no cluster em relação ao valor médio do cluster (centroide). Se tiverem  $K$  clusters,  $ESS$  é definida como a soma de  $ESS_k$  ou  $ESS = ESS_1 + ESS_2 + \dots + ESS_k$ . A cada etapa, a união de todos os possíveis pares de clusters é considerada, e os dois clusters cuja combinação resulta no menor aumento em  $ESS$  (mínima perda de informação) são unidos (JOHNSON; WICHERN, 2007). No início, cada cluster consiste de um único item, e  $ESS = 0$ . Quando todos os clusters forem combinados em um único grupo de  $N$  itens, o valor de  $ESS$  é dado pela Equação 3.30:

$$ESS = \sum_{j=1}^N (x_j - \bar{x})'(x_j - \bar{x}), \quad (3.30)$$



onde  $x_j$  é a medida multivariada associada com o  $j$ -ésimo item e  $\bar{x}$  é a média de todos os itens.

**Métodos baseados em Teoria dos Grafos** O agrupamento de dados pode ser visto como um problema de corte de grafos. Cada dado é associado a um vértice em um grafo ponderado, onde os pesos das arestas entre cada elemento é grande se os elementos são similares e pequeno caso contrário. O objetivo é cortar o grafo em componentes conectados com grande peso interior, que correspondem aos clusters, através do corte das arestas que possuem um peso relativamente baixo (FORSYTH; PONCE, 2002).

Um método de agrupamento divisivo é baseado na construção de uma árvore de extensão mínima (MST - *Minimum Spanning Tree*) com os dados, e então são deletadas as arestas da árvore com as maiores distâncias para gerar os clusters. A Figura 3.15 ilustra o grafo obtido de nove pontos bidimensionais. Quebrando a ligação entre  $C$  e  $D$  com comprimento de 6 unidades (a aresta com maior distância euclidiana) faz com que sejam obtidos 2 clusters (JAIN; MURTY; FLYNN, 1999).

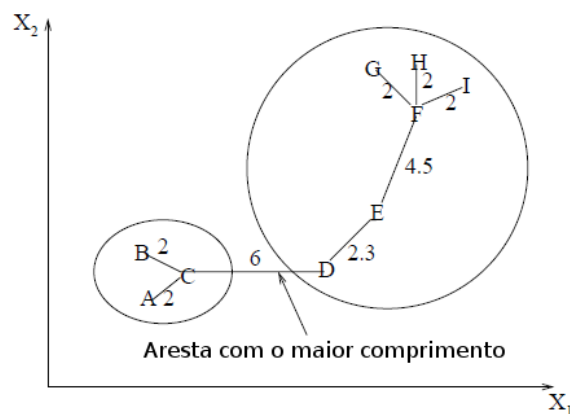


Figura 3.15: Exemplo de agrupamento de dados utilizando árvores. Removendo a aresta entre  $C$  e  $D$  com comprimento de 6 unidades (a aresta com maior distância euclidiana) faz com que os dados sejam obtidos 2 clusters (JAIN; MURTY; FLYNN, 1999).

### Métodos de Agrupamento Fuzzy

As abordagens tradicionais dos métodos de agrupamento geram partições, e cada elemento pertence a somente um cluster. São definidos agrupamentos rígidos, definindo em que grupo cada elemento ficará. O agrupamento fuzzy estende esta noção por ter um grau de associação de cada elemento com todos os clusters. As saídas desses algoritmos são clusters, mas não partições. As etapas do algoritmo de agrupamento fuzzy podem ser descritas como a seguir (JAIN; MURTY; FLYNN, 1999).

1. Selecionar uma partição fuzzy inicial de  $N$  objetos em  $K$  clusters através da seleção da matriz de pertinência  $U$   $N \times K$ . Um elemento  $u_{ij}$  dessa matriz representa o grau de pertinência do objeto  $x_i$  no cluster  $C_j$ . Geralmente  $u_{ij} \in [0, 1]$ .
2. Usando  $U$ , encontrar o valor de uma função de critério fuzzy associada com a partição correspondente. Uma possível função de critério fuzzy é:

$$E^2(X, U) = \sum_{i=1}^N \sum_{k=1}^K u_{ij} \|x_i - c_k\|^2,$$

onde  $c_k = \sum_{i=1}^N u_{ik}x_i$  é o centro do k-ésimo cluster fuzzy. Reatribuir elementos para o cluster para reduzir o valor desta função de critério e recalculer  $U$ .

3. Repetir o passo 2 até que as entradas em  $U$  não mudem significativamente.

A Figura 3.16 ilustra um exemplo de agrupamento fuzzy. Os retângulos delimitam dois clusters rígidos nos dados:  $H_1 = 1, 2, 3, 4, 5$  e  $H_2 = 6, 7, 8, 9$ . Um algoritmo de agrupamento fuzzy gera dois clusters fuzzy  $F_1$  e  $F_2$ , delimitados pelas elipses. Os elementos terão valores no intervalo  $[0, 1]$ , que indicam o valor de pertinência, para cada um dos clusters (JAIN; MURTY; FLYNN, 1999). Os clusters  $F_1$  e  $F_2$  poderiam, por exemplo, ser descritos como:

$$F_1 = \{(1, 0.9), (2, 0.8), (3, 0.7), (4, 0.6), (5, 0.55), (6, 0.2), (7, 0.2), (8, 0.0), (9, 0.0)\}$$

$$F_2 = \{(1, 0.0), (2, 0.0), (3, 0.0), (4, 0.1), (5, 0.15), (6, 0.4), (7, 0.035), (8, 1.0), (9, 0.9)\}$$

onde cada par  $(i, \mu_i)$  representa o i-ésimo elemento e seu valor de pertinência ao cluster  $\mu_i$ . Valores de pertinência maiores indicam maior confiança na atribuição do elemento para o cluster.

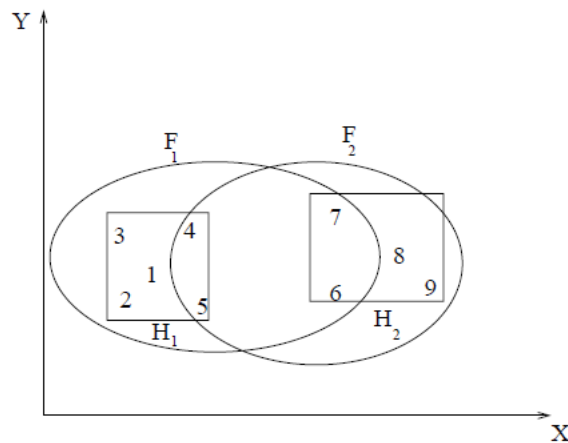


Figura 3.16: Ilustração de um agrupamento fuzzy. Os retângulos delimitam dois clusters rígidos e as elipses dois clusters fuzzy (JAIN; MURTY; FLYNN, 1999).

## 4 MÉTODO PROPOSTO PARA RASTREAMENTO E CONTAGEM DE VEÍCULOS EM VÍDEOS DE TRÁFEGO URBANO

Neste capítulo será apresentado detalhadamente o método de monitoramento de tráfego proposto. Serão apresentadas cada uma das etapas necessárias para o funcionamento do método, assim como cada uma das técnicas de processamento de imagens e visão computacional que foram utilizadas no método.

Ao longo do texto serão descritas cada uma das etapas para a obtenção das máscaras binárias contendo os objetos móveis detectados no primeiro plano do vídeo, para realizar a detecção dos veículos, para fazer o rastreamento dos veículos e, finalmente, para efetuar a contagem dos veículos.

Entre os desafios que o método proposto precisa lidar, podemos citar as tarefas de detectar e rastrear os veículos, pois os veículos andam muito próximos uns dos outros e, em situações de tráfego lento ou parado, podem ocorrer oclusões parciais dos veículos, o que poderia levar mais de um veículo a ser detectado como apenas um, além de dificultar o rastreamento.

As figuras apresentadas nessa seção para ilustrar e esclarecer os métodos utilizados foram adquiridas de diferentes vídeos, cedidos pela Empresa Pública de Transporte e Circulação (EPTC) de Porto Alegre - RS. Os vídeos foram obtidos utilizando câmeras fixas posicionadas acima da via, obtendo uma visão frontal do fluxo de tráfego, durante o período diurno. A Figura 4.1 ilustra alguns quadros dos vídeos utilizados.

O trabalho foi desenvolvido utilizando a plataforma Matlab (versão 7.13, 64 bits) e a biblioteca multiplataforma OpenCV (Open Source Computer Vision Library - versão 2.4) juntamente com as linguagens de programação C/C++. O método foi desenvolvido em um computador equipado com processador Intel Core 2 Quad Q8300 de 2.50 GHz e 8GB de memória RAM com o sistema operacional Windows 7 de 64 bits.

### 4.1 Visão Geral do Método Proposto

O objetivo do método proposto é realizar a contagem de veículos presentes nos vídeos. Para isso, podemos dividir o método nas seguintes etapas: inicialização (Seção 4.2), obtenção do *foreground* (Seção 4.3), detecção de veículos (Seção 4.7), rastreamento de veículos (Seção 4.8) e contagem de veículos (Seção 4.9).

A primeira etapa consiste em definir as regiões do vídeo que serão processadas. Nessa etapa inicial é preciso que um usuário forneça os dados de entrada. Primeiro, é exibido um quadro do vídeo que será processado, e o usuário deverá informar a região de interesse (ROI - Region of Interest), traçando um polígono sobre o quadro do vídeo. Essa ROI serve



Figura 4.1: Exemplos de vídeos utilizados, com seus respectivos códigos de identificação.

para delimitar a área de processamento do método, ou seja, os processos de detecção, rastreamento e contagem ocorrerão apenas com os objetos móveis que estiverem dentro desse polígono. A Figura 4.2 ilustra os polígonos representando as regiões de interesse dos vídeos.

Em seguida, é solicitado ao usuário que informe as posições dos laços virtuais, que são as áreas onde os veículos serão contados no vídeo quando passam sobre essas regiões. Foram utilizados retângulos para delimitar a área do laço virtual, sendo utilizados um laço em cada uma das faixas da via, como ilustrado na Figura 4.3. Existe um contador associado a cada um dos laços, e esse contador será incrementado cada vez que um veículo passar sobre o laço virtual.

Na etapa seguinte, são obtidas as máscaras binárias com os objetos detectados como objetos móveis na cena. Uma máscara binária é obtida utilizando o modelo de misturas de gaussianas (GMM) (STAUFFER; GRIMSON, 1999), e os objetos do primeiro plano são representados pelos objetos móveis na cena que possuem uma cor que difere das cores do fundo da cena. Outra máscara binária é obtida utilizando *Motion Energy Images* (MEI) (DAVIS; BOBICK, 1997). O MEI fornece uma máscara binária contendo os pixels do primeiro plano que mudaram de cor recentemente, representando onde ocorreu movimento ao longo de uma sequência definida de quadros, ajudando a descrever a forma e a trajetória dos objetos em movimento ao longo do vídeo. Uma máscara mais precisa é obtida combinando os resultados das máscaras obtidas com os métodos GMM e MEI. Assim, a nova máscara conterá somente os pixels que foram detectados como objetos móveis em ambas as máscaras, e é obtida através de uma operação *AND* entre as duas máscaras.

A Figura 4.4a e a Figura 4.4b exemplificam as máscaras binárias obtidas com o método GMM e com o método MEI, respectivamente. O resultado da combinação das máscaras é exibido na Figura 4.4c.

Após a obtenção da máscara de *foreground*, temos o início do algoritmo principal, que irá analisar a máscara e o vídeo original para fazer a detecção e rastreamento dos veículos.

Inicialmente, são lidos um quadro da máscara binária ( $MF$ ) e dois quadros do vídeo, o quadro atual ( $quadro_t$ ) e o quadro seguinte ( $quadro_{t+1}$ ). Esses dois quadros serão necessários para a obtenção dos vetores de movimento, posteriormente. A escolha da

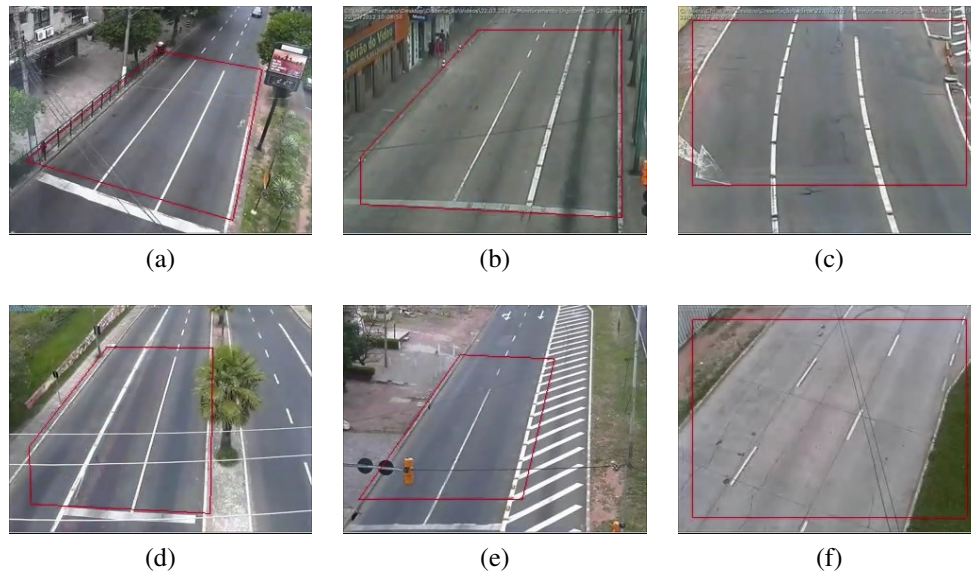


Figura 4.2: Exemplos de regiões de interesse, delimitadas pelas linhas vermelhas. (a) região de interesse do vídeo Cam06; (b) região de interesse do vídeo Cam25; (c) região de interesse do vídeo Cam41; (d) região de interesse do vídeo Cam64; (e) região de interesse do vídeo Cam68; (f) região de interesse do vídeo Cam73.



Figura 4.3: Exemplos de laços virtuais. Cada laço virtual é posicionado em uma das faixas da via, juntamente com seus respectivos contadores

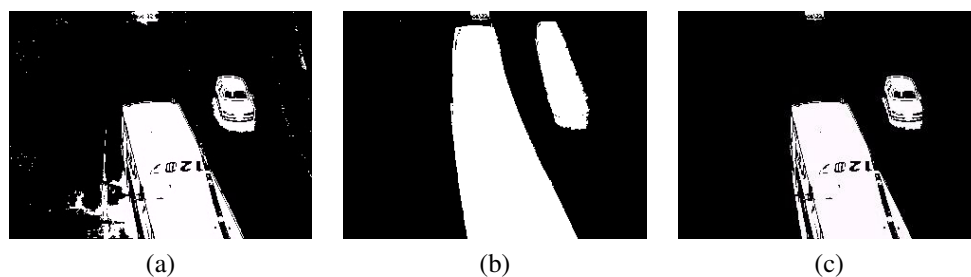


Figura 4.4: Exemplo de detecção dos objetos no primeiro plano do vídeo. (a) máscaras binária obtida com o método GMM; (b) máscaras binária obtida com o método MEI; (c) máscara combinando os resultados GMM e MEI.

utilização do  $quadro_t$  e o  $quadro_{t+1}$  para a obtenção dos vetores de movimento foi apenas uma questão de implementação, poderiam ser utilizados também o quadro atual,  $quadro_t$  e o quadro anterior,  $quadro_{t-1}$ , para a obtenção dos vetores de movimento.

Esses dois quadros passam por um processo de compensação de cor, calculando-se a cor média do fundo e dos dois quadros, sendo aplicada a devida compensação. Essa compensação de cor é utilizada para reduzir a diferença de cores entre um quadro e outro, devido às mudanças bruscas de iluminação que ocorrem nos vídeos utilizados.

O próximo passo é a detecção de cantos no  $quadro_t$ , utilizando o método dos mínimos autovalores (SHI; TOMASI, 1994), que chamamos de partículas. Essas partículas são detectadas somente nas regiões onde foram detectados objetos móveis, de acordo com a máscara binária ( $MF$ ). Após a obtenção das partículas, são obtidos os vetores de movimento para cada uma dessas partículas, utilizando a comparação de blocos com busca exaustiva, conforme visto na Seção 3.10. Assim, cada partícula será formada por quatro valores, as suas coordenadas espaciais, indicadas pelas suas posições  $x$  e  $y$  na imagem, e os valores dos vetores de movimento,  $V_x$  e  $V_y$ , indicando a posição da partícula no quadro seguinte.

As partículas serão agrupadas, formando clusters, e o centroide de cada cluster será armazenado para ser processado no próximo quadro do vídeo. As regiões do  $quadro_t$  do vídeo são divididas em: (a) regiões de partículas livres, que possuem partículas que não foram atribuídas a grupos de partículas ou veículos detectados previamente; (b) regiões de partículas agrupadas, onde as partículas fazem parte de um cluster detectado previamente; e (c) regiões de veículos, formadas por partículas que foram atribuídas aos veículos detectados. Cada uma dessas regiões será processada de uma maneira diferente.

A próxima etapa então é verificar as regiões existentes que deverão ser processadas. Se não existirem centroides referentes à regiões de partículas agrupadas ou regiões de veículos e o número de partículas for menor que um valor predeterminado, nada é feito nessa iteração, e o algoritmo vai para a próxima iteração, para processar o  $quadro_{t+1}$ . Essa condição ocorre quando não existem veículos passando no vídeo. Porém, se não existirem centroides, mas o número de partículas for maior que o valor mínimo predeterminado, temos regiões de partículas livres, e um processo de agrupamento inicial com as partículas é aplicado, gerando novos agrupamentos. Essa situação ocorre no início do vídeo, quando ainda não foram formados todos os agrupamentos ou um veículo entra na região de interesse.

Se existirem centroides referentes a regiões de partículas agrupadas ou regiões de veículos, significa que já temos veículos detectados no vídeo ou agrupamentos formados, que poderão ser detectados como veículos posteriormente. Nessa situação, também é realizado o processo de agrupamento, mas agora utilizando os valores dos centroides obtidos no quadro anterior como inicialização, que serão usados como centroides iniciais do algoritmo de agrupamento. Os clusters formados nesse processo passarão por processos de fusão e divisão de clusters, que são detalhados na Seção 4.6 e Seção 4.5, resultando em novos clusters, cujos centroides serão armazenados para serem processados no próximo quadro.

A próxima etapa é verificar se os clusters correspondem a veículos. É utilizada a envoltória convexa, discutida na Seção 3.7 para gerar uma região convexa das partículas que correspondem a cada um dos agrupamentos. Essas regiões convexas, juntamente com a máscara com os objetos do primeiro plano, são utilizadas para determinar se os clusters realmente correspondem a um veículo. São realizados testes para verificar a área do cluster, a taxa de ocupação, a proporção entre os eixos maiores e menores e também

a área da máscara de primeiro plano presente nas posições que correspondem à parte interna e à parte externa do cluster analisado. Cada teste possui uma pontuação, e ao final dos testes, os agrupamentos que atingirem um valor mínimo de pontuação, definido experimentalmente, são considerados como pertencentes a veículos.

Com os veículos detectados, inicia-se a etapa de rastreamento. O rastreamento é realizado apenas nos clusters que foram identificados como associados aos veículos e é baseado na similaridade de histogramas de cor computados para janelas de pixels centradas em cada uma das partículas associadas com a área convexa dos veículos no quadro anterior e com as mesmas partículas no quadro atual, cujas posições são dadas pelos vetores de movimento. Um valor de similaridade alto indica que são as mesmas partículas em ambos os quadros, e as partículas são mantidas. Se a similaridade entre as partículas resultar em um valor baixo, isso indica que as partículas são diferentes, e são então eliminadas do rastreamento. Uma nova envoltória convexa é criada considerando apenas as partículas que foram mantidas. Essa nova região convexa receberá um rótulo para ser usado durante o rastreamento, e esse rótulo deverá ser o mesmo rótulo atribuído a essa região nos quadros anteriores, e serve para identificar cada um dos veículos desde o momento em que ele é detectado pela primeira vez até a sua saída da região de interesse.

Os veículos rastreados também passarão por um processo de divisão de clusters. Essa etapa é realizada para casos em que veículos sobrepostos são considerados como sendo o mesmo veículo, devido à perspectiva. Quando os veículos entram na área de interesse, eles geralmente estão próximos aos pontos de fuga da cena e por isso aparecem pequenos e muito próximos, o que pode fazer com que sejam incorretamente detectados como apenas um veículo. Conforme o movimento dos veículos em direção à saída da área de interesse, mais separados e próximos da câmera eles ficarão, facilitando a detecção dos veículos.

Por fim, temos a etapa de contagem dos veículos, que é baseada em laços virtuais. Somente os veículos detectados e rastreados serão contados, ou seja, os clusters que não forem identificados como correspondentes a um veículo não serão contados, mesmo se os clusters forem persistentes ao longo de vários quadros do vídeo. A contagem é realizada em laços virtuais definidos pelo usuário, que podem ser definidos em qualquer região da área de interesse que o usuário desejar. Contudo, como são contados somente os veículos detectados e rastreados, a melhor posição para os laços virtuais é na área em que os veículos estão deixando a cena. Assim, a chance do veículo ser contado corretamente aumenta, pois os veículos terão passado pelos processos de detecção e rastreamento ao longo de um número maior de quadros até atingir os laços. Os contadores de cada laço virtual serão incrementados quando ocorrer uma intersecção entre as áreas do veículo rastreado e a do laço virtual. Cada cluster detectado e identificado como pertencente a um veículo será contado apenas uma vez. Se ocorrer uma intersecção com mais de um laço, apenas o que tiver a maior área de intersecção será considerado. A Figura 4.5 apresenta a visão geral do método proposto. A obtenção das máscaras binárias do primeiro plano, ou *foreground*, utilizando os métodos de modelo de misturas de gaussianas (GMM) (STAUFFER; GRIMSON, 1999) e Motion Energy Images (MEI) (DAVIS; BOBICK, 1997), e a combinação dos resultados desses dois métodos, foram implementadas com a linguagem de programação C/C++, juntamente com a biblioteca multiplataforma OpenCV. As outras etapas do método foram implementadas utilizando a plataforma Matlab.

Como principais contribuições desse trabalho, podemos citar a combinação de dois métodos para melhor segmentação dos objetos que estão em movimento no primeiro plano do vídeo utilizando GMM e MEI. Dessa forma é criada uma máscara binária que representa os objetos em movimento no vídeo de maneira mais confiável. Os detalhes

podem ser vistos na Seção 4.3. Também podemos citar como contribuições deste trabalho o desenvolvimento de técnicas que utilizam a máscara com os objetos móveis no primeiro plano para melhor identificar a qual agrupamento pertence cada uma das partículas e verificar se realmente os agrupamentos pertencem aos veículos. Foram desenvolvidos procedimentos para determinar se algum agrupamento apresenta correspondência com mais de um veículo na cena, fazendo a divisão do agrupamento caso necessário, e também determinar se temos mais de um agrupamento correspondendo ao mesmo veículo na cena, situação em que é necessário realizar a fusão desses agrupamentos. Essas duas principais contribuições do trabalho estão descritas na Seção 4.5 e na Seção 4.6.

Outra contribuição fundamental deste trabalho é a etapa de detecção de veículos, apresentada na Seção 4.7. Apenas o agrupamento de partículas não garante que cada agrupamento corresponda a um veículo na cena. Podem ocorrer situações em que os veículos ficam muito próximos ou sofram oclusões, fazendo com que seja formado um agrupamento que não corresponde corretamente a um veículo da cena, ou ainda erros na máscara de *foreground*, detectando regiões no *foreground* que não são veículos. Através de uma série de testes, são determinados se cada um dos agrupamentos possuem características que indiquem que eles correspondem a veículos na cena, como tamanho, proporção, entre outras.



Figura 4.5: Visão geral do método proposto.

## 4.2 Inicialização

A etapa de inicialização abrange a inicialização das variáveis e dos parâmetros fornecidos pelo usuário. O primeiro parâmetro fornecido pelo usuário é o vídeo que deverá ser processado, e, em seguida, a ROI e os laços virtuais.

Após informar o vídeo que será utilizado, é exibido um quadro do vídeo e solicitado ao usuário que trace as linhas para demarcar a região de interesse. O resultado é uma máscara



binária da região que deverá ser processada. É realizada uma operação *AND* entre a máscara binária da ROI e os quadros do vídeo, delimitando a área que será processada. A Figura 4.6 ilustra esse processo. A Figura 4.6a exibe o quadro do vídeo, a Figura 4.6b exibe as linhas traçadas pelo usuário para demarcar a região de interesse, a Figura 4.6c exibe a máscara binária obtida e a Figura 4.6d exibe o quadro final, que é o que será processado pelo método proposto.

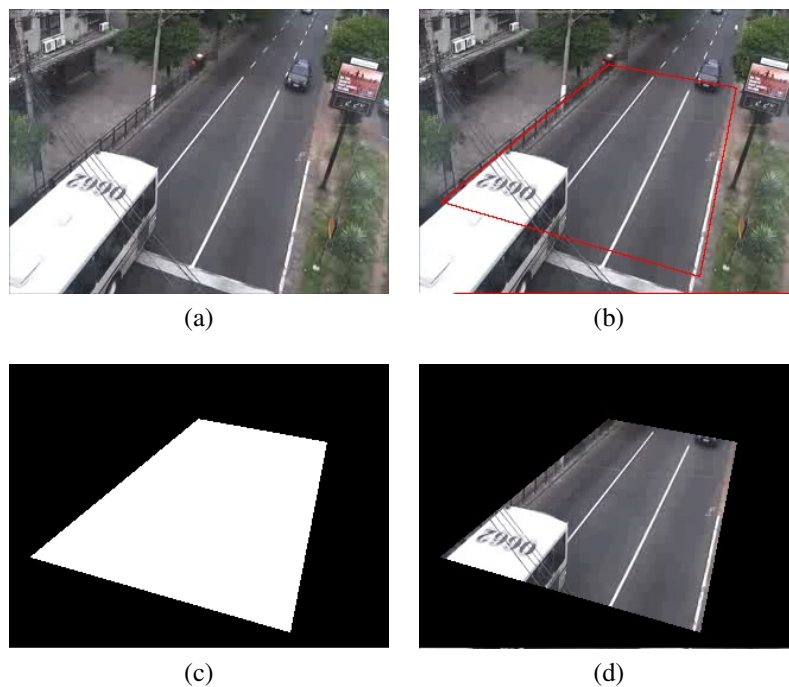


Figura 4.6: Exemplo do processo de obtenção da região de interesse. (a) quadro do vídeo; (b) linhas traçadas pelo usuário para demarcar a região de interesse; (c) máscara binária da ROI obtida; e (d) resultado da operação *AND* entre a máscara da ROI e o quadro.

A obtenção dos laços virtuais é semelhante à obtenção da ROI. Quando o usuário termina de traçar cada laço virtual e ele é inserido na imagem, é perguntado ao usuário se ele deseja inserir outro laço virtual. Quando o usuário finalizar a inserção de laços, é gerada uma máscara contendo todos os laços virtuais, cada um com uma numeração, partindo de um até o número de laços inseridos pelo usuário.

Em seguida, é obtida uma imagem do fundo da cena através do método da mediana temporal para cada canal de cor RGB, ou seja, é utilizada a mediana dos quadros do vídeo para criar uma imagem do fundo da cena, que será usada no processo de compensação de cor. A cor média de fundo é obtida através da Equação 4.1.

$$CMF_C = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N p(i, j, C), \quad (4.1)$$

onde  $CMF_C$  é cor média de fundo para cada um dos canais de cor RGB,  $M$  e  $N$  são as dimensões do quadro do vídeo e  $p(i, j, C)$  é o valor do pixel na posição  $(i, j)$  do canal de cor  $C$ .

Então, para realizar a compensação de cor para o vídeo, primeiramente é calculada a cor média do quadro processado, de acordo com a Equação 4.2. A seguir, é obtida a diferença da cor média do fundo para a cor média do quadro processado em cada um dos

canais de cor ( $CMF_C - CMQ_C$ ), e essa diferença é somada a cada um dos pixels do quadro processado, em cada um dos canais de cor  $C$ .

$$CMQ_C = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N p(i, j, C), \quad (4.2)$$

onde  $CMQ_C$  é cor média do quadro processado no canal de cor  $C$ ,  $M$  e  $N$  são as dimensões do quadro do vídeo e  $p(i, j, C)$  é o valor do pixel na posição  $(i, j)$  do canal de cor  $C$ .

### 4.3 Obtenção da Máscara com Objetos no Primeiro Plano do Vídeo

Como mencionado anteriormente, são utilizados os métodos modelo de misturas de gaussianas (GMM) (STAUFFER; GRIMSON, 1999) e Motion Energy Images (MEI) (DAVIS; BOBICK, 1997) para a obtenção da máscara binária do primeiro plano, ou *foreground*, combinando os resultados desses dois métodos. Primeiramente, será discutido o método de mistura de gaussianas, que utiliza uma mistura de gaussianas adaptativa para modelar o fundo da cena e um algoritmo de componentes conectados para segmentar os objetos que fazem parte do primeiro plano do vídeo.

Cada pixel é modelado por uma mistura de  $K$  distribuições gaussianas, e o primeiro plano é considerado como contendo objetos móveis, que possuem cores diferentes das regiões do fundo da cena, que são mais estáticos, não tendo muitas variações ao longo do tempo (STAUFFER; GRIMSON, 1999). Este método foi escolhido porque pode lidar melhor com mudanças de iluminação e com pequenas mudanças de localização de objetos do fundo da cena, como, por exemplo, o movimento das folhas de árvores.

As mudanças de cor em cada pixel dentro de uma janela temporal são representadas como vetores  $\vec{X}_{i,t} = \{R_{i,t}, G_{i,t}, B_{i,t}\}$ . A probabilidade de observar uma dada cor em um pixel  $i = (x_0, y_0)$  no tempo  $t$  é modelado por  $K$  componentes ( $K = 3$  neste trabalho). As cores assumidas por um pixel em um intervalo de tempo  $[1, t - 1]$  são denotadas como  $\{\vec{X}_{i,1}, \dots, \vec{X}_{i,t-1}\} = \{\vec{I}(x_0, y_0, j) : 1 \leq j \leq t - 1\}$ , onde  $\vec{I}$  é a sequência de vídeo. Assim, a probabilidade de observar uma dada cor no pixel  $i$  no tempo  $t$  é:

$$P(\vec{X}_{i,t}) = \sum_{k=1}^K \omega_{i,t,k} \eta(\vec{X}_{i,t}, \vec{\mu}_{i,t,k}, \Sigma_{i,t,k}), \quad (4.3)$$

onde  $\omega_{i,t,k}$ ,  $\vec{\mu}_{i,t,k}$ ,  $\Sigma_{i,t,k}$  são o peso, o vetor de médias e a matriz de covariância da  $k$ -ésima componente da GMM no tempo  $t$ , e  $\eta$  é a função densidade de probabilidade gaussiana, dada por:

$$\eta(\vec{X}_{i,t}, \vec{\mu}_{i,t,k}, \Sigma_{i,t,k}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{k,t}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{X}_{i,t} - \vec{\mu}_{k,t})^T \Sigma_{k,t}^{-1} (\vec{X}_{i,t} - \vec{\mu}_{k,t})}. \quad (4.4)$$

A matriz de covariância é estabelecida como diagonal  $\Sigma_{i,t,k} = \text{diag}[\sigma_{R_{i,t,k}}^2, \sigma_{G_{i,t,k}}^2, \sigma_{B_{i,t,k}}^2]$ , onde  $\sigma_{R_{i,t,k}}^2$ ,  $\sigma_{G_{i,t,k}}^2$ ,  $\sigma_{B_{i,t,k}}^2$  são as variâncias dos canais RGB do componente  $k$  da GMM no tempo  $t$ , respectivamente (RIBEIRO; GONZAGA, 2006). Por simplicidade, a matriz de covariância é assumida como  $\Sigma_{i,t,k} = \sigma^2 I_3$ , onde  $I_3$  é a matriz identidade.

Para criar o modelo *background*, as componentes da GMM são ordenadas em ordem decrescente de acordo com os seus valores da razão entre seus pesos e o desvio padrão  $\{\frac{\omega_{i,t,k_{ord}}}{\sigma_{i,t,k_{ord}}}\}$ , onde  $k_{ord} = 1, \dots, K$  denota as componentes da GMM ordenadas.

Assim, as componentes da GMM mais prováveis de representar o fundo (isto é, com maiores  $\omega_{i,t,k_{ord}}$ ) e mais estáticas na cena (com menor variabilidade  $\sigma_{i,t,k_{ord}}$ ) permanecem no topo da ordenação (i.e. assumindo menor  $k_{ord}$ ,  $k_{ord} \ll K$ ), enquanto as componentes menos prováveis ficam em torno do final da ordenação (i.e. assumem um valor alto para  $k_{ord}$ ) (STAUFFER; GRIMSON, 1999). Baseado na Equação 4.5, as  $k_{top}$  componentes que estão no topo são escolhidas como modelo do fundo.

Um pixel com suas cores RGB correspondendo a alguma dessas  $k_{top}$  componentes da GMM é considerado como pertencente ao fundo da cena (KAEWTRAKULPONG; BOWDEN, 2001).

Uma correspondência ocorre se a cor RGB do pixel está dentro de 2.5 desvios padrões de alguma das componentes da GMM, em todos os canais RGB. Se a cor RGB do pixel não corresponder a nenhuma das  $k_{top}$  componentes no topo da GMM ele é considerado como pertencente ao *foreground*, como detalhado a seguir.

$$B_{k_{top}} = \underset{k_{ind}}{\operatorname{argmin}} \left( \sum_{j=1}^{k_{ind}} (\omega_{i,t,j}) > T_{Background} \right), \quad (4.5)$$

onde  $T_{Background}$  é um limiar usado para selecionar um número de  $k_{top}$  componentes GMM representando o fundo da cena.  $T_{Background} = 0.5$  neste experimento.

Se um pixel RGB corresponder com alguma das  $K$  componentes, a componente que teve uma correspondência com o pixel terá o peso, a média e a sua covariância atualizada, seguindo duas etapas. Primeiro, a componente que teve uma correspondência com o pixel terá seu peso  $\omega_{i,t-1,k}$  no tempo  $t - 1$  atualizado para obter  $\omega_{i,t,k}$ , como a seguir:

$$\omega_{i,t,k} = (1 - \alpha)\omega_{i,t-1,k} + \alpha, \quad (4.6)$$

onde  $\alpha$  é uma taxa de aprendizagem, e o vetor de médias e a matriz de covariância são atualizados como a seguir:

$$\vec{\mu}_{i,t,k} = (1 - \rho)\vec{\mu}_{i,t-1,k} + \rho\vec{X}_{i,t}, \quad (4.7)$$

$$\Sigma_{i,t,k} = (1 - \rho)\Sigma_{i,t-1,k} + \rho(\vec{X}_{i,t} - \vec{\mu}_{i,t,k})^T(\vec{X}_{i,t} - \vec{\mu}_{i,t,k}), \quad (4.8)$$

onde o parâmetro  $\rho$  também é uma taxa de aprendizagem, definida por:

$$\rho = \alpha\eta(\vec{X}_{i,t}, \vec{\mu}_{i,t-1,k}, \Sigma_{i,t-1,k}). \quad (4.9)$$

A seguir, os parâmetros das distribuições que não tiveram uma correspondência são mantidas, assim  $\vec{\mu}_{i,t,k} = \vec{\mu}_{i,t-1,k}$  e  $\Sigma_{i,t,k} = \Sigma_{i,t-1,k}$ , enquanto seus pesos são atualizados como indicado na Equação 4.10:

$$\omega_{i,t,k} = (1 - \alpha)\omega_{i,t-1,k}. \quad (4.10)$$

Se nenhuma das  $K$  componentes corresponder ao pixel RGB atual, então a componente menos provável da GMM é substituída por uma nova componente, com sua média igual ao valor atual de  $\vec{X}_{i,t}$ , uma variância inicial alta e um baixo peso a priori.

A Figura 4.7 ilustra um quadro de um vídeo e a sua respectiva máscara de primeiro plano gerada utilizando o GMM. A Figura 4.7a exibe o quadro original, e a Figura 4.7b exibe a máscara binária obtida.

A seguir é discutido o método *Motion Energy Images* (MEI) (DAVIS; BOBICK, 1997), que fornece uma máscara binária com os objetos em movimento no primeiro plano



Figura 4.7: Exemplo de máscara binária obtida através do método GMM. (a) quadro original; (b) máscara binária gerada.

do vídeo. Cada quadro dessa máscara mostra, de maneira sobreposta, as regiões de movimento que foram detectadas ao longo de um número recente de quadros. Ajudando a obter o formato e a trajetória dos objetos móveis ao longo do tempo.

Seja  $I_g(x, y, t)$  uma sequência do vídeo em tons de cinza, e  $D(x, y, t)$  uma sequência binária indicando as regiões de movimento da cena, dada por:

$$D(x, y, t) = |I_g(x, y, t) - I_g(x, y, t - 1)| > Tb, \quad (4.11)$$

onde  $x$  e  $y$  são as posições dos pixels no quadro no tempo  $t$  e  $t - 1$ , e  $Tb$  é um limiar ( $Tb = 35$  neste experimento). A máscara binária gerada pelo método MEI  $E_\tau(x, y, t)$  é definida como a união dessas sequências binárias  $D(x, y, t)$  em um intervalo de tempo:

$$E_\tau(x, y, t) = \bigcup_{j=0}^{\tau-1} D(x, y, t - j), \quad (4.12)$$

onde  $\tau$  determina a extensão temporal da sequência binária MEI  $E_\tau(x, y, t)$ .

A Figura 4.8 exibe um exemplo de uma máscara binária obtida através do método MEI. A Figura 4.8a exibe o quadro original, e a Figura 4.8b exibe a máscara binária obtida.



Figura 4.8: Exemplo de máscara binária obtida através do método MEI. (a) quadro original; (b) máscara binária gerada.

Para obter uma máscara binária mais precisa dos objetos que estão em movimento em cada quadro do vídeo, as duas máscaras, obtidas pelos métodos GMM e MEI, são combinadas, formando uma única máscara binária. As máscaras são combinadas utilizando uma operação lógica *AND* entre as duas máscaras, assim, somente os pixels que foram

detectados como pertencentes a objetos em movimento em ambas as máscaras serão considerados como pixels que realmente pertencem a objetos do primeiro plano do vídeo. Algumas operações são utilizadas nas máscaras para remover pequenas regiões e fechar buracos. Foi utilizada uma operação de abertura de componentes conectados menores que 20 pixels, uma operação de fechamento, utilizando elemento estruturante elíptico de raio 4, preenchimento de regiões por difusão (*flood-fill*) utilizando conectividade 4, e outra operação de abertura de componentes conectados menores que 100 pixels. Essa nova máscara obtida tende a ser mais precisa do que a máscara gerada somente utilizando a mistura de gaussianas, eliminando erros que foram causados por mudanças de iluminação, como pode ser observado na Figura 4.9.

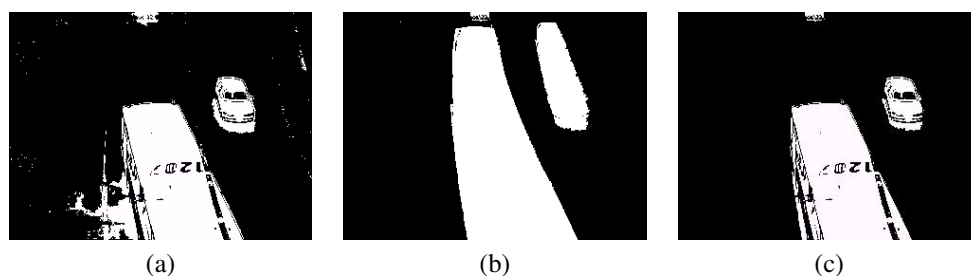


Figura 4.9: Máscara binária resultante da combinação das máscaras geradas pelos métodos GMM e MEI. (a) máscara gerada com GMM; (b) máscara gerada com MEI; (c) máscara resultante da operação AND entre (a) e (b).

#### 4.4 Detectando os Agrupamentos de Partículas

O processo de detecção das partículas ocorre de acordo com a região das partículas, que, como visto antes, estão divididas em: (a) regiões de partículas livres, que possuem partículas que não foram atribuídas a grupos de partículas ou veículos detectados previamente; (b) regiões de partículas agrupadas, onde as partículas fazem parte de um cluster detectado previamente; e (c) regiões de veículos, formadas por partículas que foram atribuídas aos veículos detectados.

As regiões são verificadas baseadas nas partículas existentes do quadro anterior. No início ainda não existem agrupamentos anteriores, então todas as partículas obtidas serão consideradas partículas livres. As partículas são detectadas utilizando o método dos mínimos autovalores (SHI; TOMASI, 1994), que é baseado na detecção de cantos na imagem, conforme discutido na Seção 3.9. As partículas são detectadas somente nas regiões detectadas como primeiro plano, isto é, em regiões que representam movimento na máscara binária de *foreground*. Uma ilustração das partículas detectadas no primeiro plano é exibida na Figura 4.10. A Figura 4.10a mostra a máscara de *foreground* e a Figura 4.10b as partículas (cantos) que foram detectadas, delimitadas pelas regiões que foram detectadas como movimento na máscara binária.

Se existirem agrupamentos obtidos no quadro anterior,  $quadro_{t-1}$ , é gerada uma área convexa ao redor das posições dadas pelos vetores de movimento da partículas, usando a envoltória convexa explanada na Seção 3.7, para cada um dos agrupamentos. Ou seja, a área convexa é gerada no  $quadro_t$  nas posições para onde as partículas do  $quadro_{t-1}$  foram deslocadas, de acordo com a estimativa de movimento realizada utilizando a comparação de blocos com busca exaustiva, discutida na Seção 3.10.

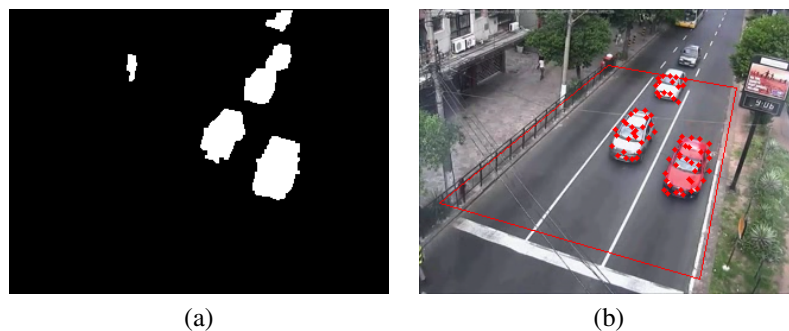


Figura 4.10: Exemplo de partículas detectadas no *foreground*. (a) máscara de primeiro plano; (b) partículas obtidas.

Dentro de cada uma dessas regiões convexas são geradas partículas aleatórias. Essas partículas aleatórias servirão para melhorar a detecção dos veículos, especialmente os mais longos e que não possuem muita textura, e que por isso não terão muitas partículas detectadas.

A seguir, são obtidos os vetores de movimento para todas as partículas, as que foram detectadas como cantos e as que forem eventualmente geradas com o ruído impulsivo. Os vetores de movimento são obtidos utilizando a comparação de blocos com busca exaustiva discutida na Seção 3.10. São geradas janelas do tamanho de  $9 \times 9$  pixels ao redor de cada partícula do  $quadro_t$ , que são então comparadas com uma região de busca de  $17 \times 17$  pixels ao redor das mesmas posições de cada partícula, mas no  $quadro_{t+1}$ . Assim são obtidas as posições estimadas de onde cada uma das partículas deve estar no próximo quadro.

A próxima etapa é realizar o agrupamento inicial das partículas livres, que são partículas que não foram atribuídas a grupos de partículas ou veículos detectados previamente. Essas partículas livres existem em duas situações. A primeira é quando não existirem agrupamentos passados dos quadros anteriores e houver um número mínimo de partículas, o que ocorre quando são processados os primeiros quadros do vídeo ou quando novos veículos entram no quadro. A segunda é quando, apesar de existirem agrupamentos formados e/ou agrupamentos identificados como veículos que foram passados do quadro anterior, também existem partículas que não pertencem a nenhum desses grupos.

Se existirem centroides referentes a regiões de partículas agrupadas ou regiões de veículos, significa que já temos veículos detectados ou agrupamentos formados, que poderão ser detectados como veículos posteriormente. Nesses casos, também serão realizados processos de agrupamento, mas agora utilizando os valores dos centroides obtidos no quadro anterior como inicialização, que serão usados como centroides iniciais do algoritmo de agrupamento. O processo de agrupamento inicial das partículas livres é descrito na seção abaixo.

#### 4.4.1 Agrupamento Inicial de Partículas

As partículas são agrupadas de acordo com as suas posições espaciais e vetores de movimento. O agrupamento é realizado utilizando o algoritmo k-means, apresentado na Subseção 3.12.1, que, como foi explanado, necessita que o número de clusters seja definido antes de sua execução. Para definir o número inicial de clusters presentes no quadro atual, é feito o procedimento descrito a seguir.

Seja  $N_{min}$  o número mínimo de partículas que podem estar associadas a uma região

do primeiro plano (i.e. veículo). Foi determinado experimentalmente que  $N_{min} = 8$  partículas/veículo é o suficiente para os vídeos testados (resolução de  $320 \times 240$  pixels). Vídeos com resoluções diferentes ou contendo veículos com outras dimensões em relação às dimensões do quadro podem necessitar de um ajuste no valor de  $N_{min}$ .

Dados  $N_{min}$  e o número total de partículas detectadas no quadro,  $N_{part}$ , podemos definir o número máximo de clusters para o quadro de vídeo processado como  $clusters_{max} = N_{part}/N_{min}$ . Em seguida, são obtidos os clusters iterativamente com o algoritmo k-means, com o número de clusters variando de um até  $clusters_{max}$ . A cada iteração, a soma das distâncias das partículas para os seus respectivos centroides é calculada (Equação 4.13), e quando a diferença absoluta das distâncias partículas-centroides em duas iterações consecutivas for menor que um limiar  $T_{Dif}$  (0.01 neste trabalho), o processo iterativo termina. A medida de distância utilizada no algoritmo k-means para obter a distância das partículas para os seus respectivos centroides que gerou os clusters com maior precisão, foi a medida do cosseno, apresentada na Subseção 3.12.1. A soma das distâncias das partículas para os seus respectivos centroides em cada iteração é calculada como:

$$SD^{cluster_k} = \sum_{i=1}^{cluster_k} \sum_{j=1}^n D_{cos}(\vec{u}_{i,j}, \vec{c}_i), \quad (4.13)$$

onde  $cluster_k$  é o número de clusters a cada iteração e  $n$  é o número de partículas em um cluster  $Q_i$  e  $D_{cos}(\vec{u}_{i,j}, \vec{c}_i)$  é a distância das partículas  $\vec{u}_i$ , pertencentes ao cluster  $Q_i$ , para o seu respectivo centroide,  $\vec{c}_i$ . A diferença absoluta das distâncias entre duas iterações consecutivas é definida como  $Dif = |SD^{cluster_k-1} - SD^{cluster_k}|$ .

A qualidade de cada cluster formado é verificada utilizando a técnica de gráfico da silhueta dos clusters (ROUSSEEUW, 1987), através da comparação da similaridade de cada partícula com as outras partículas de seu próprio cluster assim como com as partículas pertencentes a outros clusters. A silhueta mostra quais partículas estão bem ajustadas dentro de seus clusters, e é dada por:

$$S(\vec{u}) = \frac{b(\vec{u}) - a(\vec{u})}{\max\{a(\vec{u}), b(\vec{u})\}}, \quad (4.14)$$

onde  $S(\vec{u})$  é o valor da silhueta para a partícula  $\vec{u}$ , e descreve o quão similar essa partícula é das partículas de seu próprio cluster, quando comparada com as partículas dos outros clusters.  $a(\vec{u})$  é a distância média da partícula  $\vec{u}$  para as outras partículas no mesmo cluster que  $\vec{u}$ , e  $b(\vec{u})$  é a distância média mínima da partícula  $\vec{u}$  para as partículas de algum outro cluster, mais próximo, ou seja, distância média da partícula  $\vec{u}$  para as partículas no cluster mais próximo do cluster contendo a partícula  $\vec{u}$ . O cluster com a menor dissimilaridade média é chamado de cluster vizinho. Os valores da silhueta variam entre  $-1$  e  $+1$ . Um valor alto de silhueta indica que a partícula  $\vec{u}$  combina bem com o seu próprio cluster, e tem uma baixa correspondência com os clusters vizinhos. Um valor de silhueta próximo a  $-1$  indica que seria mais apropriado atribuir a partícula  $\vec{u}$  para o cluster vizinho. Neste trabalho, as partículas com valores de silhueta menores que zero são consideradas como não sendo similares às outras partículas de seus respectivos clusters, e são eliminadas. Qualquer medida de dissimilaridade pode ser usada, sendo mais comum o uso de medidas de distância, como a *city block*, que foi a utilizada neste trabalho.

As partículas que foram eliminadas passam recursivamente por esse processo de agrupamento inicial para serem agrupadas em novos clusters, que se juntarão aos obtidos anteriormente.

A Figura 4.11 mostra o diagrama do processo de agrupamento inicial, onde  $cluster_k$  é o número de clusters em cada iteração,  $T_{Dif}$  é um limiar para a diferença absoluta das distâncias partículas-centroides,  $clusters_{max}$  é o número máximo de clusters permitidos, dado por  $N_{part}/N_{min}$ .

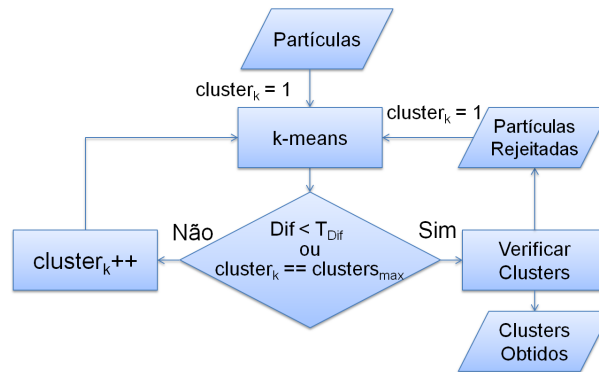


Figura 4.11: Ilustração do processo de agrupamento inicial.

A hipótese que as partículas associadas a um veículo ocorrem próximas uma das outras e se movem de maneira coerente, não pode ser verificada em alguns casos práticos. Um exemplo é o caso de veículos longos, que podem apresentar pouca textura, como ilustrado na Figura 4.12. A Figura 4.12 apresenta um caso em que teremos mais de um grupo de partículas associado ao mesmo veículo. Devido ao comprimento do veículo, são gerados grupos de partículas com uma grande distância espacial entre eles (Figura 4.12a). Também podem ocorrer deslocamentos das partículas devido à deformação perspectiva, levando a vetores de movimento diferentes para as partículas, mesmo que pertençam ao mesmo veículo (Figura 4.12b).

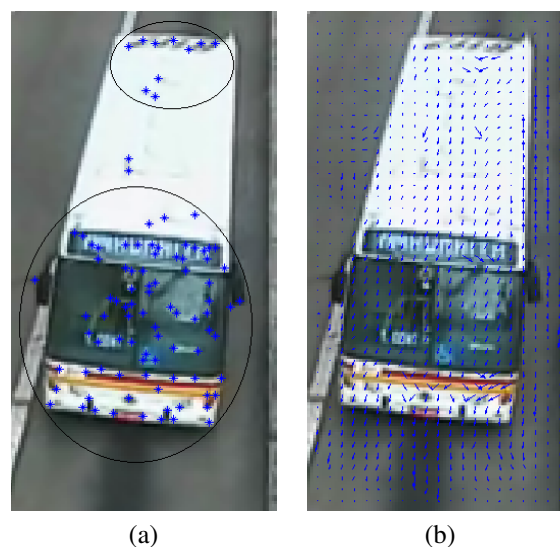


Figura 4.12: Exemplo de agrupamento inadequado causado pela região com pouca textura. (a) veículo com poucas partículas associadas a ele; (b) vetores de movimento diferentes ao longo do mesmo veículo; Esses fatores levam à obtenção de dois agrupamentos para o mesmo veículo.

Portanto, é necessário um processo para verificar e fazer a fusão desses clusters que



parecem estar associados com veículos diferentes, mas de fato pertencem ao mesmo veículo. Esse processo, denominado fusão de clusters, usa a máscara com os objetos de *foreground* para verificar se existe mesmo uma região de fundo detectada na posição da máscara correspondente à posição do cluster. Caso não exista, significa que os clusters pertencem ao mesmo veículo, e os clusters serão unidos em apenas um. Em outros casos, pode ser necessário fazer o processo inverso, dividir um cluster que foi associado a apenas um objeto móvel, mas na verdade existem dois objetos móveis. Essa situação ocorre quando os veículos estão muito próximos, gerando partículas com posições espaciais muito próximas e com vetores de movimento similares. Esse processo, denominado divisão de clusters, também utiliza a máscara de *foreground* para verificar a existência de uma região de fundo detectada na posição da máscara correspondente à posição do cluster. Caso exista, isso significa que existem objetos na máscara de *foreground* que, apesar de estarem próximos, estão separados um do outro, e o cluster deve então ser dividido em dois.

Os processo de divisão e fusão de clusters serão descritos detalhadamente a seguir, na Seção 4.5 e na Seção 4.6.

## 4.5 Divisão de Clusters

A divisão de cluster é a próxima etapa a ser aplicada aos agrupamentos presentes no quadro atual. A divisão dos clusters é necessária para tratar situações em que dois veículos estão muito próximos, movendo-se na mesma direção e com velocidades semelhantes, pois nessas situações ocorre o agrupamento incorreto das partículas deles como um único cluster. A Figura 4.13 ilustra o processo de divisão de cluster.

No processo de divisão, para todos os clusters  $Q_i = \{u_{i1}, \dots, u_{in}\}$ ,  $i = \{1, \dots, nc\}$ , onde  $\{u_{i1}, u_{i2}, \dots, u_{in}\}$  é um vetor representando as partículas no cluster  $Q_i$  e  $nc$  é o número total de clusters, o seguinte teste é realizado para verificar se o cluster  $Q_i$  deve ser dividido. A Figura 4.13a mostra as partículas sobrepostas ao quadro processado, de acordo com as suas coordenadas. Como podemos observar na imagem, existem dois veículos na região desse cluster, e, portanto, deveriam existir dois clusters, cada um associado com um dos veículos.

Seja  $E_{conv} = envoltoria\_convexa(Q_i)$  a área da envoltória convexa, ou seja, a área convexa determinada pelo menor polígono convexo que contém todos os pontos de um conjunto, do cluster  $Q_i$  (Figura 4.13b) e  $AB = E_{conv} - (E_{conv} \cap MF)$ ,  $AB \subseteq E_{conv}$  a área que não está na máscara de *foreground* ( $MF$ ), na posição do cluster  $Q_i$ . A Figura 4.13c mostra a área do primeiro plano em  $E_{conv}$ , e a Figura 4.13d mostra as duas áreas sobrepostas, permitindo vermos, em azul,  $AB$ , a área do cluster que não corresponde a um objeto detectado na máscara de *foreground*. Se  $\#\{AB\}/\#\{E_{conv}\} > T_{div}$  (um limiar de divisão, 0.15 neste trabalho), então  $Q_j^{t+1} = (Q_i^t)'$ ,  $Q_{j+1}^{t+1} = Q_i^t - (Q_i^t)' = (Q_i^t)''$ ,  $j = \{1, \dots, nc'\}$ , com  $nc' > nc$ , onde  $\#\{AB\}$ ,  $\#\{E_{conv}\}$  representam o número de pixels nas regiões  $AB$  e  $E_{conv}$ , respectivamente, e  $(Q_i^t)'$ ,  $(Q_i^t)''$  são obtidos através do agrupamento das partículas em  $Q_i^t$  com o método k-means, com  $k = 2$ . Isto é, se mais de 15% da área em  $E_{conv}$  não está presente em  $MF$ , o cluster  $Q_i$  é dividido em dois, usando o algoritmo k-means, especificando que serão criados dois clusters com as partículas do cluster analisado,  $Q_i$ . O valor do limiar  $T_{div}$  é baseado em uma razão, e deve funcionar bem com vídeos com diferentes resoluções e distâncias em relação à câmera, mas pode ser ajustado caso necessário. Esse limiar indica a área máxima aceitável de background presente na região convexa do cluster para que ele não seja dividido. Conforme esse limiar aumenta, mais

área de background será tolerada dentro da região  $E_{conv}$  sem que o cluster seja dividido. A Figura 4.13e ilustra os clusters após o processo de divisão, onde agora cada veículo é corretamente associado às partículas de um cluster.

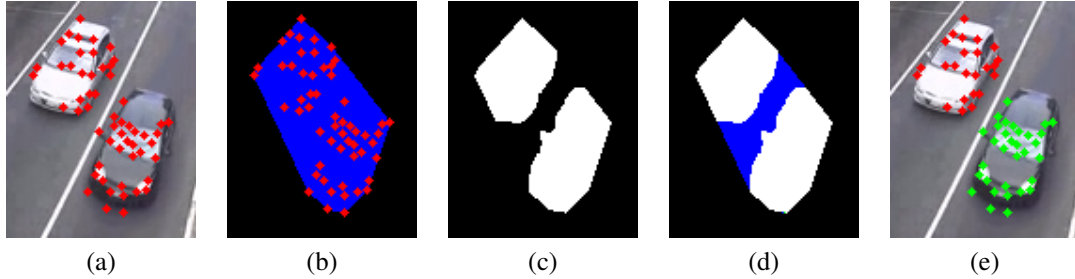


Figura 4.13: Exemplo de divisão de clusters. (a) partículas sobrepostas ao quadro processado que foram associadas a apenas um cluster; (b) área convexa ( $E_{conv}$ ) ao redor das partículas em (a); (c) área de *foreground* contida em  $E_{conv}$ ; (d) área do cluster que não corresponde a um objeto detectado na máscara de *foreground* ( $AB$ ), em azul; (e) resultado da divisão dos cluster. Como podemos observar, a razão entre as áreas  $AB$  e  $E_{conv}$  é maior que o limiar de divisão, e os clusters são divididos em dois.

## 4.6 Fusão de Clusters

A fusão dos clusters ocorre de maneira similar ao processo de divisão visto na seção anterior. A fusão de cluster é um processo necessário para lidar com situações como a ilustrada na Figura 4.12, em que veículos longos e com regiões homogêneas, sem a presença de muita textura, não apresentam muitas partículas detectadas e também são gerados vetores de movimento diferentes ao longo do veículo. Isso faz com que sejam associados mais de um cluster a um mesmo veículo. A Figura 4.14 ilustra o processo de fusão de clusters.

Todos os clusters são testados, aos pares, para verificar se eles precisam ser fundidos. Assim, inicialmente é verificado se a distância euclidiana entre os centroides dos dois clusters testados,  $Q_i$  e  $Q_j$ , é menor que a soma das distâncias máximas das partículas para os centroides de seus respectivos clusters, com  $i = \{1, \dots, nc - 1\}$ ,  $j = \{i + 1, \dots, nc\}$  e  $i \neq j$ , como indicado abaixo:

$$D_E(\vec{c}_i, \vec{c}_j) < maxdist(\vec{u}_i, \vec{c}_i) + maxdist(\vec{u}_j, \vec{c}_j), \quad (4.15)$$

onde  $D_E(\vec{c}_i, \vec{c}_j)$  é a distância euclidiana entre os centroides dos clusters  $Q_i$  e  $Q_j$ ,  $\vec{c}_i$  e  $\vec{c}_j$ , respectivamente, que é dada pela Equação 4.16;  $maxdist(\vec{u}_i, \vec{c}_i)$  é a distância máxima de uma partícula  $\vec{u}_i$  no cluster  $Q_i$  para o seu centroide  $\vec{c}_i$ , e  $maxdist(\vec{u}_j, \vec{c}_j)$  é a distância máxima de uma partícula  $\vec{u}_j$  no cluster  $Q_j$  para o seu centroide,  $\vec{c}_j$ , indicado na Equação 4.17.

$$D_E(\vec{c}_i, \vec{c}_j) = \sqrt{(\vec{c}_i - \vec{c}_j)^2}, \quad (4.16)$$

$$maxdist(\vec{u}, \vec{c}) = \max_{k=1, \dots, n} \left( \sqrt{(u_k - \vec{c})^2} \right), \quad (4.17)$$

onde  $\vec{u} = (u_1, u_2, \dots, u_n)$  representa o conjunto de partículas no cluster  $Q$ , e  $\vec{c}$  o centroide do cluster  $Q$ .

Se o teste da Equação 4.15 for verdadeiro para os clusters  $Q_i$  e  $Q_j$ , o seguinte teste é realizado para determinar se os clusters serão unificados:

Seja  $E_{conv} = envoltoria\_convexa(Q_i \cup Q_j)$  a menor área convexa contendo todas as partículas de  $Q_i \cup Q_j$ , como ilustrado na Figura 4.14b, e  $AF = E_{conv} \cap MF$ ,  $AF \subseteq E_{conv}$  a área de intersecção entre a máscara de *foreground* ( $MF$ ) e a  $E_{conv}$ , isto é, a área de *foreground* contida em  $E_{conv}$ , como ilustrado na Figura 4.14c.

Se  $\#\{AF\}/\#\{E_{conv}\} > T_{fund}$  ( $T_{fund} = 0.85$  neste trabalho), então  $Q_{max(i,j)}^{t+1} = Q_i^t \cup Q_j^t$  e  $Q_{min(i,j)}^{t+1} = \emptyset$ , onde  $max(i, j)$  e  $min(i, j)$  são os índices máximo e mínimo entre  $i$  e  $j$ , respectivamente,  $\#\{AF\}$  e  $\#\{E_{conv}\}$  são o número de pixels nas regiões  $AF$  e  $E_{conv}$ , respectivamente. Isto é, se a área de  $AF$  corresponde a pelo menos 85% da área total da área de  $E_{conv}$ , os clusters  $Q_i$  e  $Q_j$  são unificados em um único cluster (Figura 4.14e). O valor do limiar  $T_{fund}$  é baseado em uma razão, e também deve funcionar bem com vídeos com diferentes resoluções e distâncias em relação à câmera, mas pode ser ajustado caso necessário. Esse limiar indica a área mínima aceitável de  $E_{conv}$  que deve corresponder ao *foreground* para que os clusters sejam fundidos. Conforme esse limiar aumenta, mais área de *foreground* precisará existir na região  $E_{conv}$  para que os clusters sejam fundidos. A Figura 4.13d mostra as duas áreas,  $E_{conv}$  e  $AF$ , sobrepostas, permitindo ver a área do cluster que corresponde a um objeto detectado na máscara de *foreground*,  $AF$ , em branco, correspondendo a mais de 85% da área de  $E_{conv}$ , em azul.

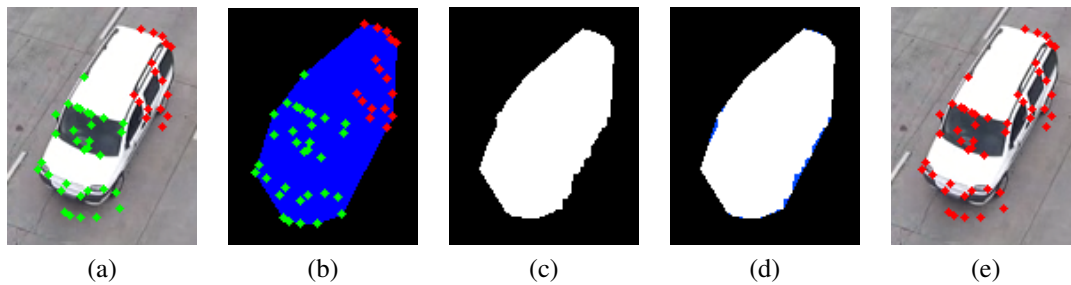


Figura 4.14: Exemplo de fusão de clusters. (a) partículas sobrepostas ao quadro processado que foram associadas a dois clusters. (b) área convexa ( $E_{conv}$ ) ao redor das partículas em (a); (c) área de *foreground* ( $AF$ ) contida em  $E_{conv}$ ; (d) área de *foreground* ( $AF$ ), em branco, intersectando  $E_{conv}$ , em azul; (e) resultado da fusão dos cluster. Como podemos observar, a razão entre as áreas  $AF$  e  $E_{conv}$  é maior que o limiar de fusão, e os clusters são unificados.

## 4.7 Detecção de Veículos

O agrupamento das partículas é apenas o passo inicial para a detecção dos veículos. Para realizar a detecção dos veículos, são utilizados procedimentos que utilizam os agrupamentos de partículas e a máscara binária de *foreground* para verificar a forma de cada cluster e a correspondência do cluster com um objeto detectado na máscara binária, com a finalidade de determinar se cada um dos agrupamentos realmente corresponde a veículos na cena. A Figura 4.15 e a Figura 4.16 ilustram o processo de detecção de veículos, que será descrito a seguir.

No processo de detecção de veículos é verificado a área do cluster, que deve ser grande o bastante para corresponder a um veículo, a taxa de ocupação, que determina a distribuição das partículas dentro do cluster, e a razão entre o eixo maior e o eixo menor do cluster,

pois os veículos possuem um formato retangular, mantendo certa proporção entre os dois eixos. Também são verificados os objetos do *foreground* detectados na máscara binária nas coordenadas que correspondem às regiões dentro e ao redor da área do cluster, sendo que a região interna do cluster deve possuir uma grande área de intersecção com os objetos detectados na máscara binária e a região externa uma pequena área de intersecção com os objetos da máscara, pois a maior parte do objeto deve estar na região interna do cluster.

Para todos os clusters  $Q_i, i = \{1, \dots, nc\}$ , onde  $nc$  é o número total de clusters, são realizados testes para verificar se o cluster  $Q_i$  corresponde a um veículo. Os testes são realizados de acordo com as definições dadas a seguir. Sejam  $E_{conv} = envoltoria\_convexa(Q_i)$  a menor área convexa contendo todas as partículas de  $Q_i$ , como ilustrado na Figura 4.15b e na Figura 4.15c,  $C_{dilatadas} = VIZ(\vec{u}_{ij}) \oplus B$  a vizinhança das partículas  $\vec{u}_{ij}, j = \{1, \dots, n\}$  no cluster  $Q_i$  dilatada por um elemento estruturante  $B$  de formato circular e raio 5 (Figura 4.15d),  $CD_{env} = envoltoria\_convexa(C_{dilatadas})$  a menor área convexa contendo toda a vizinhança dilatada das partículas em  $Q_i$ , como ilustrado na Figura 4.15e,  $E_{maior} = eixo\_maior(E_{conv})$  e  $E_{menor} = eixo\_menor(E_{conv})$  o comprimento, em pixels, do maior e menor eixo da região  $E_{conv}$ , respectivamente, como ilustrado na Figura 4.15f,  $AF_{int} = E_{conv} \cap MF$  a correspondente área de *foreground* da intersecção entre  $E_{conv}$  e  $MF$  (a máscara de *foreground*), como indicado na Figura 4.16c, e  $AF_{ext} = MF - E_{conv}$  a correspondente área de *foreground* ao redor da região  $E_{conv}$ , exibido na Figura 4.16d.

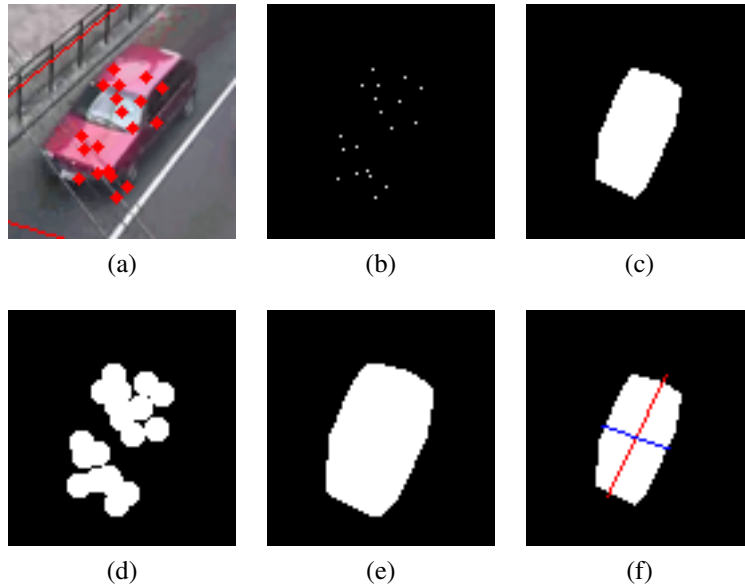


Figura 4.15: Exemplo de análise da forma do cluster. (a) cluster analisado; (b) máscara com as partículas do cluster analisado; (c) área convexa contendo as partículas em (b) para obter a área do cluster ( $E_{conv}$ ); (d) partículas dilatadas usadas para obter a taxa de ocupação  $C_{dilatadas}$ ; (e) área convexa contendo as partículas dilatadas ( $CD_{env}$ ), a taxa de ocupação  $C_{dilatadas}$  é calculada como a área das partículas dilatadas (d) dividida por essa área convexa (e); (f) o eixo maior ( $E_{maior}$ ) dividido pelo eixo menor ( $E_{menor}$ ) deve ser menor que um limiar  $T_{eixos}$ .

Os testes possuem uma pontuação associada, que será acumulada durante o processo. Quando um teste resultar em um valor verdadeiro, a pontuação é adicionada, e quando

resultar em falso, a pontuação do teste é subtraída. Os seguintes testes são realizados para verificar se o cluster  $Q_i$  corresponde a um veículo:

- $T_{areamin} < \#\{E_{conv}\} < T_{areamax}$ , o valor da área (número de pixels) de  $E_{conv}$  deve estar dentro de um intervalo de valores máximo e mínimo de área para ser considerado um veículo (esse intervalo varia de acordo com a resolução do vídeo e a distância da câmera para os veículos na cena). (pontuação: 1)
- $\#\{CD_{env}\}/\#\{C_{dilatadas}\} > T_{ocup}$ , a taxa de ocupação fornece a distribuição das partículas dentro do cluster, indicando se o cluster está densamente ocupado (mais provável de corresponder a um veículo) ou tem partículas mais esparsas (menos provável de corresponder a um veículo).  $T_{ocup} = 0.8$ , neste trabalho. (pontuação: 1)
- $E_{maior}/E_{menor} < T_{eixos}$ , verifica a razão entre o eixo maior e o eixo menor do cluster, onde  $T_{eixos} = 3$  neste experimento. (pontuação: 1)
- $\#\{AF_{int}\}/\#\{E_{conv}\} > T_{int}$ , para corresponder a um veículo, a parte interna desta área convexa ( $E_{conv}$ ) deve ter uma grande área de intersecção com a região do primeiro plano ( $AF_{int}$ ).  $T_{int} = 0.9$  neste experimento. (pontuação: 1)
- $\#\{AF_{ext}\}/\#\{E_{conv}\} < T_{ext}$ , para corresponder a um veículo, a parte ao redor da região  $E_{conv}$  deve ser pequena, o que significa que  $E_{conv}$  cobre a maior parte do objeto móvel na máscara de primeiro plano.  $T_{ext} = 0.1$  neste trabalho. (pontuação: 1)

Os valores dos limiares de área,  $T_{areamin}$  e  $T_{areamax}$  podem variar de acordo com a resolução do vídeo e a distância da câmera para os veículos na cena. Para ajustar esses valores para outros vídeos, deve-se ter  $T_{areamin}$  como o menor valor de área (menor número de pixels) ocupado pelos menores veículos, e  $T_{areamax}$  como o maior valor de área (maior número de pixels) ocupado por veículos grandes. A razão entre os eixos maior e menor é um valor fixo, e, neste trabalho, o valor três, indicando que um veículo pode ter o seu comprimento até três vezes maior que a sua largura, obteve bons resultados. Caso o vídeo a ser processado apresente veículos com dimensões atípicas deve-se ajustar para um valor apropriado. Os outros valores de limiares presentes nos testes acima são baseados em porcentagens, com valores variando de zero a um, com valores próximos a um indicando que os valores dessas proporções devem ser altos (próximos de 100%), e valores próximos à zero indicando que devem ser baixos (próximos de 0%). Portanto, devem funcionar bem com outros vídeos com diferentes resoluções e distâncias em relação à câmera, mas podem ser ajustados caso necessário, preferencialmente escolhendo limiares acima de 0.7 para os valores que possuem um valor de proporção alto, e abaixo de 0.3 para os o valor de proporção deve ser baixo.

Ao final de todos os testes, se a pontuação acumulada for maior que zero, o cluster  $Q_i$  é considerado como associado a um veículo, passando, portanto, a ser denotado por  $V_i$ . Cada um dos clusters associados aos veículos detectados são envolvidos por um retângulo e passam a ter um número de identificação único, como exibido na Figura 4.17.

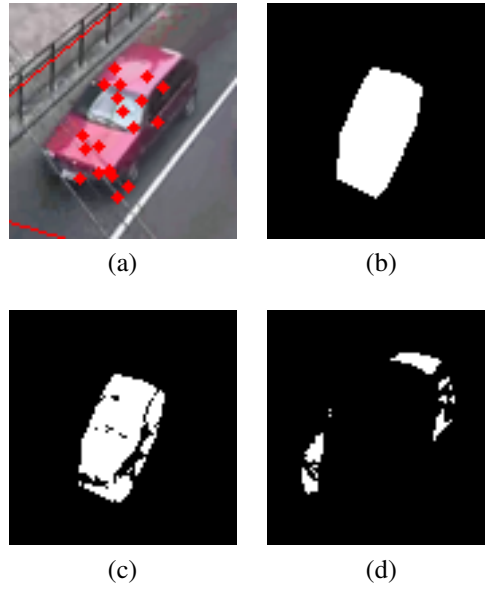


Figura 4.16: Exemplo de análise da máscara de *foreground* na região do cluster. (a) cluster analisado; (b) área convexa ( $E_{conv}$ ) das partículas no cluster analisado; (c) área do *foreground* ( $AF_{int}$ ) intersectando a área correspondente à parte interna de  $E_{conv}$ ; (d) área do *foreground*  $AF_{ext}$  correspondente à parte externa ao redor de  $E_{conv}$ .



Figura 4.17: Exemplo veículo detectado.

## 4.8 Rastreamento de Veículos

O rastreamento de veículos é realizado somente nos clusters que correspondem a veículos ( $V_i$ ), ou seja, aqueles que foram identificados como associados aos veículos, como visto na seção anterior. O rastreamento é baseado na similaridade entre histogramas de cor, discutido na Seção 3.8, computados para pequenas janelas de pixels centradas em cada uma das partículas associadas com a área convexa formada pelos veículos ( $V_i$ ) no  $quadro_{t-1}$  e nas partículas no  $quadro_t$ , cujas posições são dadas de acordo com os vetores de movimento de cada partícula em  $V_i$ . Neste trabalho foram usadas janelas de pixels de tamanho  $9 \times 9$ , em vídeos com resoluções ou distâncias da câmera em relação aos veículos diferentes dos utilizados neste trabalho, poderá ser necessário modificar o tamanho desta janela de pixels, de modo que a janela de pixels seja grande o suficiente para abranger detalhes ao redor da partícula, mas sem abranger a região ao redor dos veículos quando a partícula estiver nas bordas.

Sejam  $V_i^{t-1} = \{u_{i,1}^{\vec{t-1}}, \dots, u_{i,npp}^{\vec{t-1}}\}$  os clusters que correspondem a veículos no quadro anterior no tempo  $t-1$ ,  $V_i^t = \{u_{i,1}^{\vec{t}}, \dots, u_{i,npc}^{\vec{t}}\}$  os clusters que correspondem a veículos no quadro atual (tempo  $t$ ), com  $i = \{1, \dots, nv\}$ , onde  $\{u_{i,1}^{\vec{t-1}}, \dots, u_{i,npp}^{\vec{t-1}}\}$  e  $\{u_{i,1}^{\vec{t}}, \dots, u_{i,npc}^{\vec{t}}\}$  são as partículas dos veículos no  $quadro_{t-1}$  ( $V_i^{t-1}$ ) e no  $quadro_t$  ( $V_i^t$ ), respectivamente,  $nv$  é o número de clusters que correspondem a veículos,  $npp$  e

$npc$  são os números de partículas em  $V_i^{t-1}$  e  $V_i^t$ , respectivamente. Então, janelas de pixels centradas nas posições dadas pelas coordenadas das partículas são extraídas do *quadro* <sub>$t-1$</sub> , denotada  $W_{i,p} = janela(u_{i,p}^{t-1})$ , e do *quadro* <sub>$t$</sub> , denotada  $W_{i,c} = janela(u_{i,c}^t)$ , com  $p = \{1, \dots, npp\}$  e  $c = \{1, \dots, npc\}$ , onde  $W_{i,p} = janela(u_{i,p}^{t-1})$  e  $W_{i,c} = janela(u_{i,c}^t)$  fornecem as janelas de pixels centradas em cada uma das coordenadas das partículas associadas com a área convexa formada pelos veículos no quadro anterior ( $V_i^{t-1}$ ) e no quadro atual ( $V_i^t$ ),  $npp$  e  $npc$  indicam o número de partículas nos clusters  $V_i^{t-1}$  e  $V_i^t$ , respectivamente.

A seguir, é obtido o histograma de cor para cada uma das janelas de pixels. Para isso, é realizada uma requantização uniforme da imagem, reduzindo as 256 cores em cada canal RGB da imagem para 8 cores em cada canal, ou seja, cada canal é dividido em 8 intervalos iguais, resultando em uma imagem final com 512 cores. Assim, é obtido o histograma de cor normalizado para cada janela de imagem no quadro anterior ( $W_{i,p}$ ), denotado  $H_{i,p} = hist(W_{i,p})$  e a janela correspondente no quadro atual ( $W_{i,c}$ ), denotado  $H_{i,c} = hist(W_{i,c})$ . Os histogramas extraídos da vizinhança de cada partícula no *quadro* <sub>$t$</sub>  são então comparadas com os histogramas dessas partículas no *quadro* <sub>$t-1$</sub>  usando o coeficiente de Bhattacharyya (COMANICIU; RAMESH; MEER, 2000), apresentado na Seção 3.8 e indicado na Equação 4.18. Um valor alto de similaridade (acima de 0,8 neste trabalho) indica que a partícula é mesma em ambos os quadros, já um valor baixo, indica que são partículas diferentes, ou seja, que a partículas do quadro atual não corresponde a partículas do frame anterior, sendo então descartada do rastreamento.

$$s_{hist}(H_{i,p}, H_{i,c}) = \sum_{j=1}^L \sqrt{(H_{i,p})_j (H_{i,c})_j}, \quad (4.18)$$

onde  $H_{i,p}$  é o histograma extraído da vizinhança da partícula  $u_{i,p}^{t-1}$  no quadro anterior no tempo  $t-1$ ,  $H_{i,c}$  o histograma extraído da vizinhança da partícula correspondente  $u_{i,c}^t$  no quadro atual no tempo  $t$ ,  $L$  é o número de intervalos de cores do histograma (512).

As partículas no cluster  $V_i$  no tempo  $t$  cujas similaridades com partículas no tempo  $t-1$ , em termos de histogramas de cor, forem menores que um limiar de similaridade  $T_{Sim}$  ( $T_{Sim} = 0.8$  neste trabalho) são descartadas. Um novo cluster de partículas, nomeado  $T_i = \{u_{i,1}^{\vec{t}}, \dots, u_{i,nt}^{\vec{t}}\}$ , onde  $nt$  é o número de veículos rastreados, é obtido com as partículas do cluster de partículas  $V_i$  que não foram eliminadas, isto é,  $T_i$  contém as partículas de  $V_i$  que não foram descartadas no teste  $s_{hist}(H_{i,p}, H_{i,c}) > T_{Sim}$ . Este novo cluster de partículas,  $T_i$ , significa que o cluster de partículas associado a um veículo detectado no tempo  $t-1$  corresponde ao mesmo veículo no tempo  $t$ , e é portanto um veículo rastreado. Como as posições das partículas entre dois quadros consecutivos são dadas pelos vetores de movimento, sobreposições podem ocorrer entre dois clusters  $T_i$  e  $T_j$ . Para determinar a qual cluster as partículas na região de sobreposição pertencem, cada partícula na região de sobreposição é testada com a finalidade de distribuir elas para um dos cluster,  $T_i$  ou  $T_j$ , e isso é realizado utilizando a distância de Mahalanobis, apresentada na Equação 4.19, e cada partícula  $u_{i,k}^{\vec{t}}$ , onde  $k$  são os índices das partículas na região de sobreposição, é atribuída ao cluster considerado como o mais próximo dessa partícula  $u_{i,k}^{\vec{t}}$ .

$$Md(\vec{u}) = \sqrt{(\vec{u} - \vec{\mu}_T)^T \Sigma_u^{-1} (\vec{u} - \vec{\mu}_T)} \quad (4.19)$$

onde  $\vec{u}$  representa a partícula no cluster de partículas associado com um veículo  $T$ ,  $\vec{\mu}_T$  é o vetor de médias das partículas no cluster  $T$ ,  $\Sigma_u$  é a matriz de covariância das partículas

no cluster  $T$ , e  $T$  indica transposição.

A Figura 4.18 mostra um exemplo de similaridade de histograma de cor entre duas partículas, uma no  $quadro_{t-1}$  e outra no  $quadro_t$ , onde  $W_{i,p}$  (Figura 4.18a) e  $W_{i,c}$  (Figura 4.18b) são as janelas de imagem de  $9 \times 9$  pixels centradas em cada uma dessas partículas. Podemos observar pelo gráfico que o histograma da imagem  $W_{i,p}$  ( $H_{i,p}$ ), que pode ser observado na Figura 4.18c em azul, não é muito similar ao histograma da imagem  $W_{i,c}$  ( $H_{i,c}$ ), que pode ser observado na Figura 4.18d em vermelho. A Figura 4.18e exibe os histogramas  $H_{i,p}$  e  $H_{i,c}$  sobrepostos, onde podemos observar facilmente que eles não possuem grande similaridade, possuindo, de acordo com a Equação 4.18, um valor de similaridade  $s_{hist}(H_{i,p}, H_{i,c}) = 0,6631$ , que é menor que o limiar de similaridade  $T_{Sim} = 0,8$ , sendo, portanto, consideradas partículas diferentes. Na Figura 4.19 temos um exemplo de similaridade de histograma de cor entre outras duas partículas, uma no  $quadro_{t-1}$  e outra no  $quadro_t$ , onde  $W_{i,p}$  (Figura 4.19a) e  $W_{i,c}$  (Figura 4.19b) são as janelas de imagem de  $9 \times 9$  pixels centradas nessas partículas. Podemos observar pelo gráfico que o histograma da imagem  $W_{i,p}$  ( $H_{i,p}$ ), que pode ser observado na Figura 4.19c em azul, não é muito similar ao histograma de  $W_{i,c}$  ( $H_{i,c}$ ), que pode ser observado na Figura 4.19d em vermelho. A Figura 4.19e exibe os histogramas  $H_{i,p}$  e  $H_{i,c}$  sobrepostos, onde podemos observar que agora eles possuem grande similaridade, de acordo com a Equação 4.18, um valor de similaridade  $s_{hist}(H_{i,p}, H_{i,c}) = 0,9948$ , que é maior que o limiar de similaridade  $T_{Sim} = 0,8$ .



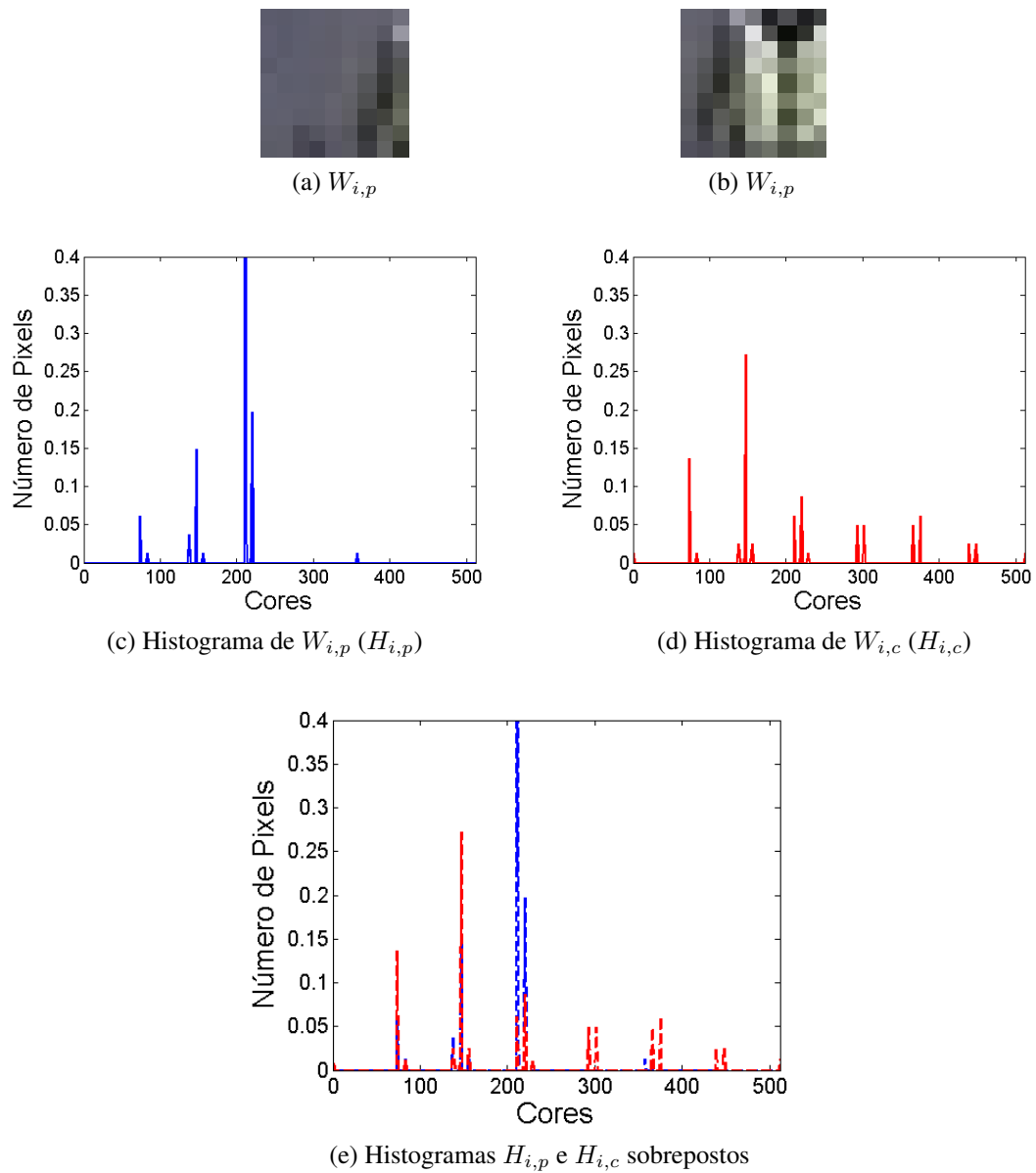


Figura 4.18: Exemplo de similaridade entre histogramas de cor. Neste exemplo  $H_{i,p}$  e  $H_{i,c}$  não são muito similares de acordo com a Equação 3.13, possuindo um valor de similaridade de  $s_{hist}(H_{i,p}, H_{i,c}) = 0,6631$ .

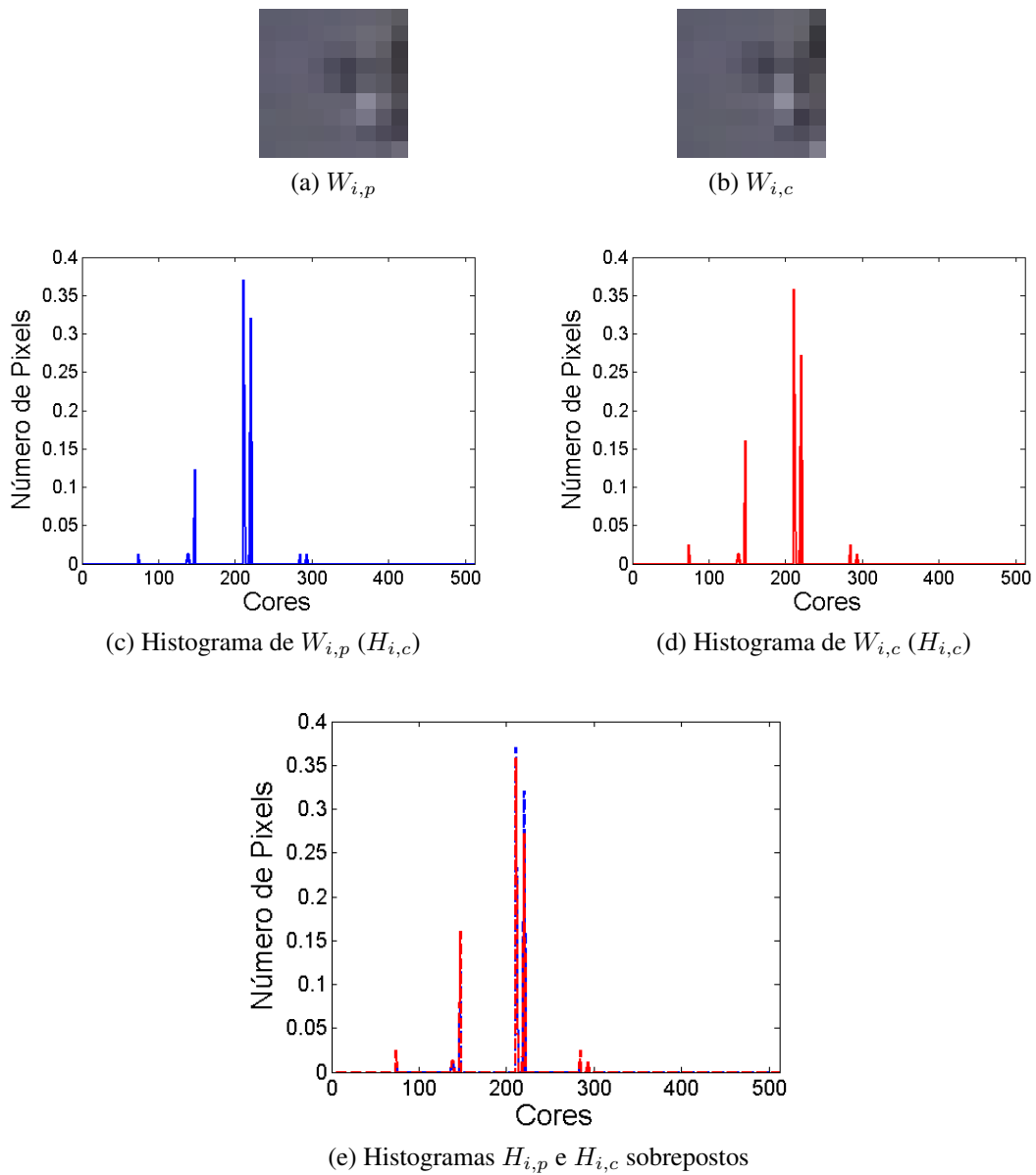


Figura 4.19: Exemplo de similaridade entre histogramas de cor. Neste exemplo  $H_{i,p}$  e  $H_{i,c}$  são similares de acordo com a Equação 3.13, possuindo um valor de similaridade de  $s_{hist}(H_{i,p}, H_{i,c}) = 0,9948$ .

## 4.9 Contagem de Veículos

Nesta etapa, é realizada a contagem dos veículos. Somente os agrupamentos que foram detectados como veículos e rastreados serão contados, os agrupamentos que não foram identificados como veículos não serão contados. Os veículos são contados somente quando passam sobre laços virtuais definidos pelo usuário, que podem ser definidos em qualquer posição dentro da área de interesse, como pode ser observado na Figura 4.20. No entanto, uma vez que apenas os veículos detectados e rastreados são contados, o melhor lugar para incluirmos um laço virtual é na área onde os veículos estão deixando a cena. Dessa maneira, as chances de um veículo ser contado aumentam, pois o veículo terá sido detectado e rastreado ao longo de um maior número de quadros antes de atingir o laço virtual.

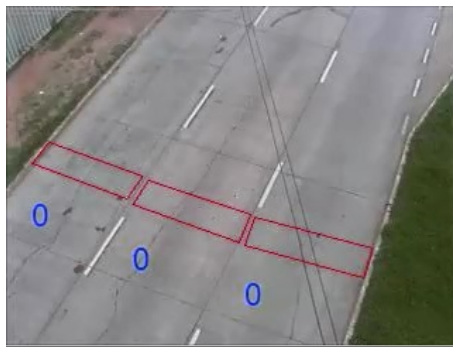


Figura 4.20: Exemplo de três laços virtuais posicionados sobre as pistas.

A Figura 4.21 exemplifica o processo de incremento do contador de veículos do laço virtual. A contagem dos veículos é baseada na intersecção das regiões convexas formadas pelas partículas rastreadas associadas a um veículo com as regiões poligonais que definem os laços virtuais. Isso pode ser definido formalmente da seguinte maneira: Seja  $CT = envoltoria\_convexa(T_i)$  a menor área convexa contendo todas as partículas do cluster associado a um veículo,  $T_i$ ,  $i = \{1, \dots, nt\}$ , onde  $nt$  é o número de veículos rastreados, e  $Laço_L = \{Laço_1, \dots, Laço_{nLaços}\}$ , que representa as regiões dos laços virtuais, onde  $nlaços$  é o número de laços virtuais. O incremento do laço virtual é realizado de acordo com  $argmax_L(Laço_L \cap CT)$ . Isto é, o laço virtual ( $Laço_L$ ) que tiver a maior área de intersecção com um grupo de partículas rastreadas (veículos)  $CT$  tem o seu contador incrementado, e essas partículas rastreadas não ativarão outros laços virtuais. Cada agrupamento de partículas rastreado e detectado como correspondente a um veículo é contado apenas uma vez, mesmo quando passam sobre mais de um laço virtual ao mesmo tempo, como ilustrado na Figura 4.22. Na Figura 4.22c podemos ver que as partículas correspondentes ao veículo sobre os laços na Figura 4.22a estão sobre os laços 1 e 2. Após ser obtida a envoltória convexa dessas partículas ( $CT$ ), ilustrado na Figura 4.22d, é verificado com qual laço  $CT$  tem a maior área de intersecção (Figura 4.22e). Como podemos observar, laço virtual com maior intersecção com  $CT$  é o  $Laço_3$ , que é então ativado e tem o seu contador incrementado (Figura 4.22f).

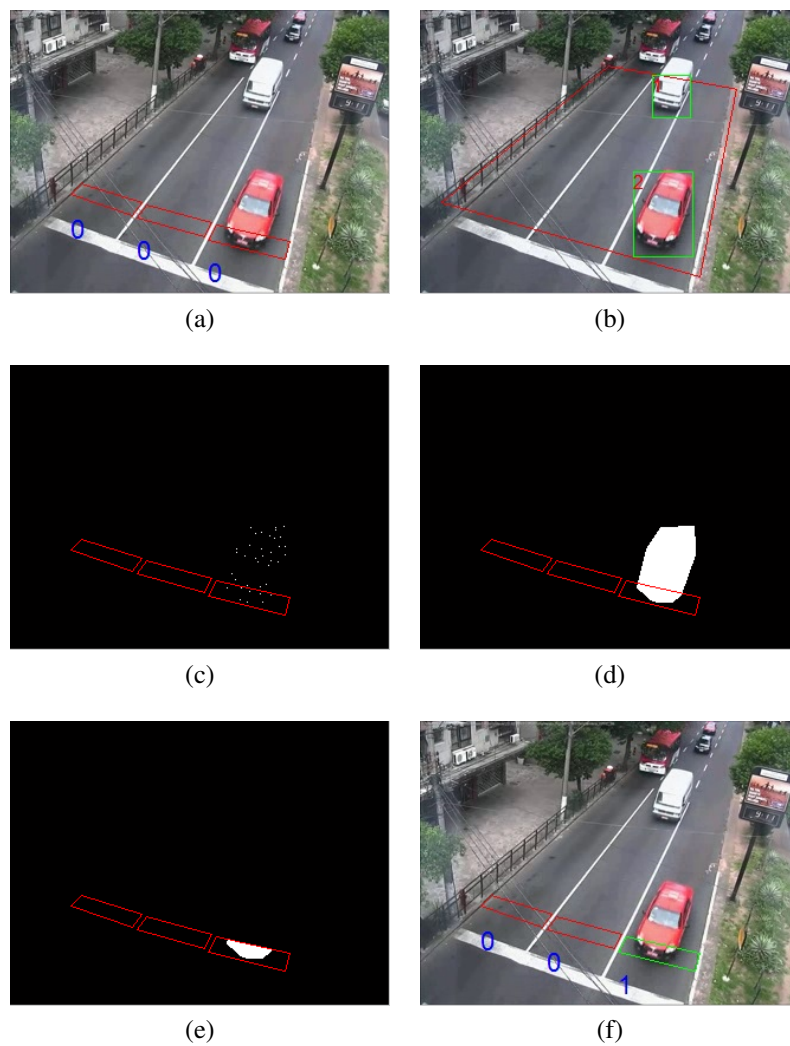


Figura 4.21: Ilustração do processo de incremento do contador de veículos do laço virtual. (a) veículo entrando no laço virtual; (b) retângulo envolvendo o veículo detectado; (c) partículas correspondentes ao veículo rastreado; (d) região convexa formada pelas partículas ( $CT$ ); (e) área de intersecção entre a região convexa correspondente ao veículo ( $CT$ ) com o laço virtual definido pelo usuário ( $Laço_3$ ); (f) ativação do laço virtual com maior área de intersecção, o laço muda de cor para indicar que foi ativado e o contador é incrementado.

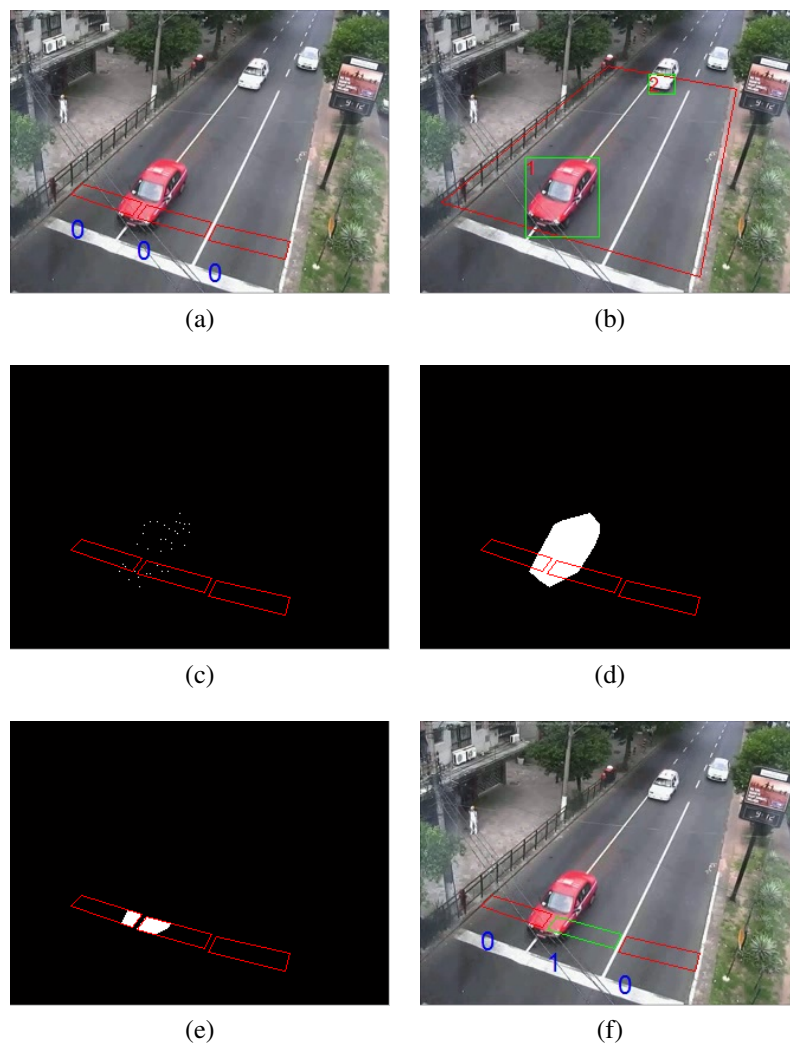


Figura 4.22: Ilustração do processo de incremento do contador de veículos do laço virtual. (a) veículo entrando no laço virtual; (b) caixa envolvendo o veículo detectado; (c) partículas correspondentes ao veículo rastreado; (d) região convexa formada pelas partículas ( $CT$ ); (e) área de intersecção entre a região convexa correspondente ao veículo ( $CT$ ) com os laços virtuais ( $Laço_1$  e  $Laço_2$ ); (f) ativação do laço virtual com maior área de intersecção ( $Laço_1$ ).

## 5 RESULTADOS EXPERIMENTAIS

Este trabalho consiste no desenvolvimento de um sistema automático para realizar a contagem de veículos em tráfego urbano por câmeras de vídeos utilizando técnicas de processamento de imagens e visão computacional. Para isso, são utilizadas partículas, obtidas através de um método de detecção de cantos em imagens, que são agrupadas de maneira que cada um dos agrupamentos formados corresponda a um veículo presente no vídeo, empregando métodos para fazer a correta detecção dos veículos e também o rastreamento, que é essencial para garantir que um mesmo veículo seja contado apenas uma vez. Essa contagem é realizada quando os veículos passam sobre marcações na pista que foram definidas pelo usuário, chamadas de laços virtuais.

Esta seção apresenta os resultados obtidos com o método proposto, sendo também realizada uma comparação com dois outros métodos encontrados na literatura, um com abordagem semelhante, proposto por Kim (2008), que combina subtração de fundo, rastreamento de partículas (feições) e algoritmos de agrupamento, e outro baseado na obtenção de objetos de interesse e rastreamento de componentes conectados, proposto por Sanchez et al. (2011), apresentados na Seção 2.3. Esses métodos foram escolhidos por serem similares ao método proposto, utilizando partículas (KIM, 2008) e uma máscara que detecta os objetos em movimento no primeiro plano do vídeo (SANCHEZ et al., 2011). Para gerar os resultados das comparações foram utilizados um aplicativo do método proposto por Kim, disponibilizado pelo autor, e uma implementação em Matlab para o método proposto Sanchez et al., implementado de acordo com o artigo.

Uma das vantagens em relação aos métodos comparativos é a utilização de uma máscara de *foreground* mais robusta, o que faz com que os objetos em movimento na cena sejam identificados de maneira mais precisa. Outra vantagem em relação aos métodos comparativos é a realização de etapas de verificação dos veículos, que verifica se cada agrupamento formado corresponde a um veículo, e também as etapas de fusão e divisão de clusters, que corrige falhas que possam ocorrer no agrupamento inicial das partículas, atribuindo as partículas para os agrupamentos corretos. A etapa de rastreamento das partículas, que leva em consideração as posições estimadas pelos vetores de movimento, também garante que em situações onde o tráfego está parado ou lento o método ainda consiga manter um rastreamento razoável dos veículos, evitando que sejam contados como novos veículos quando o movimento reiniciar.

Como mencionado anteriormente, a obtenção das máscaras binárias de *foreground*, utilizando os métodos de modelo de misturas de gaussianas (GMM) e Motion Energy Images (MEI), foram implementadas utilizando a linguagem de programação C/C++, juntamente com a biblioteca multiplataforma OpenCV. Após a obtenção das máscaras, são executadas as etapas seguintes do método, que foram implementadas utilizando a plataforma Matlab. Os vídeos utilizados nos experimentos e comparações foram adquiridos de

seis diferentes câmeras, cedidos pela Empresa Pública de Transporte e Circulação (EPTC) de Porto Alegre - RS, com resolução originalmente variando entre  $1280 \times 960$  e  $640 \times 480$  pixels, que foram redimensionados para  $320 \times 240$  pixels para reduzir o tempo de processamento, com cada vídeo possuindo em média 8 minutos de duração e taxa de 25 quadros por segundo, totalizando mais de 80 mil quadros, e consistem em vídeos diurnos com iluminação relativamente uniforme, obtidos através de câmeras fixas posicionadas acima da via, de modo que o fluxo de veículos ocorra na direção vertical, sendo que na parte superior do vídeo os veículos estão mais afastados da câmera, e na parte inferior mais próximos da câmera. Serão apresentados exemplos de detecção e rastreamento para cada um dos vídeos, juntamente com os seus respectivos resultados e comparações.

O tempo de processamento do método é em média 8 segundos por quadro. Isso se deve ao fato de estar implementado em Matlab. As etapas de maior complexidade são as etapas de fusão/divisão de clusters e de detecção dos veículos. Nessas etapas, é necessário verificar todos os agrupamentos existentes na cena, e também envolve o cálculo de distâncias entre partículas e entre cluster. Com a otimização do código e a utilização de uma plataforma mais rápida que o Matlab, como C/C++, por exemplo, é esperado que o método possa executar em tempo real, sendo que muitos dos testes e verificações realizados em cada uma das etapas não precisam executar sequencialmente, como está sendo executado neste trabalho.

Para cada vídeo é apresentado uma tabela contendo o número de veículos detectados sobre os laços virtuais em cada uma das pistas da via para o método proposto e para os métodos de comparação. Em seguida, são apresentadas outras duas tabelas, uma com o número de veículos que não foram detectados e outra com o número de veículos que foram contados mais de uma vez. Por fim, é exibida uma tabela com o resultado geral, considerando os vídeos de todas as câmeras. A Seção 5.1 apresenta a discussão a respeito dos resultados obtidos, sendo descritas também as situações em que ocorreram falhas no método proposto e as possíveis correções que poderiam ser realizadas.

## 5.1 Resultados e Discussão

A seguir, serão exibidos exemplos de cada uma das câmeras utilizadas nos testes. Serão exibidos quadros dos vídeos processados, a máscara de *foreground* obtida e as posições da região de interesse e dos laços virtuais de cada uma das câmeras. Serão apresentados também exemplos de rastreamento e contagem dos veículos, assim como falhas e outras situações que ocorreram em cada um dos vídeos.

Para cada uma das câmeras serão apresentadas três tabelas. A primeira mostrará os resultados obtidos para a contagem com o método proposto e também com os métodos comparativos. Os resultados são comparados com os valores da contagem real, obtidos através de inspeção visual. A tabela apresenta para cada uma das pistas a contagem de veículos, a diferença da contagem para o valor real, o erro em relação ao valor real e a porcentagem do erro. Esses valores também são apresentados para o total da via, ou seja, o somatório de todas as pistas, onde a porcentagem do erro é calculada como o número total de erros sobre o número real de veículos no vídeo.

Outras duas tabelas apresentam o número de veículos que não foram detectados em cada um dos métodos, e o número de veículos que foram detectados múltiplas vezes, ou seja, os veículos que acabaram sendo divididos e contados como se fossem mais de um veículo. São apresentados os valores totais e também os valores para cada uma das pistas, separadamente. Para melhor analisar os resultados, os veículos foram divididos em três

tipos: motocicletas (veículos pequenos), carros (veículos médios) e caminhões (veículos grandes). O valor entre parênteses representa a contagem real. Assim, é possível saber a quantidade de veículos de cada tipo nas pistas e quais os tipos de veículos que estão tendo problemas para serem contados.

A Figura 5.1a mostra um quadro de exemplo do vídeo Cam06, e na Figura 5.1b a sua respectiva máscara de *foreground*. A Figura 5.1c exibe as linhas, em vermelho, que delimitam a região de interesse. A ROI foi posicionada de maneira a cobrir todas as pistas da via. Cada uma das pistas recebeu um laço virtual, que foi posicionado dentro da área de interesse, cobrindo toda a extensão de cada pista da via, próximo do ponto onde os veículos deixam a área de interesse, como exibido na Figura 5.1d. Isso faz com que as chances do veículo ser contado corretamente aumente, pois ele já terá sido detectado e rastreado ao longo de vários quadros.

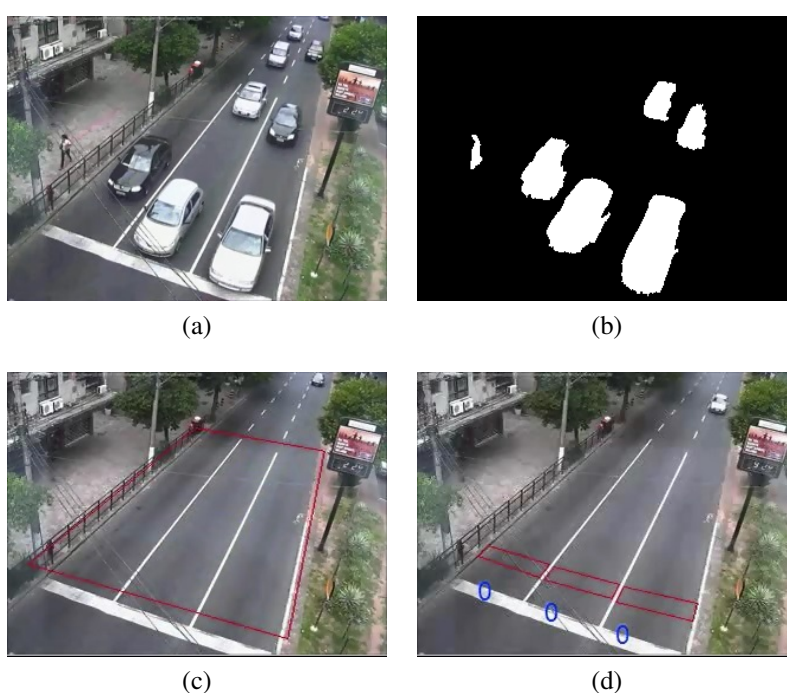


Figura 5.1: Vídeo Cam06. (a) uma sequência do vídeo; (b) máscara de *foreground*; (c) região de interesse; (d) laços virtuais.

A Figura 5.2a exibe um exemplo de rastreamento de veículos na Cam06. São exibidos quadros mostrando desde o momento que o veículo entra na região de interesse até o momento que ele sai. Cada veículo que está sendo rastreado possui um retângulo, em verde, ao seu redor, juntamente com um número identificador, em vermelho, que permanece o mesmo até o veículo sair da região de interesse. Na Figura 5.3 é ilustrado o processo de contagem dos veículos na Cam06. Quando o veículo passa sobre a região dos laços virtuais, o laço que tiver a maior área de intersecção com esse veículo é ativado, o que é indicado pela mudança de cor do laço, de vermelho para verde, e o valor do contador associado com o laço que foi ativado é incrementado.

A Tabela 5.1 mostra os resultados obtidos para a contagem dos veículos da Cam06 com o método proposto e também com os métodos comparativos.

A Tabela 5.2 apresenta o número de veículos que não foram detectados em cada um dos métodos, e a Tabela 5.3 apresenta o número de veículos que foram detectados múltiplas vezes, ou seja, os veículos que acabaram sendo divididos e contados como se fossem



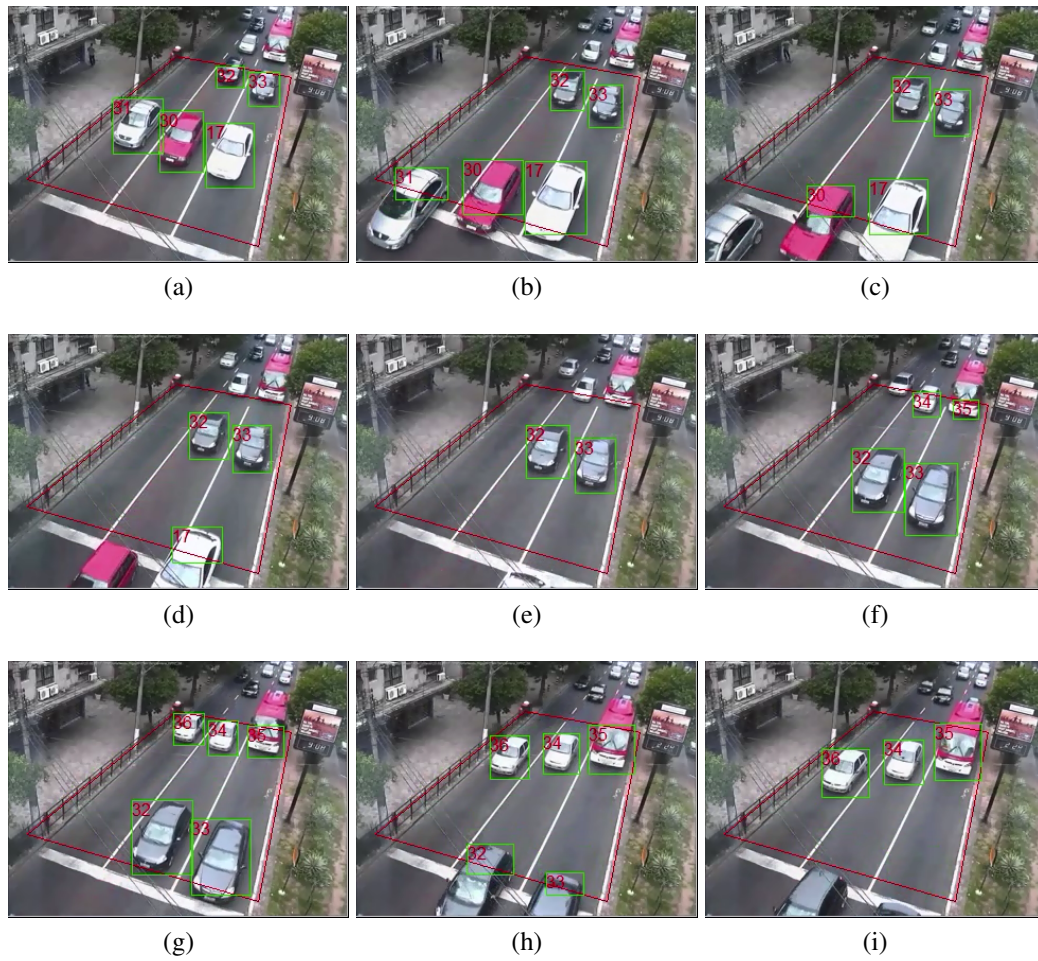


Figura 5.2: Exemplo de rastreamento na Cam06. (a) - (i) ilustram o rastreamento de veículos em uma sequência de quadros da Cam06.

Tabela 5.1: Contagem de veículos para a Cam06.

Método		Pista 1	Pista 2	Pista 3	Total
Inspecao Visual	Contagem	43	92	89	<b>224</b>
KIM	Contagem	82	103	80	<b>265</b>
	Diferença	39	11	-9	<b>41</b>
	Erro	39	11	9	<b>59</b>
	Erro (%)	90.70%	11.96%	10.11%	<b>26.34%</b>
SANCHEZ et al.	Contagem	51	100	96	<b>247</b>
	Diferença	8	8	7	<b>23</b>
	Erro	8	8	7	<b>23</b>
	Erro (%)	18.60%	8.70%	7.87%	<b>10.27%</b>
Método Proposto	Contagem	41	92	92	<b>225</b>
	Diferença	-2	0	3	<b>1</b>
	Erro	2	0	3	<b>5</b>
	Erro (%)	4.65%	0.00%	3.37%	<b>2.23%</b>

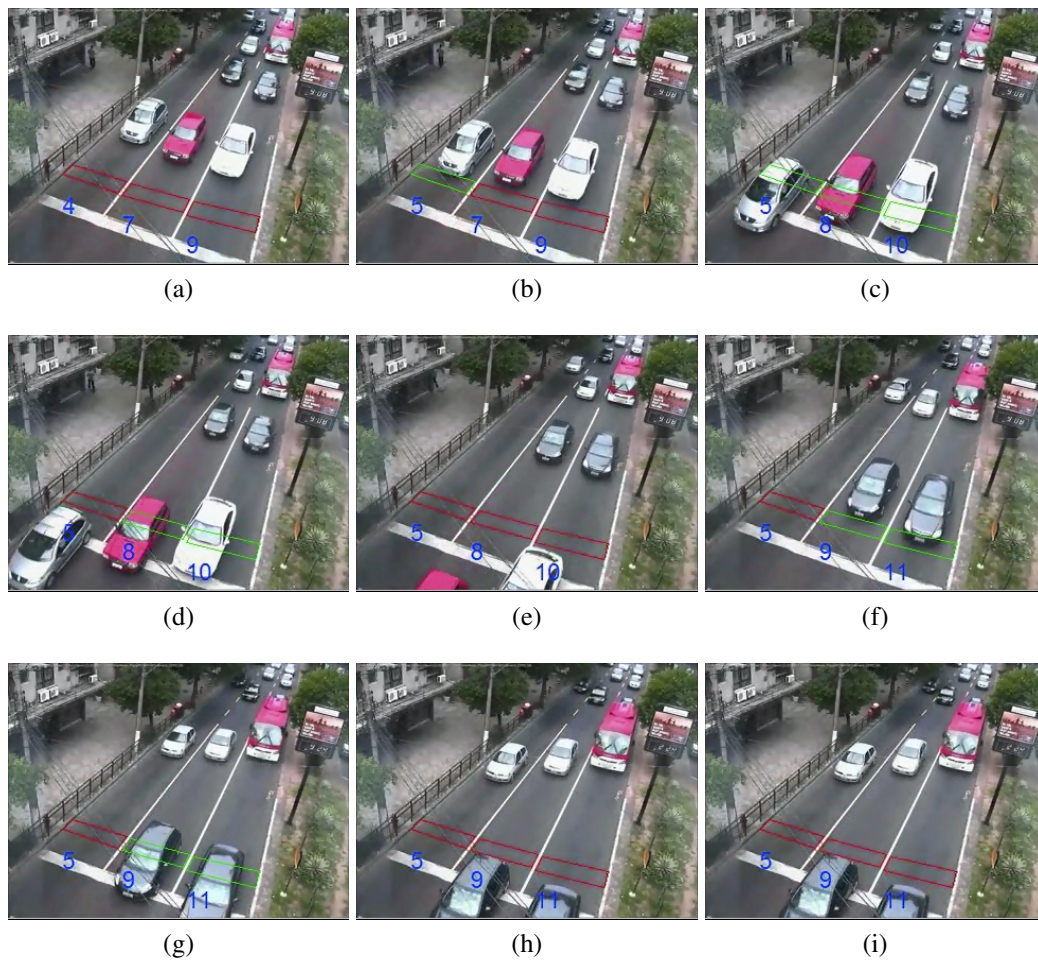


Figura 5.3: Exemplo do processo de contagem de veículos na Cam06. (a) - (i) exemplificam o processo de contagem dos veículos sobre os laços virtuais.

mais de um veículo. São apresentados os valores totais e também os valores para cada uma das pistas, separadamente. Os valores entre parênteses representam a contagem real para cada tipo de veículo.

Tabela 5.2: Veículos não detectados na Cam06.

Método	Tipo de Veículo	Pista 1	Pista 2	Pista 3	Total Veículos
KIM	Motocicletas	1 (5)	0 (3)	2 (3)	3
	Carros	0 (26)	4 (83)	9 (80)	13
	Caminhões	0 (12)	0 (6)	0 (6)	0
	Total	1	4	11	16
SANCHEZ et al.	Motocicletas	5 (5)	0 (3)	0 (3)	5
	Carros	3 (26)	5 (83)	9 (80)	17
	Caminhões	2 (12)	1 (6)	1 (6)	4
	Total	10	6	10	26
Método Proposto	Motocicletas	0 (5)	1 (3)	0 (3)	1
	Carros	2 (26)	2 (83)	1 (80)	5
	Caminhões	1 (12)	0 (6)	0 (6)	1
	Total	3	3	1	7

Tabela 5.3: Veículos detectados múltiplas vezes na Cam06.

Método	Tipo de Veículo	Pista 1	Pista 2	Pista 3	Total Veículos
KIM	Motocicletas	0 (5)	0 (3)	0 (3)	0
	Carros	8 (26)	8 (83)	1 (80)	17
	Caminhões	32 (12)	7 (6)	1 (6)	40
	Total	40	15	2	57
SANCHEZ et al.	Motocicletas	0 (5)	1 (3)	1 (3)	2
	Carros	11 (26)	5 (83)	14 (80)	30
	Caminhões	7 (12)	8 (6)	2 (6)	17
	Total	18	14	17	49
Método Proposto	Motocicletas	0 (5)	0 (3)	0 (3)	0
	Carros	1 (26)	0 (83)	4 (80)	5
	Caminhões	0 (12)	3 (6)	0 (6)	3
	Total	1	3	4	8

Como pode ser visto na Figura 5.2a, existe um semáforo na via, o que torna mais difícil a contagem correta dos veículos, pois quando os veículos param devido ao sinal de trânsito, eles ficam muito próximos uns dos outros, podendo inclusive ocorrer situações de oclusões de veículos, principalmente se estiverem presentes na cena veículos de grandes dimensões. O método proposto por Sanchez et al. tende a considerar os veículos parados como parte do fundo e o método proposto por Kim tende a dividir os veículos grandes em vários objetos, contando múltiplas vezes o mesmo veículo.

A Tabela 5.1 apresenta o resultados da contagem com cada um dos métodos. Nela observamos que o método proposto obteve o menor valor de erro, 2,23%, e que o método proposto por Kim apresentou um valor bastante alto. Analisando a Tabela 5.2 e a Tabela 5.3, notamos que isso se deve ao elevado número de veículos detectados mais de uma vez na pista 1, especialmente veículos grandes, enquanto deixou de contar apenas um veículo.

O método proposto por Sanchez et al. também obteve um número elevado de veículos detectados múltiplas vezes em todas as pistas, assim como um número elevado de veículos não detectados, fazendo com que o número de veículos não detectados fosse compensado pelo número de veículos detectados mais de uma vez. Por isso, o valor do erro total ficou menor que no método proposto por Kim. Isso ocorre quando os veículos param no sinal de trânsito, e o método proposto por Sanchez et al. detecta os veículos muito próximos como sendo um único objeto, principalmente os menores. Outro motivo para as detecções múltiplas ocorridas é a utilização de um modelo de fundo adaptativo por Sanchez et al., pois quando os veículos ficam parados, eles são considerados como parte do fundo, mesmo os que já foram detectados e contados. Quando os veículos retornam o movimento, o método considera esses veículos como parte do *foreground* novamente, e os mesmos veículos são detectados e contados mais uma vez.

A Figura 5.4a mostra um quadro de exemplo do vídeo Cam25, e a Figura 5.4b a sua respectiva máscara de *foreground*. Podemos observar que quando os veículos estão parados no sinal, eles ficam muito próximos uns dos outros, fazendo com que os objetos da máscara binária fiquem conectados, só voltando à normalidade quando os veículos reiniciam o movimento. A Figura 5.4c exhibe as linhas, em vermelho, que delimitam a região de interesse. A ROI foi posicionada de maneira a cobrir todas as pistas da via. Cada uma das pistas recebeu um laço virtual, posicionado próximo do ponto onde os veículos deixam a área de interesse, como exibido na Figura 5.4d. Essa câmera apresenta pequenos movimentos e vibrações, por isso, os laços virtuais tiveram a sua largura diminuída para não ficarem próximos das linhas que dividem as pistas, pois isso poderia fazer com que as linhas fossem detectadas como veículos.

A Figura 5.5 exhibe um exemplo de rastreamento de veículos na Cam25. Nesse exemplo podemos ver que os veículos são detectados como sendo apenas um único veículo quando entram na ROI, mas quando se aproximam da parte inferior do vídeo, passam a ser detectados corretamente, com cada veículo rastreado recebendo um retângulo, em verde, ao seu redor, e um número identificador diferente, em vermelho. Na Figura 5.6 é apresentada uma sequência de quadros que mostra as faixas da pista sendo detectadas como veículos, devido à vibração da câmera. Apesar disso, esses erros de detecção não são contados como veículos, pois não ocorrem sobre os laços virtuais.

A Tabela 5.4 mostra os resultados obtidos para a contagem dos veículos da Cam25 com o método proposto e também com os métodos comparativos.

Na Tabela 5.5 é apresentado o número de veículos que não foram detectados em cada um dos métodos e a Tabela 5.6 apresenta o número de veículos que foram detectados múltiplas vezes.

Na Figura 5.4b é apresentado um exemplo da máscara de *foreground* quando os veículos estão parados. Nela podemos observar que diversos veículos são detectados como sendo um único objeto na máscara. Isso ocorre devido à proximidade dos veículos, que ficam muito próximos enquanto esperam no semáforo, com alguns até mesmo sofrendo oclusões pelos veículos maiores.

Outra situação onde os veículos acabam fundidos em um só é ilustrada na Figura 5.5.

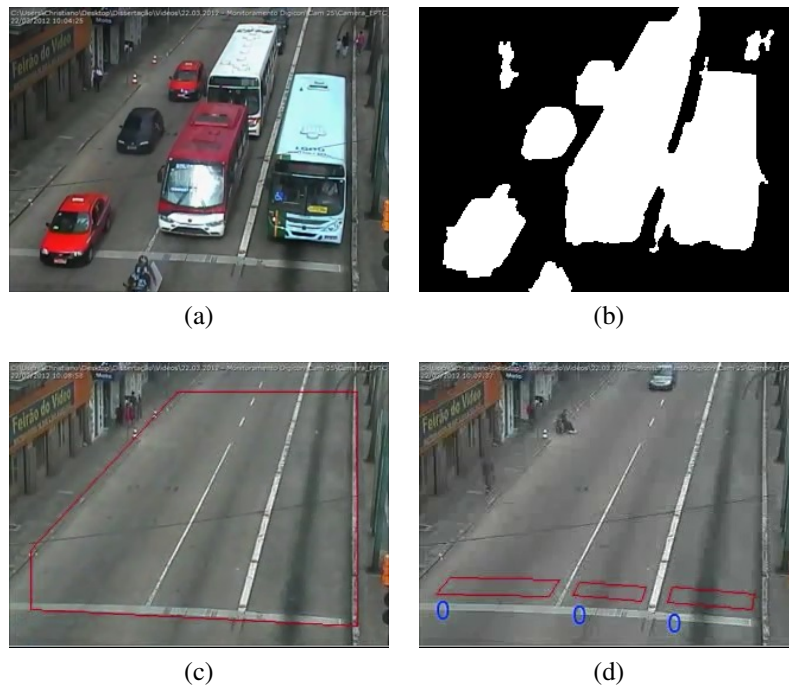


Figura 5.4: Vídeo Cam25. (a) uma sequência do vídeo; (b) máscara de *foreground*; (c) região de interesse; (d) laços virtuais.

Tabela 5.4: Contagem de veículos para a Cam25.

Método		Pista 1	Pista 2	Pista 3	Total
Inspecao Visual	Contagem	97	68	24	<b>189</b>
	Contagem	85	65	39	<b>189</b>
KIM	Diferença	-12	-3	15	<b>0</b>
	Erro	12	3	15	<b>30</b>
	Erro (%)	12.37%	4.41%	62.50%	<b>15.87%</b>
	Contagem	118	74	22	<b>214</b>
SANCHEZ et al.	Diferença	21	6	-2	<b>25</b>
	Erro	21	6	2	<b>29</b>
	Erro (%)	21.65%	8.82%	8.33%	<b>15.34%</b>
	Contagem	95	69	25	<b>189</b>
Método Proposto	Diferença	-2	1	1	<b>0</b>
	Erro	2	1	1	<b>4</b>
	Erro (%)	2.06%	1.47%	4.17%	<b>2.12%</b>



Figura 5.5: Exemplo de rastreamento na Cam25. (a) - (i) ilustram o rastreamento de veículos em uma sequência de quadros da Cam25.

Tabela 5.5: Veículos não detectados na Cam25.

Método	Tipo de Veículo	Pista 1	Pista 2	Pista 3	Total Veículos
KIM	Motocicletas	9 (26)	1 (6)	0 (0)	10
	Carros	4 (63)	3 (54)	0 (0)	7
	Caminhões	2 (8)	1 (8)	0 (24)	3
	Total	15	5	0	20
SANCHEZ et al.	Motocicletas	4 (26)	0 (6)	0 (0)	4
	Carros	16 (63)	9 (54)	0 (0)	25
	Caminhões	1 (8)	2 (8)	5 (24)	8
	Total	21	11	5	37
Método Proposto	Motocicletas	4 (26)	0 (6)	0 (0)	4
	Carros	3 (63)	0 (54)	0 (0)	3
	Caminhões	0 (8)	1 (8)	2 (24)	3
	Total	7	1	2	10

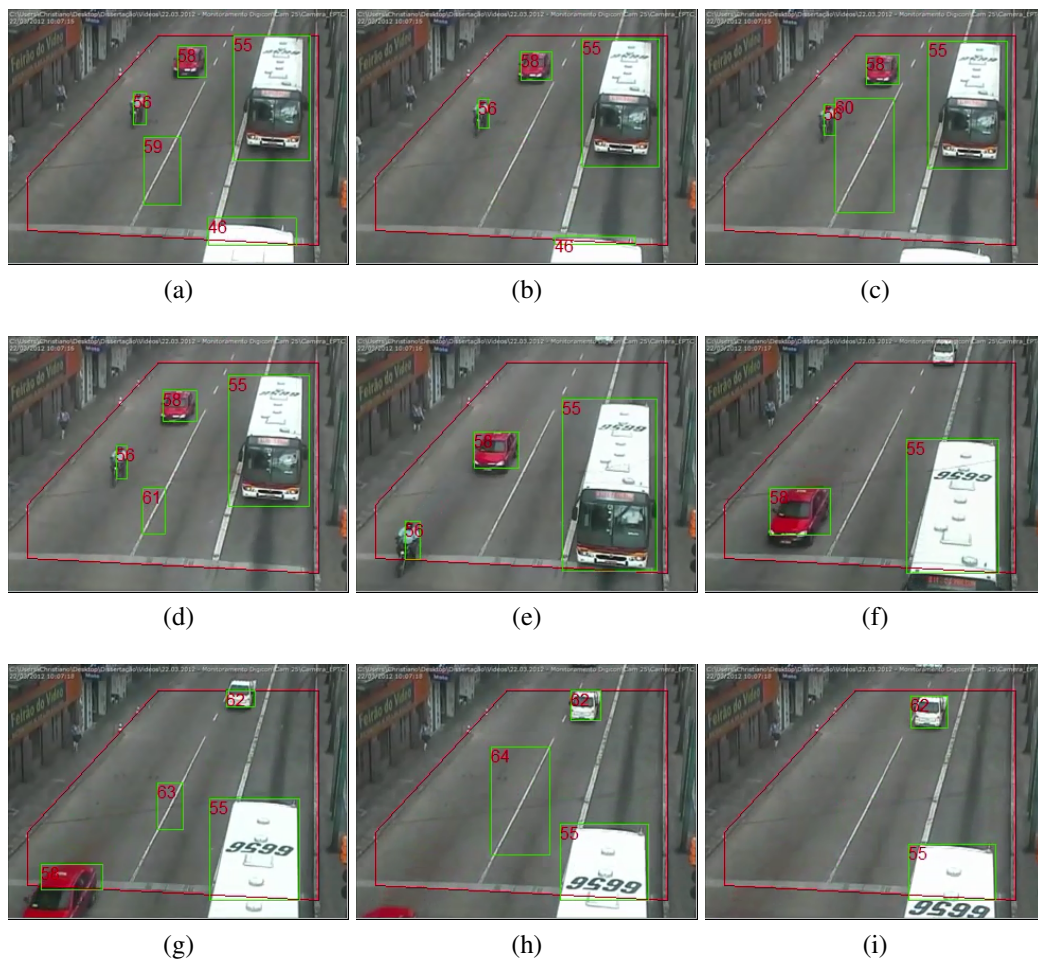


Figura 5.6: Exemplo de rastreamento Cam25. (a) - (i) ilustram o rastreamento de veículos em uma sequência de quadros da Cam25. Devido à vibração da câmera, as faixas são detectadas como objetos móveis na via. Entretanto, como não passam sobre os laços virtuais, as faixas não são contadas como veículos.

Tabela 5.6: Veículos detectados múltiplas vezes na Cam25.

Método	Tipo de Veículo	Pista 1	Pista 2	Pista 3	Total Veículos
KIM	Motocicletas	1 (26)	0 (6)	0 (0)	1
	Carros	0 (63)	1 (54)	0 (0)	1
	Caminhões	2 (8)	1 (8)	15 (24)	18
	Total	3	2	15	20
SANCHEZ et al.	Motocicletas	3 (26)	4 (6)	0 (0)	7
	Carros	34 (63)	10 (54)	1 (0)	45
	Caminhões	5 (8)	3 (8)	2 (24)	10
	Total	42	17	3	62
Método Proposto	Motocicletas	2 (26)	0 (6)	0 (0)	2
	Carros	2 (63)	2 (54)	0 (0)	4
	Caminhões	1 (8)	0 (8)	3 (24)	4
	Total	5	2	3	10

Quando os veículos estão entrando na região de interesse, eles estão mais longe da câmera, isso faz com que os objetos detectados possuam um tamanho menor e também apareçam muito próximos uns dos outros, o que contribui para que os veículos sejam fundidos. Apesar disso, quando os veículos vão se aproximando da saída da ROI, onde estão posicionados os laços virtuais, eles passam a ser detectados como veículos diferentes e quando passam sobre os laços virtuais já estão corretamente detectados, sendo então contados sem erros.

A Figura 5.6 apresenta uma ocasião onde ocorrem falhas de detecção devido a vibrações na câmera. Com essas vibrações, ocorrem pequenos deslocamentos nas imagens, fazendo com que as faixas sejam detectadas como objetos móveis na máscara de *foreground* em alguns quadros do vídeo. Para evitar que essas faixas fossem contadas como veículos, optou-se por reduzir um pouco a largura dos laços virtuais, assim, mesmo que ocorra algum deslocamento nos quadros, as faixas não entraram na área dos laços virtuais, e, portanto, não são incorretamente contadas como veículos.

Podemos observar na Tabela 5.4 que o método proposto apresentou um erro de contagem menor que os métodos comparativos. O método proposto por Kim obteve uma contagem maior na pista 3, isso é causado pelo fato desse método dividir veículos grandes em diversos objetos menores. O que pode ser verificado na Tabela 5.6, que indica que a maioria dos veículos detectados múltiplas vezes são caminhões, e estão na pista 3. Pela Tabela 5.5 e pela tabela 3 vemos que o método de Sanchez et al. obteve um número equilibrado de veículos não detectados e detectados múltiplas vezes. Ele deixou de detectar diversos veículos na pista 1, mas em compensação contou múltiplas vezes os veículos dessa pista. Por detectar o fundo da cena de maneira adaptativa, os veículos que ficam parados durante certo tempo são considerados parte do fundo da cena, e, quando o retornam o movimento, são contados novamente.

A Figura 5.7a mostra um quadro de exemplo do vídeo Cam41, e a Figura 5.7b a sua respectiva máscara de *foreground*. A Figura 5.7c exibe as linhas, em vermelho, que delimitam a região de interesse. A ROI não cobre inteiramente a via, pois existe um defeito na obtenção do vídeo, que causa um reflexo nas imagens, que pode ser observado na Figura 5.8. Assim, a ROI foi definida um pouco acima dessa região para evitar a interferência do problema do vídeo na contagem. As pistas recebem os laços virtuais, que foram posicionados dentro da área de interesse, cobrindo toda a extensão da via, próximo do ponto onde os veículos deixam a área de interesse, como exibido na Figura 5.7d.

A Figura 5.9 exibe um exemplo de rastreamento de veículos na Cam41. Nesse exemplo podemos ver que os veículos são detectados como sendo apenas um único veículo quando entram na ROI, mas quando se aproximam da parte inferior do vídeo, passam a ser detectados corretamente, com cada veículo rastreado recebendo um retângulo verde ao seu redor, e um número identificador diferente, em vermelho. Na Figura 5.10 é apresentada uma sequência de quadros que mostra uma falha na contagem. Um veículo longo é detectado como dois veículos diferentes.

A Tabela 5.7 mostra os resultados obtidos para a contagem dos veículos da Cam41 com o método proposto e também com os métodos comparativos. Os resultados são comparados com os valores da contagem real, obtidos através de inspeção visual.

A Tabela 5.8 apresenta o número de veículos que não foram detectados em cada um dos métodos, e a Tabela 5.9 apresenta o número de veículos que foram detectados múltiplas vezes.

Na Figura 5.7 observamos que a Cam41 não apresenta semáforos, e os veículos andam em um fluxo contínuo, não precisando parar em sinais de trânsito, dessa forma, não



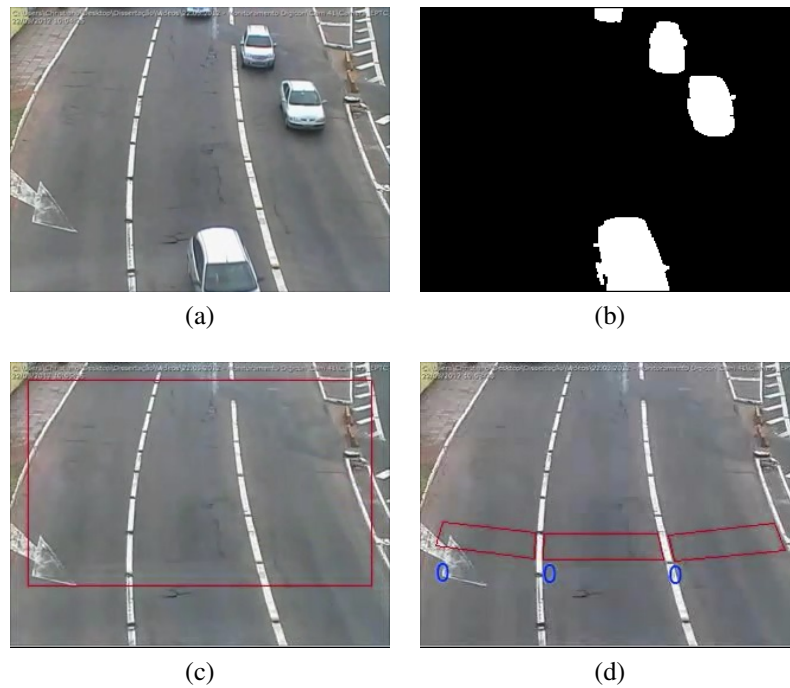


Figura 5.7: Vídeo Cam41. (a) uma sequência do vídeo; (b) máscara de *foreground*; (c) região de interesse; (d) laços virtuais.

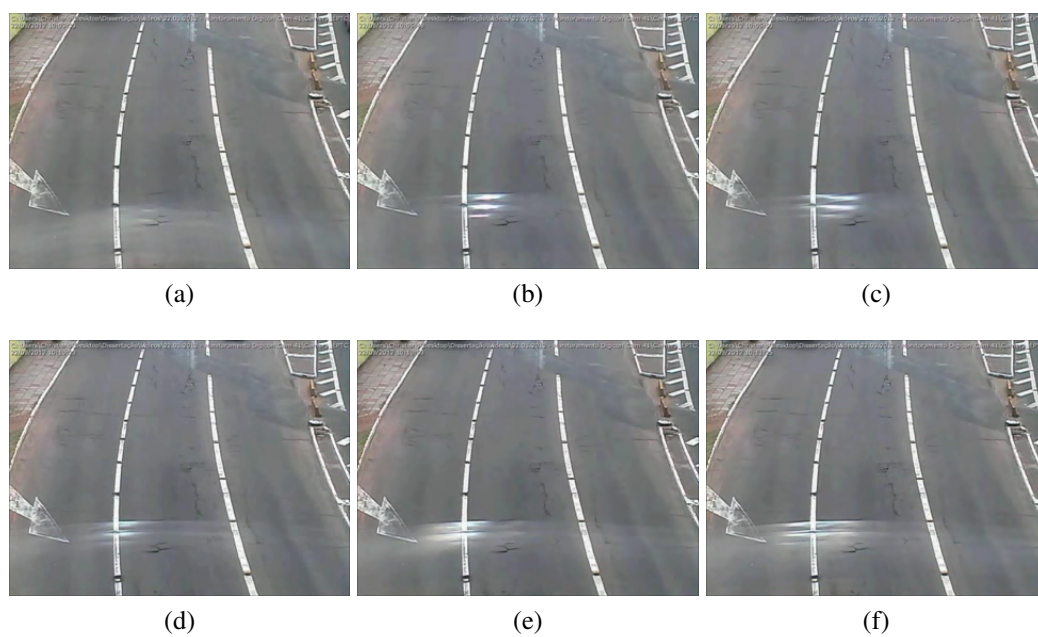


Figura 5.8: Defeito na obtenção dos vídeos da Cam41. (a) - (f) exibem quadros da Cam41 ilustrando os reflexos nas imagens.

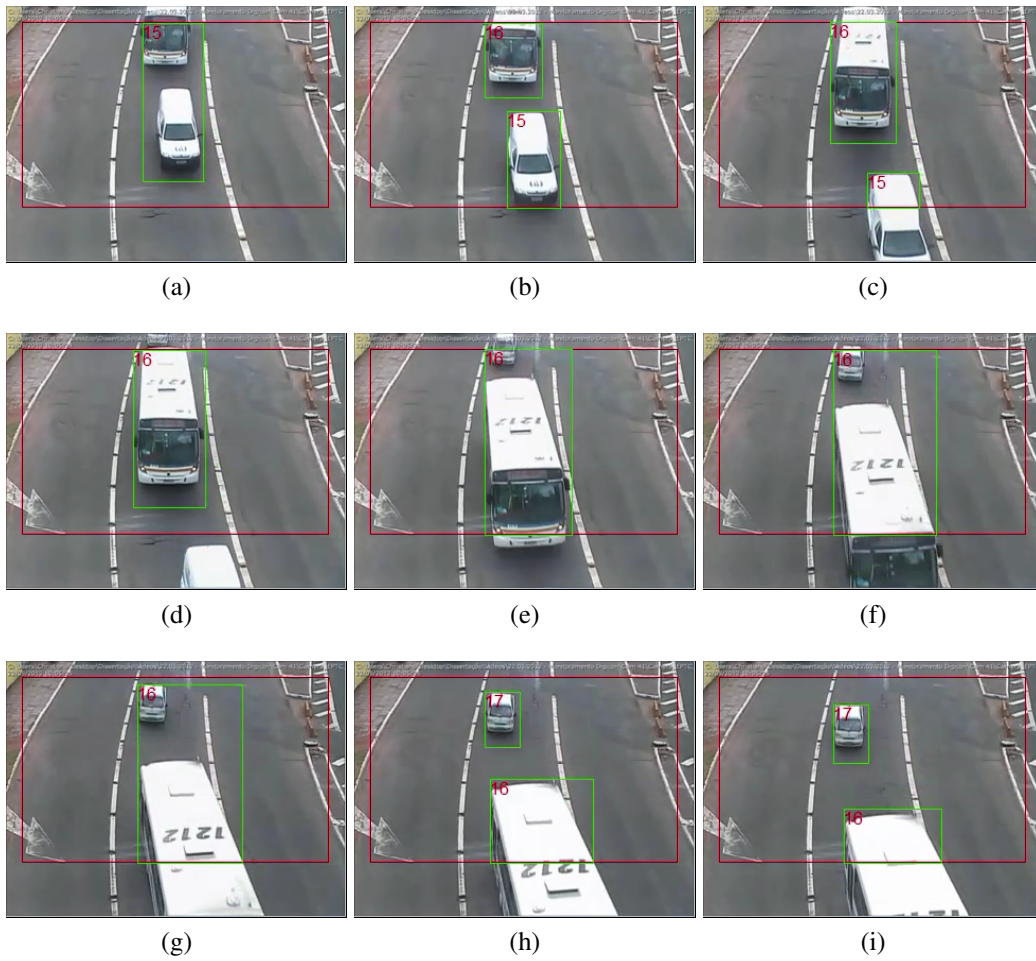


Figura 5.9: Exemplo de rastreamento na Cam41. (a) - (i) ilustram o rastreamento de veículos em uma sequência de quadros da Cam41 onde os veículos são detectados como sendo apenas um único veículo quando entram na ROI, mas quando se aproximam da parte inferior do vídeo, passam a ser detectados corretamente.

Tabela 5.7: Contagem de veículos para a Cam41.

Método		Pista 1	Pista 2	Pista 3	Total
Inspecao Visual	Contagem	10	157	84	<b>251</b>
	Contagem	10	185	81	<b>276</b>
KIM	Diferença	0	28	-3	<b>25</b>
	Erro	0	28	3	<b>31</b>
	Erro (%)	0.00%	17.83%	3.57%	<b>12.35%</b>
	Contagem	11	132	83	<b>226</b>
SANCHEZ et al.	Diferença	1	-25	-1	<b>-25</b>
	Erro	1	25	1	<b>27</b>
	Erro (%)	10.00%	15.92%	1.19%	<b>10.76%</b>
	Contagem	10	162	82	<b>254</b>
Método Proposto	Diferença	0	5	-2	<b>3</b>
	Erro	0	5	2	<b>7</b>
	Erro (%)	0.00%	3.18%	2.38%	<b>2.79%</b>

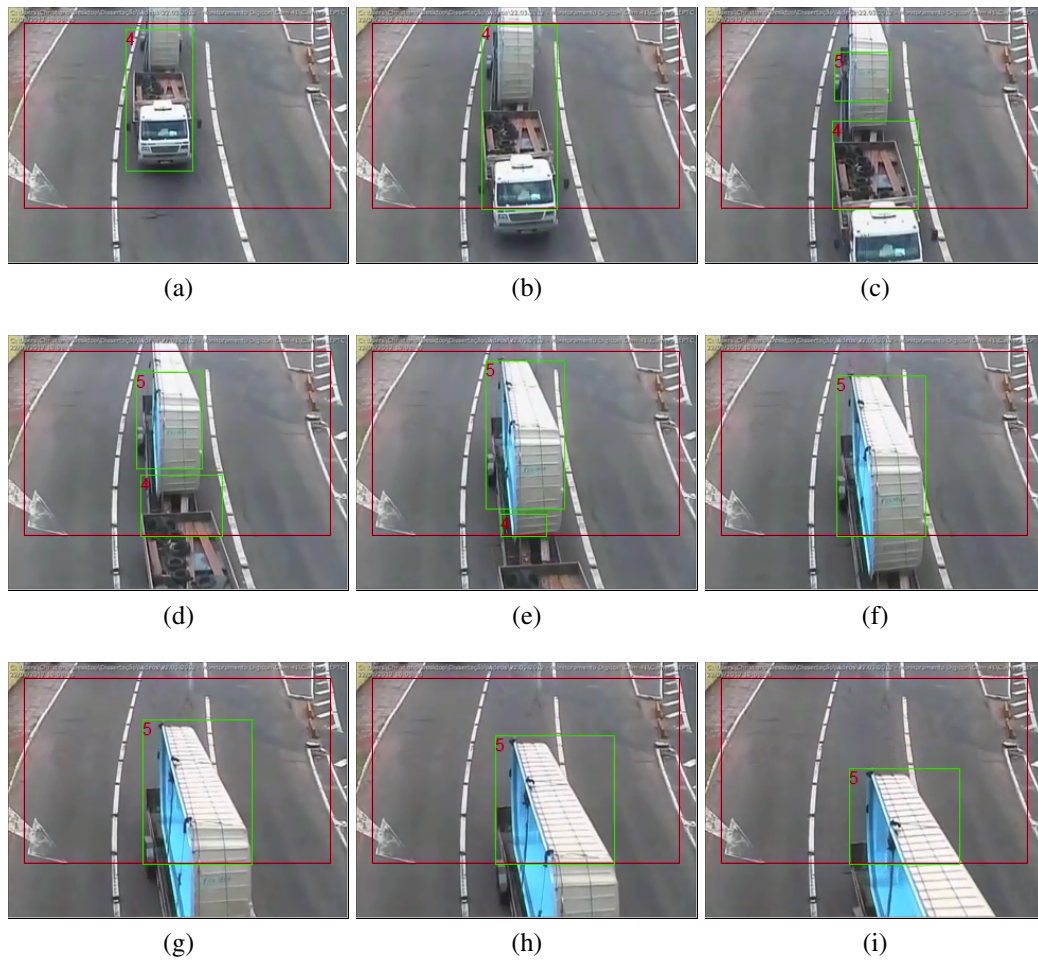


Figura 5.10: Exemplo de falha de contagem na Cam41. (a) - (i) ilustram o rastreamento de veículos em uma sequência de quadros da Cam41. Um veículo longo é detectado como dois veículos distintos.

Tabela 5.8: Veículos não detectados na Cam41.

Método	Tipo de Veículo	Pista 1	Pista 2	Pista 3	Total Veículos
KIM	Motocicletas	0 (1)	2 (6)	0 (5)	2
	Carros	0 (9)	5 (122)	4 (70)	9
	Caminhões	0 (0)	0 (29)	1 (9)	1
	Total	0	7	5	12
SANCHEZ et al.	Motocicletas	0 (1)	0 (6)	0 (5)	0
	Carros	0 (9)	18 (122)	1 (70)	19
	Caminhões	0 (0)	7 (29)	0 (9)	7
	Total	0	25	1	26
Método Proposto	Motocicletas	0 (1)	0 (6)	0 (5)	0
	Carros	0 (9)	1 (122)	2 (70)	3
	Caminhões	0 (0)	0 (29)	0 (9)	0
	Total	0	1	2	3

Tabela 5.9: Veículos detectados múltiplas vezes na Cam41.

Método	Tipo de Veículo	Pista 1	Pista 2	Pista 3	Total Veículos
KIM	Motocicletas	0 (1)	0 (6)	0 (5)	0
	Carros	0 (9)	1 (122)	0 (70)	1
	Caminhões	0 (0)	34 (29)	2 (9)	36
	Total	0	35	2	37
SANCHEZ et al.	Motocicletas	0 (1)	0 (6)	0 (5)	0
	Carros	0 (9)	0 (122)	0 (70)	0
	Caminhões	1 (0)	0 (29)	0 (9)	1
	Total	1	0	0	1
Método Proposto	Motocicletas	0 (1)	0 (6)	0 (5)	0
	Carros	0 (9)	3 (122)	0 (70)	3
	Caminhões	0 (0)	3 (29)	0 (9)	3
	Total	0	6	0	6

temos as dificuldades apresentadas em outras câmeras, como a oclusão causada quando os veículos estão parados e próximos uns dos outros.

A Figura 5.9 também ilustra a situação em que os veículos são considerados como sendo um objeto único quando entram na região de interesse, mas sendo, em seguida, detectados corretamente, quando se aproximam da câmera.

Como podemos observar na Tabela 5.7, o método proposto obteve o melhor resultado na contagem. Todos os métodos tiveram uma taxa de erro maior na pista central, pois é a que possui o maior número de veículos trafegando, e eles também entram muito próximos na área de interesse, fazendo com que inicialmente sejam contados como um único objeto. Por esse motivo, os métodos comparativos contaram alguns veículos como sendo um veículo único, deixando de detectar diversos veículos na pista central, principalmente o método proposto por Sanchez et al., como pode ser verificado na Tabela 5.8. Verificando a Tabela 5.9, percebemos que o método proposto por Kim apresentou dificuldades para contar corretamente os veículos grandes, como ônibus e caminhões, tendendo a dividir esses veículos em diversos objetos menores, fazendo com que os mesmos veículos sejam contados mais de uma vez.

A Figura 5.11a mostra um quadro de exemplo do vídeo Cam64, e a Figura 5.11b a sua respectiva máscara de *foreground*. Figura 5.11c exhibe as linhas que delimitam a região de interesse, em vermelho. A região de interesse cobre todas as pistas do lado esquerdo do vídeo. Cada uma das pistas recebeu um laço virtual, de modo que os laços cubram toda a extensão da via, como exibido na Figura 5.11d.

A Figura 5.12 exhibe um exemplo onde a detecção de veículos falha na Cam64. Neste exemplo, os veículos estavam parados no início do vídeo, isso faz com que não seja detectado movimento na cena e os veículos não aparecem na máscara de *foreground*. Quando o movimento inicia, são detectados os objetos no *foreground*, mas ainda assim um pouco fragmentados. Conforme os veículos passam, os objetos começam a ser detectados corretamente na máscara de *foreground*, e os veículos passam ser detectados corretamente, com cada veículo rastreado recebendo ao seu redor um retângulo verde, e um número identificador único, em vermelho. Na Figura 5.13 é apresentada outra situação desafiadora

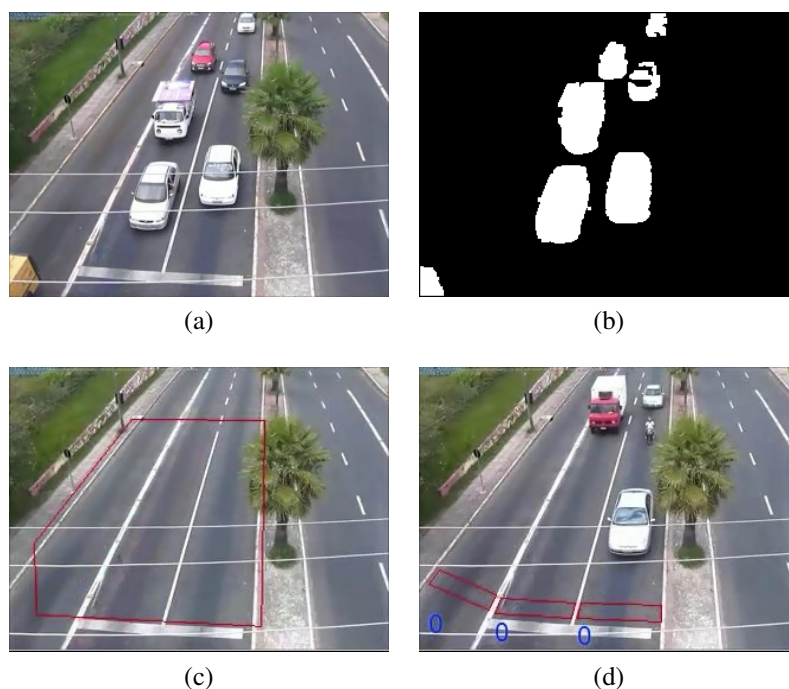


Figura 5.11: Vídeo Cam64. (a) uma sequência do vídeo; (b) máscara de *foreground*; (c) região de interesse; (d) laços virtuais.

para essa câmera. Devido à posição da câmera, os veículos da primeira pista, localizada na parte esquerda do vídeo, sofrem oclusões, principalmente quando veículos maiores, como ônibus, estão na pista central.

A Tabela 5.10 mostra os resultados obtidos para a contagem dos veículos da Cam64 com o método proposto e também com os métodos comparativos.

Na Tabela 5.11 é apresentado o número de veículos que não foram detectados em cada um dos métodos, enquanto a Tabela 5.12 apresenta o número de veículos que foram detectados múltiplas vezes.

A via monitorada pela Cam64, ilustrada na Figura 5.11, apresenta um semáforo. Como visto anteriormente, isso causa problemas de detecção e oclusões, devido aos veículos ficarem muito próximos quando parados. Uma outra dificuldade encontrada é o fato do vídeo já iniciar com os veículos parados, não tendo portanto nenhum movimento detectado na máscara de *foreground*. Quando os veículos começam o movimento, eles começam a ser detectados corretamente, mas alguns veículos, principalmente os que já estavam próximos ou sobre os laços virtuais, são detectados e contados incorretamente, como ilustrado na Figura 5.12.

Na Cam64, temos também uma situação em que a pista 1, mais a esquerda do vídeo, fica livre enquanto os veículos estão parados no semáforo nas pistas 2 e 3. Assim, os veículos da pista 1 sofrem oclusões quando existem veículos grandes parados na faixa central.

A Tabela 5.10 mostra que todos os métodos tiveram mais dificuldades com as pistas 2 e 3, devido ao fato dos veículos ficarem parados e muito próximos. O método proposto por Kim apresentou um grande número de erros na contagem, principalmente na pista 2, onde, conforme a Tabela 5.12, diversos veículos foram detectados múltiplas vezes, principalmente os veículos grandes. O método proposto por Sanchez et al. deixou de contar diversos veículos, como pode ser observado na Tabela 5.11, principalmente motocicletas,



Figura 5.12: Exemplo de falha de detecção na Cam64. (a) - (i) ilustram o rastreamento de veículos na Cam64 em uma situação onde ocorrem problemas de detecção dos veículos. O problema ocorre devido aos veículos estarem parados no início do vídeo, o que faz com que a máscara de *foreground* fique fragmentada, voltando ao normal conforme os veículos se movem.



Figura 5.13: Exemplo oclusões na Cam64. (a) - (i) ilustram o rastreamento de veículos na Cam64 em uma situação onde ocorrem problemas de oclusão de veículos. Devido à posição da câmera, os veículos da primeira pista sofrem oclusões quando veículos maiores, como ônibus, estão na pista central.

Tabela 5.10: Contagem de veículos para a Cam64.

Método		Pista 1	Pista 2	Pista 3	Total
Inspecao Visual	Contagem	72	36	54	<b>162</b>
	Contagem	67	48	43	<b>158</b>
KIM	Diferença	-5	12	-11	<b>-4</b>
	Erro	5	12	11	<b>28</b>
	Erro (%)	6.94%	33.33%	20.37%	<b>17.28%</b>
	Contagem	67	33	61	<b>161</b>
SANCHEZ et al.	Diferença	-5	-3	7	<b>-1</b>
	Erro	5	3	7	<b>15</b>
	Erro (%)	6.94%	8.33%	12.96%	<b>9.26%</b>
	Contagem	73	41	55	<b>169</b>
Método Proposto	Diferença	1	5	1	<b>7</b>
	Erro	1	5	1	<b>7</b>
	Erro (%)	1.39%	13.89%	1.85%	<b>4.32%</b>

Tabela 5.11: Veículos não detectados na Cam64.

<b>Método</b>	<b>Tipo de Veículo</b>	<b>Pista 1</b>	<b>Pista 2</b>	<b>Pista 3</b>	<b>Total Veículos</b>
KIM	Motocicletas	2 (14)	0 (1)	3 (10)	5
	Carros	8 (50)	2 (27)	8 (42)	18
	Caminhões	0 (8)	0 (8)	1 (2)	1
	<b>Total</b>	<b>10</b>	<b>2</b>	<b>12</b>	<b>24</b>
SANCHEZ et al.	Motocicletas	8 (14)	1 (1)	7 (10)	16
	Carros	13 (50)	6 (27)	7 (42)	26
	Caminhões	0 (8)	3 (8)	0 (2)	3
	<b>Total</b>	<b>21</b>	<b>10</b>	<b>14</b>	<b>45</b>
Método Proposto	Motocicletas	0 (14)	0 (1)	3 (10)	3
	Carros	3 (50)	1 (27)	1 (42)	5
	Caminhões	0 (8)	0 (8)	0 (2)	0
	<b>Total</b>	<b>3</b>	<b>1</b>	<b>4</b>	<b>8</b>

Tabela 5.12: Veículos detectados múltiplas vezes na Cam64.

<b>Método</b>	<b>Tipo de Veículo</b>	<b>Pista 1</b>	<b>Pista 2</b>	<b>Pista 3</b>	<b>Total Veículos</b>
KIM	Motocicletas	0 (14)	0 (1)	0 (10)	0
	Carros	1 (50)	5 (27)	1 (42)	7
	Caminhões	4 (8)	9 (8)	0 (2)	13
	<b>Total</b>	<b>5</b>	<b>14</b>	<b>1</b>	<b>20</b>
SANCHEZ et al.	Motocicletas	1 (14)	0 (1)	2 (10)	3
	Carros	8 (50)	5 (27)	17 (42)	30
	Caminhões	7 (8)	2 (8)	2 (2)	11
	<b>Total</b>	<b>16</b>	<b>7</b>	<b>21</b>	<b>44</b>
Método Proposto	Motocicletas	0 (14)	0 (1)	1 (10)	1
	Carros	3 (50)	1 (27)	1 (42)	5
	Caminhões	1 (8)	5 (8)	3 (2)	9
	<b>Total</b>	<b>4</b>	<b>6</b>	<b>5</b>	<b>15</b>



que passam muito próximas dos veículos. Esse método também teve bastante veículos detectados mais de uma vez, como apontado na Tabela 5.12. Comparando a Tabela 5.12 e a Tabela 5.11, podemos notar que o método de Sanchez et al. obteve um valor de erro menor devido ao número equilibrado entre veículos não detectados e veículos detectados múltiplas vezes.

Apesar de ter obtido o melhor resultado, o método proposto também apresentou um número elevado de veículos detectados múltiplas vezes, devido aos veículos pararem no semáforo, fazendo com que os objetos da máscara de *foreground* fiquem todos conectados, como um único objeto, e, quando os veículos reiniciam o movimento, a máscara de *foreground* fique fragmentada por alguns instantes, causando múltiplas detecções dos veículos próximos aos laços virtuais.

A Figura 5.14a mostra um quadro de exemplo do vídeo Cam68, e a Figura 5.14b a sua respectiva máscara de *foreground*. A Figura 5.14c exibe as linhas, em vermelho, da região de interesse. Ao contrário dos outros vídeos, este possui apenas duas faixas de tráfego, e cada uma recebeu um laço virtual, posicionado na parte inferior da imagem, onde os veículos deixam a área de interesse, como exibido na Figura 5.14d.

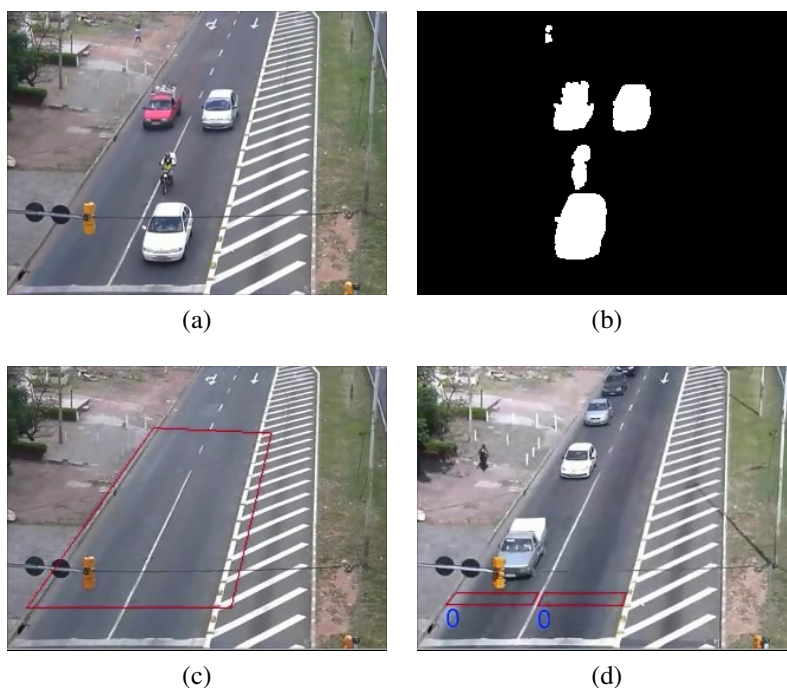


Figura 5.14: Vídeo Cam68. (a) uma sequência do vídeo; (b) máscara de *foreground*; (c) região de interesse; (d) laços virtuais.

A Figura 5.15 exibe um exemplo de rastreamento de veículos na Cam68. Aqui os veículos também estavam parados, e também ocorre uma fragmentação da máscara de *foreground*, fazendo com que os objetos da cena sejam detectados incorretamente. Entretanto, como podemos observar na imagem, assim que os veículos começam a se movimentar, e se aproximar da parte inferior do vídeo, são todos detectados e contados corretamente, com cada veículo rastreado recebendo ao seu redor um retângulo, em verde, e um número identificador diferente, em vermelho. Na Figura 5.16 é apresentada uma sequência de quadros onde os veículos estão parados e muito próximos, e todos os veículos são detectados como um único objeto, principalmente por causa do ônibus, que, devido ao seu tamanho, faz com que ele cubra uma parte dos veículos da pista ao lado. Conforme os

veículos se movimentam, alguns dos veículos são detectados e contados corretamente, porém, pelo menos três veículos não são contados

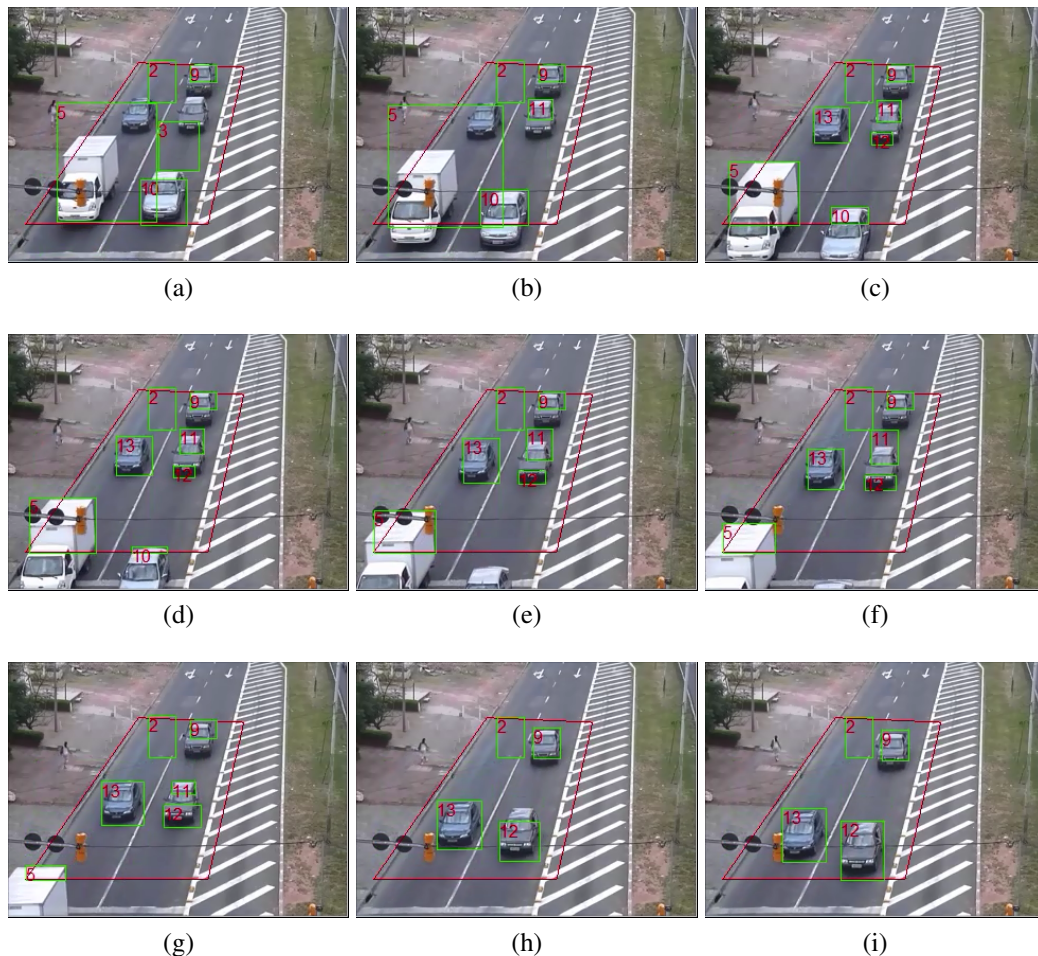


Figura 5.15: Exemplo de rastreamento na Cam68. (a) - (i) ilustram o rastreamento de veículos em uma sequência de quadros da Cam68. Os veículos estavam parados devido ao sinal de trânsito, fazendo com que os objetos da cena sejam detectados incorretamente. Entretanto, como podemos observar nas figuras, assim que os veículos começam a se movimentar, e se aproximam da parte inferior do vídeo, são todos detectados e contados corretamente.

Na Tabela 5.13 são exibidos os resultados obtidos para a contagem dos veículos da Cam68 utilizando o método proposto e também com os métodos comparativos.

A Tabela 5.14 apresenta o número de veículos que não foram detectados, utilizando cada um dos métodos, enquanto a Tabela 5.15 apresenta o número de veículos que foram contados mais de uma vez.

Na Cam68 os veículos também param devido ao sinal de trânsito, causando problemas semelhantes aos da Cam64. Quando os veículos estão parados, eles são detectados como um único objeto, como ilustrado na Figura 5.16, e, no momento que os veículos se movimentam, ocorrem fragmentações na máscara de *foreground*. A pista da esquerda também apresenta situações de oclusão quando existem veículos grandes na pista da direita, fazendo com que seja detectado um único veículo.

De acordo com a Tabela 5.13, o método proposto obteve o menor valor de erro para a Cam68, 1,30%, enquanto o método proposto por Kim obteve a maior diferença para o

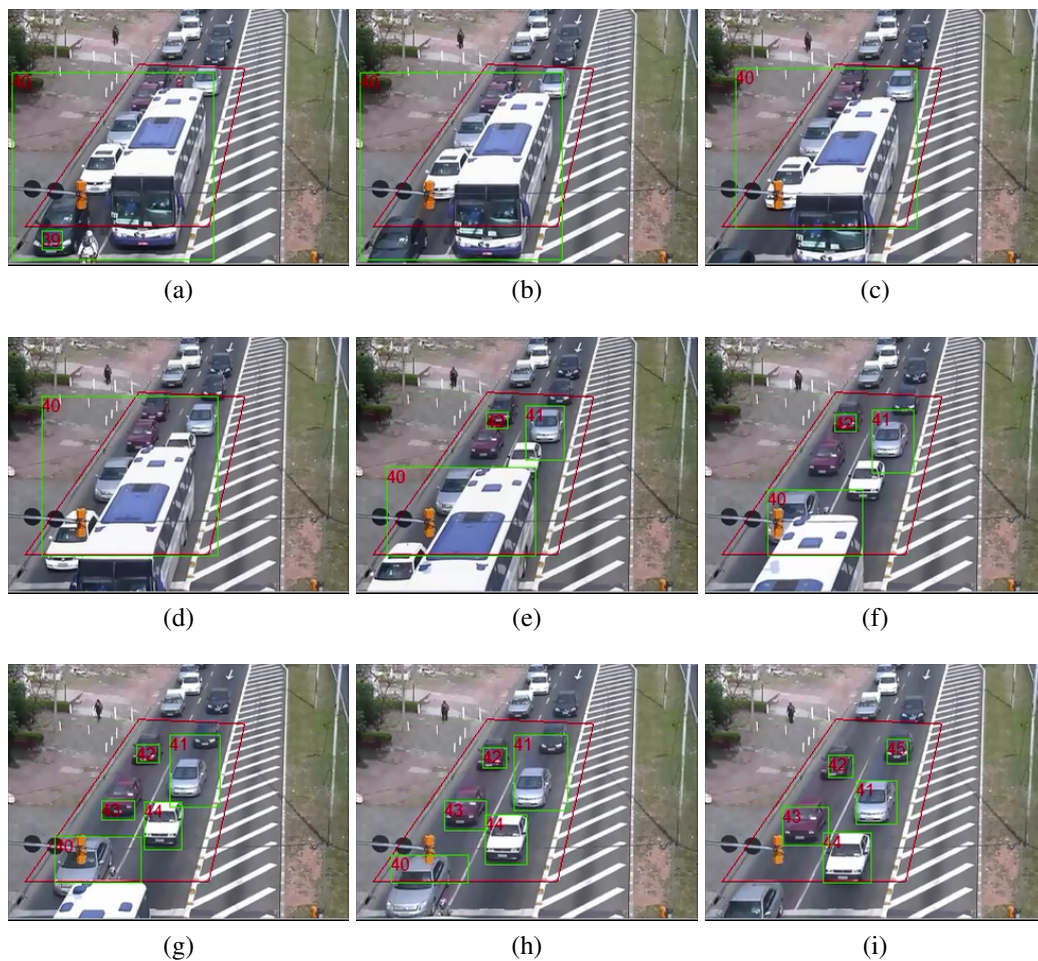


Figura 5.16: Exemplo de falha de contagem na Cam68. (a) - (i) mostram uma sequência de quadros da Cam68 onde os veículos estão parados e muito próximos. Os veículos são detectados como um único objeto e, conforme os veículos se movimentam, alguns dos veículos são detectados e contados corretamente, porém, pelo menos três veículos não são contados.

Tabela 5.13: Contagem de veículos para a Cam68.

<b>Método</b>		<b>Pista 1</b>	<b>Pista 2</b>	<b>Total</b>
Inspecao Visual	Contagem	66	88	<b>154</b>
	Contagem	83	91	<b>174</b>
KIM	Diferença	17	3	<b>20</b>
	Erro	17	3	<b>20</b>
	Erro (%)	25.76%	3.41%	<b>12.99%</b>
	Contagem	66	96	<b>162</b>
SANCHEZ et al.	Diferença	0	8	<b>8</b>
	Erro	0	8	<b>8</b>
	Erro (%)	0.00%	9.09%	<b>5.19%</b>
	Contagem	65	89	<b>154</b>
Método Proposto	Diferença	-1	1	<b>0</b>
	Erro	1	1	<b>2</b>
	Erro (%)	1.52%	1.14%	<b>1.30%</b>

Tabela 5.14: Veículos não detectados na Cam68.

<b>Método</b>	<b>Tipo de Veículo</b>	<b>Pista 1</b>	<b>Pista 2</b>	<b>Total Veículos</b>
KIM	Motocicletas	1 (7)	2 (17)	3
	Carros	0 (43)	3 (64)	3
	Caminhões	1 (16)	0 (7)	1
	Total	2	5	7
SANCHEZ et al.	Motocicletas	4 (7)	11 (17)	15
	Carros	14 (43)	8 (64)	22
	Caminhões	2 (16)	2 (7)	4
	Total	20	21	41
Método Proposto	Motocicletas	2 (7)	2 (17)	4
	Carros	6 (43)	1 (64)	7
	Caminhões	1 (16)	0 (7)	1
	Total	9	3	12

Tabela 5.15: Veículos detectados múltiplas vezes na Cam68.

Método	Tipo de Veículo	Pista 1	Pista 2	Total Veículos
KIM	Motocicletas	0 (7)	0 (17)	0
	Carros	5 (43)	0 (64)	5
	Caminhões	14 (16)	8 (7)	22
	Total	19	8	27
SANCHEZ et al.	Motocicletas	3 (7)	0 (17)	3
	Carros	11 (43)	17 (64)	28
	Caminhões	6 (16)	12 (7)	18
	Total	20	29	49
Método Proposto	Motocicletas	1 (7)	0 (17)	1
	Carros	5 (43)	0 (64)	5
	Caminhões	2 (16)	4 (7)	6
	Total	8	4	12

valor real, principalmente na pista 1. Observando a Tabela 5.15 vemos que isso ocorre devido ao elevado número de veículos que foram detectados múltiplas vezes, principalmente caminhões, que são divididos em vários objetos. Já o método de Sanchez et al., apresentou para esse vídeo um número equilibrado de veículos não detectados e veículos detectados múltiplas vezes. Esse método apresentou um grande número de motocicletas não detectadas, devido ao posicionamento da câmera ser mais afastado da pista e, novamente, contou multiplamente diversos veículos em ambas as pistas, devido ao fato do método proposto por Sanchez et al. detectar o fundo da cena adaptativamente, fazendo com que os veículos que ficam parados durante certo tempo sejam considerados parte do fundo da cena, e, quando o retornam o movimento, são contados novamente.

A Figura 5.17a exibe um quadro de exemplo do vídeo Cam73, enquanto a Figura 5.17b exibe a sua respectiva máscara de *foreground*. Na Figura 5.17c são exibidas as linhas, em vermelho, que delimitam a região de interesse. Esse vídeo possui um ângulo de visão mais elevado, capturando uma vista do topo da cena, por isso os laços virtuais deste vídeo foram posicionados na parte central da imagem, para que pudessem cobrir totalmente as pistas externas, como exibido na Figura 5.17d.

A Figura 5.18 ilustra o rastreamento de veículos na Cam73. Neste vídeo, por ter uma visão mais do topo da cena, alguns veículos ocupam toda a região de interesse. Na imagem podemos observar que o veículo é maior que a área de interesse, entretanto, ele é detectado e contado corretamente. Na Figura 5.19 é apresentada uma sequência de quadros onde ocorre uma detecção múltipla de um veículo, que é dividido em duas partes e contado como sendo dois veículos diferentes.

A Tabela 5.16 exibe os resultados obtidos para a contagem dos veículos da Cam73 utilizando o método proposto e também os métodos comparativos.

Na Tabela 5.17 é apresentado o número de veículos que não foram detectados, utilizando cada um dos métodos, enquanto na Tabela 5.18 é apresentado o número de veículos que foram contados mais de uma vez.

A Figura 5.17 ilustra a Cam73, que é um dos cenários onde não existem veículos parados nas sinalizações de trânsito, não apresentando tantos problemas relacionados com

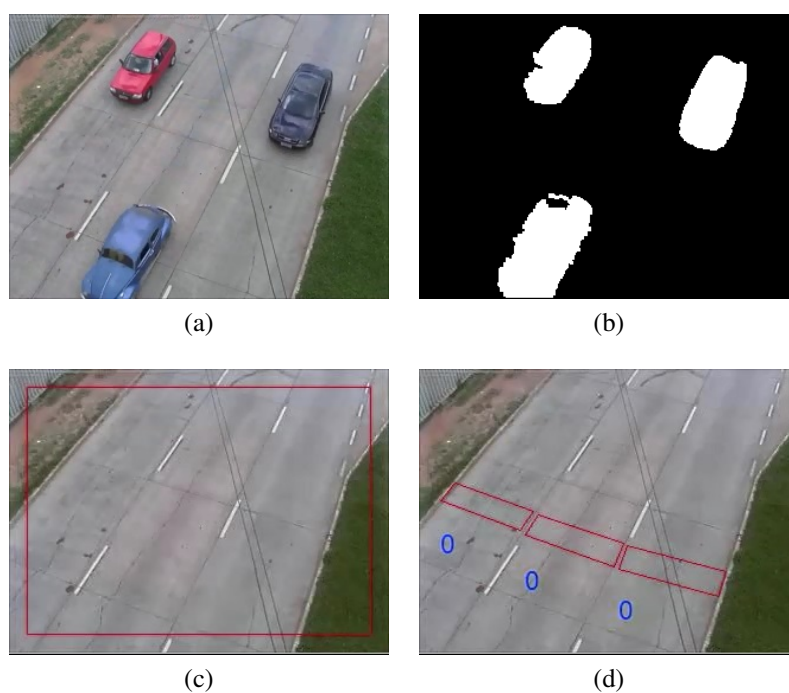


Figura 5.17: Vídeo Cam73. (a) uma sequência do vídeo; (b) máscara de *foreground*; (c) região de interesse; (d) laços virtuais.

Tabela 5.16: Contagem de veículos para a Cam73.

Método		Pista 1	Pista 2	Pista 3	Total
Inspecao Visual	Contagem	25	48	28	<b>101</b>
	Contagem	24	43	18	<b>89</b>
KIM	Diferença	-1	-5	-10	<b>-16</b>
	Erro	1	5	10	<b>16</b>
	Erro (%)	4.00%	10.42%	35.71%	<b>15.84%</b>
	Contagem	21	49	28	<b>98</b>
SANCHEZ et al.	Diferença	-4	1	0	<b>-3</b>
	Erro	4	1	0	<b>5</b>
	Erro (%)	16.00%	2.08%	0.00%	<b>4.95%</b>
	Contagem	25	51	29	<b>105</b>
Método Proposto	Diferença	0	3	1	<b>4</b>
	Erro	0	3	1	<b>4</b>
	Erro (%)	0.00%	6.25%	3.57%	<b>3.96%</b>

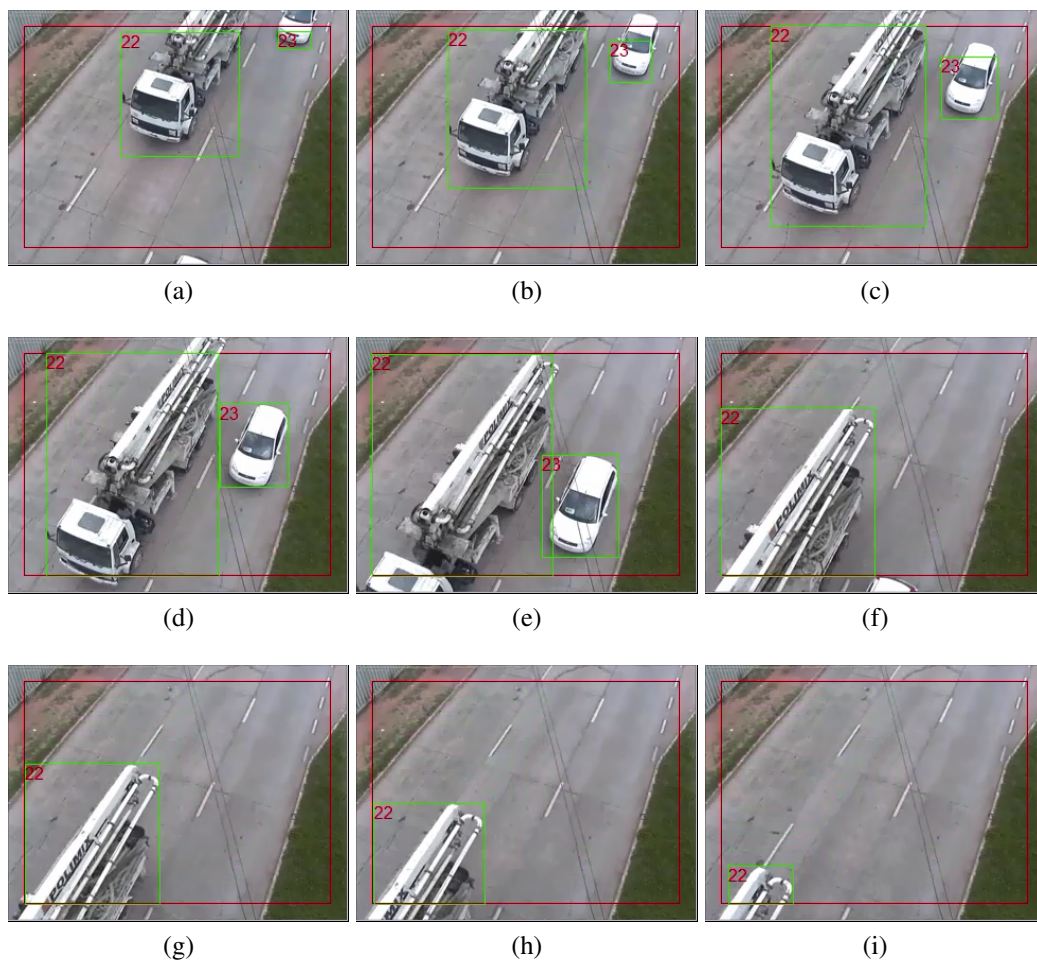


Figura 5.18: Exemplo de rastreamento de veículo maior que a região de interesse na Cam73. (a) - (i) ilustram o rastreamento de veículos em uma sequência de quadros da Cam73. Por ter um visã mais do topo da cena, alguns veículos ocupam quase toda a região de interesse.

Tabela 5.17: Veículos não detectados na Cam73.

Método	Tipo de Veículo	Pista 1	Pista 2	Pista 3	Total Veículos
KIM	Motocicletas	0 (1)	1 (3)	0 (1)	1
	Carros	5 (17)	6 (34)	10 (25)	21
	Caminhões	0 (7)	0 (11)	0 (2)	0
	Total	5	7	10	22
SANCHEZ et al.	Motocicletas	0 (1)	1 (3)	0 (1)	1
	Carros	1 (17)	0 (34)	1 (25)	2
	Caminhões	3 (7)	0 (11)	0 (2)	3
	Total	4	1	1	6
Método Proposto	Motocicletas	0 (1)	0 (3)	0 (1)	0
	Carros	0 (17)	0 (34)	0 (25)	0
	Caminhões	0 (7)	0 (11)	0 (2)	0
	Total	0	0	0	0

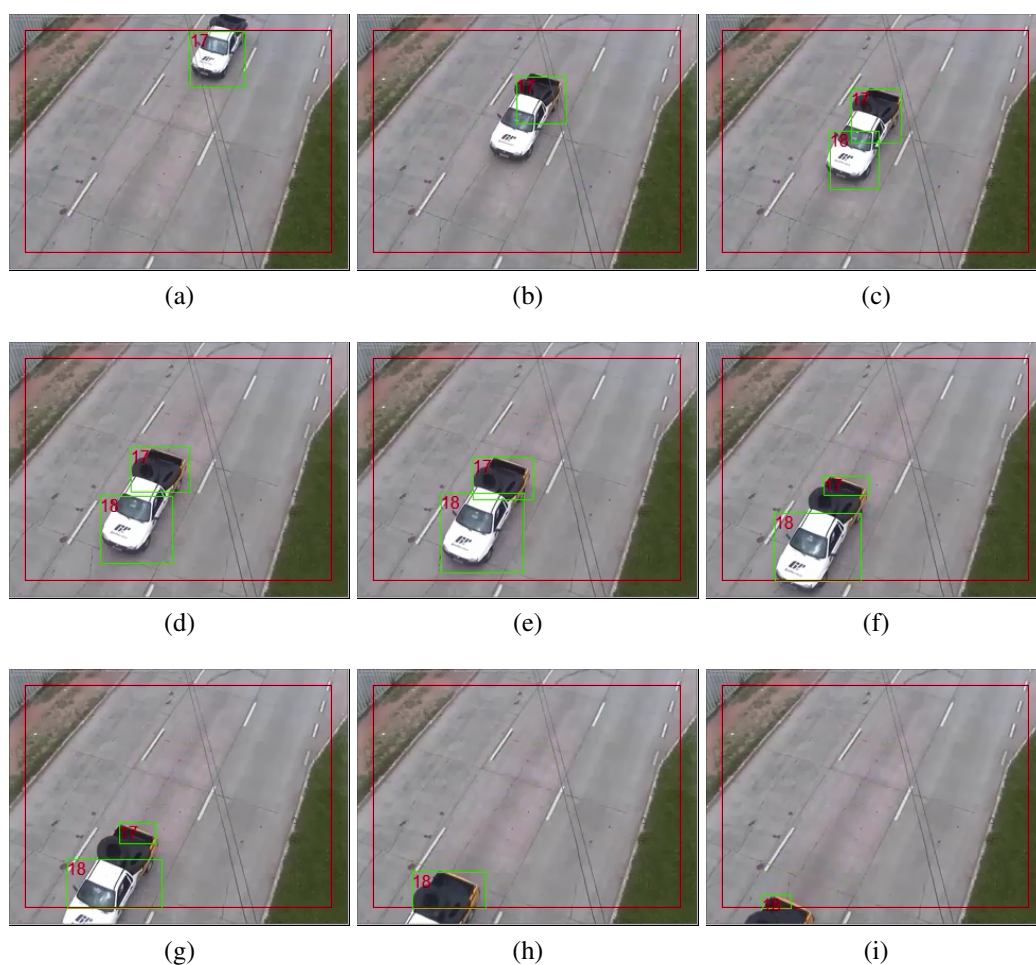


Figura 5.19: Exemplo de detecção múltipla de veículos na Cam73. (a) - (i) ilustram uma sequência de quadros da Cam73 onde um veículo é dividido em duas partes e contado como sendo dois veículos diferentes.

Tabela 5.18: Veículos detectados múltiplas vezes na Cam73.

Método	Tipo de Veículo	Pista 1	Pista 2	Pista 3	Total Veículos
KIM	Motocicletas	0 (1)	0 (3)	0 (1)	0
	Carros	0 (17)	0 (34)	0 (25)	0
	Caminhões	4 (7)	2 (11)	0 (2)	6
	Total	4	2	0	6
SANCHEZ et al.	Motocicletas	0 (1)	0 (3)	0 (1)	0
	Carros	0 (17)	0 (34)	0 (25)	0
	Caminhões	0 (7)	2 (11)	1 (2)	3
	Total	0	2	1	3
Método Proposto	Motocicletas	0 (1)	0 (3)	0 (1)	0
	Carros	0 (17)	2 (34)	0 (25)	2
	Caminhões	0 (7)	1 (11)	1 (2)	2
	Total	0	3	1	4



veículos sendo contados múltiplas vezes, principalmente no método Sanchez et al..

De acordo com a Tabela 5.16, o método proposto apresentou um erro de 3.96%, e o método de Sanchez et al. ficando também bastante próximo, com 4.95% de erro, obtendo um erro maior na pista 1, de 16%. O método proposto por Kim obteve 15.84% de erro, sendo a pista 3 a com o maior número de erros.

Observando a Tabela 5.17 e a Tabela 5.18, notamos que esses erro são causados por veículos que não foram detectados, sendo poucos os veículos que foram detectados múltiplas vezes. Inclusive o método de Kim, que geralmente apresenta um elevado número de erros causados pela detecção múltipla, apresentou um número maior de erros causados por veículos não detectados. Um dos motivos para esses erros de detecção é o posicionamento da câmera. Como visto na Figura 5.17, a câmera captura uma visão mais de topo da cena, e também está bem próxima da via. Por isso, juntamente com o fato de veículos não precisarem parar devido a alguma sinalização na via, os veículos passam muito rápido pelo vídeo, não ficando um número de quadros suficientes para serem detectados corretamente. Assim, os métodos comparativos tiveram dificuldades para detectar alguns veículos.

A Tabela 5.19 reúne a contagem obtida em todos os vídeos, com o método proposto e com os métodos de comparação, mostrando ao final o resultado geral. Nela são apresentas para cada um dos vídeos a contagem de veículos, a diferença da contagem para o valor real, o erro em relação ao valor real e a porcentagem do erro. Esses valores também são apresentados para o total dos vídeos, ou seja, o somatório de todas as pistas de todos os vídeos, onde a porcentagem do erro é calculada como o número total de erros sobre o número real de veículos no vídeo.

Na Tabela 5.20 estão os resultados gerais do número de veículos não detectados, e na Tabela 5.21 os resultados gerais dos veículos detectados mais de uma vez, para todos os vídeos e utilizando o método proposto e os métodos de comparação.

O método proposto apresenta um erro de 2,68%, ou seja, um erro de contagem de 29 veículos para um total de 1081 veículos. Podemos observar que o método proposto tem uma taxa de erro e uma diferença absoluta menor em relação ao método comparativo proposto por Kim, que possui um erro de contagem de 184 veículos (17,02%), e ao método proposto por Sanchez et al., que obteve uma diferença de detecção de 107 veículos (9,9%). Um ponto importante, é que o valor menor para a diferença de contagem do método comparativo proposto por Sanchez et al., 27 veículos contados a mais do que o número total de veículos, na verdade se deve ao número mais balanceado de falhas de detecção e múltiplas detecções de veículos, sendo esses erros de detecção e de detecção múltipla maiores que os outros métodos testados, como pode ser observado na Tabela 5.20 e na Tabela 5.21.

Como vemos na Tabela 5.20, o método proposto perde menos veículos na avaliação geral que os métodos comparativos, deixando de contar apenas 40 veículos. O número de veículos não detectados no método proposto é no mínimo duas vezes menor que nos métodos comparativos de Kim e Sanchez et al., que não detectaram 101 e 181 veículos, respectivamente.

Quando se trata de detectar veículos longos, o método proposto também apresenta resultados melhores que os métodos comparativos. O processo de fusão de clusters assegura que agrupamentos de partículas que são diferentes, mas na verdade pertencem ao mesmo veículo, sejam unidos, na maioria das vezes, com êxito para detectar um veículo, e isso ocorre devido ao auxílio da máscara de *foreground*, que identifica os objetos em movimento nas regiões correspondentes aos clusters.

Tabela 5.19: Resultado geral da contagem de veículos.

Vídeo	Pistas	Inspeção Visual	Método Proposto				KIM				SANCHEZ et al.			
			Cont.	Dif.	Erro	Erro (%)	Cont.	Dif.	Erro	Erro (%)	Cont.	Dif.	Erro	Erro (%)
Cam06	Pista 1	43	41	-2	2	4.65%	82	39	39	90.70%	51	8	8	18.60%
	Pista 2	92	92	0	0	0.00%	103	11	11	11.96%	100	8	8	8.70%
	Pista 3	89	92	3	3	3.37%	80	-9	9	10.11%	96	7	7	7.87%
	<b>Total</b>	<b>224</b>	<b>225</b>	<b>1</b>	<b>5</b>	<b>2.23%</b>	<b>265</b>	<b>41</b>	<b>59</b>	<b>26.34%</b>	<b>247</b>	<b>23</b>	<b>23</b>	<b>10.27%</b>
Cam25	Pista 1	97	95	-2	2	2.06%	85	-12	12	12.37%	118	21	21	21.65%
	Pista 2	68	69	1	1	1.47%	65	-3	3	4.41%	74	6	6	8.82%
	Pista 3	24	25	1	1	4.17%	39	15	15	62.50%	22	-2	2	8.33%
	<b>Total</b>	<b>189</b>	<b>189</b>	<b>0</b>	<b>4</b>	<b>2.12%</b>	<b>189</b>	<b>0</b>	<b>30</b>	<b>15.87%</b>	<b>214</b>	<b>25</b>	<b>29</b>	<b>15.34%</b>
Cam41	Pista 1	10	10	0	0	0.00%	10	0	0	0.00%	11	1	1	10.00%
	Pista 2	157	162	5	5	3.18%	185	28	28	17.83%	132	-25	25	15.92%
	Pista 3	84	82	-2	2	2.38%	81	-3	3	3.57%	83	-1	1	1.19%
	<b>Total</b>	<b>251</b>	<b>254</b>	<b>3</b>	<b>7</b>	<b>2.79%</b>	<b>276</b>	<b>25</b>	<b>31</b>	<b>12.35%</b>	<b>226</b>	<b>-25</b>	<b>27</b>	<b>10.76%</b>
Cam64	Pista 1	72	73	1	1	1.39%	67	-5	5	6.94%	67	-5	5	6.94%
	Pista 2	36	41	5	5	13.89%	48	12	12	33.33%	33	-3	3	8.33%
	Pista 3	54	55	1	1	1.85%	43	-11	11	20.37%	61	7	7	12.96%
	<b>Total</b>	<b>162</b>	<b>169</b>	<b>7</b>	<b>7</b>	<b>4.32%</b>	<b>158</b>	<b>-4</b>	<b>28</b>	<b>17.28%</b>	<b>161</b>	<b>-1</b>	<b>15</b>	<b>9.26%</b>
Cam68	Pista 1	66	65	-1	1	1.52%	83	17	17	25.76%	66	0	0	0.00%
	Pista 2	88	89	1	1	1.14%	91	3	3	3.41%	96	8	8	9.09%
	<b>Total</b>	<b>154</b>	<b>154</b>	<b>0</b>	<b>2</b>	<b>1.30%</b>	<b>174</b>	<b>20</b>	<b>20</b>	<b>12.99%</b>	<b>162</b>	<b>8</b>	<b>8</b>	<b>5.19%</b>
Cam73	Pista 1	25	25	0	0	0.00%	24	-1	1	4.00%	21	-4	4	16.00%
	Pista 2	48	51	3	3	6.25%	43	-5	5	10.42%	49	1	1	2.08%
	Pista 3	28	29	1	1	3.57%	18	-10	10	35.71%	28	0	0	0.00%
	<b>Total</b>	<b>101</b>	<b>105</b>	<b>4</b>	<b>4</b>	<b>3.96%</b>	<b>85</b>	<b>-16</b>	<b>16</b>	<b>15.84%</b>	<b>98</b>	<b>-3</b>	<b>5</b>	<b>4.95%</b>
<b>Geral</b>	<b>1081</b>	<b>1096</b>	<b>15</b>	<b>29</b>	<b>2.68%</b>	<b>1147</b>	<b>66</b>	<b>184</b>	<b>17.02%</b>	<b>1108</b>	<b>27</b>	<b>107</b>	<b>9.90%</b>	

Tabela 5.20: Resultado geral dos veículos não detectados.

Vídeo	Método Proposto				KIM				SANCHEZ et al.			
	Pista 1	Pista 2	Pista 3	Total	Pista 1	Pista 2	Pista 3	Total	Pista 1	Pista 2	Pista 3	Total
Cam06	3	3	1	7	1	4	11	16	10	6	10	26
Cam25	7	1	2	10	15	5	0	20	21	11	5	37
Cam41	0	1	2	3	0	7	5	12	0	25	1	26
Cam64	3	1	4	8	10	2	12	24	21	10	14	45
Cam68	9	3	-	12	2	5	-	7	20	21	-	41
Cam73	0	0	0	0	5	7	10	22	4	1	1	6
<b>Geral</b>	<b>22</b>	<b>9</b>	<b>9</b>	<b>40</b>	<b>33</b>	<b>30</b>	<b>38</b>	<b>101</b>	<b>76</b>	<b>74</b>	<b>31</b>	<b>181</b>

Tabela 5.21: Resultado geral dos veículos detectados múltiplas vezes.

Vídeo	Método Proposto				KIM				SANCHEZ et al.			
	Pista 1	Pista 2	Pista 3	Total	Pista 1	Pista 2	Pista 3	Total	Pista 1	Pista 2	Pista 3	Total
<b>Cam06</b>	1	3	4	<b>8</b>	40	15	2	<b>57</b>	18	14	17	<b>49</b>
<b>Cam25</b>	5	2	3	<b>10</b>	3	2	15	<b>20</b>	42	17	3	<b>62</b>
<b>Cam41</b>	0	6	0	<b>6</b>	0	35	2	<b>37</b>	1	0	0	<b>1</b>
<b>Cam64</b>	4	6	5	<b>15</b>	5	14	1	<b>20</b>	16	7	21	<b>44</b>
<b>Cam68</b>	8	4	-	<b>12</b>	19	8	-	<b>27</b>	20	29	-	<b>49</b>
<b>Cam73</b>	0	3	1	<b>4</b>	4	2	0	<b>6</b>	0	2	1	<b>3</b>
<b>Geral</b>	<b>18</b>	<b>24</b>	<b>13</b>	<b>55</b>	<b>71</b>	<b>76</b>	<b>20</b>	<b>167</b>	<b>97</b>	<b>69</b>	<b>42</b>	<b>208</b>

No entanto, as regiões de agrupamentos semelhantes ao fundo, ou cujos agrupamentos são formados longe um do outro, podem ser divididos e considerados como mais do que um veículo, no método proposto. Entretanto, o método proposto pode dividir um veículo em dois ou, no máximo, três objetos detectados, enquanto que o segundo método comparativo, proposto por Kim, pode dividir veículos, especialmente os mais longos, em mais de cinco objetos diferentes, resultando em um erro de contagem muito maior, devido aos veículos contados múltiplas vezes.

O valor alto da contagem no método proposto por Sanchez et al. ocorre pela divisão de veículos longos em vários objetos e a contagem múltipla de veículos parados sobre os laços virtuais, nos sinais de trânsito. Devido ao fato de o método utilizar uma estratégia de detecção de *background* adaptativa, veículos que permanecem parados por longos períodos de tempo podem ser considerados como parte do fundo da cena, e quando o movimento reinicia, eles são considerados novos veículos, sendo contados novamente.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um método que realiza a detecção e o rastreamento de veículos com o objetivo de obter a contagem de veículos que passam sobre laços virtuais distribuídos na pista, em posições definidas pelo usuário. Como apontado pelos resultados experimentais, o método comprovou ser eficiente na contagem de veículos utilizando vídeos de tráfego urbano, e também obteve resultados melhores que os métodos comparativos, Kim (2008) e Sanchez et al. (2011).

Na contagem geral, o método proposto foi melhor em todos os vídeos, obtendo o valor total de erro de apenas 29 veículos, enquanto o método proposto por Kim obteve 184 e o proposto por Sanchez et al. 107. O método proposto foi melhor também em relação ao número geral de veículos não detectados e de veículos detectados múltiplas vezes, não detectando apenas 40 veículos, e detectando múltiplas vezes 55 veículos.

Alguns veículos podem deixar de serem detectados, principalmente veículos muito pequenos, como motocicletas, pois muitas vezes não apresentam um número mínimo de partículas para formar um agrupamento, e também trafegam entre dois veículos ou muito próximas de outros veículos, fazendo com que não sejam detectadas corretamente. Contudo, veículos grandes, às vezes até mesmo maiores que a área de interesse, são detectados corretamente, e em situações em que não são detectados corretamente, são divididos em no máximo duas ou três partes.

Uma situação difícil de lidar é quando os veículos param na via, devido aos sinais de trânsito. Nessa situação, os veículos ficam muito próximos, o que dificulta a detecção correta dos mesmos. Entretanto, assim que o movimento é retomado, os veículos passam a ser detectados corretamente. Em algumas situações, também pode ocorrer a contagem múltipla dos veículos que estiverem parados sobre os laços virtuais, pois quando o movimento reinicia, a máscara de *foreground* fica um pouco fragmentada por alguns instantes. Outra situação que impõe limitações para o método proposto é a oclusão. Em virtude do ângulo de captura de alguns vídeos, ocorrem algumas situações de oclusões quando os veículos estão muito próximos, geralmente quando estão parados nos sinais de trânsito, ou quando existem veículos grandes, como ônibus, nas pistas mais próximas da câmera, como visto na Cam64 e na Cam68.

Os resultados dos experimentos sugerem que os métodos de contagem e rastreamento de veículos podem ser incorporados aos sistemas existentes de monitoramento de tráfego. No futuro, pretende-se melhorar o método proposto através do desenvolvimento de novos métodos de detecção de veículos e novas técnicas para o rastreamento das partículas que possam lidar adequadamente com problemas causados por oclusões. Também pretende-se realizar mais testes e avaliar o desempenho do método em outras condições climáticas e também outras posições de câmeras.

## 6.1 Trabalhos Futuros

Infelizmente, não existe uma abordagem consolidada capaz de lidar com todas as dificuldades encontradas no monitoramento de tráfego. Essas dificuldades podem ser causadas, por exemplo, por distorções na lente da câmera, mudanças de iluminação, sombras dos veículos, que são detectadas como se fizessem parte dos veículos, por congestionamentos, que fazem com que o tráfego fique lento ou parado, o que impossibilita a utilização de abordagens de detecção baseadas em movimento, além do fato de os veículos andarem muito próximos uns dos outros ocasionar a oclusão de alguns veículos (LI et al., 2013). Outro desafio é ter um método robusto que possa operar de maneira contínua, com câmeras que funcionam dia e noite, sendo necessário processar vídeos que foram capturados através de câmeras de visão noturna e também com as mais diversas condições climáticas.

Todos esses fatores fazem com que encontrar uma solução para cada uma destas situações seja realmente uma tarefa desafiadora, tendo agora que desenvolver métodos mais robustos que lidem com situações e cenários mais complexos e desafiadores, onde os métodos e abordagens tradicionais falham em obter resultados satisfatórios.

Em trabalhos futuros, pretende-se resolver algumas dessas limitações. Futuramente, serão investigados meios para lidar com câmeras que operam de maneira contínua, sofrendo os efeitos das mudanças de iluminação e sombras ao longo do dia, mudanças climáticas, e também a utilização de câmeras noturnas. Outras situações que serão investigadas futuramente são situações de tráfego lento, onde não se tem informação de movimento, e as situações em que a proximidade entre os veículos ocasiona oclusões.

Após resolver essas situações, é planejada a criação de um sistema de monitoramento de fluxo de tráfego, onde será possível determinar trajetórias, contagem e tipo de veículos em cada pista, extrair características e outras informações relevantes dos veículos, que podem ser utilizadas para diversos fins, como detectar violações de trânsito, congestionamentos, entre outros.

Com o desenvolvimento desse sistema, pretende-se fazer também a classificação dos veículos. Através da análise do vídeo será possível detectar automaticamente os veículos que trafegam na pista. Será realizado o rastreamento de cada veículo, permitindo traçar a rota e determinar a trajetória do veículo, a contagem de veículos, e a classificação de cada tipo de veículo. Os veículos detectados serão classificados em diferentes categorias, como, por exemplo, carro, ônibus, motocicleta, etc.

## 6.2 Publicações

Durante o desenvolvimento deste trabalho, foram elaborados artigos científicos para publicação em eventos e periódicos da área, relatando os resultados obtidos e as contribuições do trabalho desenvolvido durante o mestrado. Nesta seção, serão apresentadas resumidamente cada uma dessas publicações.

### ARTIGOS PUBLICADOS EM CONFERÊNCIAS:

1. *Tracking and Counting Vehicles in Traffic Video Sequences Using Particle Filtering*. Bouvie, C.; Scharcanski, J.; Barcellos, P.; Lopes Escouto, F.. Publicada em: **2013 IEEE Instrumentation and Measurement Technology Conference (I2MTC 2013)**.

Resumo do trabalho: Este trabalho apresenta um novo método para rastrear e contar veículos em seqüências de vídeo de tráfego. O método proposto utiliza processamento de imagem, filtragem de partículas, e coerência de movimento para agupar partículas do vídeos, a formando regiões convexas que são analisadas para potenciais veículos. Esta análise leva em consideração a forma convexa dos objetos e informações do *background* da cena para fundir, ou dividir, os agrupamentos. Depois que um veículo é detectado, ele é rastreado utilizando similaridade de histogramas de cor em janelas centradas nas posições das partículas (BOUVIE; SCHARCANSKI; BARCELLOS; ESCOUTO, 2013).

#### ARTIGOS SUBMETIDOS PARA PUBLICAÇÃO EM PERIÓDICOS:

1. *Counting and Tracking Vehicles in Urban Traffic Videos*. Barcellos, P.; Bouvie, C.; Lopes Escouto, F.; Scharcanski, J.. Submetido para: **IEEE Transactions on Instrumentation and Measurement**.

Resumo do trabalho: Este artigo apresenta um método para o rastreamento e contagem de veículos em vídeos de trânsito. Usando técnicas de processamento de imagens e agrupamento de partículas, o método proposto usa coerência de movimento e adjacência espacial para agrupar partículas nas sequências de vídeo. Uma máscara de *foreground* é criada usando o *Gaussian Mixture Model* e *Motion Energy Images* para determinar os locais onde as partículas devem ser geradas, e as formas convexas dos agrupamentos detectados são então analisados para a detecção de veículos. Esta análise leva em consideração a forma convexa do agrupamento de partículas (objetos), e a máscara de *foreground* para unir ou dividir os agrupamentos obtidos. Depois que um veículo é detectado, ele é rastreado utilizando similaridade de histogramas de cor em janelas centradas nos locais de partículas. A contagem de veículos ocorre através da intersecção dos veículos rastreados com laços virtuais definidos pelo usuário. Os testes foram realizados utilizando diferentes vídeos de tráfego, totalizando mais de 80.000 quadros. Os resultados preliminares foram comparados com métodos semelhantes disponíveis na literatura, obtendo resultados equivalentes ou superiores (BARCELLOS; BOUVIE; SCHARCANSKI; FABIANO, 2013).

## REFERÊNCIAS

BARCELLOS, P. et al. Counting and Tracking Vehicles in Urban Traffic Videos. **IEEE Transactions on Instrumentation and Measurement**, [S.l.], 2013. Em processo de submissão.

BARJATYA, A. Block matching algorithms for motion estimation. **IEEE Transactions Evolution Computation**, [S.l.], v.8, n.3, p.225–239, 2004.

BERG, M. de et al. **Computational Geometry Algorithms and Applications**. [S.l.: s.n.], 2008. 386p.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006, 2006. 738p.

BOUTTEFROY, P. L. M. et al. Vehicle tracking by non-drifting mean-shift using projective kalman filter. In: INTELLIGENT TRANSPORTATION SYSTEMS, 2008. ITSC 2008. 11TH INTERNATIONAL IEEE CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2008. p.61–66.

BOUVIE, C. et al. Tracking and Counting Vehicles in Traffic Video Sequences Using Particle Filtering. In: INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE (I2MTC), 2013 IEEE INTERNATIONAL. **Proceedings...** [S.l.: s.n.], 2013. p.812–815.

BUCH, N.; VELASTIN, S. a.; ORWELL, J. A Review of Computer Vision Techniques for the Analysis of Urban Traffic. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.], v.12, n.3, p.920–939, Sept. 2011.

CHAN, Y.-M. et al. Vehicle detection and tracking under various lighting conditions using a particle filter. **IET Intelligent Transport Systems**, [S.l.], v.6, n.1, p.1, 2012.

CHEN, T.-H.; LIN, Y.-F.; CHEN, T.-Y. Intelligent Vehicle Counting Method Based on Blob Analysis in Traffic Surveillance. In: SECOND INT. CONF. INNOVATIVE COMPUTING, INFORMATION AND CONTROL ICICIC '07. **Proceedings...** [S.l.: s.n.], 2007.

CHEN, Y.-L. et al. A real-time vision system for nighttime vehicle detection and traffic surveillance. **IEEE Transactions on Industrial Electronics**, [S.l.], v.58, n.5, p.2030–2044, 2011.

- CHEN, Z.; ELLIS, T.; VELASTIN, S. Vehicle type categorization- a comparison of classification schemes. **International IEEE Conference on Intelligent Transportation Systems**, [S.l.], 2011.
- COMANICIU, D.; RAMESH, V.; MEER, P. Real time tracking of non-rigid objects using mean shift. In: **IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION**, 2000. **Proceedings...** [S.l.: s.n.], 2000. v.2, p.142–149.
- CUCCHIARA, R. et al. Detecting Moving Objects, Ghosts and Shadows in Video Streams. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v.25, n.10, p.1337–1342, 2003.
- DAVIS, J. W.; BOBICK, A. F. The representation and recognition of human movement using temporal templates. **Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, [S.l.], v.23, p.928–934, 1997.
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification (2Nd Edition)**. [S.l.]: Wiley-Interscience, 2000.
- DUFAUX, F.; MOSCHENI, F. Motion estimation techniques for digital TV A review and a new contribution. **Proceedings of the IEEE**, [S.l.], 1995.
- FERIS, R. S. et al. Large-Scale Vehicle Detection , Indexing , and Search in Urban Surveillance Videos. **IEEE Transactions on Multimedia**, [S.l.], v.14, n.1, p.28–42, 2012.
- FILHO, O. M.; NETO, H. V. **Processamento Digital de Imagens**. [S.l.]: Brasport, 1999. 331p.
- FORSYTH, D. A.; PONCE, J. **Computer Vision: a modern approach**. [S.l.]: Prentice Hall Professional Technical Reference, 2002.
- GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing (2nd Edition)**. [S.l.]: Prentice Hall, 2002.
- HU, W. et al. Active contour-based visual tracking by integrating colors, shapes, and motions. **IEEE Transactions on Image Processing**, [S.l.], v.22, n.5, p.1778–92, May 2013.
- JAIN, A. K. **Fundamentals of Digital Image Processing**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.
- JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data Clustering: a review. **ACM Computing Surveys (CSUR)**, New York, NY, USA, v.31, n.3, p.264–323, Sept. 1999.
- JOHNSON, R. A.; WICHERN, D. W. **Applied Multivariate Statistical Analysis (6th Edition)**. [S.l.]: Pearson, 2007.
- JOUBERT, N. **Background modelling and subtraction for object detection in video**. 2009. Tese (Doutorado em Ciência da Computação) — Stellenbosch University.
- KAEWTRAKULPONG, P.; BOWDEN, R. An improved adaptive background mixture model for real-time tracking with shadow detection. In: **European Workshop on Advanced Video-Based Surveillance Systems**. [S.l.]: Springer, 2001.



KANHERE, N. K.; BIRCHFIELD, S. T. Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.], v.9, n.1, p.148–160, 2008.

KASTRINAKI, V.; ZERVAKIS, M.; KALAITZAKIS, K. A survey of video processing techniques for traffic applications. **Image and Vision Computing**, [S.l.], v.21, p.359–381, 2003.

KIM, Z. Real time object tracking based on dynamic feature grouping with background subtraction. In: COMPUTER VISION AND PATTERN RECOGNITION, 2008. CVPR 2008. IEEE CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2008. p.1–8.

LAI, J.-C.; HUANG, S.-S.; TSENG, C.-C. Image-based vehicle tracking and classification on the highway. In: GREEN CIRCUITS AND SYSTEMS (ICGCS), 2010 INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2010. p.666–670.

LI, Y. et al. Vehicle Detection Based on the andor Graph for Congested Traffic Conditions. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.], v.14, n.2, p.984–993, June 2013.

LIANG, W.; WEIMING, H.; TIE-NIU, T. A survey of visual analysis of human motion. **Chinese Journal of Computers**, [S.l.], v.25, n.3, p.225–237, 2002.

LO, B. P. L.; VELASTIN, S. A. Automatic congestion detection system for underground platforms. In: INTERNATIONAL SYMPOSIUM ON INTELLIGENT MULTIMEDIA, VIDEO AND SPEECH PROCESSING, 2001. **Proceedings...** [S.l.: s.n.], 2001. p.158–161.

MAGALHAES, H. A. **Análise em alta resolução de perfis magnéticos de sensores a laço indutivo para classificação de veículos automotores**. 2008. Tese (Doutorado em Ciência da Computação) — UFMG.

MEI, X.; LING, H. Robust visual tracking and vehicle classification via sparse representation. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v.33, n.11, p.2259–72, Nov. 2011.

MOSSI, J. M. et al. Real-time traffic analysis at night-time. **2011 18th IEEE International Conference on Image Processing**, [S.l.], p.2941–2944, Sept. 2011.

PURNAMA, I. K. E. et al. Real time vehicle counter system for Intelligent Transportation System. In: INSTRUMENTATION, COMMUNICATIONS, INFORMATION TECHNOLOGY, AND BIOMEDICAL ENGINEERING (ICICI-BME), 2009 INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2009. p.1–4.

RIBEIRO, H.; GONZAGA, A. Hand Image Segmentation in Video Sequence by GMM : a comparative analysis. **Computer Graphics and Image Processing, 2006. SIBGRAPI '06. 19th Brazilian Symposium on**, [S.l.], p.1–8, 2006.

ROBERT, K. Video-based traffic monitoring at day and night vehicle features detection tracking. In: INT. IEEE CONF. INTELLIGENT TRANSPORTATION SYSTEMS ITSC '09, 12. **Proceedings...** [S.l.: s.n.], 2009. p.1–6.

ROSS, D. A. et al. Incremental Learning for Robust Visual Tracking. **International Journal of Computer Vision**, [S.l.], v.77, p.125–141, 2007.

ROUSSEEUW, P. J. Silhouettes a graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics**, [S.l.], v.20, p.53–65, Nov. 1987.

SANCHEZ, A. et al. Video-Based Distance Traffic Analysis Application to Vehicle Tracking and Counting. **Computing in Science & Engineering**, [S.l.], v.13, n.3, p.38–45, 2011.

SCHARCANSKI, J. et al. A particle-filtering approach for vehicular tracking adaptive to occlusions. **Vehicular Technology, IEEE Transactions on**, [S.l.], v.60, n.2, p.381–389, 2011.

SETCHELL, C. J. **Applications of Computer Vision to Road-traffic Monitoring**. 1997. 185p. Tese (Doutorado em Ciência da Computação) — University of Bristol. (September).

SHEN, Y. **The Research and Applications of Visual Tracking Based on Mean Shift**. 2008. Tese (Doutorado em Ciência da Computação) — Jiangsu University of Science and Technology.

SHI, J.; TOMASI, C. Good features to track. **Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR94**, [S.l.], v.94, p.593–600, 1994.

SONKA, M.; HLAVAC, V.; BOYLE, R. **Image Processing, Analysis, and Machine Vision**. [S.l.: s.n.], 2007. 800p. v.3.

STAUFFER, C.; GRIMSON, W. Adaptive background mixture models for real-time tracking. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1999. **Proceedings...** IEEE Comput. Soc, 1999. v.2, p.246–252.

SZELISKI, R. **Computer Vision Algorithms and Applications**. [S.l.: s.n.], 2010. 832p. v.5.

TAKEUCHI, A.; MITA, S.; MCALLESTER, D. On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods. In: INTELLIGENT VEHICLES SYMPOSIUM (IV), 2010 IEEE. **Proceedings...** [S.l.: s.n.], 2010. p.1014–1021.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition, Fourth Edition**. [S.l.: s.n.], 2008. 164–165p. v.11.

TIAN, B. et al. Video processing techniques for traffic flow monitoring A survey. In: INTELLIGENT TRANSPORTATION SYSTEMS (ITSC), 2011 14TH INTERNATIONAL IEEE CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2011. p.1103–1108.

TSENG, B. L.; LIN, C.-Y.; SMITH, J. R. Real-time video surveillance for traffic monitoring using virtual line analysis. In: MULTIMEDIA AND EXPO, 2002. ICME'02. PROCEEDINGS. 2002 IEEE INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2002. v.2, p.541–544.

TU, Q.; XU, Y.; ZHOU, M. Robust vehicle tracking based on scale invariant feature transform. In: INFORMATION AND AUTOMATION, 2008. ICIA 2008. INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2008. p.86–90.

VIOLA, P.; JONES, M. Robust real-time object detection. **International Journal of Computer Vision**, [S.l.], 2001.

WANG, G.; XIAO, D.; GU, J. Review on vehicle detection based on video for traffic surveillance. In: IEEE INTERNATIONAL CONFERENCE ON AUTOMATION AND LOGISTICS. **Proceedings...** [S.l.: s.n.], 2008. n.September, p.2961–2966.

WARD, J. H. Hierarchical Grouping to Optimize an Objective Function. **Journal of the American Statistical Association**, [S.l.], v.58, n.301, p.236–244, 1963.

XIE, L. et al. Real-time vehicles tracking based on Kalman filter in a video-based ITS. In: COMMUNICATIONS, CIRCUITS AND SYSTEMS, 2005. 2005 INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2005. v.2.

ZHONG, Q.; JUNPING, Z. Method of removing moving shadow based on texture. In: CONTROL AND DECISION CONFERENCE, 2008. CCDC 2008. **Proceedings...** [S.l.: s.n.], 2008. p.2254–2256.