

ARTHUR FERREIRA BAILÃO

MÉTODO AUTOMÁTICO DE CONTAGEM  
VOLUMÉTRICA DE VEÍCULOS BASEADO EM  
VISÃO COMPUTACIONAL

ORIENTADOR: LEISE KELLI DE OLIVEIRA

Belo Horizonte

Maio de 2013



*TODO: dedicatoria*



# Agradecimentos

TODO: Deus, familia, amigos, professores



# Resumo

A Contagem Volumétrica consiste em quantificar o volume de veículos trafegando em uma determinada via, durante um intervalo de tempo definido. A informação coletada é de grande importância pois seus resultados são subsídios básicos para estudos econômicos, projetos rodoviários e planejamento de tráfego. Neste trabalho é proposto e avaliado um método para contagem volumétrica de veículos baseado em Visão Computacional. Se diferencia de outros trabalhos pela simplicidade de implementação, aproveitando de recursos de *software* já consolidados na área, com resultados semelhantes aos de soluções mais complexas.

**Palavras-chave:** Visão Computacional, Contagem Volumétrica de Veículos, Engenharia de Transporte.





# Abstract

TODO: traduzir

**Keywords:** Computer Vision, Vehicle Counting, Transportation Engineering.



# Lista de Figuras

|     |  |    |
|-----|--|----|
| 2.1 | O ser humano é capaz de determinar a forma e a transparência de cada pétala de uma flor [Szeliski, 2010]. . . . .  | 6  |
| 2.2 | Para um computador, o retrovisor de um carro é apenas uma matriz composta por pixels [Bradski & Kaehler, 2008]. . . . .  | 7  |
| 2.3 | Convenção dos eixos para representação de imagens digitais [Gonzalez & Woods, 2000]. . . . .   | 8  |
| 2.4 | (a) Cubo de cores RGB. Os pontos ao longo da diagonal principal têm valores da escala de cinza que vão do preto (0,0,0) ao branco (255,255,255) [Gonzalez & Woods, 2000]; (b) Representação em cores do cubo RGB [Schouten, 2003]. . . . . | 10 |
| 2.5 | (a) Imagem original RGB; (b) Conversão para escala de cinza. . . . .   | 11 |
| 2.6 | Histograma de uma imagem <i>grayscale</i> gerado pelo <i>software</i> GIMP [2013]. .   | 11 |
| 2.7 | Filtragem linear por convolução: a imagem da esquerda convolução com o filtro no centro produz a imagem da direita. Os pixels azuis indica a vizinhança usada para gerar o pixel verde na saída [Szeliski, 2010]. . . . .                  | 12 |
| 2.8 | (a) Imagem original; (b) Aplicação de filtragem linear gaussiana [Laganière, 2011]. Percebe-se o comportamento passa-baixa do filtro atenuando componentes de alta frequência na imagem (bordas). . . . .                                  | 13 |
| 2.9 | Segmentação de uma maçã utilizando limiarização binária [OpenCV Development Team, 2013]. . . . .   | 14 |
| 3.1 | Seis laços indutivos instalado em uma rodovia, dois por faixa [Goldner, 2009].   | 21 |
| 3.2 | Representação esquemática de um cabo piezoelétrico [Goldner, 2009]. . . .  | 23 |
| 3.3 | Esquema ilustrativo da arquitetura de um sistema de monitoramento do tráfego baseado em sensores micro-ondas [Goldner, 2009]. . . . .  | 25 |
| 3.4 | Tipos de montagem para um sensor ultrassônico [Goldner, 2009]. . . . .   | 26 |

|     |   |    |
|-----|---|----|
| 4.1 | (a) Posicionamento da câmera para captura de imagens. A área em destaque simboliza . . . . .                        | 29 |
| 4.2 | Fluxograma com a representação global do método de contagem. . . . .  | 30 |
| 4.3 | Fluxograma do método de rastreamento de <i>keypoints</i> utilizando <i>trackers</i> e contagem de veículos. . . . . | 33 |

# Lista de Tabelas

|     |  |    |
|-----|--|----|
| 3.1 | Comparação entre resultados de tempo de execução do método pelo <i>laptop</i> e celular [Feitosa, 2012]. . . . . | 28 |
| 5.1 | Resultados de contagem volumétrica. . . . .  | 35 |
| 5.2 | Tempo real gasto na execução do método de contagem. . . . .  | 35 |



# Sumário

|  |             |
|--|-------------|
| <b>Agradecimentos</b>                                    | <b>v</b>    |
| <b>Resumo</b>  | <b>vii</b>  |
| <b>Abstract</b>  | <b>ix</b>   |
| <b>Lista de Figuras</b>                                  | <b>xi</b>   |
| <b>Lista de Tabelas</b>                                  | <b>xiii</b> |
| <b>1 Introdução</b>                                      | <b>1</b>    |
| 1.1 Objetivos . . . . .                                  | 3           |
| 1.2 Apresentação da Empresa . . . . .                    | 3           |
| 1.3 Organização do Texto . . . . .                       | 3           |
| <b>2 Visão Computacional</b>                             | <b>5</b>    |
| 2.1 Imagem digital . . . . .                             | 8           |
| 2.2 Modelo RGB de cores . . . . .                        | 9           |
| 2.3 Escala de cinza . . . . .                            | 9           |
| 2.4 Histograma . . . . .                                 | 11          |
| 2.5 Filtragem linear . . . . .                           | 12          |
| 2.6 Limiarização . . . . .                               | 14          |
| 2.7 Biblioteca OpenCV . . . . .                          | 15          |
| 2.8 Aplicações existentes . . . . .                      | 16          |
| <b>3 Contagem de Veículos e Monitoramento do Tráfego</b> | <b>19</b>   |
| 3.1 Detector de laço indutivo . . . . .                  | 21          |
| 3.2 Tubo pneumático . . . . .                            | 22          |
| 3.3 Sensor magnético . . . . .                           | 22          |
| 3.4 Sensor piezoelétrico . . . . .                       | 23          |

|          |   |           |
|----------|---|-----------|
| 3.5      | Câmera de vídeo . . . . .                                     | 24        |
| 3.6      | Sensores micro-ondas . . . . .                                | 24        |
| 3.7      | Sensores ultrassônicos . . . . .                              | 25        |
| 3.8      | Aplicações de Visão Computacional . . . . .                   | 26        |
| 3.8.1    | Detecção de placa de veículos . . . . .                       | 26        |
| 3.8.2    | Contagem volumétrica utilizando dispositivos móveis . . . . . | 27        |
| <b>4</b> | <b>Metodologia</b>  | <b>29</b> |
| 4.1      | Entrada de dados . . . . .                                    | 29        |
| 4.2      | Pré-processamento . . . . .                                   | 30        |
| 4.3      | Subtração de <i>background</i> . . . . .                      | 31        |
| 4.4      | Binarização . . . . .   | 31        |
| 4.5      | Detecção de <i>blobs</i> . . . . .                            | 32        |
| 4.6      | Rastreamento e contagem . . . . .                             | 32        |
| <b>5</b> | <b>Testes e Resultados</b>                                    | <b>35</b> |
| <b>6</b> | <b>Considerações Finais</b>                                   | <b>37</b> |
|          | <b>Referências Bibliográficas</b>                             | <b>39</b> |



# Capítulo 1

## Introdução

O tráfego de veículos representa um fenômeno de grande importância socioeconômica, principalmente nos grandes centros urbanos onde os deslocamentos, no menor tempo possível, são uma necessidade cotidiana. Projetar sistemas viários que absorvam toda a frota veicular desses grandes centros e minimizam os congestionamentos exige ferramentas cujo desenvolvimento ainda representa objeto de estudo para diversos grupos de pesquisadores.

Vários fatores interferem na qualidade do tráfego, sendo o principal o crescente número de veículos nos centros urbanos. Segundo FENABRAVE [2013] - Federação Nacional da Distribuição de Veículos Automotores, as vendas de veículos cresceram cerca de 381% entre janeiro de 2003 e janeiro de 2011. A facilitação ao crédito pessoal, os longos financiamentos oferecidos pelas concessionárias e a redução do IPI (Imposto sobre Produtos Industrializados) incentivaram o consumo de carros no Brasil, explicando essa tendência ascendente no número de automóveis.

A falta de planejamento urbano e o crescente aumento de veículos, a péssima qualidade do transporte público no Brasil e o excessivo número de acidentes de trânsito são umas das principais causas de congestionamentos. Nos últimos anos, milhões de pessoas perderam tempo e dinheiro devido a problemas relacionados ao trânsito [Tancredi, 2012]. Diante desse fato, a administração pública deve priorizar o assunto mobilidade, como tem sido feito nas principais regiões metropolitanas brasileiras. Através de pesquisas pode-se levantar dados que contribuem para análise e simulação do tráfego. Informações como fluxo de veículos, volume de tráfego e matriz O/D (origem/destino) são de grande relevância no estudo dessa área, formando uma base teórica sólida para ser utilizada na resolução de problemas dessa natureza.

Na área de simulação do tráfego existem várias empresas atuantes: PTV [2013] é atualmente a líder mundial de mercado em simulação de fluxo de tráfego multi-modal

de microregiões. Outras companhias, TRL Software [2013] e CITILABS [2013], também fornecem produtos com capacidades de predição, simulação de congestionamentos, atrasos, acidentes, além de gerar relatórios detalhados e animações 2D e 3D. SUMO - *Simulation of Urban MObility* é uma alternativa *open source*<sup>1</sup> e gratuita para simulação do tráfego, diferente dos produtos proprietários e de alto custo citados anteriormente [Behrisch et al., 2011].

Segundo DNIT [2011], o PNCT - Plano Nacional de Contagem de Trânsito vem armazenando nos últimos anos importantes dados de volume de tráfego e já possui uma significativa série histórica desses dados. A informação coletada é de grande importância pois seus resultados são subsídios básicos para estudos econômicos, projetos rodoviários e planejamento de tráfego, além do papel essencial no estabelecimento de critérios para o cumprimento das seguintes finalidades:

- Planejar o sistema rodoviário;
- Programar necessidades e prioridades de melhorias no sistema rodoviário;
- Estabelecer as tendências de tráfego no futuro;
- Avaliar o fluxo existente de tráfego em relação ao sistema rodoviário atual;
- Justificar e planejar o policiamento;
- Estudos de localização de postos de pesagem, socorro médico emergencial, etc.;
- Projetar pavimento, obras de arte, seção transversal e outros elementos de rodovia [DNIT, 2011].

Existem métodos manuais e automáticos para realização das contagens volumétrica e classificatória, que normalmente são de alto custo financeiro. De acordo com a forma de instalação dos equipamentos de contagem, tais métodos podem ser invasivos ou não-invasivos. Os métodos invasivos necessitam de instalações junto ou sob a camada asfáltica. Já os métodos não-invasivos normalmente utilizam câmeras, sensores ou instalações sobre o solo [Goldner, 2009].

Este trabalho tem como propósito desenvolver um método computacional de simples implementação, configuração, instalação e operação, capaz de realizar a contagem volumétrica de veículos de forma não-invasiva e que possa ser utilizado como uma alternativa de baixo custo financeiro, se comparado aos sistemas de contagem volumétrica convencionais.

---

<sup>1</sup>O termo código aberto, ou *open source* em inglês, foi criado pela OSI e refere-se a *software* livre. Mais informações em <http://opensource.org>

## 1.1 Objetivos

O objetivo principal do trabalho é desenvolver um sistema de visão computacional que auxilie na análise das condições do tráfego urbano. São os objetivos específicos deste estudo:

- Entender a utilização de um sistema de visão para problemas de transporte;
- Realizar a contagem volumétrica dos veículos através de um método não-invasivo de simples implementação;
- Analisar as condições do tráfego urbano através de imagens coletadas por uma câmera pessoal.

## 1.2 Apresentação da Empresa

TODO: pegar material com o Henrique.

## 1.3 Organização do Texto

A monografia está dividida em seis capítulos:

- Capítulo 1 é introdutório e são apresentados alguns pontos que motivam e justificam o desenvolvimento do projeto, além de uma listagem dos objetivos específicos que definem o escopo do mesmo.
- Capítulo 2 introduz a teoria de visão computacional, contextualizando o leitor. Trata-se de uma apresentação da tecnologia com foco em conceitos e técnicas de processamento de imagens.
- Capítulo 3 apresenta as tecnologias de contagem volumétrica e classificatória existentes. É também apresentado uma pesquisa bibliográfica sobre sistemas de visão computacional inseridos no contexto da Engenharia de Transporte.
- Capítulo 4 apresenta a metodologia para contagem volumétrica automática de veículos. As técnicas utilizadas, a sequência de operações, as imagens de processamento, as condições de captura e a estrutura do software são mostrados e explicados.

- Capítulo 5 apresenta os resultados e a validação das contagens. Além disso, uma análise comparativa entre os resultados obtidos pelas imagens fornecidas pela BHTRANS e capturadas por uma câmera doméstica.
- Capítulo 6 são feitas as conclusões e as considerações finais do trabalho, bem como recomendações para trabalhos futuros.

## Capítulo 2

# Visão Computacional

Segundo Shapiro & Stockman [2001], visão computacional é um campo da Ciência da Computação que inclui métodos para a aquisição, processamento e análise de imagens. Morris [2004] define processamento de imagem e visão computacional separadamente. Para Morris, processamento de imagem é a manipulação de uma imagem digital, gerando como resultado uma nova representação da mesma imagem, ao passo que visão computacional é a extração de informações numéricas ou simbólicas de imagens. Em linhas gerais, visão computacional é a tecnologia que transforma imagens do mundo real em uma representação que computadores são capazes de interpretar e processar e, com isso, produzir informações úteis para aplicações de Engenharia. Contempla uma base teórica e tecnológica para a construção de sistemas artificiais que obtêm informações de imagens ou quaisquer dados multidimensionais. Mas como os computadores podem entender o mundo visual dos humanos? Quais vantagens e quais tipos de aplicações podem existir a partir desse conceito?

Segundo Szeliski [2010], o ser humano é perfeitamente capaz de perceber a estrutura tridimensional do mundo ao seu redor. Ao visualizar a Figura 2.1, percebe-se que a visão humana interpreta variações de transparência e sombra, além de diferenciar o objeto do *background*<sup>1</sup> com facilidade. Isso acontece porque o cérebro humano divide o sinal de visão em muitos canais, gerando um fluxo de diferentes tipos de informação. Ele é capaz de identificar quais são as partes importantes de uma imagem a serem examinadas e ao mesmo tempo suprimir a atenção para regiões menos importantes [Szeliski, 2010]. Além disso, o cérebro possui um sistema de realimentação poderoso que implementa um controle em malha fechada, composto por sensores (visão, audição, olfato, tato e paladar) e atuadores (íris para controlar a entrada de luminosidade nos olhos) [Bradski & Kaehler, 2008].

---

<sup>1</sup>Segundo plano ou plano de fundo



Figura 2.1: O ser humano é capaz de determinar a forma e a transparência de cada pétala de uma flor [Szeliski, 2010].

Diante da facilidade com que o ser humano enxerga o mundo ao seu redor é intuitivo pensar que o processamento de imagens por visão computacional é simples. Mas isso não é verdade. Em um sistema de visão, o computador recebe, na maioria das vezes, apenas uma matriz de números inteiros, em que cada posição é denominada pixel, como mostrado na Figura 2.2. Todos aqueles padrões de interpretação de informação presentes no cérebro não existem aqui. Além disso, deve-se considerar os ruídos existentes num sistema de visão, que diminuem ainda mais a quantidade de informação dos dados. Esse tipo de problema pode ser causado devido a variações no ambiente (luminosidade, clima, reflexos, movimentações), imperfeições na captura de imagem (lente, configuração mecânica), ruídos elétricos no sensor óptico da câmera e compressão das imagens após a captura [Bradski & Kaehler, 2008].

Mesmo com todos esses desafios, por incrível que pareça, é possível desenvolver sistemas baseados em visão computacional bastante robustos e com alto grau de confiabilidade. Isso começa a se tornar possível quando as imagens capturadas estão inseridas no contexto de uma determinada aplicação. Por exemplo: se um sistema de visão computacional tem por objetivo rastrear carros, não faz nenhum sentido buscar os veículos em áreas que não sejam as ruas, avenidas e rodovias. Caso um objeto seja encontrado em uma área verde ou no azul do céu, a probabilidade de que esse

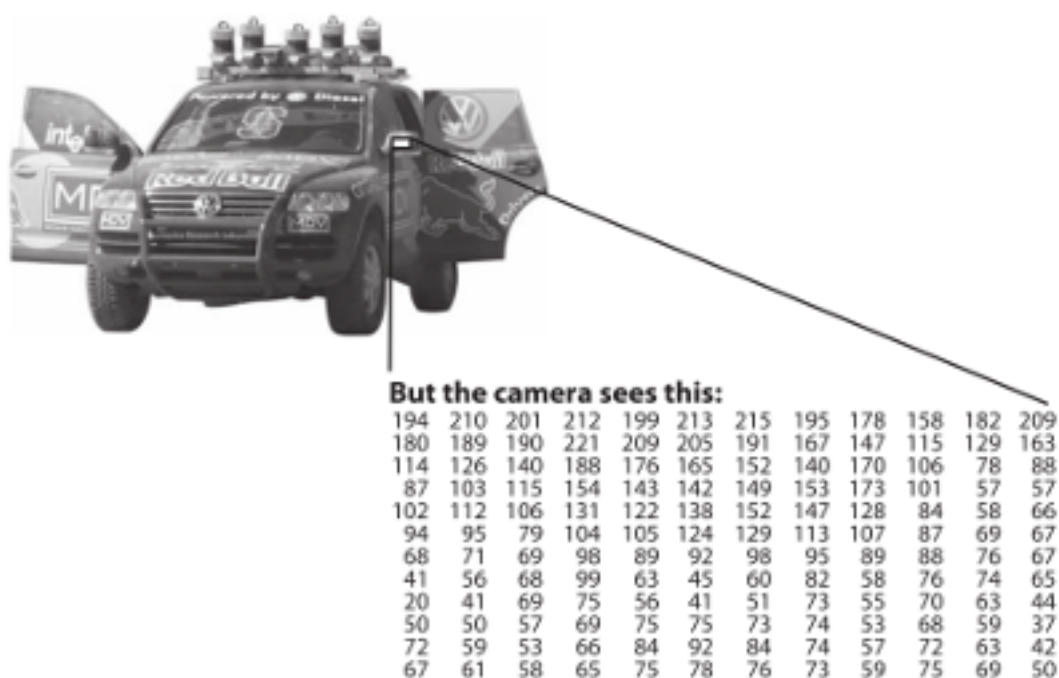


Figura 2.2: Para um computador, o retrovisor de um carro é apenas uma matriz composta por pixels [Bradski & Kaehler, 2008].

objeto seja um carro é muito baixa. É uma análise óbvia mas de suma importância no processamento de imagens.

O uso de métodos estatísticos em visão computacional também são primordiais, indo de encontro ao problema de ruídos discutido anteriormente. Considerar a média dos pixels no tempo é uma abordagem estatística, que pode vir a ser uma solução para problemas que envolvem imagens ruidosas. Outra técnica bastante comum é a construção de modelos das câmeras, capazes de caracterizá-las matematicamente através de seus parâmetros intrínsecos e extrínsecos. Os parâmetros internos da câmera como distância focal, distorções de lente e tamanho do pixel correspondem aos parâmetros intrínsecos. Os parâmetros extrínsecos estão relacionados à orientação e posição da câmera em relação a um sistema de referência no mundo. Com esses parâmetros fica simples corrigir imperfeições nas imagens, que podem ocorrer devido a lente ou alguma configuração mecânica [Bradski & Kaehler, 2008].

Essas e mais uma série de técnicas, descritas nas seções seguintes, são objetos de estudo no campo de visão computacional. Elas em conjunto são combinadas e organizadas de maneira a transformar quaisquer dados multidimensionais em informações de alto nível, como por exemplo ausência e presença de um componente, dimensão e cor de objetos. Como regra geral: quanto mais restrito é o escopo de uma aplicação

de visão computacional, mais o problema pode ser simplificado e mais confiável será a solução final [Bradski & Kaehler, 2008].

A seguir é realizado uma breve revisão bibliográfica sobre conceitos, técnicas e métodos de processamento de imagens digitais aplicados ao campo da visão computacional.

## 2.1 Imagem digital

De acordo com Gonzalez & Woods [2000], uma imagem pode ser definida como uma função bidimensional de intensidade de luz  $f(x, y)$ , em que  $x$  e  $y$  são coordenadas espaciais de um plano, e a amplitude de  $f$  é proporcional ao brilho (ou níveis de cinza) da imagem naquele ponto. A Figura 2.3 ilustra a convenção dos eixos adotada nesse e na maioria dos trabalhos da área.

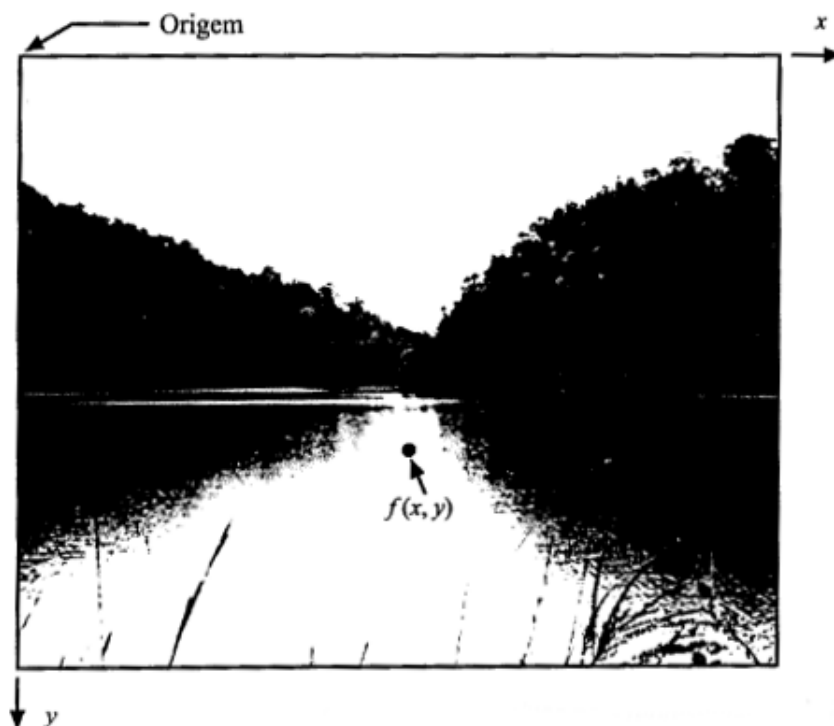


Figura 2.3: Convenção dos eixos para representação de imagens digitais [Gonzalez & Woods, 2000].

Uma imagem digital é a discretização de uma imagem  $f(x, y)$ , tanto em coordenadas espaciais quanto em intensidade de brilho. A OpenCV 2 [Bradski, 2000], uma biblioteca de visão computacional descrita na Seção 2.7, considera a imagem digital como sendo uma matriz cujos índices de linhas e colunas determinam um ponto na



imagem, e o valor correspondente do elemento da matriz identifica o nível de cinza naquele ponto. Os elementos dessa matriz digital são chamados de pixels, embora existam outros nomes como elementos da imagem, elementos da figura ou pels [Gonzalez & Woods, 2000].

## 2.2 Modelo RGB de cores

Segundo Gonzalez & Woods [2000], é comprovado que o olho humano possui entre 6 e 7 milhões de cones<sup>2</sup>, que podem ser divididos em três principais categorias de sensoria-mento, aproximadamente correspondentes às cores vermelho, verde e azul. Desse total de cones existentes, cerca de 65% são sensíveis à luz vermelha, 33% são sensíveis à luz verde e apenas 2% deles são sensíveis à luz azul, sendo os cones azuis os que possuem maior grau de sensibilidade à luz. Dessa forma, as cores são vistas pelo olho humano como combinações das chamadas cores primárias: vermelho (R, *red*), verde (G, *green*) e azul (B, *blue*), fato que inspirou estudos do modelo RGB de cores.

O modelo RGB é um sistema de coordenadas cartesianas, onde o sub-espço das cores de interesse é representado pelo cubo da Figura 2.4a. Nesse modelo, as cores são pontos sobre ou dentro do cubo, definidas por vetores que estendem-se a partir da origem. A escala de cinza é definida pela diagonal principal desse cubo e assume valores que vão do preto, localizado na origem, ao branco, no vértice mais distante [Gonzalez & Woods, 2000].

Em representações computacionais desse modelo os componentes R, G e B, denominados canais de cor, assumem valores inteiros de uma escala que vai de 0 a 255, possibilitando que cada valor seja armazenado por 1 *byte*<sup>3</sup>. Dessa forma, é possível gerar exatamente 16.777.216 (ou  $256^3$ ) combinações de cores diferentes. A Figura 2.4b ilustra o gradiente formado na superfície do cubo RGB quando os pontos assumem valores inteiros num intervalo de 0 a 255.

## 2.3 Escala de cinza

A escala de cinza é muito utilizada em processamento de imagens digitais por ter menor custo computacional. Em imagens RGB são necessários 3 *bytes* (um por canal) para definir a cor de um pixel, enquanto que em imagens *grayscale*<sup>4</sup> a representação

---

<sup>2</sup>Células fotorreceptoras, localizadas na retina do olho, que são responsáveis pela visão das cores.

<sup>3</sup>Unidade de medida de armazenamento de memória em computação. Cada *byte* contém 8 *bits*, suficiente para armazenar  $2^8$  números binários.

<sup>4</sup>Escala de cinza.

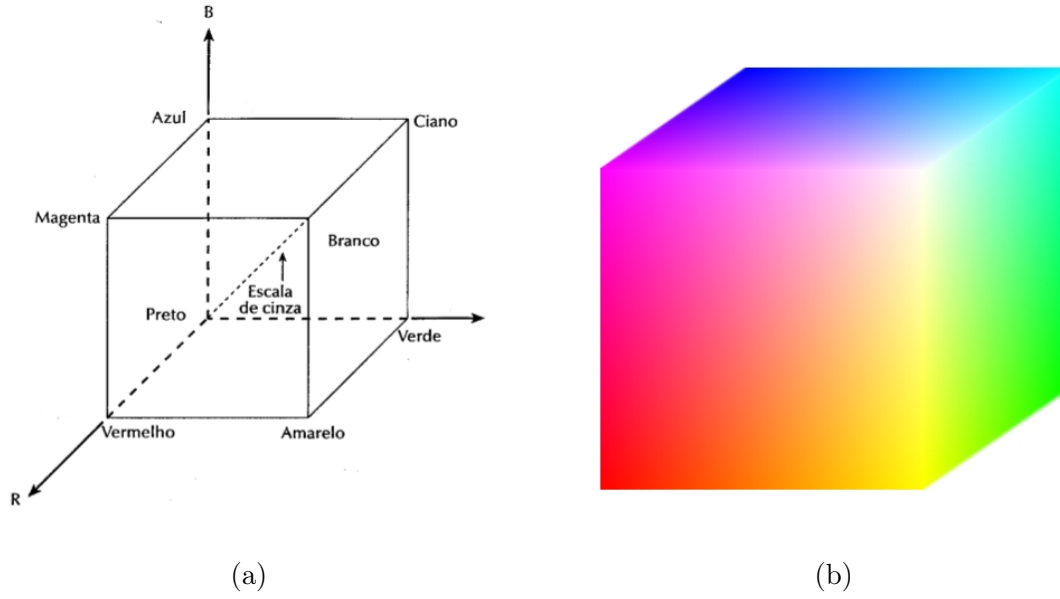


Figura 2.4: (a) Cubo de cores RGB. Os pontos ao longo da diagonal principal têm valores da escala de cinza que vão do preto (0, 0, 0) ao branco (255, 255, 255) [Gonzalez & Woods, 2000]; (b) Representação em cores do cubo RGB [Schouten, 2003].

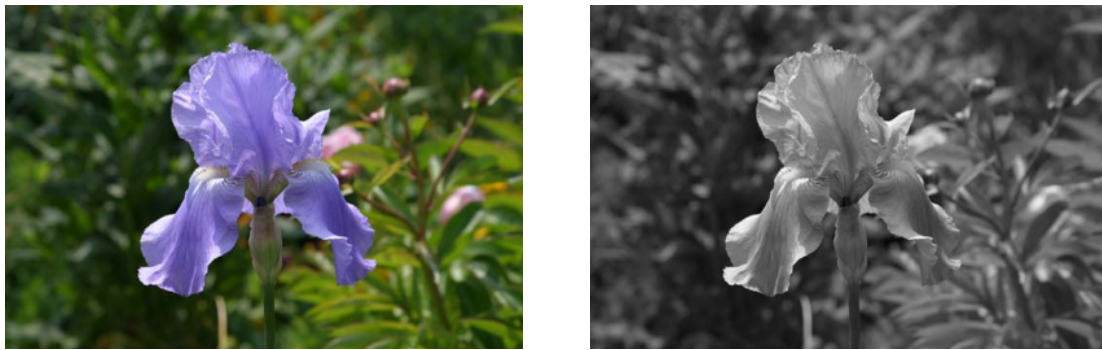
da cor de um pixel utiliza apenas 1 *byte*, um terço de uma imagem RGB. De acordo com Gonzalez & Woods [2000], sistemas de visão computacional em que a cor não é um requisito, é essencial converter as imagens RGB para *grayscale* antes de qualquer operação, como mostrado na Figura 2.5, diminuindo assim o peso do processamento digital.

Para obter uma imagem *grayscale* a partir de uma imagem RGB deve-se levar em conta a média da intensidade de cada canal. Essa conversão é dada por

$$C_{xy} = (k_R \times R_{xy}) + (k_G \times G_{xy}) + (k_B \times B_{xy}), \quad (2.1)$$

$$k_R, k_G, k_B > 0 \text{ e } k_R + k_G + k_B = 1,$$

onde  $x$  e  $y$  representam a posição do pixel no plano espacial da imagem segundo a convenção de eixos descrita na Seção 2.1;  $C_{xy}$  a intensidade resultante dos pixels na escala de cinza;  $R_{xy}$ ,  $G_{xy}$  e  $B_{xy}$  a intensidade dos pixels dos canais vermelho, verde e azul, respectivamente; e  $k_R$ ,  $k_G$  e  $k_B$  constantes relacionadas à sensibilidade de incidência de luz de cada canal de cor [Johnson, 2006].



(a)

(b)

Figura 2.5: (a) Imagem original RGB; (b) Conversão para escala de cinza.

## 2.4 Histograma

Histograma é um gráfico matemático que representa a distribuição de frequências de uma massa de medições [Lancaster, 1973]. Normalmente é representado por retângulos na vertical, onde cada retângulo corresponde a um intervalo e sua altura é proporcional ao número de ocorrências naquele intervalo.

Em visão computacional os histogramas são utilizados para representar o número de ocorrências de cada cor em uma imagem, como ilustra a Figura 2.6. Normalmente é gerado um histograma para cada canal de cor em imagens coloridas, já que pode existir uma infinidade de cores possíveis. O histograma pode proporcionar um melhor entendimento da imagem pois facilita a visualização de parâmetros como contraste e luminosidade.

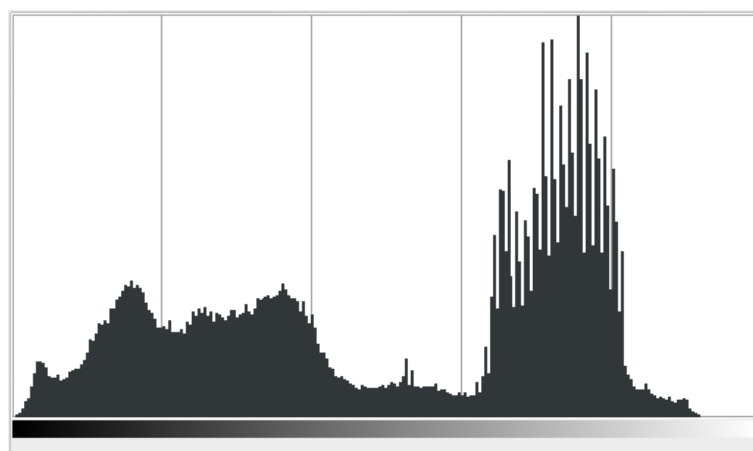


Figura 2.6: Histograma de uma imagem *grayscale* gerado pelo *software* GIMP [2013].

## 2.5 Filtragem linear

A filtragem linear é um tipo de operação local que leva em conta pixels vizinhos de um determinado ponto para calcular seu novo valor, como ilustrado na Figura 2.7. Operadores locais, ou operadores de vizinhança<sup>5</sup> como são mais conhecidos, são muito utilizados no processamento digital de imagens e podem ser usados como filtros espaciais lineares para gerar efeitos de borramento, acentuação de bordas, remoção de ruídos, entre outros.

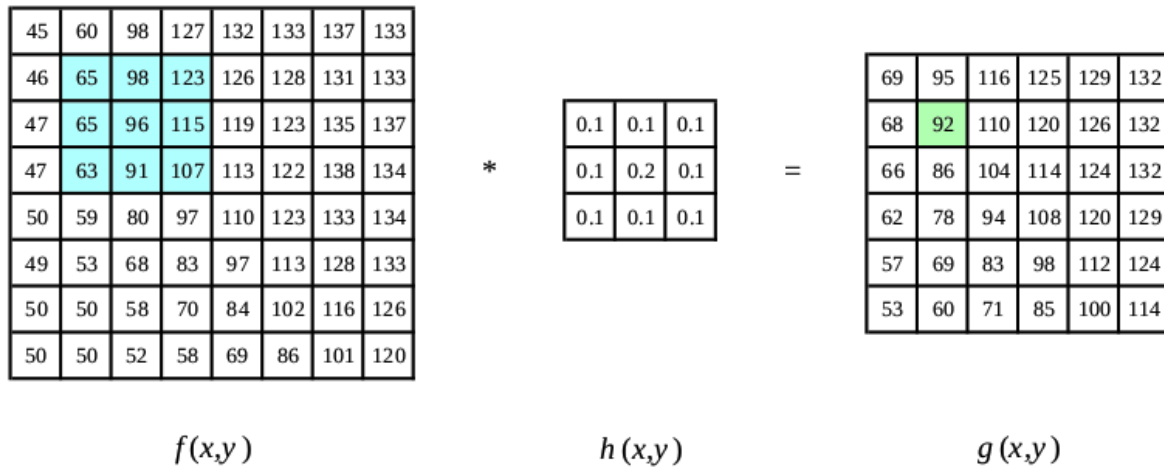


Figura 2.7: Filtragem linear por convolução: a imagem da esquerda convolução com o filtro no centro produz a imagem da direita. Os pixels azuis indica a vizinhança usada para gerar o pixel verde na saída [Szeliski, 2010].

De acordo com Szeliski [2010], filtros lineares são os operadores de vizinhança mais usados, em que o valor de saída  $g(i, j)$  é determinado por uma soma ponderada de valores de pixels de entrada  $f(i + k, j + l)$  (Figura 2.7),

$$g(i, j) = \sum_{k,l} f(i + k, j + l)h(k, l). \quad (2.2)$$

A ponderação é dada pelo *kernel* ou máscara  $h(k, l)$ , que contém os coeficientes do filtro, responsáveis pelas características de suavização, borramento ou remoção de ruídos inerentes a cada tipo de filtragem. A Equação (2.2) é conhecida como operador de **correlação**, que na forma compacta pode ser escrito como

$$g = f \otimes h. \quad (2.3)$$

---

<sup>5</sup>Tradução livre de *neighborhood operators*

Uma variação muito comum de (2.2) é

$$g(i, j) = \sum_{k, l} f(i - k, j - l)h(k, l) = \sum_{k, l} f(k, l)h(i - k, j - l), \quad (2.4)$$

onde o sinal dos deslocamentos em  $f$  foram invertidos. Essa é a chamada operação de **convolução**,

$$g = f * h. \quad (2.5)$$

As operações de correlação e convolução são equivalentes, a não ser pela fase de  $180^\circ$  que gera uma rotação no processo de filtragem por correlação. Elas são a base matemática para implementação de qualquer tipo de filtro linear, como por exemplo, **gaussiano**.



(a)



(b)

Figura 2.8: (a) Imagem original; (b) Aplicação de filtragem linear gaussiana [Laganière, 2011]. Percebe-se o comportamento passa-baixa do filtro atenuando componentes de alta frequência na imagem (bordas).

Segundo OpenCV Development Team [2013], o filtro gaussiano é talvez o mais útil entre todos. É largamente utilizado no processamento de imagens digitais para remoção de ruídos e redução de detalhes, como ilustrado na Figura 2.8, onde pode-se verificar um suave borramento e a diminuição de detalhes nas bordas, características típicas de um filtro passa-baixa. A filtragem é feita através da convolução de cada pixel da imagem de entrada por um *kernel* cujos valores obedecem uma distribuição gaussiana bidimensional, representada por

$$h(k, l) = Ae^{\frac{-(k - \mu_k)^2}{2\sigma_k^2} + \frac{-(l - \mu_l)^2}{2\sigma_l^2}}, \quad (2.6)$$

onde  $\mu$  é a média e  $\sigma$  o desvio padrão. Nessa configuração, os pixels mais próximos de  $\mu$  têm maior peso no resultado, portanto são mais significativos para o cálculo da intensidade de cada ponto. O peso dos pixels vizinhos decai com a distância e é influenciado também pelo desvio padrão  $\sigma$ .

## 2.6 Limiarização

Segundo Bradski & Kaehler [2008], a operação de limiarização, também conhecida por *thresholding*, é o método de segmentação em imagens digitais mais simples que existe. Essa técnica pode ser aplicada para separar regiões correspondentes a objetos que se deseja analisar, como iustrado na Figura 2.9.



Figura 2.9: Segmentação de uma maçã utilizando limiarização binária [OpenCV Development Team, 2013].

A separação se dá pela diferença de intensidade entre os pixels do objeto e os pixels pertencentes ao plano de fundo. Para diferenciar os pixels de interesse é feito uma comparação do nível de cinza de cada pixel com um limiar (ou valor de *threshold*), que pode ser uma constante determinada de acordo com as necessidades do problema ou uma variável em algoritmos de limiarização adaptativos. Uma vez que os pixels de interesse foram segmentados, basta determinar um valor comum para identificá-los, sendo mais comum o branco 255 (Figura 2.9).

No caso geral, a operação de limiarização binária é dada por

$$dst(x, y) = \begin{cases} maxVal & \text{se } src(x, y) > thresh \\ 0 & \text{caso contrário.} \end{cases} \quad (2.7)$$

Portanto, se a intensidade do pixel  $src(x, y)$  é maior que o limiar  $thresh$ , então a nova intensidade do pixel  $dst(x, y)$  será  $maxVal$ ; caso contrário, é atribuído a  $dst(x, y)$  o valor 0 (preto) [Bradski & Kaehler, 2008].

## 2.7 Biblioteca OpenCV

OpenCV (*Open Source Computer Vision*) [Bradski, 2000] é uma biblioteca de visão computacional *open source*<sup>6</sup> disponível em: <http://sourceforge.net/projects/opencvlibrary/>. O código é escrito em C e C++ e pode ser compilado e executado em ambientes Linux, Windows e Mac OS X. Existem também interfaces em Python, Ruby, Matlab e outras linguagens. Contém mais de 500 algoritmos otimizados para processamento de imagens e vídeos, cobrindo diversas áreas da visão computacional, como: inspeção industrial, imagens médicas, segurança, interface com o usuário, calibração de câmera, visão estéreo, robótica e *machine learning* [Bradski & Kaehler, 2008].

Segundo Laganière [2011], desde o lançamento da OpenCV em 1999, ela vem sendo largamente adotada como a principal ferramenta de desenvolvimento pela comunidade de pesquisadores e programadores em visão computacional. A biblioteca foi originalmente desenvolvida pela Intel [Intel, 2013], por um time liderado por Gary Bradski e com o propósito de avançar em pesquisas na área de visão. Depois de uma série de versões *Beta*, a versão 1.0 foi lançada em 2006. O segundo maior lançamento aconteceu em 2009 com a OpenCV 2, propondo importantes modificações em sua estrutura e especialmente a nova interface C++. Atualmente a OpenCV encontra-se na versão 2.4.

Bradski & Kaehler [2008] afirmam que a licença *open source* da OpenCV permite que aplicações comerciais possam ser construídas utilizando parte ou toda a biblioteca, sem a necessidade de que o código da aplicação seja aberto. Devido a essa política liberal de uso, gigantes como IBM, Microsoft, Intel, SONY, Siemens, Google e outros, além dos centros de pesquisa Stanford, MIT, CMU, Cambridge, INRIA e outros, utilizam a biblioteca em seus projetos e pesquisas. A OpenCV foi peça chave no sistema

---

<sup>6</sup>O termo código aberto, ou *open source* em inglês, foi criado pela OSI e refere-se a *software* livre. Mais informações em <http://opensource.org>

de visão de um robô desenvolvido em Stanford, conhecido como Stanley, que ganhou a corrida de carros autônomos \$2M DARPA Grand Challenge.

## 2.8 Aplicações existentes

Invent Vision [2013] possui diversas aplicações concebidas utilizando os conhecimentos de visão computacional. O fato dessa tecnologia ser não intrusiva, ou seja, não altera em nada o meio em que está sendo utilizada, torna os sistemas de visão realizáveis em grande parte dos processos industriais, urbanos e ambientais.

As aplicações em análise dimensional se caracterizam por efetuarem medidas em objetos, sendo elas lineares e/ou angulares. A análise dimensional por imagem é vantajosa pois possibilita que as medidas sejam feitas à distância, quando não é possível ou desejável o contato direto com o objeto. Câmeras de alta resolução se aplicam nesses projetos, garantindo medições precisas e com repetibilidade. Muito comum na siderurgia, esse tipo de aplicação possibilita a medição em objetos que se encontram em altas temperaturas. As câmeras infravermelho tem sido utilizadas nesse tipo de aplicação.

O uso de reconhecimento de padrões permite a identificação de características de um produto comparando-o com um modelo predeterminado. Essas características a serem inspecionadas são escolhidas de modo a identificar e diferenciar um tipo de modelo de outro. Assim é possível dizer se um objeto está conforme um padrão ou não.

Inspeção de nível é bastante comum na indústria de bebidas e na indústria farmacêutica para inspeção de enchimento de ampolas, vidros de medicamentos ou qualquer recipiente translúcido que contenha líquido.

A inspeção por análise de cores possibilita a criação de soluções para separar produtos por cor ou verificar se a tonalidade está igual a uma amostra. Tem a vantagem de ser um método determinístico em relação a uma inspeção de cores realizada por seres humanos, onde pode existir subjetividade. No entanto, variações de luminosidade podem dificultar a realização desse tipo de inspeção.

Existem também aplicações de rastreabilidade que envolvem leituras de códigos de barras ou de códigos bidimensionais como *Data Matrix*<sup>7</sup> e o *QR Code*<sup>8</sup>. Permitem

---

<sup>7</sup>Código de barras matricial bi-dimensional que consiste em células brancas ou pretas arranjadas em forma de quadrado ou retângulo. Se caracteriza pelas bordas em formato de L. Armazena no máximo 2335 caracteres alfanuméricos.

<sup>8</sup>*Quick Response Code* é um código de barras bi-dimensional criado pela empresa japonesa Denso-Wave em 1994. Pode armazenar até 7089 caracteres, dependendo do tipo de dado. Possui redundância em sua codificação, possibilitando correção e recuperação de informação.



identificar e rastrear os produtos, bem como armazenar grandes quantidades de dado. São sistemas frequentemente utilizados na indústria automobilística mas são aplicáveis em qualquer projeto que necessite de controle de produção.

As aplicações para contagem, seleção e classificação podem instrumentar diversas indústrias, como por exemplo a indústria farmacêutica para contagem de medicamentos. Também é comum na produção de componentes eletrônicos para verificação de pinos e conectores. Na engenharia de transporte podem contribuir na análise do tráfego realizando contagem volumétrica de veículos e pessoas.



## Capítulo 3

# Contagem de Veículos e Monitoramento do Tráfego

De acordo com DNIT [2011], Departamento Nacional de Infra-Estrutura de Transportes, a contagem volumétrica consiste em quantificar o volume de veículos que trafegam por um determinado trecho da rodovia, durante um intervalo de tempo definido. A contagem de veículos é importante para o cumprimento de diversas finalidades, dentre elas: planejamento do sistema rodoviário, estabelecimento de tendências de tráfego futuro, determinação do volume de viagens de forma a proporcionar justificativa econômica aos investimentos programados, avaliação do fluxo existente de tráfego em relação ao sistema rodoviário atual, planejamento e justificativa do policiamento, realização de análise estatística de acidentes, estudos de localização de postos de pesagem, socorro médico emergencial, entre outros.

O volume de veículos, obtido através da contagem, pode ser classificado como [Goldner, 2009]:

- **Volume de tráfego:** quantidade de veículos em tráfego na seção de uma via, em um período de tempo definido.
- **AADT ou VMDA:** volume médio diário de veículos durante o ano, ou seja, a somatória anual, dividido por 365.
- **ADT ou VMD:** volume diário do tráfego ou volume médio diário. Também é usado para intervalos de tempo diferentes, representando o volume total durante dado período, dividido pela quantidade de dias do período. Assim tem-se:
  - **VMDm:** volume médio diário mensal. Número total de veículos em um mês, dividido pelo número de dias do mês.

- **VMDs**: volume médio diário semanal. Número total de veículos em uma semana, dividido por 7. É sempre acompanhado pelo nome do mês a que se refere.
- **VMDd**: Volume médio diário em um dia de semana. Deve ser sempre acompanhado pela indicação do dia de semana e do mês correspondente.
- **Composição do tráfego**: porcentagem dos diferentes tipos de veículos que compõem o tráfego, por exemplo, automóveis, caminhões, ônibus e motos.
- **Volume abreviado**: volume do fluxo de um período menor do que 1 hora.
- **Variações do volume de tráfego**: mudanças no volume em um determinado período, divididas em: variações sazonais ou mensais ao longo do ano; variações diárias ao longo da semana; variações horárias ao longo do dia; e variações dentro da hora.

Goldner [2009] define dois métodos de contagem: **manual** e **mecânica**. A contagem manual utiliza recursos humanos, ou seja, pesquisadores observam o fluxo de veículos e manualmente fazem suas anotações. Esse procedimento é vantajoso pela precisão e variedade nas informações de tráfego obtidas, como tipo e tamanho dos veículos, além de proporcionar flexibilidade, simplicidade e rapidez de execução. No entanto, é um método com limitações de cobertura e de alto custo. A contagem mecânica utiliza detectores de tráfego de instalações permanente ou móvel. Possui algumas vantagens, dentre elas: baixo custo operacional, grande amplitude de tempo de cobertura e boa precisão. Por outro lado não fornecesse muitas informações e demanda investimento inicial alto.

A classificação feita por Goldner peca ao usar o termo contagem mecânica para caracterizar métodos não manuais. Grande parte dos equipamentos de contagem volumétrica são dispositivos eletrônicos, por exemplo sensores micro-ondas, ultra-sônicos e piezoelétricos, descaracterizando o uso dessa nomenclatura. Até mesmo o tubo pneumático, que pode ser considerado um dispositivo mecânico, utiliza sinais elétricos para transmissão dos dados. Por esse motivo, será adotado o termo **contagem automática** para descrever os métodos de contagem que não dependem diretamente da participação humana durante o processo de obtenção dos dados.

Por sua vez, os métodos automáticos de contagem podem ser classificados como **invasivos** e **não-invasivos**. Os métodos invasivos necessitam de instalações junto ou sob a camada asfáltica. Já os métodos não-invasivos não modificam a estrutura da via, com instalações acima ou às margens da faixa de tráfego [Goldner, 2009].

Nas seções seguintes são apresentados alguns equipamentos automáticos de contagem volumétrica, descritos em Goldner [2009], de Almeida [2010] e DNIT [2011], bem como algumas aplicações de visão computacional inseridas na área de Engenharia de Transporte.

### 3.1 Detector de laço indutivo

São os sensores mais utilizados para a coleta de dados de tráfego. É composto basicamente por um circuito elétrico oscilador, uma unidade eletrônica de processamento, cabos de transmissão e o laço indutivo propriamente dito, que é formado por uma ou mais voltas de um cabo isolado instalado sob o pavimento da via (Figura 3.1), caracterizando-o como equipamento invasivo.



Figura 3.1: Seis laços indutivos instalado em uma rodovia, dois por faixa [Goldner, 2009].

O laço funciona como indutor de um circuito oscilador de frequência fixa, normalmente entre 10 KHz e 200 KHz. Quando objetos de metal transitam nas proximidades do equipamento instalado no solo, acontece uma interação entre os campos magnéticos induzidos no laço e no objeto metálico, que normalmente é um veículo, provocando variações na indutância do laço e, portanto, variações na frequência de oscilação do circuito ressonante. A presença de um veículo na região do laço provoca um aumento na frequência de oscilação do circuito, comportamento que é interpretado pela unidade de processamento capaz de realizar a contagem.

Cada veículo que passa pelo laço indutivo possui um perfil magnético característico que depende de vários fatores, dentre eles, seu tamanho, formato, condutividade

e orientação em relação ao laço. Isso possibilita, além da contagem, realizar uma classificação veicular através de métodos de clusterização por redes neurais, como descrito em [de Almeida, 2010].

A vantagem de se usar laços indutivos para monitoramento é o bom funcionamento na captura de dados básicos do tráfego, como volume e velocidade, além de apresentar baixo custo se comparado à equipamentos não-invasivos. Por outro lado, o processo de instalação do equipamento junto ao solo gera a paralisação no tráfego local e muitas vezes causa transtornos no trânsito.

## 3.2 Tubo pneumático

Os sensores de contagem volumétrica por tubo pneumático funcionam a partir da pressão exercida pelos pneus de um veículo assim que passam sobre o tubo de borracha. A massa de ar é então deslocada ao longo do tubo até o ponto de conversão, onde um interruptor converte o sinal pneumático em sinal elétrico, para que esse possa ser transmitido a um *software* de análise ou a um contador.

O tubo é instalado sobre a via, perpendicular à direção do fluxo do tráfego, caracterizando-se como um equipamento invasivo. É normalmente utilizado para contagem de tráfego em períodos curtos, classificação dos veículos por número de eixos e espaçamento, medição de velocidade e pesquisas acadêmicas.

Apesar de ser a primeira tecnologia de detecção de tráfego, inventada em 1920, é ainda muito utilizada devido ao baixo custo e a simples instalação e operação. No entanto, o equipamento é impreciso na contagem de caminhões ou ônibus muito largos, além de apresentar baixa durabilidade e sensibilidade à temperatura.

## 3.3 Sensor magnético

O princípio de funcionamentos dos sensores magnéticos é baseado em variações das linhas de fluxo do campo magnético terrestre. Tal efeito é conhecido como anomalia magnética, causada pela aproximação de objetos metálicos (veículos) da área de cobertura do sensor. Por indução eletromagnética, variações de tensão são geradas e posteriormente amplificadas, digitalizadas e finalmente transmitidas ao controlador eletrônico que realiza a contagem.

São utilizados para medir volume, direção, presença e velocidade dos veículos. Podem ser divididos em dois tipos: os magnetômetros de indução, que se baseiam em mudanças nas linhas de fluxo do campo magnético devido ao movimento, não detectam

veículos parados na via; e os magnetômetros de eixo duplo, que detectam mudanças nos componentes horizontais e verticais do campo magnético terrestre de acordo com a densidade do metal do veículo, e são capazes de identificar objetos em movimento ou parados.

A utilização destes sensores é vantajosa devido a pouca sensibilidade à temperatura ambiente e ao tráfego intenso. Suas desvantagens são a pequena zona de detecção, a dificuldade em considerar veículos parados e a sua característica de instalação invasiva, fazendo-se necessárias paralizações do tráfego e interferências na camada asfáltica.

### 3.4 Sensor piezoelétrico

Um material piezoelétrico é capaz de converter energia cinética em energia elétrica. Esses materiais geram uma tensão quando submetidos a choques mecânicos ou vibrações. Portanto, quando um veículo passa sobre um detector, o sensor gera uma tensão proporcional ao peso do veículo. É construído a partir de um cabo piezoelétrico do tipo coaxial, com um núcleo de metal, seguido pelo material piezoelétrico e uma camada externa de metal, como ilustrado na Figura 3.2.

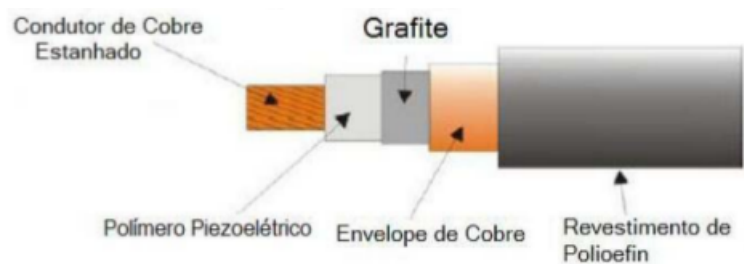


Figura 3.2: Representação esquemática de um cabo piezoelétrico [Goldner, 2009].

Devido a sua grande precisão, esse tipo de sensor é capaz de medir volume do tráfego, velocidade, peso e ainda classificar os veículos baseado no espaçamento e contagem de eixos. Sua instalação é feita junto à via, caracterizando-o como invasivo, e não é recomendado em solos irregulares. Apresenta alto custo se comparado a outros equipamentos invasivos, mas em contrapartida são mais precisos e adquirem informações mais completas.

### 3.5 Câmera de vídeo

As câmeras de vídeo são utilizadas no monitoramento do tráfego, na maioria das vezes, como uma simples ferramenta de vigilância urbana, com capacidades de transmissão e gravação de imagens. Nesse cenário, as imagens devem ser interpretadas por um operador humano, que tem a função de extrair dados de tráfego. Com o avanço das áreas de processamento digital de imagens e visão computacional, surgiram diversos sistemas de gerenciamento do tráfego capazes de extrair automaticamente informações de tráfego importantes, como velocidade, volume, presença, ocupação, densidade, movimentos de conversão, mudança de faixa, aceleração, classificação de veículos e outros [Martinsky, 2007; Feitosa, 2012].

Um sistema inteligente de processamento de imagens de vídeo consiste em uma ou mais câmeras, um *hardware*<sup>1</sup> para digitalização e processamento e um *software* para interpretação e conversão de imagens em dados de tráfego, numéricos ou simbólicos. Segundo Feitosa [2012], esse tipo de sistema pode, com apenas uma câmera, substituir vários equipamentos invasivos, como laços indutivos, e proporcionar detecção de veículos em múltiplas faixas com menores custos de manutenção.

Dentre as vantagens de uso de câmeras está a instalação, que é totalmente não-invasiva e não gera qualquer tipo de transtorno ao trânsito. Uma vez capturadas, as imagens do tráfego podem ser analisadas posteriormente por profissionais capazes de, por exemplo, realizar a contagem volumétrica manual ou periciar um acidente. Mudanças naturais na cena, como chuvas, tempo nublado, o dia e a noite, podem interferir na visibilidade, dificultando tanto a interpretação humana quanto o processamento digital por visão computacional. Deslocamentos indesejados da câmera e problemas de oclusão também são comuns.

### 3.6 Sensores micro-ondas

Esse tipo de equipamento transmite radiação de micro-ondas de baixa energia na direção de uma área do pavimento utilizando uma antena localizada na parte superior do radar. Quando um veículo atravessa um feixe, uma parte dessa energia é refletida de volta para o detector, também localizado na parte superior do radar, e dados como velocidade e presença são calculados por um controlador. A Figura 3.3 ilustra a configuração mais comum de um sistema de monitoramento baseado em sensores micro-ondas.

Esses sensores podem ser de dois tipos:

---

<sup>1</sup>Parte física de computadores ou quaisquer equipamentos eletrônicos.



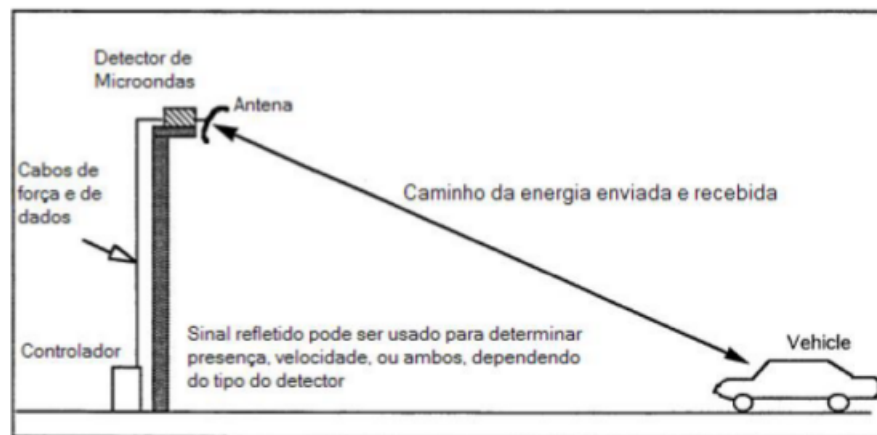


Figura 3.3: Esquema ilustrativo da arquitetura de um sistema de monitoramento do tráfego baseado em sensores micro-ondas [Goldner, 2009].

- **Doppler:** mede a presença de um veículo em função do movimento relativo entre uma fonte sonora e seu receptor e que, por efeito *Doppler*, varia a frequência recebida na volta. Como é um método baseado em movimento, não considera veículos parados ou em regime de “anda-pára”.
- **Radar:** utiliza um sinal de frequência ou fase modulada para calcular o atraso de tempo da onda refletida, obtendo a distância do veículo. Nesse método é possível detectar a presença de veículos parados, além de medir velocidade, monitorar filas e ocupação.

Os sensores micro-ondas são não-invasivos e não interferem no tráfego de veículos no processo de instalação. Geralmente não apresentam sensibilidade em más condições climáticas e podem ser construídos para suportar múltiplas pistas. No entanto, um dos tipos não é capaz de detectar veículos parados.

### 3.7 Sensores ultrassônicos

Esses detectores transmitem ondas de pressão de energia sonora acima da frequência audível humana, que é de 20 KHz. Estas ondas sonoras refletem no pavimento ou no veículo, são captadas por um receptor e processadas para fornecer informações de passagem e presença de veículos. A montagem não-invasiva do equipamento pode ser feita acima ou ao lado da via, como ilustra a Figura 3.4.

Existem dois tipos de sensores ultrassônicos:

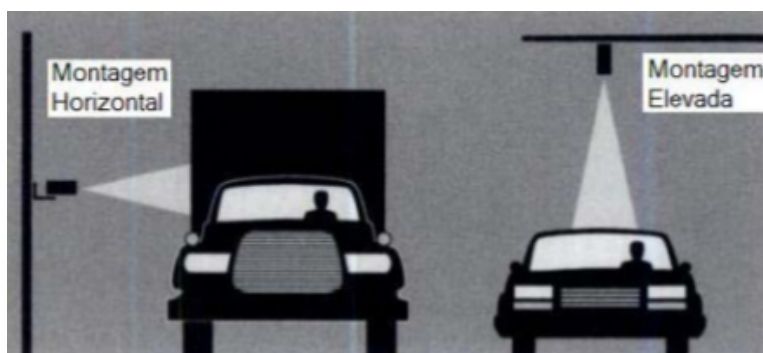


Figura 3.4: Tipos de montagem para um sensor ultrassônico [Goldner, 2009].

- **Pulso ultrassônico:** são emitidos pulsos de energia com largura e período conhecidos. Se o tempo do pulso refletido for menor que o valor conhecido, a presença do veículo é acusada, fornecendo informações de altura, largura, ocupação, volume e classificação.
- **Onda ultrassônica contínua:** usam o princípio de *Doppler*, que leva em conta a variação da frequência causada pela velocidade relativa entre onda emitida e refletida, para determinar a presença, volume e velocidade veicular.

O uso de sensores ultrassônicos é vantajoso pois a instalação é não-invasiva e pode monitorar mais de uma pista ao mesmo tempo. Como desvantagem, as mudanças de temperatura e ventanias podem afetar o seu desempenho.

## 3.8 Aplicações de Visão Computacional

Com o desenvolvimento de pesquisas na área de processamento de imagens digitais nos últimos tempos, impulsionado pelas facilidades de implementação de aplicações de visão computacional utilizando a biblioteca OpenCV [Bradski, 2000], cada vez mais trabalhos de monitoramento do tráfego que utilizam visão computacional são publicados. A seguir, uma breve revisão bibliográfica de projetos de Engenharia de Transporte que utilizam conceitos de visão computacional.

### 3.8.1 Detecção de placa de veículos

Martinsky [2007]

### 3.8.2 Contagem volumétrica utilizando dispositivos móveis

Feitosa [2012] propõe em sua Tese de Mestrado em Ciência da Computação pela UFG - Universidade Federal de Goiás, um método para contagem volumétrica de veículos baseado em visão computacional. A pesquisa é focada em otimização de algoritmos de processamento de imagens para execução em dispositivos móveis.

Alguns requisitos são estabelecidos nesse trabalho: as imagens são capturadas da lateral de uma via, com uma câmera fixada em uma baixa altura, identificando e contando veículos que trafegam em ambas as direções; as imagens utilizadas são de baixa resolução; e o processo de contagem é realizado em tempo real.

Apesar de o objetivo do projeto ser a contagem volumétrica via dispositivos móveis, o estudo inicial das técnicas de processamento de imagem e visão computacional é feito em um *laptop*<sup>2</sup>. A metodologia adotada pode ser dividida em seis etapas principais. São elas:

1. **Entrada de dados:** os *frames*<sup>3</sup> do vídeo são disponibilizados em tempo real ou por um arquivo em disco, que foi gravado anteriormente.
2. **Pré-processamento de imagem:** conversão para escala de cinza, caso ainda não esteja nesse formato; equalização do histograma; e suavização da imagem por filtragem linear.
3. **Reconhecimento da imagem referência de fundo:** obtenção de um modelo adaptativo da imagem de fundo da cena, utilizando um método de rápido processamento.
4. **Definição da área de movimento:** determinação da região da cena com maior movimentação que, provavelmente, é a área de tráfego de veículos. Essa região de interesse também elimina movimentações indesejadas, como por exemplo, o movimento da vegetação.
5. **Segmentação de objetos:** subtrai a imagem referência de fundo do *frame* corrente, segmentando os objetos de interesse. Também descarta objetos sem interesse.
6. **Acompanhamento de objetos segmentados:** rastreamento dos objetos segmentados ao longo dos quadros, evitando que o mesmo seja contado mais de uma vez.

---

<sup>2</sup>Computador portátil, que no Brasil também é conhecido como *notebook*

<sup>3</sup>Cada um dos quadros de um vídeo.

O método proposto foi implementado em forma de aplicativo na linguagem Java, utilizando a biblioteca de processamento de imagens JavaCV, que utiliza a maior parte das funções da OpenCV [Bradski, 2000].

Os testes foram realizados a partir de 15 amostras de vídeos, sendo 14 com resolução  $640 \times 480$  e apenas 1 com resolução  $320 \times 240$ , e duração média de 09:30 minutos. Para avaliar o desempenho do método, num primeiro momento foi feita uma contagem volumétrica manual dos veículos em cada amostra, para que assim o percentual de acerto do método automático, executado no *laptop*, fosse calculado. Além disso, o tempo de execução de cada amostra foi medido.

Em seguida, o aplicativo desenvolvido em Java foi adaptado para ser executado por um dispositivo móvel com sistema operacional Android. Os mesmos testes foram realizados para as 15 amostras e o percentual de acerto permaneceu o mesmo, resultado esperado tendo em vista que os algoritmos de processamento de imagem e visão computacional não mudaram. Nesse novo cenário, o tempo de execução também foi medido. A Tabela 3.1 lista as medições de cinco amostras.

Tabela 3.1: Comparação entre resultados de tempo de execução do método pelo *laptop* e celular [Feitosa, 2012].

| Amostra | % acerto | Duração  | Tempo celular | Tempo <i>laptop</i> |
|---------|----------|----------|---------------|---------------------|
| 1       | 90,58    | 00:15:08 | 01:00:56      | 00:03:54            |
| 2       | 92,45    | 00:05:15 | 00:19:55      | 00:01:36            |
| 3       | 91,30    | 00:02:44 | 00:11:55      | 00:00:27            |
| 4       | 98,15    | 00:10:18 | 00:48:18      | 00:03:39            |
| 5       | 97,71    | 00:13:07 | 01:09:27      | 00:03:48            |

Os percentual de acerto obtido nas contagens foi satisfatórios e aplicativo desenvolvido para executar em um *laptop* foi adaptado e portado para um dispositivo móvel Android com sucesso. No entanto, o tempo de execução do método no dispositivo móvel não foi satisfatório, inviabilizando que essa aplicação funcione em tempo real.

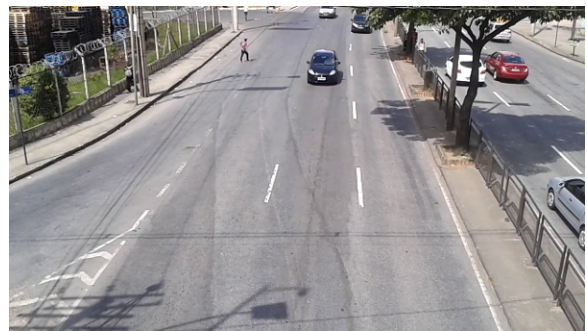
# Capítulo 4

## Metodologia

Como visto no Capítulo 2, existem diversas técnicas de processamento digital de imagens que possibilitam que aplicações de visão computacional sejam capazes de extrair informações a partir de uma ou mais imagens de entrada.



(a)



(b)

Figura 4.1: (a) Posicionamento da câmera para captura de imagens. A área em destaque simboliza

### 4.1 Entrada de dados

```
#include <iostream>

#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
```

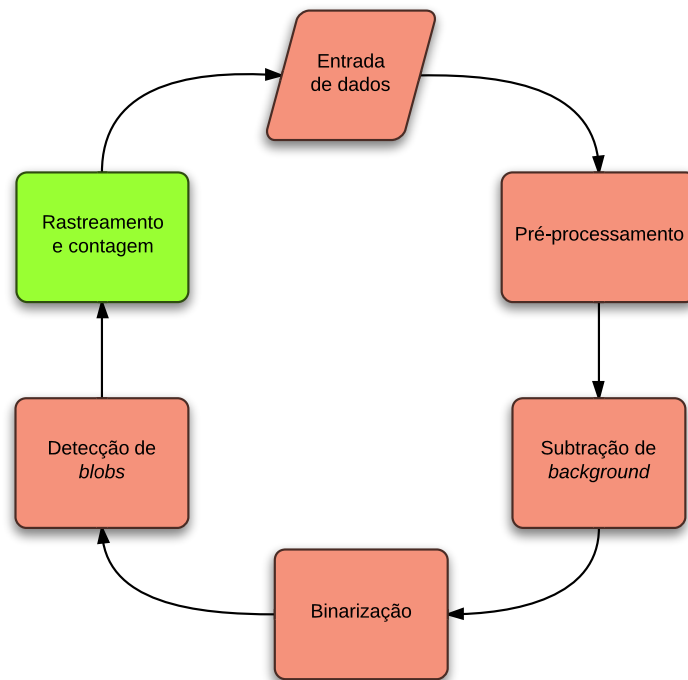


Figura 4.2: Fluxograma com a representação global do método de contagem.

```
int main(int argc, char const *argv[])
{
    cv::VideoCapture capture;
    if(!capture.isOpened()) {
        std::cout << "can not open camera or video file" << std::endl;
        return;
    }
    ...
    for(;;) {
        cv::Mat frame;
        capture >> frame;

        if(frame.empty()) break;
        ...
    }

    return 0;
}
```

## 4.2 Pré-processamento

```

...
for (;;) {
    ...
    cv::Mat gray;
    cv::cvtColor(frame, gray, CV_BGR2GRAY);

    cv::GaussianBlur(gray, gray, cv::Size(7, 7), 3);
    ...
}
...

```

### 4.3 Subtração de *background*

```

...
cv::BackgroundSubtractorMOG2 model;
for (;;) {
    ...
    cv::Mat foreground;
    model(gray, foreground);
    ...
}
...

```

### 4.4 Binarização

```

void bin(cv::Mat &src)
{
    //conta os pixels cinza
    int sumGray = 0, sumWhite = 0;
    for(int i = 0; i < src.rows; i++) {
        const uchar* ptri = src.ptr<uchar>(i);
        for(int j = 0; j < src.cols; j++) {
            if(ptri[j] != 0 && ptri[j] != 255)
                sumGray++;
            else if(ptri[j] == 255)
                sumWhite++;
        }
    }

    if(sumWhite == 0) {
        src.setTo(cv::Scalar(0));
    }
    else if((float)sumGray/(float)sumWhite > 10.0)

```

```

        cv::threshold(src , src , 250, 255, CV_THRESH_BINARY);
    else
        cv::threshold(src , src , 5, 255, CV_THRESH_BINARY);
}

...
for (;;) {
    ...
    bin(foreground);

    cv::Mat morph;
    cv::Mat element = cv::getStructuringElement(cv::MORPH_ELLIPSE,
                                                cv::Size(5,5));
    cv::morphologyEx(foreground , morph, CV_MOP_CLOSE, element ,
                    cv::Point(-1,-1), 3);
    ...
}
...

```

## 4.5 Detecção de *blobs*

```

...
cv::SimpleBlobDetector::Params params;
params.filterByInertia = false;
params.filterByConvexity = false;
params.filterByColor = true;
params.blobColor = 255;
params.filterByCircularity = false;
params.filterByArea = true;
params.minArea = 500.0f;
params.maxArea = 80000.0f;

cv::Ptr<cv::FeatureDetector> detector =
    new cv::SimpleBlobDetector(params);

...
for (;;) {
    ...
    std::vector<cv::KeyPoint> keypoints;
    detector.detect(morph, keypoints)
}
...

```

## 4.6 Rastreamento e contagem



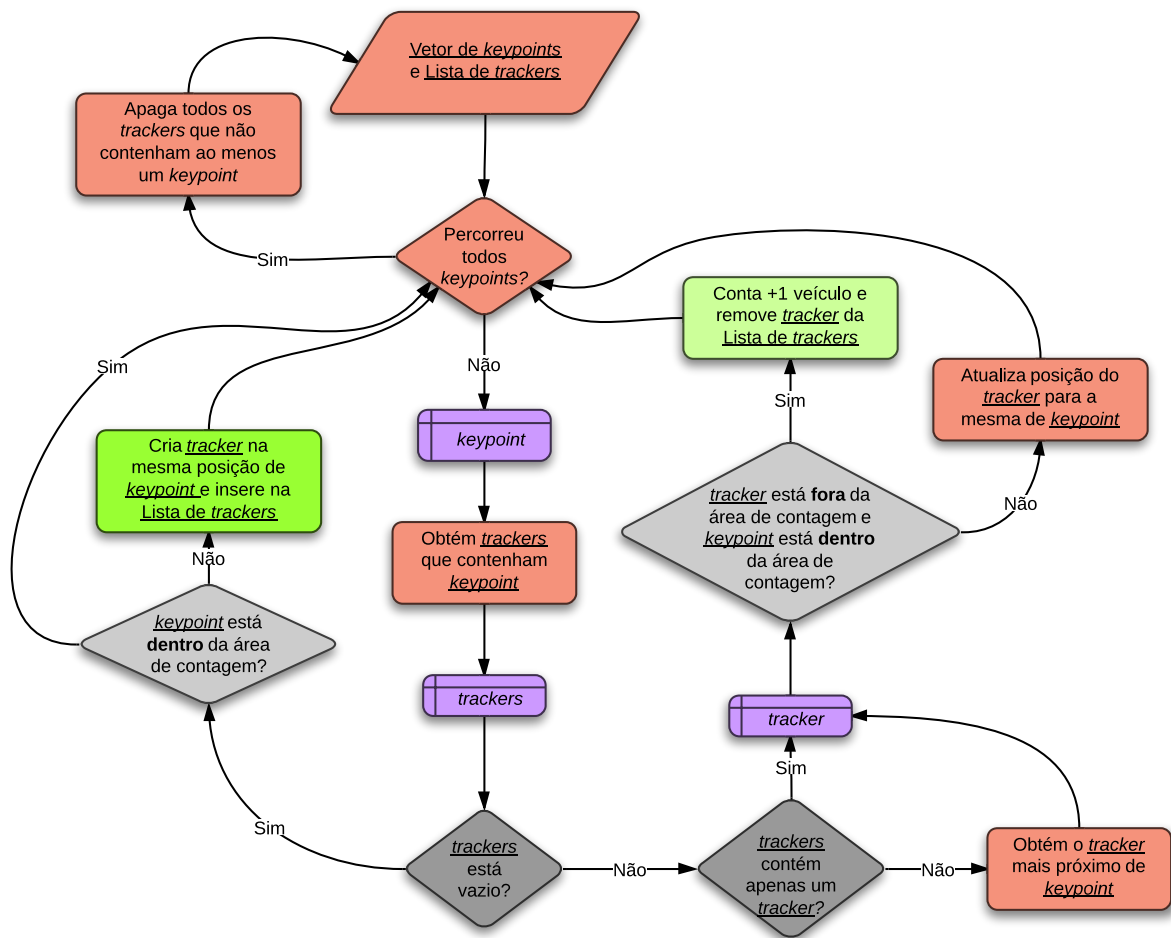


Figura 4.3: Fluxograma do método de rastreamento de *keypoints* utilizando *trackers* e contagem de veículos.



## Capítulo 5

### Testes e Resultados

Tabela 5.1: Resultados de contagem volumétrica.

| Nome do vídeo                    | Resolução         | Manual | Autom. | % acerto |
|----------------------------------|-------------------|--------|--------|----------|
| carlos_luz_centro_vga            | $640 \times 480$  | 164    | 156    | 95,12    |
| carlos_luz_centro_hd_resized     | $640 \times 360$  | 150    | 153    | 98,04    |
| carlos_luz_centro_hd             | $1280 \times 720$ | 150    | 145    | 96,67    |
| carlos_luz_pampulha_vga_1        | $640 \times 480$  | 208    | 173    | 83,17    |
| carlos_luz_pampulha_vga_2        | $640 \times 480$  | 169    | 138    | 81,66    |
| carlos_luz_pampulha_hd_2_resized | $640 \times 360$  | 201    | 175    | 87,06    |
| carlos_luz_pampulha_hd_2         | $1280 \times 720$ | 201    | 170    | 84,58    |

Tabela 5.2: Tempo real gasto na execução do método de contagem.

| Nome do vídeo                    | Resolução         | Duração  | Tempo gasto |
|----------------------------------|-------------------|----------|-------------|
| carlos_luz_centro_vga            | $640 \times 480$  | 00:05:23 | 00:05:47    |
| carlos_luz_centro_hd_resized     | $640 \times 360$  | 00:05:25 | 00:05:16    |
| carlos_luz_centro_hd             | $1280 \times 720$ | 00:05:29 | 00:17:48    |
| carlos_luz_pampulha_vga_1        | $640 \times 480$  | 00:05:43 | 00:06:04    |
| carlos_luz_pampulha_vga_2        | $640 \times 480$  | 00:05:19 | 00:05:33    |
| carlos_luz_pampulha_hd_2_resized | $640 \times 360$  | 00:06:19 | 00:05:38    |
| carlos_luz_pampulha_hd_2         | $1280 \times 720$ | 00:06:21 | 00:19:00    |



## Capítulo 6

### Considerações Finais



# Referências Bibliográficas

- Behrisch, M.; Bieker, L.; Erdmann, J. & Krajzewicz, D. (2011). SUMO - Simulation of Urban MObility: An Overview. Em *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pp. 63–68, Barcelona, Spain.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Bradski, G. & Kaehler, A. (2008). *Learning OpenCV*. O'Reilly.
- CITILABS (2013). Cube Dynasim 4. <http://www.citilabs.com/cube-dynasim-4>.
- de Almeida, F. A. M. (2010). Classificação Automática de Veículos pelo Perfil Magnético através de Técnicas de Aprendizagem de Máquina. Dissertação de mestrado, Universidade Estadual do Ceará.
- DNIT (2011). Plano Nacional de Contagem de Trânsito. <http://www.dnit.gov.br/rodovias/operacoes-rodoviararias/postos-de-contagem/plano-nacional-de-contagem-de-transito>.
- Feitosa, F. C. C. (2012). Um estudo prático para contagem volumétrica automática de veículos usando Visão Computacional. Dissertação de mestrado, Universidade Federal do Goiás.
- FENABRAVE (2013). Federação Nacional da Distribuição de Veículos Automotores. <http://www3.fenabrave.org.br:8082/plus/>.
- GIMP (2013). The GNU Image Manipulation Program. <http://www.gimp.org/>.
- Goldner, L. G. (2009). Engenharia de Tráfego 1º Módulo. Relatório técnico, Universidade Federal de Santa Catarina.
- Gonzalez, R. C. & Woods, R. E. (2000). *Processamento de Imagens Digitais*. Edgard Blücher Ltda.

- Intel (2013). Intel Corporation. <http://www.intel.com/>.
- Invent Vision (2013). Aplicações. <http://www.inventvision.com.br/>.
- Johnson, S. (2006). *Stephen Johnson on Digital Photography*. O'Reilly.
- Laganière, R. (2011). *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing Ltd.
- Lancaster, H. (1973). *An introduction to medical statistics*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley. ISBN 9780471512509.
- Martinsky, O. (2007). Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems. Dissertação de mestrado, BRNO University of Technology.
- Morris, T. (2004). *Computer Vision and Image Processing*. Palgrave Macmillan.
- OpenCV Development Team (2013). OpenCV 2.4.5.0 documentation. <http://docs.opencv.org/>.
- PTV (2013). Vissim. <http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/>.
- Schouten, T. (2003). Inhoud college Image Processing - Chapter 2: Fundamental Aspects. <http://www.cs.ru.nl/~ths/rt2/col/h2/2fundENG.html>.
- Shapiro, L. G. & Stockman, G. C. (2001). *Computer Vision*. Prentice Hall.
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.
- Tancredi, P. R. (2012). Monitoramento do Acesso de Veículos de Carga em Vias Urbanas. Dissertação de mestrado, Universidade de São Paulo.
- TRL Software (2013). ARCADY. [https://www.trlsoftware.co.uk/products/junction\\_signal\\_design/arcady](https://www.trlsoftware.co.uk/products/junction_signal_design/arcady).