UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
Departamento de Engenharia Elétrica

Francisco Assis da Silva

**Georreferenciamento automático de placas de sinalização com
imagens obtidas com um sistema móvel de mapeamento**

Tese apresentada à Escola de Engenharia de São Carlos da
Universidade de São Paulo, como parte dos requisitos para
obtenção do título de Doutor em Ciências, Programa de
Engenharia Elétrica.

Área de Concentração: Processamento de Sinais e
Instrumentação
Orientadora: Prof^ª. Dr^ª. Maria Stela Veludo de Paiva

São Carlos
2012

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

S586g Silva, Francisco Assis da
Georreferenciamento automático de placas de
sinalização com imagens obtidas com um sistema móvel de
mapeamento. / Francisco Assis da Silva ; orientadora
Maria Stela Veludo de Paiva. São Carlos, 2012.

Tese (Doutorado - Programa de Pós-Graduação em
Engenharia Elétrica e Área de Concentração em
Processamento de Sinais e Instrumentação)-- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2012.

1. Reconhecimento de objetos. 2. Georreferenciamento
direto. 3. Mapeamento móvel. 4. SIFT. I. Título.

FOLHA DE JULGAMENTO

Candidato: Bacharel **FRANCISCO ASSIS DA SILVA**.

Título da tese: "Georreferenciamento automático de placas de sinalização com imagens obtidas com um sistema móvel de mapeamento".

Data da defesa: 27/06/2012

Comissão Julgadora:

Resultado:

Profa. Dra. **Maria Stela Veludo de Paiva (Orientadora)**
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Prof. Associado **Adilson Gonzaga**
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Prof. Dr. **Ricardo Luis Barbosa**
(Universidade Estadual Paulista "Júlio de Mesquita Filho/UNESP-Sorocaba)

APROVADO

Prof. Dr. **Almir Olivette Artero**
(Universidade Estadual Paulista "Júlio de Mesquita Filho/UNESP-Presidente Prudente)

APROVADO

Dr. **Lúcio André de Castro Jorge**
(Empresa Brasileira de Pesquisa Agropecuária/EMBRAPA)

APROVADO

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Titular **Denis Vinicius Coury**

Suplente do Presidente da Comissão de Pós-Graduação em Exercício:
Prof. Titular **Denis Vinicius Coury**

Dedico este trabalho primeiramente a Deus, por ter me concedido a oportunidade e a capacidade de ter realizado este trabalho.

À minha mãe, pela criação que eu recebi, que me ensinou a ser humilde, e que sempre me incentivou a não desistir de alcançar os objetivos que eu almejei na minha vida.

Agradecimentos

À professora orientadora Dra. Maria Stela Veludo de Paiva, que acreditou em mim mesmo eu estando tão longe, que muito me ensinou, e que contribuiu muito para meu crescimento profissional e científico.

Ao professor e amigo Dr. Ricardo Luís Barbosa, que sempre confiou no meu trabalho desde a época da faculdade e que não mediu esforços para me auxiliar na realização deste trabalho. Tenho muito para te agradecer por tudo que fez por mim.

Ao professor Dr. Almir Olivette Artero com o qual aprendi muitas coisas e que me ajudou muito com sua inteligência e amizade em momentos decisivos do trabalho.

Ao Dr. Rodrigo Bezerra de Araújo Gallis pesquisador da Empresa Cartovias Engenharia Cartográfica que realizou os levantamentos para coleta de dados com o Sistema Móvel de Mapeamento nas vias públicas da cidade de Presidente Prudente, com o qual aprendi muito da área de Cartografia.

À professora Dra. Arlete Aparecida Correia Meneguette da FCT/Unesp de Presidente Prudente, que me ajudou com a base cartográfica.

Aos alunos de Ciência da Computação da Faculdade de Informática de Presidente Prudente (FIPP) e programadores da empresa Cartovias, Alan Kazuo Hiraga e Victor Hugo Masutani, que sempre me ajudaram na configuração das câmaras fotográficas e na realização dos levantamentos.

Aos meus amigos professores colegas de trabalho da Faculdade de Informática de Presidente Prudente que sempre me incentivaram para a realização deste trabalho.

*“Cada sonho que você deixa pra trás,
é um pedaço do seu futuro
que deixa de existir.”*

Steve Jobs

Resumo

SILVA, F. A. **Georreferenciamento automático de placas de sinalização com imagens obtidas com um sistema móvel de mapeamento**. 2012. 231 f. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012.

A detecção e reconhecimento de objetos em ambiente não controlado tem aplicações diversas no campo da visão computacional, e juntamente com o georreferenciamento de objetos de forma automática propicia uma variedade de aplicações, como por exemplo, o mapeamento da sinalização de trânsito. Os sinais de trânsito são muito importantes por proverem regras de navegação nas ruas e estradas. Um sistema para a determinação das posições geográficas de placas de sinalização de trânsito em áreas urbanas de forma automática constitui uma ferramenta útil para a gestão municipal podendo servir para tomadas de decisão, como por exemplo, fluxo de tráfego e definição de sinalização nas vias terrestres. Do ponto de vista prático, um sistema com estas características tem uma grande complexidade na implementação o que caracteriza um grande desafio. Diante do contexto exposto, nesta tese, é tratada a computação da detecção, o reconhecimento de sinais e o georreferenciamento de placas de trânsito. A implementação deste trabalho consistiu na coleta de conjuntos de dados e a aplicação de algoritmos para a extração dos descritores de pontos chave e para realizar a correspondências dos pontos chave entre duas imagens (imagem de uma via contendo uma ou mais placas e imagem de um *template* de uma placa de sinalização). Uma vez obtidos apenas os pontos em comuns referentes aos seus descritores, na sequência foram aplicados algoritmos para a detecção, reconhecimento e georreferenciamento das placas de trânsito. Para a obtenção do conjunto de dados foi utilizado um sistema móvel de mapeamento terrestre, equipado com sensores de imageamento digital, que além de obter conjuntos de sequências de imagens, também capturam informações de navegação e posicionamento. Para a detecção e reconhecimento foram utilizados algoritmos já consolidados na literatura (SIFT e BBF) e também algoritmos definidos e implementados para a realização da metodologia proposta. Para a extração de pontos chave condizentes com a placa a ser detectada, foi desenvolvido um algoritmo, pelo fato dos algoritmos citados na literatura não serem adequados para imagens que apresentam poucos pontos de correspondência, como é o caso do algoritmo RANSAC. Foi também definido e implementado um algoritmo para o reconhecimento de caracteres para o caso de placas de sinalização que especificam limite de velocidade. Com o conhecimento das fotocoordenadas centrais referentes às placas detectadas e reconhecidas e os dados de navegação e posicionamento, é realizado o georreferenciamento a fim de determinar as posições das placas no terreno por meio das equações de colinearidade. Foram realizados experimentos iniciais comprovando que a metodologia proposta é adequada para os objetivos definidos. As taxas de acerto na detecção e reconhecimento das placas de sinalização atingiram valores superiores a 80%, mesmo utilizando imagens com cenas complexas. O trabalho desenvolvido contribui com a metodologia proposta destinada à determinação das posições das feições dos sinais de trânsito em áreas urbanas, e na Área de Visão Computacional, contribui com novos algoritmos para a detecção e reconhecimento de placas de sinalização, bem como um novo algoritmo para o reconhecimento de caracteres.

Palavras-chave: Reconhecimento de objetos, Georreferenciamento direto, mapeamento móvel, SIFT.

Abstract

SILVA, F. A. **Automatic georeferencing of traffic signs with images took from a mobile mapping system.** 2012. 231 f. Dr. Thesis – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012.

The detection and object recognition in uncontrolled environment has several applications in the field of computer vision, and together with automatic georeferencing of objects provides a variety of applications, for example, the mapping of traffic signs. Traffic signs are very important because they provide navigation rules in streets and roads. A system for the automatic determining of the geographic positions of traffic sign plates in urban areas constitutes a useful tool for municipal management, it can be used for decision making, such as traffic flow and sign location on roads. From a practical point of view, a system with these characteristics has a great complexity in the implementation that characterizes a great challenge. Considering the exposed context, this thesis treats the computation of detection, recognition and georeferencing of traffic signs. The implementation of this work consisted in collecting data sets and application of algorithms for extracting keypoint features and performing the keypoint matching between two images (image of a road containing one or more plates and image of a template from a traffic sign). Once only the points in common in relation to their descriptors had been obtained, in the sequence, some algorithms were applied to the detection, recognition and georeferencing of traffic signs. To obtain the data set a land-base mobile mapping system was used, equipped with digital imaging sensors, which in addition to obtaining sets of image sequences, they also capture navigation information and positioning. For detection and recognition algorithms already established in literature (SIFT and BBF) were used and algorithms defined and implemented to the realization of the proposed methodology were also used. For the extraction of keypoints suitable with the plate to be detected, an algorithm was developed, because of the algorithms mentioned in literature are not appropriate for images that have few points of matching such as the RANSAC algorithm. An algorithm for recognition of characters for the case of signs which specify the speed limit was also defined and implemented. With the knowledge of the central photo coordinates referring to plates detected and recognized and navigation and positioning data, the georeferencing is performed to determine the positions of the plates on the ground through the collinearity equations. Initial experiments were performed demonstrating that the proposed methodology is appropriate for the defined goals. The hit rates of detection and recognition of sign plates reached values above 80%, even using images with complex scenes. The developed work contributes with the proposed methodology destined to the determination of traffic signs positions in urban areas, and in the Computer Vision Area, it contributes with new algorithms for the detection and recognition of traffic signs and a new algorithm for character recognition.

Key words: Object recognition, direct georeferencing, mobile mapping, SIFT.

Lista de Figuras

Figura 2.1: Representação do processo para construção das imagens resultantes da diferença de Gaussianas DoG em cada oitava no espaço escala. Fonte: Adaptado de Lowe (2004).	41
Figura 2.2: Exemplo do processo de construção de uma oitava.....	42
Figura 2.3: Pirâmide de imagens filtradas pelas Gaussianas mostrando o processo de construção das oitavas.....	43
Figura 2.4: Exemplo do processo de detecção de extremos (máximos ou mínimos) das imagens resultantes da diferença Gaussiana. Fonte: Adaptado de Lowe (2004).	44
Figura 2.5: Resultado dos experimentos para determinar a escolha do número de amostras em escalas vizinhas por oitava para detectar pontos chave. Fonte: (LOWE, 2004).	45
Figura 2.6: Resultado do experimento para determinação do valor prévio de σ para cada oitava. Fonte: (LOWE, 2004).	46
Figura 2.7: Gradientes da imagem e descritor de pontos chave. Fonte: (LOWE, 2004).	51
Figura 2.8: Exemplo de uma árvore <i>k-d tree</i> . Na esquerda o modo que o plano é subdividido e à direita a correspondente árvore binária. Fonte: (BERG et al., 2008).	55
Figura 2.9: Exemplo de busca BBF em uma árvore <i>k-d tree</i> com 8 pontos. Fonte: (BEIS; LOWE, 1997).....	58
Figura 2.10: Estimação de uma reta ajustada a um conjunto de pontos 2D.....	60
Figura 2.11: Pseudocódigo do algoritmo RANSAC para calcular um melhor ajuste na correspondência dos descritores SIFT.	63
Figura 2.12: Imagem “box.png” detectada na imagem “box_in_cene.png” com 45 pontos com extração dos <i>inliers</i> pelo algoritmo RANSAC.	65
Figura 2.13: Imagem de desenho detectada com 39 pontos (<i>inliers</i>) extraídos pelo algoritmo RANSAC.....	66
Figura 2.14: Placa “Velocidade máxima permitida 40 Km/h” detectada e reconhecida com 11 pontos chave com extração dos <i>inliers</i> pelo algoritmo RANSAC.	67
Figura 2.15: Placa de sinalização “Parada obrigatória” detectada e reconhecida com 11 pontos chave com extração dos <i>inliers</i> pelo algoritmo RANSAC.....	67
Figura 2.16: Pontos chave correspondentes entre as imagens na tentativa de reconhecimento da placa de sinalização "Proibido virar à direita".....	68

Figura 2.17: Pontos chave correspondentes entre as imagens na tentativa de reconhecimento da placa de sinalização "Siga em frente ou à direita".	69
Figura 2.18: Pontos chave correspondentes entre as imagens na tentativa de reconhecimento da placa de sinalização "Estacionamento regulamentado".	69
Figura 2.19: Exemplos de algumas placas de sinalização que não apresentaram bons resultados no reconhecimento usando o algoritmo RANSAC para extrair os <i>inliers</i> .	70
Figura 3.1: Coordenadas geodésicas (latitude (ϕ) e longitude (λ)) e altitude geométrica (h)) de um ponto qualquer P . Fonte: Adaptado de Monico (2000).	76
Figura 3.2: Distribuição das cartas topográficas ao milionésimo no Brasil. Fonte: (CÂMARA et al., 2001).	77
Figura 3.3: Sistema de Referência do GPS. Fonte: (MONICO, 2000).	79
Figura 3.4: Constelação dos satélites GPS. Fonte: (MONICO, 2000).	80
Figura 3.5: Exemplo de uma amostra do arquivo de <i>log</i> INSPVAA.	84
Figura 3.6: Translação da posição da antena GPS para o centro perspectivo de cada imagem do estéreo par. Fonte: (SILVA et al., 2003).	84
Figura 3.7: Sistemas de fotocoordenadas (x_f, y_f) e da imagem (x_p, y_p).	86
Figura 3.8: Geometria da condição de colinearidade para o caso terrestre. Fonte: Adaptado de Wolf (1983).	87
Figura 3.9: Geometria da condição de colinearidade para o caso terrestre com um par de imagens. Fonte: (WOLF, 1983).	88
Figura 3.10: SIMMOV no Campo de Calibração.	92
Figura 3.11: Pontos do Campo de Calibração de Câmaras da FCT/Unesp. Fonte: (GALO, 2011).	92
Figura 3.12: Resultados do processo de calibração das câmaras.	93
Figura 4.1: Procedimento de detecção. Fonte: (REITERER et al., 2009)	99
Figura 4.2: Esquema utilizado no projeto de Barcelona. Fonte: (BARÓ et al., 2009a)	114
Figura 5.1: Sistema Móvel de Mapeamento.	122
Figura 5.2: Câmara Nikon D300S.	123
Figura 5.3: a) Aparelho GPS. b) Unidade de Medida Inercial – IMU.	123
Figura 5.4: Fluxograma da metodologia utilizada para detecção, reconhecimento e georreferenciamento das placas de sinalização.	126
Figura 5.5: Par de imagens esquerda (a) e direita (b) capturadas pelo SIMMOV.	127
Figura 5.6: Classes do pacote “ <i>CamadaApresentacao</i> ” e suas relações.	131

Figura 5.7: Classes do pacote “ <i>CamadaLogica</i> ” e do pacote “ <i>CamadaNegocio</i> ” com suas relações.....	132
Figura 5.8: Classes do pacote “ <i>Cartografia</i> ”.....	133
Figura 5.9: Classes do pacote “ <i>ColetaPontos</i> ”.....	133
Figura 5.10: Classes do pacote “ <i>ExifMetadata</i> ”.....	134
Figura 5.11: Classes do pacote “ <i>Lowe</i> ” e do pacote “ <i>ReconhecePlacas</i> ”.....	135
Figura 5.12: Classes do pacote “ <i>OpenCV</i> ”.....	136
Figura 5.13: Classes do pacote “ <i>ReconheceCaracteres</i> ”.....	136
Figura 5.14: Modelo Entidade Relacionamento do banco de dados do Sistema <i>AutoTSR</i>	140
Figura 5.15: Classes “ <i>ListaTemplate</i> ” e “ <i>NoListaTemplate</i> ”.....	141
Figura 5.16: Nodo controlador da lista encadeada.....	142
Figura 5.17: Nodo da lista, usado para armazenar as informações de um <i>template</i> de uma placa de sinalização.....	142
Figura 5.18: Diagrama de Sequência do algoritmo de Detecção e Reconhecimento das placas.....	143
Figura 5.19: Nodo controlador da lista duplamente encadeada.....	145
Figura 5.20: Nodo da lista, usado para armazenar os resultados da detecção e reconhecimento de uma placa de sinalização.....	146
Figura 5.21: Placas de sinalização com o mesmo significado, porém com formas ligeiramente diferentes.....	147
Figura 5.22: Classes “ <i>ListaImagens</i> ” e “ <i>NoImagens</i> ”.....	147
Figura 5.23: Correspondência entre os pontos chave após avaliação dos descritores do SIFT.....	149
Figura 5.24: Conjunto de pontos chave do <i>template</i> cujos descritores correspondem aos descritores da imagem de entrada.....	152
Figura 5.25: Conjunto de pontos chave da imagem de entrada cujos descritores correspondem aos descritores dos pontos chave do <i>template</i>	152
Figura 5.26: Pontos chave candidatos a <i>inliers</i> após a avaliação dos ângulos de orientação dos gradientes locais.....	153
Figura 5.27: Ponto chave candidato a <i>inlier</i> classificado como <i>outlier</i>	155
Figura 5.28: Conjunto de pontos chave classificados como <i>inliers</i>	155
Figura 5.29: Pontos chave (<i>inliers</i>) com as direções dos gradientes representadas pelas setas saindo dos pontos.....	156

Figura 5.30: Classe “ <i>DeteccaoReconhecimento</i> ” pertencente ao pacote “ <i>ReconhecePlacas</i> ”	157
Figura 5.31: Placa “Proibido virar à direita” detectada e reconhecida com 5 pontos chave..	158
Figura 5.32: Placa “Siga em frente ou à direita” detectada e reconhecida com 3 pontos chave.	158
Figura 5.33: Placa “Estacionamento regulamentado” detectada e reconhecida com 6 pontos chave.	159
Figura 5.34: Sequência de pixels na imagem usada.	161
Figura 5.35: a) Exibição dos registros da classe 1 em preto e da classe 2 em cinza; b) conjunto de dados contendo 4 registros com treze atributos em duas classes; c) Exibição dos registros da classe 1 e d) classe 2.....	161
Figura 5.36: Regras usadas para fazer a descrição dos caracteres.....	162
Figura 5.37: Diagrama de classes do algoritmo de Reconhecimento de Caracteres.	164
Figura 5.38: Diagrama de Sequência do estágio de treinamento do algoritmo de Reconhecimento de Caracteres.....	165
Figura 5.39: Diagrama de Sequência do algoritmo de Reconhecimento de Caracteres no estágio de reconhecimento dos dígitos numéricos.....	166
Figura 5.40: Trinta e nove caracteres (números: 0, 1, ..., 9) escritos à mão. Fonte: (FRANK; ASUNCION, 2010).	167
Figura 5.41: Visualização obtida usando: a) todas as transições de todos os caracteres nas dez classes; b) apenas os registros da classe do caractere “0”; c) apenas os registros da classe do caractere “1”; d) apenas os registros da classe do caractere “2”; e) apenas os registros da classe do caractere “9”.	167
Figura 5.42: Matriz de confusão obtida com a classificação usando as transições entre os atributos adjacentes.....	168
Figura 5.43: a) Conjunto de caracteres usados no treinamento do classificador (30 fontes diferentes); b) Conjunto usado no teste (outras 10 fontes diferentes).	169
Figura 5.44: Conjunto de caracteres numéricos usados no treinamento do classificador para o reconhecimento dos dígitos numéricos das placas de velocidade.	170
Figura 5.45: Parte do conjunto de 5120 atributos, representando a sequência de pixels para cada classe.	170
Figura 5.46: Exemplo de uma imagem contendo uma placa de sinalização de velocidade detectada e reconhecida.....	171

Figura 5.47: Exemplo de uma imagem segmentada de uma placa de sinalização de velocidade detectada e reconhecida.....	171
Figura 5.48: Diagrama de Sequência do estágio de Georreferenciamento das placas de sinalização.	172
Figura 6.1: Placas de sinalização de trânsito usadas como <i>templates</i>	174
Figura 6.2: Erro no reconhecimento das placas contendo o sinal “Estacionamento regulamentado”.....	176
Figura 6.3: Exemplos de reconhecimento apesar de fatores complicadores utilizando a metodologia proposta.....	176
Figura A.1: Mapa contendo o posicionamento e numeração sequencial das placas detectadas e reconhecidas (Parte 1/2).....	198
Figura A.2: Mapa contendo o posicionamento e numeração sequencial das placas detectadas e reconhecidas (Parte 2/2).....	199
Figura C.1: Imagem com a placa “Proibido virar à esquerda” não detectada.....	227
Figura C.2: Imagem com a placa “Proibido retornar à esquerda” não detectada.....	228
Figura C.3: Imagem com as placas “Proibido virar à esquerda” e “Proibido retornar à esquerda” não detectadas.	228
Figura C.4: Imagem com a placa “Proibido virar à esquerda” não detectada.....	228
Figura C.5: Imagem com a placa “Estacionamento regulamentado” não detectada.	229
Figura C.6: Imagem com a placa “Estacionamento regulamentado” não detectada.	229
Figura C.7: Imagem com a placa “Estacionamento regulamentado” não detectada.	229
Figura C.8: Imagem com a placa “Estacionamento regulamentado” não detectada.	230
Figura C.9: Imagem com a placa “Siga em frente ou à direita” não detectada.....	230
Figura C.10: Imagem com a placa “Proibido retornar à esquerda” não detectada.....	230
Figura C.11: Imagem com a placa “Parada obrigatória” não detectada.	231

Lista de Tabelas

Tabela 1: Tabela comparativa entre os sistemas GPS e INS. Fonte: (RUY, 2004).....	82
Tabela 2: Arquivo de <i>log</i> INSPVAA do GPS/INS. Fonte: (SPAN-SE, 2011).	83
Tabela 3: Pontos chave com as coordenadas (x , y) e orientação (θ) após a computação da correspondência dos descritores SIFT (<i>matching</i>) e comparação das orientações θ	151
Tabela 4: Transições que não ocorrem entre atributos adjacentes no conjunto de dados ilustrado na Figura 5.35 (b).	162
Tabela 5: Resultados no reconhecimento das placas de sinalização.	175

Lista de Abreviaturas e Siglas

API:	<i>Application Programming Interface</i>
ASCII:	<i>American Standard Code for Information Interchange</i>
AutoTSR:	<i>Automatic Traffic Sign Recognition</i>
BBF:	<i>Best-Bin-First</i>
BCT:	<i>Bilateral Chinese Transform</i>
BIH:	<i>Bureau International de l'Heure</i>
CCD:	<i>Contracting Curve Density</i>
CCPA:	<i>Candid covariance-free incremental principal component analysis</i>
CPU:	<i>Central Processing Unit</i>
CSS:	<i>Curvature Scale Space</i>
CTP:	<i>Conventional Terrestrial Pole</i>
CTRS:	<i>Conventional Terrestrial Reference System</i>
DGPS:	<i>Differential GPS</i>
DLL:	<i>Dynamic Link Library</i>
DMI:	<i>Distance Measuring Instrument</i>
DoG:	<i>Difference of Gaussian</i>
ECOC:	<i>Error-Correcting Output Code</i>
Exif:	<i>Exchangeable image file format</i>
GLOH:	<i>Gradient Location and Orientation Histogram</i>
GPS:	<i>Global Positioning System</i>
GPS/INS:	<i>Global Positioning System/Inertial Navigation System</i>
HOG:	<i>Histogram of Oriented Gradients</i>
HSV:	<i>Hue, Saturation, Value</i>
IBGE:	<i>Instituto Brasileiro de Geografia e Estatística</i>
IGC:	<i>Instituto Geográfico e Cartográfico</i>
IMPCA:	<i>Image Principal Components Analysis</i>
IMU:	<i>Inertial Measurement Unit</i>
INS:	<i>Inertial Navigation System</i>
INSPVAA:	<i>INS Position, Velocity and Attitude</i>
JEIDA:	<i>Japan Electronics and Information Technology Industries Association</i>

MLP:	<i>Multi-Layer Perceptron</i>
MPCA:	<i>Modular Principal Components Analysis</i>
OMG:	<i>Object Management Group</i>
PCA-SIFT:	<i>Principal Component Analysis Scale Invariant Feature Transform</i>
RANSAC:	<i>RANdom SAmple Consensus</i>
RBMC:	Rede Brasileira de Monitoramento Contínuo
RGB:	<i>Red, Green, Blue</i>
ROI:	<i>Region of Interest</i>
RSR:	<i>Road Sign Detection and Recognition</i>
SIFT:	<i>Scale Invariant Feature Transform</i>
SIG:	Sistemas de Informação Geográfica
SIMMOV:	Sistema Móvel de Mapeamento
SINS:	<i>Strap Down Inertial Navigation System</i>
SQL:	<i>Structured Query Language</i>
SURF:	<i>Speeded Up Robust Features</i>
SVM:	<i>Support Vector Machine</i>
SMM:	Sistemas Móveis de Mapeamento
SMMD:	Sistemas Móveis de Mapeamento Digital
SMMT:	Sistema Móvel de Mapeamento Terrestre
SPS:	<i>Standard Positioning Service</i>
TM:	Transversa de Mercator
TSR:	<i>Traffic Sign Recognition</i>
UML:	<i>Unified Modeling Language</i>
UTC:	<i>Universal Time Coordinated</i>
UTM:	Universal Transversa de Mercator
WGS84:	<i>World Geodetic System – 1984</i>

Sumário

Resumo	11
Abstract	13
Lista de Figuras.....	15
Lista de Tabelas.....	21
Lista de Abreviaturas e Siglas.....	23
1 Introdução	25
1.1 Considerações Iniciais.....	25
1.2 Justificativas	28
1.2.1 Contribuição	29
1.3 Objetivos.....	31
1.4 Estrutura do Trabalho	31
2 Técnicas para detecção e reconhecimento de placas de sinalização.....	35
2.1 Considerações Iniciais.....	35
2.2 Introdução	35
2.3 Algoritmo SIFT	38
2.3.1 Detecção de extremos no espaço escala	39
2.3.2 Localização exata dos pontos chave.....	46
2.3.3 Determinação de Orientação dos Descritores	49
2.3.4 Descritor local.....	50
2.4 Correspondência dos descritores SIFT.....	52
2.4.1 Árvore <i>k-d tree</i>	54
2.4.2 Algoritmo de busca BBF	56
2.5 Extração dos <i>inliers</i>	59
2.5.1 Algoritmo RANSAC aplicado ao ajuste na correspondência dos descritores SIFT.....	63
2.5.2 Aplicação para detecção e reconhecimento de objetos	64
2.5.3 Problemas no reconhecimento de placas utilizando o algoritmo RANSAC.....	68
2.6 Considerações Finais	71

3	Aspectos Gerais de Cartografia e Sistemas de Coordenadas.....	73
3.1	Considerações Iniciais	73
3.2	Sistemas Móveis de Mapeamento.....	73
3.3	Sistema de Referência	75
3.3.1	Sistemas de Coordenadas Geodésicas.....	75
3.3.2	Sistema UTM.....	76
3.3.3	Sistema de Referência WGS84.....	78
3.4	Sistema de Posicionamento Global (GPS)	79
3.5	Sistema de Navegação Inercial.....	81
3.6	Obtenção dos dados do GPS/INS	82
3.7	Interseção fotogramétrica	85
4	Trabalhos Relacionados.....	95
4.1	Considerações Iniciais	95
4.2	Trabalhos Relacionados com relação à detecção, classificação e reconhecimento de placas de sinalização de trânsito.....	95
4.2.1	Abordagem baseada no algoritmo SIFT.....	96
4.2.2	Abordagem baseada em propriedades de cor.....	102
4.2.3	Abordagem baseada em propriedades de forma.....	104
4.2.4	Abordagem baseada em propriedades de cor e forma	106
4.2.5	Outras abordagens.....	108
4.3	Trabalhos com aplicações de informações georreferenciadas.....	112
4.4	Trabalhos Relacionados com relação ao reconhecimento de caracteres.....	116
4.5	Considerações Finais.....	118
5	Metodologia da detecção, reconhecimento e determinação das coordenadas das placas de sinalização	121
5.1	Considerações Iniciais	121
5.2	Materiais e Métodos	122
5.3	Descrição técnica do Sistema <i>AutoTSR</i>	127
5.3.1	Modelagem e descrição das classes do Sistema <i>AutoTSR</i> em UML	129
5.3.2	Banco de Dados.....	137

5.3.3	Estrutura de Dados contendo os <i>templates</i> recuperados do banco de dados	140
5.4	Detecção e reconhecimento das placas de sinalização.....	142
5.4.1	Algoritmo de extração dos <i>inliers</i> para detecção e reconhecimento das placas de sinalização	148
5.4.2	Resultados com o algoritmo desenvolvido para extração dos <i>inliers</i>	157
5.5	Reconhecimento dos dígitos numéricos das placas de limite de velocidade	159
5.5.1	Algoritmo para o reconhecimento de caracteres	160
5.6	Determinação das coordenadas do terreno das placas de sinalização	172
6	Resultados e Discussões	173
6.1	Avaliação dos Resultados.....	177
6.2	Discussões	177
7	Conclusões e Sugestões.....	179
7.1	Sugestões de Trabalhos Futuros	181
7.2	Artigos Publicados e Submetidos	183
7.2.1	Artigos Publicados.....	183
7.2.2	Artigos Aceitos para Publicação	183
7.2.3	Artigos Submetidos	184
7.2.4	Participação como co-autor em Artigos Publicados	184
7.2.5	Participação como co-autor em Artigos Aceitos para Publicação.....	184
	Referências Bibliográficas.....	185
	Apêndice A – Mapa contendo as placas detectadas e reconhecidas	197
	Apêndice B – Resultados da Detecção, Reconhecimento e Georreferenciamento das placas de Sinalização.....	201
	Apêndice C – Imagens de entrada com as placas não detectadas pela metodologia proposta	227

Capítulo 1

Introdução

1.1 Considerações Iniciais

As aplicações que utilizam informações geográficas estão avançando rapidamente e os sistemas móveis terrestres de mapeamento são utilizados cada vez mais para aquisição de imagens e dados além de serem úteis na atualização das informações geográficas.

A integração da Cartografia com tecnologias computacionais proporcionou o desenvolvimento dos Sistemas de Informação Geográfica (SIG's), que são ferramentas gerenciais para organizar, processar e administrar grandes volumes de informações geográficas para atividades que exigem tomadas de decisões. Tais sistemas necessitam de atualização periódica e por isso deve haver um grande dinamismo na atualização destes dados; desse modo também há a necessidade de técnicas confiáveis para realizar a coleta das informações. Estas ferramentas são assistidas pela utilização de banco de dados para o gerenciamento das informações existentes (OLIVEIRA, 2003).

Os dados manipulados pelos SIG's podem ser entidades ou fenômenos geográficos distribuídos na superfície terrestre, podendo pertencer a sistemas naturais ou criados pelo homem, tais como tipos de solos, vegetação, cidades, redes de telefonia, fluxo de veículos etc. Os mapas têm sido as principais fontes de dados para SIG's, e o levantamento de campo, o principal processo de coleta de dados (GALLIS, 2002).

Não é novidade dizer que o Brasil ainda é um país de carências, e que as dimensões continentais do país ainda não foram adequadamente mapeadas, verificando-se grandes porções do território sem cobertura cartográfica. É comum também observar que as cartas topográficas de regiões já mapeadas podem estar desatualizadas. Devido a esta crescente necessidade de atualização de dados, houve um grande avanço nas pesquisas, buscando novas técnicas de coleta de informações espaciais. Uma delas são os Sistemas Móveis de Mapeamento (SMM), que vem se destacando principalmente devido à sua eficiência e economia. Estes sistemas proporcionam sequências de imagens, que permitem a análise e medição de pontos para fins de levantamento e mapeamento fotogramétrico ou simples

observação para avaliação das condições das vias de transporte e de seu entorno. Essas condições são físicas, geométricas e ambientais, como, por exemplo, as condições do pavimento, largura, rampas, acostamentos ou calçadas, arborização, sinalização horizontal e vertical, segurança para pedestres e condutores (OLIVEIRA, 2003). Os Sistemas Móveis de Mapeamento representam uma metodologia de levantamento de dados que integra sensores de posicionamento e imageamento, permitindo o mapeamento pormenorizado do mundo real.

A tarefa de encontrar as correspondências entre duas imagens da mesma cena ou objeto é parte das aplicações de Visão Computacional. Calibração de câmara, reconstrução 3D, registro de imagens e reconhecimento de objetos são apenas alguns exemplos (TREIBER, 2010) (GALO, 2011) (GRAUMAN; LEIBE, 2011). A busca por correspondências de imagens digitais pode ser dividida em três etapas principais. Primeiramente, “pontos de interesse” são selecionados em locais distintos da imagem, tais como cantos, borrões (*blobs*), e junções *T*. A propriedade mais valiosa de um detector de pontos de interesse é a repetitividade, isto é, ele encontra de forma confiável os mesmos pontos de interesse sob diferentes condições de visualização. Em seguida, a vizinhança de cada ponto de interesse é representada por um vetor de características. Esse descritor deve ser distinto e, ao mesmo tempo, robusto ao ruído, permitir detecção de erros e deformações geométricas e fotométricas. Finalmente, os vetores de descritores são combinados entre imagens diferentes. A correspondência é muitas vezes baseada em uma distância entre os vetores, por exemplo, a distância Mahalanobis ou Euclidiana. A dimensão do descritor tem um impacto direto sobre o tempo que ele consome em seu processamento, e um menor número de dimensões é, portanto desejável (BAY et al., 2008).

A grande variabilidade da aparência de sinais de trânsito em ambientes não controlados faz a detecção e classificação desses sinais um problema desafiador em Visão Computacional. O reconhecimento de sinais de trânsito vem sendo estudado para diversos propósitos, tais como direção autônoma e assistida (HANDMANN et al., 1998) (FRANKE et al., 1998). O reconhecimento dos sinais de trânsito permite que um motorista possa ser avisado sobre ações impróprias e situações potencialmente perigosas (BARÓ et al., 2009c).

De acordo com Gonzalez e Woods (2002), um sistema de Visão Computacional deve ser dotado de algumas características, como por exemplo, a habilidade para extrair apenas informações importantes a partir de uma cena repleta de informações, entre as quais podem existir algumas irrelevantes. O sistema também deve ter a habilidade para fazer inferências a partir de informações incompletas e, principalmente, conseguir identificar um objeto ou padrão em uma imagem com a maior independência possível em relação a fatores como

mudanças de posição, tamanho, orientação e variações no ponto de vista (transformações geométricas).

Reconhecimento de padrão é uma parte integrante da maioria dos sistemas de inteligência de máquina para a tomada de decisão, cujo objetivo é a classificação de objetos em um número de categorias ou classes, e muito importante na área de visão de máquina. Um sistema de visão de máquina capta imagens através de uma câmara e as analisa para produzir descrições do que é imagiado (THEODORIDIS; KOUTROUMBAS, 2009).

Reconhecimento de objetos em imagens é uma das principais áreas de aplicação de Processamento de Imagens e Visão Computacional e é usado em muitas aplicações diferentes. Por muitas décadas tem sido – e ainda é – uma área de extensa pesquisa. O procedimento comum da maioria dos sistemas é que, dado algum conhecimento sobre a aparência de certos objetos, uma ou mais imagens são examinadas para avaliar quais objetos estão presentes e onde (TREIBER, 2010).

Segundo Arlicot et al. (2009) sinais de trânsito são muito importantes por proverem regras de navegação nas ruas e estradas. Suas propriedades visuais são muito fortes, porque eles foram projetados para ser facilmente reconhecidos pela visão humana. A detecção e reconhecimento de sinais de trânsito enquadram-se na subárea de reconhecimento e classificação de objetos dentro da área de Visão Computacional.

Uma aplicação capaz de realizar a detecção e o reconhecimento de placas de sinalização de trânsito em um ambiente urbano de forma automática evidencia as características expostas por Gonzalez e Woods (2002), Theodoridis e Koutroumbas (2009), e Treiber (2010).

Reconhecimento visual tem uma variedade de aplicações potenciais incluindo muitas áreas de inteligência artificial e de recuperação de informação, por exemplo, busca de imagem baseada em conteúdo, mineração de dados em vídeo, ou identificação de objetos para robôs móveis. Progressos significativos para reconhecimento de objetos têm sido feitos com o desenvolvimento de descritores locais invariantes. Estes descritores permitem ao algoritmo encontrar as estruturas locais da imagem de uma forma repetível e as codificar em uma representação que é invariante a uma série de transformações geométricas, tais como translação, rotação, escala e pequenas alterações de ponto de vista. Os descritores resultantes então formam a base da metodologia utilizada para o reconhecimento de objetos específicos e categorias de objetos (GRAUMAN; LEIBE, 2011).

O propósito de descritores locais invariantes é prover uma representação que permita realizar a correspondência de estruturas locais entre imagens de forma eficiente. Isto é feito a

partir da obtenção de um conjunto de medidas locais que capturam a essência das imagens de entrada e que codifica suas estruturas de interesse. O processo de extração de descritores deve ser repetível e preciso, de forma que os mesmos descritores extraídos de duas imagens devem representar o mesmo objeto (GRAUMAN; LEIBE, 2011).

Durante os últimos anos, métodos utilizando descritores locais se tornaram os preferidos em diferentes campos de visão de computacional. Hoje eles são aplicados em muitas tarefas como imagens panorâmicas (BROWN; LOWE, 2007) ou reconhecimento de objetos (LOWE, 2004). Recentemente, descritores locais são utilizados no contexto de TSR (*Traffic Sign Recognition*) de modos diferentes, sendo aplicados tanto para a detecção de sinais de trânsito, quanto para a sua classificação, devido a seus pontos fortes e discriminativos (HÖFERLIN; HEIDEMANN, 2010).

Técnicas de descritores locais detectam primeiramente características de uma determinada imagem e então um conjunto de descritores é computado para estas características (MIKOLAJCZYK et al., 2005). Os descritores representam as partes características de uma imagem. Existem muitos algoritmos na literatura para a extração de descritores locais, mas em muitos trabalhos o algoritmo SIFT (*Scale Invariant Feature Transform*) é predominantemente utilizado, por ser invariante a transformações de escala, rotação e pontos de vista.

Em geral, a identificação de sinais de trânsito é alcançada por meio de duas fases principais: detecção e reconhecimento. Primeiramente, é aplicado um algoritmo para a detecção do sinal de trânsito, sendo localizada a região da placa. Posteriormente, o algoritmo de reconhecimento utiliza esta região localizada para identificar o tipo do sinal de trânsito.

1.2 Justificativas

A necessidade de georreferenciar uma grande quantidade de informações localizadas nas ruas e rodovias, como postes, placas, pontos de ônibus, lixeiras etc., faz com que a busca por algoritmos automáticos ou semi-automáticos para esse fim ganhem destaque na literatura, pois reduz o tempo de processamento e consequentemente aumenta a produtividade dessa tarefa. As informações produzidas a partir do georreferenciamento das placas de sinalização de trânsito são importantes e necessárias para auxiliar aos gestores de administrações

municipais no planejamento e tomadas de decisões, bem como para navegação autônoma de veículos.

Nesse sentido, um sistema que possibilite a captura das informações das placas de sinalização de trânsito de forma automática, rápida e precisa, possibilita uma enorme gama de aplicações. Embora existam trabalhos na literatura para essa finalidade (AMLACHER; PALETTA, 2008) (BARÓ et al., 2009a) (REITERER et al., 2009), o trabalho proposto nesta tese apresenta uma metodologia que integra tecnologias para a aquisição de dados (pares de imagens e informações de posicionamento e de navegação), e contribui com técnicas de Visão Computacional para detectar e reconhecer placas de sinalização de trânsito em vias, a partir de imagens capturadas em ambientes abertos contendo fundo complexo e georreferenciar as informações de interesse de forma automática.

1.2.1 Contribuição

Um acervo de imagens, vídeos e informações geográficas permite a administradores e engenheiros ter ferramentas para auxiliar na gestão municipal que tenha como objetivo principal melhorar os padrões de qualidade de vida nas cidades. Os Sistemas Móveis de Mapeamento (SMM) desempenham um papel importante na coleta dos dados com informações geográficas. A partir de sensores de imageamento digital, navegação e posicionamento integrados, possibilitam o georreferenciamento dessas informações com rapidez, eficiência e economia de recursos.

As aplicações para as informações obtidas com o georreferenciamento são muitas, e entre elas podem-se destacar aplicações destinadas aos avanços da tecnologia de navegação autônoma de veículos, tema associado com rodovias inteligentes. Este novo ramo da engenharia de transportes requer que o mundo real, ruas e rodovias e seus arredores, estejam repletos de dispositivos ópticos e eletrônicos, cujos dados e informações sejam detectados pelos sensores embarcados no veículo, que por sua vez possui um banco de dados com informações.

A construção de bancos de dados geográficos deste tipo é de grande importância e caracteriza uma das muitas aplicações de um sistema que utiliza os dados obtidos com um sistema móvel de mapeamento. Muitos tipos diferentes de dados relacionados aos caminhos imageados podem ser derivados por meio de operações de medição nas imagens ou busca por

determinação de posições de feições de interesse, proporcionando informações para decisões mais rápidas e eficientes.

Partindo desse princípio, este trabalho vem contribuir com a metodologia proposta destinada ao fornecimento das determinações das posições das feições dos sinais de trânsito em áreas urbanas. Estes sinais de trânsito são principalmente placas de sinalização afixadas em calçadas de ruas e avenidas de cidades.

Em um estágio anterior da computação do georreferenciamento para a determinação das posições (coordenadas no terreno) das placas de sinalização, é realizada a detecção e o reconhecimento dessas placas. Muitos trabalhos para esta atividade podem ser encontrados na literatura, tanto a partir de imagens digitais, quanto a partir de sequências de vídeo (o capítulo 4 descreve alguns desses trabalhos). Nesses trabalhos, encontra-se o emprego das mais diversas técnicas de processamento de imagens e reconhecimento de objetos, entretanto, mesmo assim, a técnica e os algoritmos construídos e empregados neste trabalho representam uma contribuição científica para a atividade de detecção e reconhecimento de placas de sinalização de trânsito. A técnica proposta é baseada no algoritmo SIFT (LOWE, 2004), que após a computação da correspondência dos pontos chave de duas imagens (imagem do *template* da placa de sinalização e imagem de entrada pertencente à sequência de imagens obtidas pelo Sistema Móvel de Mapeamento), é executado o processo de eliminação dos pontos chave que não condizem com o *template*. Este algoritmo foi implementado neste trabalho de doutorado com características específicas voltadas à detecção e reconhecimento das placas na tentativa de maximizar os resultados de forma automática e minimizar os falsos positivos. Um de seus estágios é a extração dos pontos chave que condizem com o *template*. Embora exista na literatura o algoritmo RANSAC (FISCHLER; BOLLES, 1981) para esta atividade, que é amplamente empregado em diversos trabalhos de reconhecimento de objetos e confecção de imagens panorâmicas, este apresentou problemas nos experimentos realizados quando os objetos a serem detectados e reconhecidos possuem poucos detalhes, como é o caso de boa parte das placas de sinalização. Com poucos detalhes, implica em extrair poucos pontos chave e em consequência disso ocorrem muitas falhas na detecção e reconhecimento.

Também como contribuição científica, foi implementado um algoritmo para o reconhecimento dos dígitos numéricos das placas de limites de velocidade baseado na classificação supervisionada dos caracteres. Esta atividade é importante para completar o estágio da computação do reconhecimento das placas de sinalização, quando, a partir do conhecimento dos números, torna-se possível conhecer o limite de velocidade na via.

1.3 Objetivos

O presente trabalho de doutorado tem como objetivo geral propor uma metodologia para o reconhecimento automático de placas de sinalização em um ambiente urbano e fazer o georreferenciamento automático dessas placas para utilização em um Sistema de Informações Geográficas. A detecção, reconhecimento e georreferenciamento das placas de sinalização são feitas a partir de sequências de pares de imagens (esquerda e direita) obtidas com um par de câmaras fotográficas instaladas sobre o teto de um veículo, utilizado para realizar a coleta de imagens e dados (levantamento). As imagens capturadas contêm os objetos de interesse em cenas com fundo complexo, podendo apresentar grande variação de iluminação.

Sucintamente os objetivos específicos deste trabalho são relacionados na sequência:

- Extrair os pontos chave das placas de sinalização detectadas em uma sequência de imagens, com o algoritmo SIFT;
- Detectar e reconhecer as placas de sinalização, com a combinação dos pontos chave extraídos das imagens do levantamento em correspondência com os *templates* de imagens de placas de sinalização, armazenados previamente em um banco de dados;
- Determinar as coordenadas no terreno das placas reconhecidas (georreferenciamento);
- Representar as placas de sinalização em uma base cartográfica.

1.4 Estrutura do Trabalho

Após este capítulo introdutório, a presente tese está organizada conforme a estrutura que segue.

No capítulo 2, são elucidados os aspectos relevantes às técnicas para detecção e reconhecimento de placas de sinalização. Um primeiro enfoque é dado ao algoritmo SIFT, escolhido a partir de investigações realizadas na literatura. Nesse enfoque, é abordada a importância desse algoritmo para este trabalho e seus estágios para a extração dos descritores dos pontos chave das imagens utilizadas, obtidas por meio de levantamento. Após, são

abordados os algoritmos para a correspondência dos descritores SIFT entre as imagens do levantamento com as imagens dos *templates* das placas de sinalização de trânsito. Ainda neste capítulo encontra-se a descrição do algoritmo RANSAC utilizado para o ajuste na correspondência dos descritores SIFT para o reconhecimento de objetos e problemas encontrados com o uso do RANSAC no processo de detecção e reconhecimento de placas de sinalização.

O capítulo 3 aborda alguns aspectos gerais de cartografia, a fim de apresentar os aspectos relevantes utilizados neste trabalho. Nesse contexto, são abordados os Sistemas Móveis de Mapeamento, Sistemas de Referência contemplando os Sistemas de Coordenadas Geodésicas, o Sistema UTM, o Sistema de Referência WGS84 e o Sistema de Posicionamento Global (GPS). Também são abordadas a obtenção dos dados do GPS/INS e a interseção fotogramétrica.

No capítulo 4 são apresentados os resultados das investigações realizadas na literatura, a fim de obter conhecimento de trabalhos relacionados ao apresentado nesta tese, considerando até mesmo abordagens diferentes. Inicialmente são abordados os trabalhos relacionados à detecção, classificação e reconhecimento das placas de sinalização de trânsito, considerando abordagens baseadas no algoritmo SIFT. Também são mencionados trabalhos baseados em propriedades de cor, forma, operadores morfológicos, métodos de aprendizagem SVM, redes neurais, conjuntos *fuzzy*, entre outros. Ainda neste capítulo, um enfoque é dado aos trabalhos investigados a respeito de informações georreferenciadas e trabalhos relacionados ao reconhecimento de caracteres.

O capítulo 5 apresenta a metodologia empregada neste trabalho para a detecção, reconhecimento e determinação das coordenadas do terreno das placas de sinalização. É apresentada a modelagem e a descrição do Banco de Dados, assim com uma descrição do Sistema implementado usando a notação UML. Neste capítulo são abordados os algoritmos definidos e implementados para a detecção e reconhecimento das placas de sinalização e para o reconhecimento dos dígitos numéricos das placas de limite de velocidade. Um enfoque também é dado à determinação das coordenadas do terreno das placas reconhecidas.

No capítulo 6 são apresentados os resultados obtidos e discussões a partir do experimento realizado. O Apêndice A mostra o posicionamento das placas detectadas e reconhecidas em um mapa da área na qual foi realizado o levantamento. Os resultados da detecção, reconhecimento e georreferenciamento das placas de sinalização, gerados de forma automática pelo Sistema, são apresentados no Apêndice B. No Apêndice C, são mostradas as

imagens de entrada obtidas por meio de levantamento, que possuem placas não detectadas pela metodologia proposta.

Por fim, o capítulo 7 apresenta as conclusões desta tese de doutorado, aspectos que ficaram em aberto como sugestão para o direcionamento de trabalhos futuros, e a relação dos artigos publicados, frutos da contribuição deste trabalho.

Capítulo 2

Técnicas para detecção e reconhecimento de placas de sinalização

2.1 Considerações Iniciais

Neste capítulo e em suas seções e subseções são descritos os algoritmos utilizados, existentes na literatura de Visão Computacional, para a realização da detecção e o reconhecimento de placas de sinalização a partir de pares de imagens. Inicialmente é dada ênfase à explicação do algoritmo SIFT proposto por Lowe (2004) que é utilizado para a extração dos descritores dos pontos chave das placas de sinalização. Posteriormente são descritos os algoritmos para a tarefa de reconhecimento de objetos em imagens. Nesta atividade estão envolvidas a comparação dos descritores das imagens e a eliminação de pontos chave de ambas as imagens que não possuem correspondência entre si. No capítulo 4 e em suas seções e subseções podem ser encontrados trabalhos correlatos às atividades de detecção e reconhecimento de sinais das placas de sinalização envolvendo inclusive abordagens diferentes.

2.2 Introdução

Uma fase inicial para a detecção de sinais de trânsito em imagens, empregada neste trabalho, efetua a correspondência entre duas imagens; uma dessas imagens sendo a que possivelmente possui um ou mais sinais de trânsito e a outra contém o sinal desejado, representando o *template* de busca.

A correspondência de imagens (*image matching*) é um aspecto fundamental de muitos problemas de Visão Computacional, incluindo reconhecimento de objetos ou de cenas, obtenção da estrutura 3D de múltiplas imagens, montagem automática de mosaicos,

correspondência estéreo e acompanhamento de movimentos (LOWE, 2004). Uma abordagem para se trabalhar com correspondência de imagens é se usar descritores locais para se representar uma imagem. Descritores são vetores de características de uma imagem ou de determinadas regiões de uma imagem e podem ser usados para se comparar regiões em imagens diferentes. Esses descritores são utilizados em aplicações que se destinam a obter correspondência e reconhecimento de imagens.

Existem diversas técnicas para se descrever regiões locais em uma imagem. O mais simples descritor é um vetor com as intensidades dos pixels da imagem. A medida de correlação cruzada pode ser então usada para computar a similaridade entre duas regiões. Porém, a alta dimensionalidade de tal descritor aumenta a complexidade computacional da comparação. Então, esta técnica é principalmente usada para se encontrar correspondências ponto a ponto entre duas imagens. A vizinhança de um ponto também pode ser escalada de modo a reduzir sua dimensão. Outro descritor simples é a distribuição de intensidades de uma região representada por seu histograma.

Trabalhos recentes na literatura têm se concentrado em descritores invariáveis a transformações nas imagens. Algoritmos baseados em métodos que não detectam apenas pontos de interesse ou as características locais de regiões nas imagens, tem se mostrado bastante eficientes na resolução de problemas relacionadas à área de reconhecimento de padrões. Algoritmos como o SIFT (*Scale Invariant Feature Transform*), proposto por Lowe (2004), o SURF (*Speeded Up Robust Features*), proposto por Bay et al. (2008) e suas respectivas variações, como o PCA-SIFT (*Principal Component Analysis Scale Invariant Feature Transform*) proposto por Ke e Sukthankar (2004), U-SURF (BAY et al., 2008), SURF-128 (BAY et al., 2008), são algoritmos que englobam diversas metodologias relacionadas aos processos de identificação de pontos de interesse, geração de descritores e correspondência entre os pontos de interesse.

Alguns dos fatores que permitiram que esses algoritmos ganhassem popularidade devem-se ao fato dos mesmos agregarem características que os tornam invariantes a transformações como rotação, translação, escala, alterações inerentes às condições de iluminação do ambiente onde as imagens são adquiridas e à existência de ruídos gerados no processo de aquisição ou gravação das imagens (MACHADO et al., 2008).

Existe uma variedade de outros métodos disponíveis na literatura que possuem o mesmo objetivo que o SIFT e o SURF. No trabalho de Mikolajczyk e Schmid (2005) foi realizado um levantamento sobre outros detectores de pontos de interesse e descritores. Neste trabalho foi constatado que o SIFT apresenta um desempenho superior em relação aos demais

algoritmos mencionados em termos de invariância. Os autores do SURF, Bay et al. (2008), afirmam que o seu algoritmo é superior ao SIFT em termos de eficiência de tempo de processamento e que o mesmo apresenta um bom desempenho no processo de reconhecimento de objetos, considerando a qualidade dos pontos de características. O artigo de Bauer et al. (2007) também realiza um estudo comparativo entre os dois algoritmos e suas variações, descrevendo os aspectos positivos e negativos inerentes a cada um deles. Segundo Ihara et al. (2009), os descritores SIFT podem ser usados para reconhecimento de sinal de trânsito porque eles são robustos a mudanças na escala e rotação, mas o processamento em tempo real é difícil, pois o custo computacional da extração e correspondência SIFT é alto.

O algoritmo SIFT é um algoritmo que foi apresentado originalmente por Lowe (1999; 2004) para a extração de descritores. Ao contrário de muitos outros algoritmos com esse objetivo, o SIFT não busca segmentar regiões ou objetos para produzir um conjunto de descritores, mas sim, identificar pontos estáveis. Nesse caso em particular, pontos considerados estáveis são pontos na imagem que tendem a ser invariantes à escala e orientação, isto é, pontos que podem ser reconhecidos como similares independentemente do valor absoluto da escala e orientação que os descrevem. Além destas qualidades, o algoritmo SIFT foi escolhido para ser utilizado neste trabalho, devido à sua eficiência demonstrada em diversos trabalhos encontrados na bibliografia, como parte do processo para o reconhecimento de objetos em cenas repletas com diversos tipos de objetos.

A grande variabilidade da aparência de sinal apresentada nas imagens das placas de sinalização obtidas em ambientes não controlados também justifica a escolha do SIFT. Ambiente não controlado, neste trabalho, refere-se ao local de obtenção das imagens. Nas imagens, os objetos de interesse, as placas de sinalização, podem apresentar diferentes escalas, diferentes inclinações (são consideradas neste trabalho apenas inclinações suaves, pois não se deseja reconhecer placas muito fora da posição correta), diferentes tipos de iluminação (sombra, ao Sol), reflexo de luminosidade (reflexão da luz do Sol) e problemas de oclusão (árvores, automóveis etc.). Este último problema, o da oclusão de objetos, que pode atrapalhar diretamente a detecção e mais ainda o reconhecimento das placas de sinalização, não foi objeto de estudo deste trabalho de doutorado.

As seções 2.3 e subseções de 2.3.1 a 2.3.4 seguintes descrevem a estrutura e o funcionamento do algoritmo SIFT para obtenção de pontos chave e extração dos seus descritores característicos. A seção 2.4 e suas subseções descrevem atividades de correspondência dos descritores SIFT. A seção 2.5 e suas subseções descrevem a extração dos

inliers por meio do algoritmo RANSAC, que elimina possíveis pontos falsos correspondentes entre as imagens.

2.3 Algoritmo SIFT

SIFT (*Scale Invariant Feature Transform*) é um algoritmo de Visão Computacional desenvolvido e publicado por David G. Lowe em 1999 (LOWE, 1999). Lowe é professor do departamento de Ciência da Computação na *University of British Columbia*, em Vancouver, Canadá.

O algoritmo SIFT é dividido basicamente em duas partes, a do detector e a do descritor. O detector SIFT é baseado nos cálculos de diferença de Gaussianas e o descritor SIFT utiliza histogramas de gradientes orientados para descrever a vizinhança local dos pontos de interesse.

SIFT permite a detecção e extração de descritores que tem muitas propriedades que são desejáveis para correspondência (*matching*) de diferentes imagens de um objeto ou cena. Esses descritores são razoavelmente invariantes a mudanças de iluminação, ruído de imagem, rotação, escala e pequenas mudanças de ponto de vista.

Os descritores fornecidos pelo SIFT são bem localizados em ambos os domínios, o domínio do espaço e o da frequência, reduzindo assim a probabilidade de não haver correspondência dos descritores por oclusão ou ruído. Os descritores são altamente distintos, permitindo que uma determinada característica possa ser corretamente combinada (*matched*) com alta probabilidade em um banco de dados de descritores de diversas imagens, provendo uma base para o reconhecimento de um objeto ou cena. A obtenção de descritores de uma imagem pelo algoritmo SIFT é feita por meio de quatro estágios principais, onde os dois primeiros estágios descrevem a parte do detector e as duas seguintes a formação do descritor. Esses estágios são (LOWE, 2004):

- **Detecção de extremos no espaço escala:** Esse primeiro estágio da computação faz a busca em todas as escalas e locais de uma imagem. Isto é feito utilizando-se a diferença de filtros Gaussianos de modo a se identificar pontos de interesse invariáveis à escala e orientação. Este é o estágio mais custoso computacionalmente do algoritmo;

- **Localização de pontos chave:** Para cada local candidato, em que foi detectado um extremo (máximo ou mínimo), um modelo detalhado é ajustado para se determinar a localização e escala. Pontos chave, ou pontos de interesse, são então selecionados baseando-se em suas medidas de estabilidade;
- **Determinação de orientação:** Uma ou mais orientações são associadas para cada localização de ponto chave, baseadas em direções do gradiente local da imagem. Todas as operações seguintes são realizadas nos dados da imagem relativamente transformados em relação à orientação, escala e localização de cada ponto chave, assim provendo a invariância a estas transformações;
- **Descritor dos pontos chave:** Neste estágio da computação, os gradientes locais da imagem são medidos na escala selecionada, na região ao redor de cada ponto chave. Estas medidas são então transformadas para uma representação que permite observar níveis significantes de distorção de forma e mudança na iluminação. São então criados histogramas de orientação para compor o descritor.

Esses quatro estágios da computação do algoritmo SIFT são detalhados, respectivamente, nas subseções de 2.3.1 a 2.3.4 na sequência.

2.3.1 Detecção de extremos no espaço escala

Este primeiro estágio do algoritmo SIFT consiste em identificar locais e escalas, que podem ser repetidamente associados ao mesmo objeto, sob diferentes pontos de vista. A detecção de locais que são invariantes às mudanças de escala da imagem pode ser realizada por busca de características estáveis nas diferentes escalas possíveis, utilizando uma função chamada de espaço escala (WITKIN, 1983). No SIFT é usada a função Gaussiana como sendo o núcleo (*kernel*) da função espaço escala, conforme mostram Koenderink (1984) e Lindeberg (1994) – esses autores também relatam que sob uma variedade razoável de suposições a função Gaussiana é o único núcleo possível para a função espaço escala. Resumidamente, são detectados extremos (máximos ou mínimos) em uma pirâmide da imagem convoluída com a função de diferença de filtros Gaussianos. Pontos chave correspondem a estes extremos para diferentes escalas.

O espaço escala de uma imagem é definido como uma função $L(x, y, \sigma)$, que é produzida a partir da convolução de uma Gaussiana de escala variável $G(x, y, \sigma)$, com uma imagem de entrada $I(x, y)$, conforme a Equação (1).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

onde $*$ é a operação de convolução em x e y , e a função Gaussiana é dada por:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2)$$

Neste filtro Gaussiano, o valor de σ representa o parâmetro que define o fator de suavização da imagem I .

Foi proposta em Lowe (1999) uma forma eficiente para detectar pontos chave estáveis usando os extremos do espaço escala com o uso de uma função de Diferença de Gaussianas (DoG – *Difference of Gaussian*). A diferença de Gaussianas, $D(x, y, \sigma)$, é computada pela diferença de duas imagens filtradas em escalas próximas, separadas por uma constante multiplicadora k . A função DoG é definida pela Equação (3).

$$\text{DoG} = G(x, y, k\sigma) - G(x, y, \sigma) \quad (3)$$

A convolução de uma imagem com a função DoG é definida por:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (4)$$

Esta convolução é a diferença entre imagens suavizadas por um filtro Gaussiano em escalas σ e $k\sigma$. A função DoG consegue detectar variações de intensidade na imagem, elimina detalhes indesejados e realça características fortes. Variando-se σ , é possível encontrar características em diferentes escalas.

Um modo eficiente para a construção da Diferença de Gaussianas (DoG) é feito por meio de uma pirâmide de imagens, mostrado na Figura 2.1. No algoritmo SIFT, inicialmente a imagem de entrada é redimensionada para o dobro de seu tamanho original, e isto é feito para se conseguir detectar mais pontos chave. Mudando-se esta situação, ou seja, utilizando-se a imagem de entrada no seu tamanho original, consegue-se um maior desempenho do algoritmo, porém são detectados menos pontos chave. Para algumas aplicações o ganho de

desempenho pode ser mais interessante do que duplicar a imagem para se ter um maior número de pontos chave.

Para cada oitava no espaço escala, a imagem inicial sofre convoluções incrementais com filtros Gaussianos para produzir um conjunto de imagens separadas por um fator de escala constante k no espaço escala, gerando imagens suavizadas L , representadas na coluna da esquerda da Figura 2.1. É denominado de oitava o conjunto de imagens L e DoG. A Figura 2.1 ilustra duas oitavas, estando à esquerda o conjunto de imagens suavizadas pelos filtros Gaussianos e à direita o conjunto de imagens resultantes da Diferença de Gaussianas.

Figura 2.1: Representação do processo para construção das imagens resultantes da diferença de Gaussianas DoG em cada oitava no espaço escala.

Fonte: Adaptado de Lowe (2004).

A imagem inicial é convoluída progressivamente com funções Gaussianas para produzir as outras $s + 2$ imagens utilizando um fator $k = 2^{1/s}$. São produzidas $s + 3$ imagens na pilha de imagens suavizadas com o filtro Gaussiano para cada oitava, de forma que a detecção de extremos cubra toda a oitava. A imagem é inicialmente filtrada por filtro Gaussiano com escala σ (sem o fator multiplicativo k). A partir de então, são geradas imagens que são progressivamente convoluídas, cada nova imagem é filtrada com escala k vezes a escala utilizada anteriormente. Imagens em escalas adjacentes são subtraídas para produzir imagens de diferença de Gaussiana, como mostrado na direita da Figura 2.1.

Para exemplificar o processo representado na Figura 2.1 de forma simples, deseja-se um intervalo $s = 2$, e como o fator utilizado é $k = 2^{1/s}$, então, tem-se $k = \sqrt{2}$. Neste caso são utilizadas cinco imagens na pilha das imagens filtradas com Gaussianas, pois $s + 3 = 5$. Estas imagens são:

- A primeira imagem filtrada por uma Gaussiana com escala σ ,
- Outras quatro imagens filtradas com escalas multiplicadas pelo fator k , elas são:
 - A segunda imagem usa $k\sigma$ como escala para o filtro Gaussiano;
 - A terceira imagem usa a escala 2σ , obtida a partir de $k^2\sigma$, logo, $(\sqrt{2})^2\sigma = 2\sigma$;
 - A quarta imagem usa a escala $2k\sigma$, obtida a partir de $k^3\sigma$, logo, $k^2k\sigma = (\sqrt{2})^2k\sigma = 2k\sigma$;
 - A quinta imagem usa a escala $2k^2\sigma$, obtida a partir de $k^4\sigma$, logo, $k^2k^2\sigma = 2k^2\sigma$;

Desta maneira, tem-se no exemplo quatro imagens produzidas por DoG, sendo que a primeira imagem (Figura 2.2 (b)) é dada por $D(x, y, \sigma)$ e a última imagem da pilha de DoG é dada por $D(x, y, 2k\sigma)$. Para melhor entendimento, na Figura 2.2 (a), a primeira imagem à esquerda é $I(x, y, \sigma)$ e a última imagem é $L(x, y, 2k^2\sigma)$.

O processo apresentado na Figura 2.2 constrói apenas uma oitava, sendo que cada oitava representa um conjunto de imagens L e um conjunto de imagens D .

Figura 2.2: Exemplo do processo de construção de uma oitava.

Uma vez que uma oitava tiver sido processada, a imagem Gaussiana L que possuir duas vezes o valor inicial da escala σ (2σ corresponde à antepenúltima imagem da pilha

superior) é redimensionada para a metade de seu tamanho, e esta será a primeira imagem da próxima oitava, assim como pode ser visto na Figura 2.1. Cada oitava produz o mesmo número de imagens no intervalo, e a computação é reduzida para cada nova oitava com imagens redimensionadas para a metade. A quantidade de oitavas depende da dimensão da imagem original e os redimensionamentos ocorrem até atingir uma dimensão mínima de aproximadamente 4×4 pixels. No algoritmo SIFT, Lowe definiu a Equação (5), a partir das dimensões da imagem (*width* e *height*) para calcular a quantidade de oitavas a serem processadas.

$$octvs = \lfloor \log(\min(width, height)) / \log(2) - 2 \rfloor \quad (5)$$

A pirâmide de imagens filtradas pelas Gaussianas mostrando o processo de construção das oitavas é apresentada na Figura 2.3, com três oitavas e cinco imagens L . Esta pirâmide representa uma forma mais intuitiva de entendimento dos exemplos de construção das imagens resultantes da diferença de Gaussianas DoG apresentada na Figura 2.1 e do processo de construção de uma oitava apresentado na Figura 2.2. Cada oitava é constituída por cinco imagens na pilha das imagens filtradas com Gaussianas, com diferentes escalas (σ , $k\sigma$, 2σ , $2k\sigma$, $2k^2\sigma$), ocasionando diferentes níveis de detalhes. Após o processamento de uma oitava, a oitava seguinte utiliza como base a imagem que possui escala 2σ redimensionada para a metade de seu tamanho.

Figura 2.3: Pirâmide de imagens filtradas pelas Gaussianas mostrando o processo de construção das oitavas.

Fonte: Adaptado de Heckel (2008).

Para detectar máximos ou mínimos locais de DoG, cada ponto é comparado com seus oito vizinhos na imagem corrente e nove vizinhos na escala superior e nove vizinhos na escala inferior, e somente é selecionado um ponto se ele for maior ou menor que todos. Os extremos são dados por valores máximo ou mínimo locais para cada $D(x, y, \sigma)$ que corresponda a um intervalo. As escalas superior e inferior são correspondentes às imagens vizinhas em uma mesma oitava para a pilha de imagens DoG, possuem a mesma dimensão. Quando se diz escala superior e inferior, está se fazendo referência à σ .

Os extremos (máximos ou mínimos locais) das imagens produzidas pela diferença Gaussiana são detectados pela comparação de cada pixel (marcado com “X”) e seus 26 vizinhos na região 3x3 da imagem corrente e escalas adjacentes (marcadas com círculos “O”). Esse processo de detecção de extremos apresenta um custo computacional de verificação relativamente baixo, devido ao fato que muitos pontos amostrados serão eliminados a partir das primeiras verificações. As três imagens DoG apresentadas correspondem à diferença entre imagens adjacentes da pirâmide Gaussiana em um pilha. A Figura 2.4 apresenta um exemplo do processo de detecção de máximos ou mínimos das imagens resultantes da diferença Gaussiana.

Figura 2.4: Exemplo do processo de detecção de extremos (máximos ou mínimos) das imagens resultantes da diferença Gaussiana.

Fonte: Adaptado de Lowe (2004).

A partir de experimentos baseados em tarefas de correspondências usando uma coleção de 32 imagens reais variadas, incluindo cenas ao ar livre, faces humanas, fotografias aéreas e imagens industriais, Lowe (2004) busca determinar o número confiável de amostras em escalas próximas por oitava para detectar os pontos chave estáveis usando os extremos do espaço escala.

Lowe conclui que a maior repetibilidade de pontos chave detectados na mesma posição é obtida quando são amostradas três escalas por oitava das imagens produzidas pela DoG. A Figura 2.5 apresenta os resultados dos experimentos de Lowe, e nela pode-se observar o número de escalas com a maior porcentagem de repetibilidade de pontos chave detectados no mesmo local.

Figura 2.5: Resultado dos experimentos para determinar a escolha do número de amostras em escalas vizinhas por oitava para detectar pontos chave.
Fonte: (LOWE, 2004).

No experimento de determinação do valor prévio de σ , Lowe (2004) investiga a determinação ideal desse valor, que é o valor mínimo de escala da função Gaussiana aplicado à primeira imagem de cada oitava, e para cada imagem de cada oitava aplica-se o fator multiplicativo k . A linha do topo do gráfico da Figura 2.6 apresenta a repetibilidade em porcentagem de pontos chave, que são detectados variando-se o valor prévio de σ . Lowe conclui que a repetibilidade de pontos chave detectados aumenta com o aumento de σ , entretanto, o custo computacional também aumenta com esse número, por isso, no algoritmo SIFT, Lowe usa $\sigma = 1,6$, o qual provê uma boa taxa de repetibilidade com um custo computacional razoável.

Figura 2.6: Resultado do experimento para determinação do valor prévio de σ para cada oitava.
Fonte: (LOWE, 2004).

O próximo estágio da computação do algoritmo SIFT é definir a localização dos pontos chave e descartar os pontos de amostragem que não apresentam estabilidade.

2.3.2 Localização exata dos pontos chave

Todos os pontos detectados como extremos são possíveis candidatos a pontos chave, e o próximo passo é executar um ajuste detalhado aos pixels próximos em relação à escala, localização, e relação de curvaturas principais. Essa informação permite rejeitar pontos que tenham baixo contraste (são sensíveis ao ruído) ou localizados ao longo da borda. A localização dos pontos chave é especialmente importante para as últimas oitavas, porque o espaçamento amostral destas representa grandes distâncias em relação à imagem base. A implementação inicial desse enfoque (LOWE, 1999) localizava pontos chave em locais e escala de pontos de amostragem centrais. Mais tarde Brown e Lowe (2002) desenvolveram um método que consiste em ajustar uma função quadrática 3D aos pontos de amostragem

loais para determinar a localização interpolada do extremo, a fim de prover melhorias substanciais na correspondência e estabilidade (LOWE, 2004).

Para cada ponto analisado é utilizada uma expansão de Taylor da função no espaço escala $D(x, y, \sigma)$, deslocada de modo que a origem desta expansão esteja localizada no ponto de amostragem.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (6)$$

onde D e suas derivadas (primeira e segunda) são calculadas no ponto de amostragem analisado e $x = (x, y, \sigma)^T$ é o deslocamento (*offset*) desse ponto. Ou seja, D é o valor da função $D(x, y, \sigma)$ no ponto avaliado, x é o deslocamento em relação a este ponto e $D(x)$ é a aproximação do valor de $D(x, y, \sigma)$ interpolado para um ponto transladado com *offset* x .

Os coeficientes quadráticos são computados aproximando-se as derivadas através das diferenças entre pixels das imagens já filtradas. A localização do extremo, $\hat{x}$, é determinada ao se fazer a derivada de $D(x)$ (Equação 6) com relação à x , igualando o resultado a zero:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (7)$$

Como sugerido por Brown e Lowe (2002), a matriz Hessiana e a derivada de D são aproximadas pelas diferenças de pontos amostrais da vizinhança. O resultado é um sistema linear 3x3 que pode ser resolvido com custo mínimo. Se o *offset* $\hat{x}$ for maior que 0.5 em alguma dimensão, significa que o extremo está mais próximo de outro ponto amostral. Nesse caso, o ponto amostrado é mudado e a interpolação é realizada para esse novo ponto. O *offset* $\hat{x}$ final é adicionado à localização do ponto analisado para se estimar a interpolação para a localização do extremo.

O valor da função no extremo, $D(\hat{x})$, é útil para a rejeição de extremos instáveis com baixo contraste, que seriam sensíveis a ruído. Substituindo-se a Equação (7) na Equação (6) obtém-se:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (8)$$

É aconselhado por Lowe (2004) que se rejeitem valores de $|D(\hat{x})|$ inferiores a um determinado limiar (0,03), assumindo-se que os valores dos *pixels* em tons de cinza da imagem estejam entre [0, 1].

Além do procedimento apresentado para se descartar pontos, Lowe (2004) afirma que para a estabilidade, não é suficiente rejeitar pontos chave com baixo contraste. A função DoG tem uma resposta forte nas bordas, mesmo se a localização ao longo da borda não for bem determinada, isto faz com que estes pontos sejam instáveis até para valores pequenos de ruído.

Um pico não bem definido na função DoG terá uma curvatura principal grande pela extremidade das bordas, mas pequena na direção perpendicular, o que não é desejável. As curvaturas principais podem ser computadas por meio de uma matriz Hessiana 2x2 H , para a localização e escala dos pontos chave:

$$H(x, y) = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (9)$$

onde D_{xy} é a derivada de $D(x, y, \sigma)$ na localização e escala em relação a x e y ; D_{xx} é a segunda derivada em relação a x ; e D_{yy} é a segunda derivada em relação a y .

As derivadas são estimadas levando em consideração as diferenças entre os pontos de amostra vizinhos à localização e escala definida, resultando nos autovalores (*eigenvalues*) de H , que são proporcionais às curvaturas principais de D (LOWE, 2004). As derivadas podem ser aproximadas por (BELO, 2006):

$$D_{xx} = D(x + 1, y, \sigma) - 2D(x, y, \sigma) + D(x - 1, y, \sigma) \quad (10)$$

$$D_{yy} = D(x, y + 1, \sigma) - 2D(x, y, \sigma) + D(x, y - 1, \sigma) \quad (11)$$

$$D_{xy} = \left(\begin{array}{l} D(x-1, y+1, \sigma) - D(x+1, y+1, \sigma) \\ + D(x+1, y-1, \sigma) - D(x-1, y-1, \sigma) \end{array} \right) / 4 \quad (12)$$

São computadas as somas dos autovalores de H e seus produtos do determinante (LOWE, 2004):

$$\begin{aligned} \text{Tr}(H) &= D_{xx} + D_{yy} = \alpha + \beta, \\ \text{Det}(H) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \end{aligned} \quad (13)$$

onde α é o autovalor com a maior magnitude e β o de menor magnitude.

Quando o determinante é negativo, as curvaturas têm sinais diferentes, então o ponto é descartado como não sendo um extremo. Considere r como sendo a razão entre o autovalor de maior magnitude e o de menor magnitude, assim $\alpha = r\beta$. Então,

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}, \quad (14)$$

que só depende da relação dos autovalores, independente dos seus valores individuais. O valor $(r+1)^2/r$ é mínimo quando os dois autovalores forem iguais e aumenta com r . Para verificar se a razão das curvaturas principais está abaixo de um limiar (*threshold*), r , usa-se a Equação (15):

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(r+1)^2}{r} \quad (15)$$

Lowe (2004) propõe o uso de $r = 10$. Assim, eliminam-se pontos chave que não são estáveis, apesar de estarem próximos de extremidades.

2.3.3 Determinação de Orientação dos Descritores

Até o estágio anterior da computação do algoritmo SIFT, foi vista a invariância dos pontos chave no espaço escala. Ao se atribuir uma orientação consistente para cada ponto chave baseada nas propriedades da imagem, podem-se representar os descritores em relação a esta orientação, conseguindo-se assim invariância quanto à rotação (LOWE, 2004).

A escala σ do ponto chave é utilizada para selecionar a imagem suavizada L com o filtro Gaussiano, com escala mais próxima ao ponto chave e oitava referente ao ponto avaliado. Dessa maneira, todas as computações passam a ser executadas com invariância à escala. Para cada imagem $L(x, y, \sigma)$, na mesma escala σ , a magnitude do gradiente $m(x, y)$ e a orientação $\theta(x, y)$ são computadas usando diferenças de pixels, conforme as equações (16) e (17).

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (16)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (17)$$

Um histograma de orientações é construído com as orientações dos gradientes dos pontos amostrados que se encontram em uma região ao redor do ponto chave. A orientação do

histograma tem 36 regiões (*bins*) cobrindo os 360 graus de orientações. Cada amostra na vizinhança do ponto chave, adicionada ao histograma, é ponderada pela sua magnitude do gradiente e por uma janela Gaussiana circular com um σ que é 1.5 vezes o tamanho da escala σ do ponto chave.

Picos na orientação do histograma correspondem a direções dominantes para os gradientes locais. O maior pico no histograma é detectado e picos que correspondem a valores percentuais acima de 80% do maior pico são usados para se definir a orientação de cada ponto chave. Portanto, para locais com múltiplos picos similares em magnitude, haverá múltiplos pontos chave criados no mesmo local e escala, mas com diferentes orientações. Somente aproximadamente 15% dos pontos são atribuídos com orientações múltiplas, mas estes contribuem significativamente para a estabilidade da correspondência. Para se definir com maior precisão a orientação, uma parábola é ajustada entre os três valores do histograma mais próximos de cada pico, e então é interpolada a posição do pico para uma melhor precisão.

2.3.4 Descritor local

Nas operações realizadas até então, para cada oitava, foram atribuídos localização, escala e orientação para cada ponto chave, a fim de descrever uma região da imagem em um sistema de coordenadas 2D e prover invariância para esses parâmetros. Para isso, as magnitudes e orientações dos gradientes são amostradas ao redor da localização do ponto chave. O próximo passo consiste em computar o descritor que represente as regiões relativas aos pontos chave. Isto é feito para a região de imagem local que é altamente distintiva e possivelmente invariante para variações restantes, como mudança em iluminação ou ponto de vista 3D.

A computação do descritor de cada ponto chave é feita em relação à orientação definida na seção 2.3.3. Primeiramente é selecionada a imagem filtrada L por meio do filtro Gaussiano referente à escala σ e oitava relativas ao ponto chave. Para se obter invariância à orientação, as coordenadas do descritor e das orientações dos gradientes destes pontos são rotacionadas em relação ao ponto chave. Para maior eficiência, os gradientes são computados para todos os níveis da pirâmide das imagens convoluídas com a função de diferença de filtros Gaussianos, e esses são ilustrados com pequenas setas de cada local amostrado no lado esquerdo da Figura 2.7. Uma função Gaussiana é utilizada para dar peso à magnitude do

gradiente de cada ponto na vizinhança do ponto chave, σ é escolhido igual à metade da largura da janela em que será calculado o descritor.

O descritor de pontos chave é mostrado no lado direito da Figura 2.7, e para cada região, é construído um histograma de orientação de oito direções, com comprimento de cada seta correspondendo à magnitude de cada entrada do histograma.

Figura 2.7: Gradientes da imagem e descritor de pontos chave.

Fonte: (LOWE, 2004).

O peso referente à magnitude de cada pixel é atenuado pela função Gaussiana. A função Gaussiana não é aplicada de modo idêntico ao da seção anterior. É utilizado um peso α para interpolar a direção relativa no histograma.

O descritor é representado por um vetor contendo os valores de todas as orientações das entradas do histograma, correspondendo aos pesos das setas mostradas no lado direito da Figura 2.7. Essa figura ilustra uma matriz 2x2 de orientação de histogramas com 8 direções em cada posição, representando um vetor com 32 elementos descritores para cada ponto chave. Lowe (2004) afirma que os melhores resultados foram obtidos quando se usou uma matriz de 4x4 de histogramas, com 8 orientações cada, resultando em um vetor de 128 elementos descritores para cada ponto chave.

O vetor de elementos descritores é modificado para reduzir os efeitos na mudança de iluminação. O primeiro passo é normalizar o vetor, e em seguida fazer uma mudança no contraste da imagem, na qual cada valor de pixel é multiplicado por uma constante. Uma mudança no brilho na qual uma constante é adicionada para cada pixel da imagem não irá afetar os valores dos gradientes, pois eles são computados de diferentes *pixels*. Dessa forma, o descritor é invariante a pequenas mudanças de iluminação. Mudanças não lineares de iluminação podem ocorrer devido à saturação das câmaras ou devido a mudanças de iluminação que afetem superfícies 3D com diferentes orientações. Esses efeitos podem causar

uma grande mudança nas magnitudes relativas para alguns gradientes, mas afetam menos as orientações dos gradientes. Com isso consegue-se reduzir a influência das grandes magnitudes do gradiente com um valor de limiar. Após a normalização, todos os valores acima de um determinado limiar são ajustados para este limiar. Isto é feito para que direções com magnitude muito grande não dominem a representação do descritor. É sugerido usar um valor de limiar 0,2, e então o vetor é normalizado novamente para a unidade de tamanho. De acordo com Lowe (2004) o valor 0,2 foi obtido por meio de testes usando imagens com diferentes iluminações para um mesmo objeto 3D.

Para cada imagem, são construídos diversos descritores, cada um referente a um ponto chave. Quando se aplica o algoritmo SIFT em uma imagem, tem-se como resultado um conjunto de descritores, que são usados para se fazer a correspondência entre duas imagens.

2.4 Correspondência dos descritores SIFT

Um aspecto importante do algoritmo SIFT é a geração de um número grande de descritores que conseguem cobrir densamente uma imagem quanto à escalas e localização. A quantidade de pontos característicos é particularmente importante para o reconhecimento de objetos, onde a capacidade de se detectar pequenos objetos em ambientes desordenados requer ao menos três pontos característicos de cada objeto, que sejam corretamente combinados (*matched*) para uma identificação confiável.

Para a tarefa de comparação e reconhecimento de imagens, os descritores de pontos chave SIFT são primeiramente extraídos de diferentes imagens para então poderem ser comparados. Uma nova imagem é combinada por comparações individuais de cada descritor com os descritores das imagens *templates* previamente armazenadas em um banco de dados, a fim de encontrar descritores candidatos. Essa atividade é realizada usando a distância Euclidiana em todos os vetores de descritores dos pares de imagens, sendo uma a imagem de entrada e outra a imagem *template*.

Os descritores de pontos chave são altamente distintos, o que permite que um único descritor possa encontrar sua combinação com alta probabilidade em um grande banco de descritores. Entretanto, em uma imagem a ser buscada no banco de dados, muitos descritores de fundo certamente não possuem combinação, aumentando as falsas correspondências (*matches*). A correta combinação pode ser filtrada do conjunto completo de pontos chave

correspondentes pela identificação dos pontos chave que combinam com o objeto em sua localização, escala, e orientação na imagem.

Para se encontrar a correspondência (*match*) entre duas imagens, devem ser encontrados pontos em comum entre as duas. Os pontos de interesse (pontos chave) são detectados utilizando o algoritmo SIFT, onde se obtém para cada ponto, um vetor de descritores com 128 dimensões (subseção 2.3.4). Tendo-se descritores de duas imagens, a tarefa de se encontrar a correspondência de uma imagem em outra é resumida pela busca de se encontrar entre os descritores de uma imagem, os melhores candidatos a serem seus equivalentes na outra imagem. Segundo Lowe (2004), a melhor correspondência candidata para cada ponto chave é encontrada pela identificação de seu vizinho mais próximo em um banco de dados de pontos chave de imagens de treinamento ou *templates*. O vizinho mais próximo é definido como o ponto chave com menor distância Euclidiana para o vetor de descritores invariantes.

Para exemplificar a tarefa de encontrar a correspondência de forma simples entre duas imagens I_1 e I_2 , cujos descritores são definidos respectivamente por di_1 e dj_2 , sendo i, j os índices para cada um dos descritores de cada imagem, k o tamanho de cada descritor e m a magnitude de cada valor dos vetores, tem-se a seguinte representação:

$$\begin{aligned} di_1 &= (m_1 i_1, m_1 i_2, m_1 i_3, \dots, m_1 i_k) \\ dj_2 &= (m_2 j_1, m_2 j_2, m_2 j_3, \dots, m_2 j_k) \end{aligned} \quad (18)$$

A correspondência é realizada encontrando-se o melhor candidato dj_2 para determinado di_1 , e isto é feito procurando-se o vizinho mais próximo de di_1 entre todos os possíveis candidatos k em dj_2 . O melhor candidato é definido por dj_2 , que possui a menor distância Euclidiana em relação à di_1 , ou seja, deseja-se encontrar j para todo i que minimize a função:

$$|di_1 - dj_2| = \sqrt{(m_1 i_1 - m_2 j_1)^2 + (m_1 i_2 - m_2 j_2)^2 + \dots + (m_1 i_k - m_2 j_k)^2} \quad (19)$$

Dessa forma são encontrados todos os pares de descritores correspondentes nas duas imagens I_1 e I_2 . Essa busca exaustiva para encontrar a correspondência dos pontos chave baseada no vizinho mais próximo pode ser demorada, e para tal existem algumas maneiras para se acelerar essa busca.

Uma estrutura de dados usual para encontrar o vizinho mais próximo de descritores de imagens a partir de consultas é a *k-d tree* (BEIS; LOWE, 1997), que é uma forma de árvore de busca binária balanceada.

Uma árvore *k-d tree* (FRIEDMAN et al., 1977) é construída e nela é realizada a busca exaustiva para encontrar a correspondência dos pontos chave da imagem *template* e os pontos chave da imagem a ser trabalhada. O algoritmo denominado *Best-Bin-First* (BBF) (BEIS; LOWE, 1997) é utilizado para a busca, pois retorna o vizinho mais próximo com alta probabilidade. O algoritmo BBF usa uma ordenação de busca modificada para o algoritmo da árvore *k-d tree*, de forma que os descritores são procurados para suas distâncias mais próximas do local de consulta. Um bom resultado pode ser conseguido com baixo custo interrompendo a busca depois que um número específico de vizinhos tenha sido explorado.

2.4.1 Árvore *k-d tree*

Em Berg et al. (2008) é descrita a construção de uma árvore *k-d tree* a partir de um conjunto de pontos P bidimensionais (x, y) . Primeiramente o conjunto é dividido na coordenada x , depois na coordenada y , em seguida, novamente a coordenada x , e assim por diante. Na raiz o conjunto P é dividido com uma linha vertical ℓ em dois subconjuntos aproximadamente de tamanhos iguais. A linha de separação é armazenada na raiz. P_{left} , o subconjunto de pontos para a esquerda ou na linha de separação, é armazenado na subárvore esquerda, e P_{right} , o subconjunto à direita, é armazenado na subárvore direita.

No filho esquerdo da raiz, P_{left} é dividido em dois subconjuntos, com uma linha horizontal; os pontos abaixo são armazenados na subárvore esquerda do filho esquerdo, e os pontos acima são armazenados na subárvore direita. O filho esquerdo armazena a linha de divisão. Da mesma forma, o conjunto P_{right} é dividido com uma linha horizontal em dois subconjuntos, que são armazenadas na subárvore esquerda e direita do filho direito.

Nos netos da raiz, ocorre novamente uma divisão com uma linha vertical. Em geral, os nodos cuja profundidade é “par” são divididos com uma linha vertical, e os nodos cuja profundidade é “ímpar” são divididos com uma linha horizontal. A Figura 2.8 ilustra como é feita a divisão assim como a correspondente árvore binária. Uma árvore como essa é chamada de *k-d tree*, onde k representa a dimensão, portanto essa árvore em discussão seria uma *2-d tree*.

Figura 2.8: Exemplo de uma árvore *k-d tree*. Na esquerda o modo que o plano é subdividido e à direita a correspondente árvore binária.

Fonte: (BERG et al., 2008).

Pode-se construir a árvore *k-d tree* utilizando um procedimento recursivo, e nele são utilizados dois parâmetros: um conjunto de pontos P e um número inteiro (representando a profundidade utilizada nas chamadas recursivas). O parâmetro de profundidade é zero na primeira chamada. A profundidade é importante, pois determina se a divisão ocorre com uma linha vertical ou uma linha horizontal. O processo retorna a raiz da árvore *k-d tree*. A sequência de execução do algoritmo para construir uma árvore *k-d tree* é mostrada no Algoritmo 1 a seguir e foi obtida em Berg et al. (2008).

Algoritmo 1 – ConstruirKdTree($P, depth$)

Entrada: Um conjunto de pontos P e a profundidade atual $depth$.

Saída: A raiz de uma árvore *k-d tree* armazenando P .

- 1: **se** P contém somente um ponto
 - 2: **então retorne** uma folha armazenando esse ponto
 - 3: **senão se** $depth$ é par
 - 4: **então** divide P em dois subconjuntos com uma linha vertical ℓ através da coordenada x mediana dos pontos em P . Considere P_1 o conjunto de pontos para a folha esquerda de ℓ , e P_2 o conjunto de pontos para a folha direita de ℓ .
 - 5: **senão** divide P em dois subconjuntos com uma linha horizontal ℓ através da coordenada y mediana dos pontos em P . Considere P_1 o conjunto de pontos abaixo de ℓ , e P_2 o conjunto de pontos acima de ℓ .
 - 6: $v_{\text{left}} \leftarrow \text{ConstruirKdTree}(P_1, depth + 1)$
 - 7: $v_{\text{right}} \leftarrow \text{ConstruirKdTree}(P_2, depth + 1)$
 - 8: Cria um nodo v armazenando ℓ , torne v_{left} o filho esquerdo de v , e v_{right} o filho direito de v .
 - 9: **retorne** v
-

Os elementos armazenados na árvore *k-d tree* são vetores de alta dimensão em R^d . No primeiro nível da árvore (raiz), os dados são divididos em duas metades por um hiperplano ortogonal para uma dimensão escolhida, com um valor limiar. Geralmente, essa divisão é feita em média na dimensão com a maior variação no conjunto de dados. Ao comparar o vetor da consulta com o valor de particionamento, é fácil determinar a que metade os dados do vetor de consulta pertence. Cada uma das duas metades dos dados é então dividida recursivamente da mesma forma para criar uma árvore binária completamente balanceada. Na parte inferior da árvore, cada nodo da árvore corresponde a um único ponto do conjunto de dados, embora em algumas implementações, os nodos folha podem conter mais de um ponto. A altura da árvore será $\log_2 N$ onde N é o número de pontos no conjunto de dados (SILPA-ANAN; HARTLEY, 2008).

A árvore *k-d tree* foi introduzida em Bentley (1975) como uma generalização de uma árvore binária de dimensão elevada. Diversas variações na construção de uma árvore, incluindo a seleção aleatória do valor de particionamento, foram sugeridas. Mais tarde, uma árvore *k-d tree* otimizada, com um tempo logarítmico de busca, foi proposto em Friedman (1977). No entanto, essa busca em tempo logarítmico não se aplica a árvores de grande dimensão, onde o tempo de busca pode tornar-se quase linear.

Assim, embora as árvores *k-d trees* sejam eficazes e eficientes em baixas dimensões, a sua eficácia diminui para os dados com altas dimensões. Isto porque com dados de alta dimensão, uma árvore *k-d tree* geralmente leva muito tempo para encontrar a solução ideal. Busca-se encontrar o mínimo absoluto com um desempenho probabilístico dentro de limites razoáveis com o algoritmo *Best-Bin-First* (BBF).

2.4.2 Algoritmo de busca BBF

A busca pelo vizinho mais próximo é um componente importante em aplicações de Visão Computacional. Em uma aplicação típica, um grande número de descritores SIFT extraídos de uma ou muitas imagens são armazenadas em um banco de dados. Uma consulta geralmente envolve encontrar o vetor ou os vetores com descritores de melhor correspondência no banco de dados para um descritor SIFT (SILPA-ANAN; HARTLEY, 2008).

O algoritmo *Best-Bin-First* (BBF), é um algoritmo de pesquisa, que encontra o vizinho mais próximo para uma grande parte das consultas, e encontra um vizinho muito próximo, nos demais casos (BEIS; LOWE, 1997).

As folhas de uma árvore *k-d tree* formam uma partição completa do espaço de dados, com uma propriedade interessante, os *bins* (nodos da árvore) são menores em regiões de alta densidade e maiores em regiões de baixa densidade.

A busca do algoritmo BBF inicia percorrendo a árvore para encontrar o nodo que contém o ponto de consulta. Isso requer somente $d = \lceil \log_2 N \rceil$ (baseado em uma árvore construída de forma balanceada) comparações escalares, e em geral o ponto recuperado do nodo é uma boa aproximação para o vizinho mais próximo.

Toda ramificação da árvore pode ser podada se a região do espaço que ela representa está além do ponto de consulta q para o vizinho mais próximo, contendo uma distância superior a um raio de busca estabelecido. A busca termina quando todas as ramificações tenham sido podadas. Este processo pode ser muito eficiente em espaços com baixas dimensões, mas em altas dimensões há muito mais nodos adjacentes para o nodo central que deve ser examinado, e o desempenho degrada rapidamente. De forma interessante, uma grande porção dessa busca é gasta examinando nodos na qual somente uma pequena fração do seu volume poderia possivelmente prover o vizinho mais próximo.

A Figura 2.9 exemplifica uma busca BBF em uma árvore *k-d tree* com 8 pontos, onde $k = 2$. Na busca, os nodos próximos do ponto de consulta q são examinados. A figura mostra um círculo centrado em q com um certo raio determinado, e o nodo adjacente mais próximo no espaço (abaixo de q).

Figura 2.9: Exemplo de busca BBF em uma árvore *k-d tree* com 8 pontos.
Fonte: (BEIS; LOWE, 1997).

2.5 Extração dos *inliers*

Para a detecção e reconhecimento de objetos em imagens, tomando como base a correspondência dos pontos chave extraídos pelo algoritmo SIFT (LOWE, 2004), faz-se necessário eliminar os pontos que não correspondem ou não se ajustam com o modelo do objeto desejado. Apenas são preservados aqueles pontos chave que realmente condizem com o objeto desejado. Neste processo o algoritmo RANSAC (*RANdom Sample Consensus*) (FISCHLER; BOLLES, 1981) é um dos mais comumente encontrado na literatura para a obtenção de um resultado robusto. Além de ser amplamente empregado para o reconhecimento de objetos, o algoritmo RANSAC possibilita encontrar as correspondências características geometricamente consistentes usadas para resolver a homografia (matriz fundamental) entre pares de imagens. Esta atividade é realizada para a construção de panoramas de imagens utilizando descritores invariantes como é o caso dos extraídos das imagens com o algoritmo SIFT (BROWN; LOWE, 2007).

O algoritmo RANSAC proposto por Fischler e Bolles (1981) é um método de estimação robusto projetado para extração dos *inliers*¹ do conjunto total de dados de entrada. Esses dados de entrada seriam os pontos chave detectados pelo algoritmo SIFT (seção 2.3) aplicado às imagens. O RANSAC é geralmente usado em Visão Computacional, como por exemplo, para resolver simultaneamente o problema de correspondência entre pontos de duas imagens e estimar a matriz fundamental M relacionada ao par de imagens.

A principal vantagem no uso do RANSAC é que ele possui a propriedade de ser um estimador robusto. Estimadores de parâmetros clássicos, como mínimos quadrados, se baseiam em todos os dados que tomam como entrada. Desse modo, se grande parte dos dados vier de uma fonte em que haja erros inerentes, ou ruidosos, (isto é, algum tipo de *outlier*²) o modelo final estimado será prejudicado. O RANSAC, porém, foi desenvolvido com o intuito de evitar que *outliers* prejudiquem a computação do modelo final.

Como mencionado por Fischler e Bolles (1981), ao contrário das técnicas convencionais que usam muitos dados para obter uma solução inicial, e então, eliminar os *outliers*, o RANSAC utiliza apenas um conjunto com número mínimo e suficiente de pontos

¹ *inliers*: pontos de dados que se ajustam com um determinado modelo desejado dentro de uma certa tolerância de erro.

² *outliers*: pontos de dados que não se ajustam ao modelo correspondente ao objeto desejado, estão fora de uma certa tolerância de erro.

necessários para uma primeira estimativa, e procede aumentando o conjunto com pontos de dados consistentes. Uma amostra mínima depende do modelo a ser gerado. Geralmente, é o número s de dados que será utilizado para inicializar um modelo. Por exemplo, no caso do modelo a ser estimado ser uma reta, uma amostra mínima consiste de apenas dois elementos.

A Figura 2.10 ilustra um exemplo apresentado por Hartley e Zisserman (2003) que pode ser facilmente visualizado. Nele é estimada uma reta ajustada a um conjunto de pontos bidimensionais. Isso pode ser feito estimando uma transformação unidimensional $x' = ax + b$ entre os pontos correspondentes em duas linhas.

O problema que é ilustrado na Figura 2.10 (a) é encontrar a reta que minimiza a soma das distâncias a partir de um conjunto de pontos de dados 2D, sujeito à condição que nenhum dos pontos válidos diverge dessa reta por mais de t unidades. Deseja-se então a reta ajustada aos dados e a classificação de dados em *inliers* (pontos válidos) e *outliers*. O limiar t é fixo de acordo com a medida de ruído (por exemplo, $t = 3\sigma$). Esse exemplo mostra que o ajuste dos pontos é severamente afetado pelos *outliers*.

Já com o uso do algoritmo RANSAC (Figura 2.10 (b)), dois pontos são selecionados aleatoriamente, e esses pontos definem uma reta. Essa seleção aleatória é repetida um número de vezes e a reta com maior número de pontos que ficam dentro de um limiar de distância é considerada como ajuste robusto. Os pontos dentro da distância do limiar são os *inliers* (e constitui o conjunto dos consensos).

Na Figura 2.10 (a) e (b) os pontos sólidos representam os *inliers*, e os pontos abertos representam os *outliers*. Na Figura 2.10 (b) as linhas pontilhadas indicam a distância do limiar, a linha passando pelos pontos a e b constitui a linha ajustada, e o ponto c é um *outlier*.

Figura 2.10: Estimação de uma reta ajustada a um conjunto de pontos 2D.

Fonte: (HARTLEY; ZISSERMAN, 2003).

A partir do conjunto mínimo necessário de pontos selecionados aleatoriamente, o RANSAC mede a precisão do modelo a ser estimado para todos os outros pontos,

classificando-os em *inliers* ou *outliers*. Para tal, é definido um valor de limiar t , que é a distância máxima do modelo que um dado pode estar para ser considerado um *inlier*. O conjunto de *inliers* associado a um modelo é o conjunto de consenso (*consensus set*) do modelo. Assim, com várias iterações, o algoritmo é capaz de encontrar um modelo preciso para o problema em questão. Uma vantagem do RANSAC é a sua habilidade de realizar a estimativa de parâmetros de um modelo de forma robusta, ou seja, ele pode estimar parâmetros com um alto grau de acerto mesmo quando um número significativo de *outliers* esteja presente nos dados analisados (FISCHLER; BOLLES, 1981).

O tamanho do conjunto de consenso é o que determina o quão bom um modelo é. Um parâmetro T define o número necessário que deve ser alcançado para um modelo ser considerado satisfatório. Durante a execução do RANSAC, quando for obtido mais de T *inliers*, o algoritmo finaliza, gerando novamente um modelo utilizando todos os T *inliers* obtidos, que será o modelo final encontrado pelo algoritmo (refinamento do modelo). É necessário definir um número máximo N de iterações que o algoritmo irá realizar, para evitar um número infinito de iterações. Se um determinado modelo não alcançar T *inliers*, o algoritmo gera um novo modelo a partir de uma nova seleção de amostra mínima dos dados de entrada. O algoritmo prossegue até conseguir obter um modelo satisfatório, ou alcançar o número máximo de iterações. No caso de a execução do algoritmo finalizar por alcançar o número máximo de iterações, pode-se retornar o conjunto com mais *inliers*, ou retornar que o algoritmo falhou.

Dado um modelo com parâmetros, deseja-se estimá-lo, para tal, os parâmetros podem ser estimados a partir de um número S de itens em um conjunto de dados conhecido. Os parâmetros que podem ser alterados de modo a determinar um bom resultado para o RANSAC são: t , que indica o limiar de distância que uma determinada entrada pode estar do modelo para ser considerado um *inlier*; T , que representa o número de *inliers* a ser alcançado para que um modelo possa ser considerado um bom modelo, sendo que isso depende de cada caso, normalmente determinado empiricamente; e N , que representa o número máximo de iterações que devem ser realizadas.

As etapas de execução são resumidas como segue no Algoritmo 2 e foram obtidas em Fischler e Bolles (1981) e Hartley e Zisserman (2003):

Algoritmo 2 – RANSAC

Objetivo: Ajuste robusto de um modelo para um conjunto S que contém *outliers*.

Passos:

- 1: Selecione aleatoriamente o número mínimo de pontos s requeridos de um conjunto S para determinar os parâmetros do modelo.
 - 2: Determine o conjunto de pontos de dados S_i que estão dentro de um limiar de distância t do modelo (matriz fundamental). O conjunto S_i é o conjunto de consensos da amostra e define os *inliers* de S .
 - 3: Se o tamanho de S_i (número de *inliers*) é maior que T (número necessário a ser alcançado), os parâmetros modelo são re-estimados usando todos os *inliers* identificados (S_i) e termina.
 - 4: Se o tamanho de S_i é menor que T , um novo subconjunto é selecionado e repete os passos anteriores.
 - 5: Após N tentativas o maior conjunto de consensos de S_i é selecionado, e o modelo é re-estimado usando todos os pontos no subconjunto S_i .
-

O número de iterações N é escolhido grande o suficiente para assegurar uma probabilidade p (normalmente estabelecido para 0.99), que pelo menos uma das amostras aleatórias do conjunto de s pontos não inclui um *outlier*. Considere w a representação da probabilidade que algum ponto de dado selecionado é um *inlier* e $\varepsilon = 1 - w$ a probabilidade de observar um *outlier*. Pelo menos N iterações (cada uma de s pontos) são requeridas, onde (HARTLEY; ZISSERMAN, 2003):

$$(1 - w^s)^N = 1 - p \quad (20)$$

e assim o número de iterações necessárias é dado por:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - \varepsilon)^s)} \quad (21)$$

O RANSAC é capaz de tratar com eficiência um conjunto de dados que contém uma porcentagem significativa de erros. Este paradigma é particularmente aplicável à análise de cena por causa dos detectores de características locais, que frequentemente cometem erros, que podem ser de dois tipos: erros de classificação e erros de medida. Erros de classificação ocorrem quando um detector de característica identifica uma porção de uma imagem incorretamente como uma ocorrência de uma característica. Erros de medida acontecem quando o detector de característica identifica a característica corretamente, mas comete erros de cálculo em um de seus parâmetros (por exemplo, seu local na imagem). Erros de medida geralmente seguem uma distribuição normal, e então a suposição de suavização é aplicável a eles. Porém, erros de classificação têm um efeito significativo maior que erros de medida (FISCHLER; BOLLES, 1981).

2.5.1 Algoritmo RANSAC aplicado ao ajuste na correspondência dos descritores SIFT

O pseudocódigo do algoritmo RANSAC apresentado na Figura 2.11 calcula um melhor ajuste na correspondência dos descritores SIFT previamente extraídos de duas imagens. Para o problema específico de remoção de *outliers* na correspondência de duas imagens, a matriz fundamental M pode ser calculada resumidamente da seguinte forma:

- Selecionar aleatoriamente um subconjunto de quatro pontos correlacionados, retirados do conjunto total de pontos chave correlacionados entre duas imagens (pontos chave obtidos pelo algoritmo SIFT);
- Para cada matriz M computada, determinar o número de pontos com distância até a linha epipolar, ou residual, menor que um limiar;
- Selecionar a matriz M que apresenta o maior número de pontos com residual inferior ao máximo definido (*consensus*);
- Recalcular a matriz M considerando todos os *inliers*.

```

entrada:
1  features[] //vetor de descritores; somente os descritores com
   correspondência entre duas imagens são usados na computação da
   homografia. Esses elementos representam o conjunto de observações.
2  n //número de elementos no vetor de descritores.
3  m //número mínimo de correspondências necessárias para instanciar o
   modelo a ser computado, foi usado o valor 4.
4  p_badxform //probabilidade desejada que a transformação final
   retornada pelo RANSAC é corrompida pelos outliers (isto é, a
   probabilidade que em nenhuma amostra de todas os inliers são
   extraídos), foi usado o valor 0,001.
5  err_tol //correspondências dentro desta distância de um modelo
   computado são consideradas como inliers, foi usado o valor 3,0.

saída:
6  inliers[] //vetor de ponteiros para o conjunto final dos inliers.
7  n_in //número de elementos no conjunto final dos inliers.
8  M //matriz de transformação computada usando RANSAC ou NULO no
   erro ou se a transformação não pôde ser computada.
9  in_frac := 0.25 //estimativa pessimista de fração de inliers.
10 in_max := 0, k := 0, consensus[] := NULO, consensus_max[] := NULO
11 matched[] e nm := descritores com correspondência entre as duas
   imagens. nm é número de correspondências no vetor matched[].
12 se (nm < m) //número mínimo de correspondências insuficiente
13   fim
14 in_min := número mínimo de inliers calculados com uma função usando o
   número nm de supostas correspondências.

```

Figura 2.11: Pseudocódigo do algoritmo RANSAC para calcular um melhor ajuste na correspondência dos descritores SIFT.

```

15 p := (1.0 - in_fracm)k //probabilidade que m pontos sejam inliers.
16 enquanto (p > p_badxform)
17     sample[] := amostra mínima de dados selecionados aleatoriamente, 4
        amostras.
18     pts[] e mpts[] := vetor dos pontos a partir de sample[] que
        correspondem aos descritores das duas imagens.
19     M := matriz homográfica planar dos mínimos quadrados a partir dos
        pontos de correspondência pts[] e mpts[].
20     se (M <> NULO)
21         consensus[] e in := um vetor de consensos a partir do conjunto
            dos descritores com correspondência e da matriz homográfica M
            (calculada anteriormente), para um modelo dado e função de
            erro. in é o número de elementos do vetor consensus[].
22         se (in > in_max)
23             consensus_max[] := consensus[]
24             in_max := in
25             in_frac := in_max / nm
26         senão
27             consensus[] := NULO
28             sample[] := NULO
29         incrementa(k)
30     p := (1.0 - in_fracm)k //probabilidade que m pontos sejam
        inliers.
31 fim_enquanto
    //calcula a transformação final baseada no melhor conjunto de
    consensos
32 se (in_max >= in_min)
33     pts[] e mpts[] := vetor dos pontos a partir de consensus_max[] que
        correspondem aos descritores das duas imagens.
34     M := matriz homográfica planar dos mínimos quadrados a partir dos
        pontos de correspondência pts[] e mpts[].
35     consensus[] e in := um vetor de consensos a partir do conjunto dos
        descritores com correspondência e da matriz homográfica M
        (calculada anteriormente), para um modelo dado e função de erro. in
        é o número de elementos do vetor consensus[].
36     pts[] e mpts[] := vetor dos pontos a partir de consensus[] que
        correspondem aos descritores das duas imagens.
37     M := matriz homográfica planar dos mínimos quadrados a partir dos
        pontos de correspondência pts[] e mpts[].
38     se (inliers[] <> NULO)
39         inliers[] := consensus[]
40         consensus[] := NULO
41     n_in := in
42 senão se (consensus_max <> NULO)
43     se (inliers[] <> NULO)
44         inliers[] := NULO
45     n_in := 0
46 retorna inliers[], n_in, M

```

Figura 2.11: Pseudocódigo do algoritmo RANSAC para calcular um melhor ajuste na correspondência dos descritores SIFT (continuação).

2.5.2 Aplicação para detecção e reconhecimento de objetos

Nesta subseção são apresentados alguns exemplos de detecção e reconhecimento de objetos, cujo ajuste na correspondência dos descritores SIFT foi realizado com o uso do

algoritmo RANSAC. Assim como já elucidado anteriormente, o algoritmo RANSAC é um bom algoritmo para ser usado no reconhecimento de objetos, pois minimiza a influência de *outliers*, eliminando-os do conjunto de pontos chave correspondentes entre duas imagens. São preservados apenas os pontos chave que realmente condizem ao objeto buscado.

A Figura 2.12 apresenta um resultado muito bom de detecção e reconhecimento de um objeto (imagem “box.png”) em uma imagem (imagem “box_in_cene.png”), onde 45 pontos chave apresentaram correspondência entre as duas imagens. As imagens utilizadas neste experimento são imagens de teste disponibilizadas pela biblioteca de Visão Computacional OpenCV (OPENCV, 2010). Na esquerda na Figura 2.12 são mostrados os pontos chave que correspondem aos pontos chave do objeto desejado, e na direita, são mostrados os *inliers* extraídos com o RANSAC.

Figura 2.12: Imagem “box.png” detectada na imagem “box_in_cene.png” com 45 pontos com extração dos *inliers* pelo algoritmo RANSAC.

A Figura 2.13 apresenta um segundo resultado bom com extração dos *inliers* pelo algoritmo RANSAC. Neste exemplo de detecção e reconhecimento a imagem de desenho foi detectada com 39 pontos chave.

Figura 2.13: Imagem de desenho detectada com 39 pontos (*inliers*) extraídos pelo algoritmo RANSAC.

Nos exemplos da Figura 2.12 e da Figura 2.13 são exibidas as localizações exatas dos objetos detectados nas imagens de teste. Isto foi possível, pois o algoritmo RANSAC possibilita encontrar as correspondências caracteristicamente consistentes usadas para resolver a homografia (matriz fundamental) entre pares de imagens, atividade muito usada na construção de panoramas de imagens (BROWN; LOWE, 2007).

Diversos experimentos de detecção e reconhecimento foram realizados com placas de sinalização usando o algoritmo RANSAC para eliminar os *outliers* e, preservar com isso apenas os pontos chave realmente correspondentes (*inliers*) entre as imagens de entrada (imagens de ruas contendo placas) e os *templates* (imagens das placas de sinalização).

A Figura 2.14 e a Figura 2.15 apresentam dois exemplos de experimentos realizados para detecção e reconhecimento de placas de sinalização. Estes exemplos proporcionaram bons resultados com uma quantidade razoável de *inliers* extraídos, sendo 11 em cada exemplo. A Figura 2.14 apresenta o resultado da detecção e reconhecimento da placa de sinalização “Velocidade máxima permitida 40 Km/h”.

Figura 2.14: Placa “Velocidade máxima permitida 40 Km/h” detectada e reconhecida com 11 pontos chave com extração dos *inliers* pelo algoritmo RANSAC.

A Figura 2.15 apresenta o resultado da detecção e reconhecimento da placa de sinalização “Parada obrigatória” usando o algoritmo RANSAC para a extração dos pontos chave realmente correspondentes entre as duas imagens (*inliers*).

Figura 2.15: Placa de sinalização “Parada obrigatória” detectada e reconhecida com 11 pontos chave com extração dos *inliers* pelo algoritmo RANSAC.

2.5.3 Problemas no reconhecimento de placas utilizando o algoritmo RANSAC

No algoritmo RANSAC, quando uma quantidade razoável de pontos chave for classificada como *inliers* relativos ao modelo estimado, tem-se bons resultados no reconhecimento de objetos. Porém, em muitos casos em que a quantidade de pontos chave correspondentes entre as imagens que estão em conformidade com o modelo não for suficiente, tem-se a possibilidade de resultados enganosos ou o que é mais provável, o não reconhecimento do objeto.

São apresentados exemplos de experimentos que não tiveram resultados bons, devido ao fato de poucos pontos chave terem sido estimados ao modelo (pontos chave das imagens dos *templates*) ou estarem muito próximos. Com a extração de *inliers* não suficientes para o algoritmo RANSAC calcular a homografia (matriz fundamental) entre as duas imagens, as tentativas de reconhecimento das placas de sinalização mostradas na Figura 2.16, na Figura 2.17 e na Figura 2.18 falharam.

Figura 2.16: Pontos chave correspondentes entre as imagens na tentativa de reconhecimento da placa de sinalização "Proibido virar à direita".

Figura 2.17: Pontos chave correspondentes entre as imagens na tentativa de reconhecimento da placa de sinalização "Siga em frente ou à direita".

Figura 2.18: Pontos chave correspondentes entre as imagens na tentativa de reconhecimento da placa de sinalização "Estacionamento regulamentado".

A Figura 2.19 ilustra alguns exemplos de algumas placas de sinalização com problemas no reconhecimento usando o algoritmo RANSAC para extrair os pontos chave que realmente correspondem entre as duas imagens (*inliers*). Essas duas imagens seriam a imagem buscada (*template* da placa de sinalização) e a imagem de entrada contendo a placa de sinalização. Na figura apresentada, estão relacionadas as seguintes placas de sinalização: (a) Siga em frente ou à direita; (b) Proibido virar à direita; (c) Siga em frente ou à esquerda; (d) Duplo sentido de circulação; (e) Sentido proibido; (f) Proibido parar e estacionar; (g) Proibido retornar à direita; (h) Estacionamento regulamentado; (i) Curva à direita; (j) Saliência ou lombada; (k) Depressão e (l) Estreitamento de pista à esquerda.

Figura 2.19: Exemplos de algumas placas de sinalização que não apresentaram bons resultados no reconhecimento usando o algoritmo RANSAC para extrair os *inliers*.

Como boa parte das placas de sinalização apresenta características com poucos detalhes e conseqüentemente poucos pontos chave extraídos, ou em alguns casos os pontos chave extraídos estão muito próximos, foi construído um algoritmo em substituição ao algoritmo RANSAC para ser utilizado neste trabalho de doutorado com características específicas para a detecção e reconhecimento de placas de sinalização. Este algoritmo foi idealizado para trabalhar com pequenas quantidades de pontos chave, na tentativa de maximizar a detecção e o reconhecimento das placas de sinalização de forma automática e minimizar os falsos positivos. A seção 5.4 detalha todo o processo para a detecção e

reconhecimento das placas de sinalização, e mais especificamente a subseção 5.4.1 descreve o algoritmo construído para a extração dos *inliers* dos *templates* das placas de sinalização em correspondência com as imagens utilizadas nos experimentos.

2.6 Considerações Finais

Neste capítulo foram descritos os algoritmos utilizados para compor a metodologia proposta, neste trabalho, para a detecção e reconhecimento das placas de sinalização. O algoritmo SIFT proposto por Lowe (2004) foi utilizado para a extração dos descritores dos pontos chave das imagens utilizadas (imagens de entrada e *templates* das placas de sinalização). Foi visto que, para a detecção de sinais de trânsito, em uma fase inicial, é efetuada a correspondência (*matching*) entre essas imagens, utilizando para isso, o algoritmo *Best-Bin-First* (BBF) (BEIS; LOWE, 1997). Essa atividade busca encontrar pontos em comum entre duas imagens, ou seja, busca encontrar entre o conjunto de descritores de uma imagem, os melhores candidatos a serem seus equivalentes, na outra imagem.

Foi ressaltado que o algoritmo RANSAC (FISCHLER; BOLLES, 1981), embora seja um dos mais presentes nos trabalhos encontrados na literatura para identificar os pontos chave que se ajustam com o modelo do objeto desejado (identificar os *inliers*), não se mostrou adequado para o estágio de reconhecimento das placas de sinalização com as imagens utilizadas neste trabalho.

Capítulo 3

Aspectos Gerais de Cartografia e Sistemas de Coordenadas

3.1 Considerações Iniciais

Neste capítulo, inicialmente é apresentada uma descrição de Sistemas Móveis de Mapeamento, relacionando os sensores utilizados para coleta de dados. Na sequência, são descritos os Sistemas de Referência empregados neste trabalho, que são utilizados para descrever de modo único as posições dos objetos. Por fim, é dada ênfase na obtenção dos dados do GPS/INS, na interseção fotogramétrica utilizada para calcular as coordenadas das placas no terreno e no processo de calibração das câmaras.

3.2 Sistemas Móveis de Mapeamento

Na Cartografia, a integração de tecnologias digitais vem criando novas opções de abordagem de problemas recentes ou já conhecidos, propiciando o surgimento de outros produtos ou serviços distintos dos que já existem e exigindo recursos humanos qualificados. A combinação de dados espaciais e temporais, obtidos com base nestas integrações, pode ser o princípio básico para a solução de vários problemas do mapeamento. Uma das possibilidades são os Sistemas Móveis de Mapeamento (SMM), que integram sensores e dados de posicionamento e imageamento (GALLIS, 2002).

Os sistemas móveis de mapeamento são definidos como plataformas sobre as quais são integrados sistemas de posicionamento e de imageamento para prover rápida e eficientemente a coleta de informações sobre os atributos do espaço objeto a partir de imagens. Os sistemas móveis de mapeamento podem ser concebidos e especializados, nos seguintes níveis (SILVA et al., 1997):

- Orbital: com aplicação em amplas regiões observadas do espaço e com grande variedade de dados;
- Aéreo: com aplicações em áreas rurais e urbanas, limitadas ao mapeamento em escalas médias e grandes;
- Terrestre: com aplicações rodoviárias e ferroviárias.

Em geral, um Sistema Móvel de Mapeamento Terrestre (SMMT) consiste da integração de sensores embarcados em veículos terrestres, para se fazer a navegação e a coleta de dados para fins de mapeamento. Esses sensores são classificados em duas categorias: posicionamento e mapeamento (TAO et al., 2001) (BARBOSA, 2006).

Os sensores de posicionamento podem ser externos ou internos. Exemplo de sensor de posicionamento externo é o GPS (*Global Positioning System*) e de sensor interno o INS (*Inertial Navigation System*).

Os sensores de mapeamento são classificados em passivos ou ativos. Os passivos são câmaras de vídeo ou fotográficas, e um sistema de varredura *laser* (*laser scanner*) é exemplo de sensor de mapeamento ativo.

Os sensores de posicionamento (GPS e INS) podem fornecer de maneira síncrona a orientação exterior dos sensores de mapeamento (câmaras de vídeo), de tal forma que os elementos de orientação exterior são conhecidos, e a reconstrução tridimensional pode ser efetuada diretamente utilizando pontos ou feições correspondentes. Este é precisamente o conceito subjacente do georreferenciamento direto³ (TAO; LI, 2007). Um sistema inercial, além da navegação e posicionamento, também fornece, em perfil, o eixo do deslocamento do veículo.

Dentre as várias aplicações de um SMMT, observa-se o interesse crescente de empresas privadas e órgãos públicos relacionados ao planejamento, construção, conservação e utilização de rodovias, sobretudo, a partir do êxito tecnológico demonstrado pelos sistemas móveis já desenvolvidos.

³ O principal objetivo do georreferenciamento direto, ou seja, determinação direta dos parâmetros de orientação exterior é o de minimizar, ou mesmo eliminar a necessidade de pontos de controle para orientar as imagens (RUY, 2004).

3.3 Sistema de Referência

Os sistemas de referência são utilizados para descrever as posições de objetos. Quando é necessário identificar a posição de uma determinada informação na superfície da Terra são utilizados os Sistemas de Referência Terrestres ou Geodésicos. Estes por sua vez, estão associados a uma superfície que mais se aproxima da forma da Terra, e sobre a qual são desenvolvidos todos os cálculos das suas coordenadas. As coordenadas podem ser apresentadas em diversas formas: em uma superfície esférica recebem a denominação de coordenadas geodésicas e em uma superfície plana recebem a denominação da projeção às quais estão associadas, como por exemplo, as coordenadas planas UTM (IBGE, 2011).

3.3.1 Sistemas de Coordenadas Geodésicas

Independente do método utilizado para se representar ou projetar uma determinada superfície no plano, deve-se adotar uma superfície que sirva de referência, garantindo uma concordância das coordenadas na superfície esférica da Terra. Neste sistema as coordenadas são definidas sobre um elipsóide (IBGE, 2011). As definições de coordenadas geodésicas de um ponto qualquer P na superfície do elipsóide são (MONICO, 2000):

- latitude geodésica (ϕ): é o ângulo contado sobre o meridiano que passa por P , compreendido entre a normal passante por P e o plano equatorial.
- longitude geodésica (λ): é o ângulo contado sobre o plano equatorial, compreendido entre o meridiano de *Greenwich* e o ponto P .
- altitude elipsoidal (h): conhecida também por altitude geométrica, corresponde à distância de P à superfície do elipsóide medida sobre a sua normal.

Deste modo, a localização de qualquer ponto da superfície terrestre poderá ser definida, de modo único, quando se dispõe de um sistema de coordenadas único e materializado. Quando é necessário definir a posição de um determinado objeto na superfície na Terra, são utilizados os Sistemas de Referência Geodésicos. As informações posicionais apresentadas em uma superfície elipsóidica recebem a denominação de coordenadas geodésicas (MONICO, 2000). A Figura 3.1 ilustra o Sistema de Coordenadas Geodésicas.

Figura 3.1: Coordenadas geodésicas (latitude (ϕ) e longitude (λ)) e altitude geométrica (h) de um ponto qualquer P .

Fonte: Adaptado de Monico (2000).

As coordenadas referidas a um determinado Sistema de Referência Geodésico podem ser representadas no plano através das componentes Norte e Leste e são o tipo de coordenadas regularmente encontrado em mapas. Para representar as feições de uma superfície curva em plana, são necessárias formulações matemáticas chamadas de Projeções. Diferentes projeções podem ser utilizadas na confecção de mapas, e no Brasil a projeção mais utilizada é a Universal Transversa de Mercator – UTM (IBGE, 2011).

3.3.2 Sistema UTM

O sistema de projeção UTM (Universal Transversa de Mercator) é largamente utilizado no Brasil e é normatizado pelo IBGE (Instituto Brasileiro de Geografia e Estatística) para as escalas que constituem o levantamento sistemático, ou seja, entre 1:1.000.000 e 1:25.000. Para o caso de escalas maiores, não existe um padrão nacional. Há, portanto, algumas normas como, por exemplo, a do IGC (Instituto Geográfico e Cartográfico), que regulamenta o uso para o estado de São Paulo. O sistema UTM é um dos sistemas derivados da projeção conforme de Gauss, com a finalidade de reduzir as deformações nos extremos do fuso. Neste sistema o cilindro que era tangente ao meridiano central passa a ser secante ao

elipsóide de revolução, e passa-se a ter duas linhas de distorção nula e equidistantes ao meridiano central. Tal sistema de projeção tem representação entre os paralelos 84°N a 80°S (MONICO, 2000).

Segundo Brunetti (1994), o sistema UTM conserva ângulos e a forma de pequenas áreas, tem como superfície de projeção 60 cilindros transversos e secantes à superfície de referência (elipsóide), cada um com amplitude de 6°. O sistema UTM possui fusos com 6° de amplitude em longitude, limitados pelos meridianos múltiplos de 6° (... , 60°, 54°, 48°, ...), e o coeficiente de deformação (K ou m_0) no meridiano central é igual 0,9996. Na interseção dos cilindros com a superfície de referência – linhas de secância – o coeficiente de deformação linear é unitário. A Figura 3.2 apresenta a distribuição das cartas 1: 1.000.000 para o Brasil.

Figura 3.2: Distribuição das cartas topográficas⁴ ao milionésimo no Brasil.

Fonte: (CÂMARA et al., 2001).

⁴ Carta topográfica é o documento que representa, de forma sistemática, geralmente em escalas entre 1:100.000 e 1:25.000, a superfície terrestre por meio de projeções cartográficas.

A origem das coordenadas TM (Transversa de Mercator), do qual são obtidas as coordenadas UTM, em cada sistema parcial, encontra-se no cruzamento do equador com o meridiano central. No entanto, as coordenadas y a oeste do meridiano central e x ao sul do equador são negativas. Logo, para se obter coordenadas positivas, é feita uma translação, acrescentando-se as constantes $10.000.000m$ para as coordenadas N , ao sul do equador, e $500.000m$ para as coordenadas E . Assim, as coordenadas (N, E) no sistema UTM podem ser obtidas a partir das coordenadas (x, y) no sistema TM (MONICO, 2000). No Hemisfério Sul, por:

$$N = 10.000.000 + 0,9996x \quad (22)$$

$$E = 500.000 + 0,9996y \quad (23)$$

e no Hemisfério Norte, por:

$$N = 0,9996x \quad (24)$$

$$E = 500.000 + 0,9996y \quad (25)$$

O terno de coordenadas (N, E, H) ou (N, E, h) constitui um referencial híbrido, pois (N, E) são coordenadas planas e H (altitude ortométrica) ou h (altitude geométrica) estão referidas a uma superfície não plana. Na superfície onde H é definido, todos os pontos possuem o mesmo potencial do campo de gravidade da Terra e que melhor se aproxima do nível médio dos mares, o geóide (GALLIS, 2006).

3.3.3 Sistema de Referência WGS84

A projeção UTM toca a Terra em várias longitudes, denominadas Meridianos Centrais e usa um ponto de projeção no centro da Terra. O modelo matemático (*datum*⁵) é WGS84 que define um elipsóide. O *datum* WGS84 (*World Geodetic System – 1984*) é atualmente utilizado nos levantamentos geodésicos no Brasil e no mundo, porém não é oficializado como *datum* a ser empregado na representação cartográfica nacional (GALLIS, 2006). O WGS84 é a quarta versão do sistema de referência geodésico global estabelecido pelo U.S. *Department of Defense* (DoD) desde 1960, com o objetivo de fornecer

⁵ Um *datum* caracteriza-se por uma superfície de referência posicionada em relação à Terra (IBGE, 2011).

o posicionamento e navegação em qualquer parte do mundo, através de informações espaciais, conforme IBGE (2011).

Ele é o sistema de referência das efemérides operacionais do sistema GPS. Desta forma, quando um levantamento é efetuado usando um receptor GPS, as coordenadas dos pontos envolvidos são obtidas nesse sistema de referência. O WGS84 tem sua origem no centro de massa da Terra, com os eixos cartesianos X , Y , Z idênticos aos do CTRS (*Conventional Terrestrial Reference System* – Sistema de Referência Terrestre Convencional), para a época 1984. O elipsóide de referência adotado é o GRS 80, um elipsóide de revolução geocêntrico (MONICO, 2000). O sistema de referência do GPS é ilustrado na Figura 3.3.

Figura 3.3: Sistema de Referência do GPS.
Fonte: (MONICO, 2000).

3.4 Sistema de Posicionamento Global (GPS)

O GPS (*Global Positioning System*), também denominado de NAVSTAR-GPS (*NAVigation Satellite with Time And Ranging*), é um sistema de rádio-navegação desenvolvido pelo Departamento de Defesa dos Estados Unidos, para fins de navegação, posicionamento geodésico e topográfico. Este sistema é resultante da fusão de dois programas paralelos financiados pelo governo norte-americano a respeito de um sistema de orientação de abrangência global de alta precisão por satélites: o da Marinha e o da Aeronáutica. Em razão da precisão proporcionada pelo sistema, uma grande comunidade usuária emergiu dos mais variados segmentos da comunidade civil (navegação, posicionamento geodésico, agricultura,

comunidade civil etc.). O segmento espacial consiste de uma constelação de 24 satélites distribuídos em seis planos orbitais igualmente espaçados, numa altitude aproximada de 20.200 Km. O sistema foi projetado para que, no mínimo quatro satélites sejam visíveis e com isso forneça posicionamento 24 horas por dia em qualquer lugar da superfície terrestre (MONICO, 2000). A Figura 3.4 ilustra a constelação de satélites GPS.

Figura 3.4: Constelação dos satélites GPS.
Fonte: (MONICO, 2000).

O princípio básico de navegação pelo GPS consiste na medição de distâncias entre o receptor e os satélites. Conhecendo-se as coordenadas dos satélites num dado sistema de referência (WGS84 – *World Geodetic System* – 1984), é possível se obter as coordenadas geocêntricas da antena do receptor no mesmo referencial. O uso de outros sistemas de referências é possível mediante a aplicação adequada de modelos de transformações geométricas adequados (MONICO, 2000).

De acordo com Monico (2000), há diferentes técnicas para se determinar a posição de um ponto sobre a superfície terrestre, dentre as quais estão o posicionamento relativo (a posição de um ponto é determinada em relação a outro, cujas coordenadas são conhecidas, utiliza-se para isso o sistema de referência WGS84) e a técnica DGPS (*Differential GPS*), na qual as coordenadas de um ponto são determinadas a partir de correções e variações enviadas por rádio de um outro ponto de posição conhecida.

Para posicionamento relativo deve-se dispor de no mínimo dois receptores GPS, ou então utilizar uma das estações de referência (exemplo, RBMC – Rede Brasileira de Monitoramento Contínuo) como estação base (RUY, 2004).

O método de posicionamento DGPS, desenvolvido principalmente para aplicações em navegação, tem como objetivo oferecer uma precisão melhor que a proporcionada pelo GPS usando o SPS (*Standard Positioning Service*) no modo absoluto. Um conceito simples de DGPS consiste no uso de um receptor estacionário (base) em uma estação com coordenadas conhecidas, rastreando todos os satélites visíveis. Seu processamento permite o cálculo das correções posicionais, correções de pseudodistâncias ou fase da onda portadora. Estas correções são enviadas via satélite, ou através de *link* de rádio, onde são aplicadas aos pontos rastreados nas proximidades da estação, melhorando sua precisão. Maiores detalhes sobre o sistema de navegação GPS são encontrados em Monico (2000).

3.5 Sistema de Navegação Inercial

O sistema inercial é composto fundamentalmente por uma unidade de medida inercial (IMU – *Inertial Measurement Unit*) e um processador de navegação. A unidade inercial é composta por três acelerômetros que medem acelerações lineares com respeito a um referencial inercial, três giroscópios que medem razões angulares percebidas pela IMU, além de sensores auxiliares para a calibração. Os principais sensores responsáveis pela determinação da atitude são os giroscópios, que medem as inclinações da plataforma ao longo do tempo (RUY, 2004).

Os sensores inerciais têm basicamente duas funções em SMM (Sistema Móvel de Mapeamento): determinação da atitude (ângulos de orientação) do sensor de imagem (câmaras) e prover a posição quando o sinal de GPS é perdido por algum motivo, principalmente por obstrução por edificações, árvores, túneis etc. A primeira mede a rotação do sistema referencial da imagem em relação ao referencial fixo e a outra mantém a posição do sensor conhecida. De uma maneira simples, INS (*Inertial Navigation System*) é um conjunto de instrumentos projetado para registrar as variações do estado de inércia de um corpo rígido, a fim de controlar ou monitorar sua trajetória.

O sistema inercial pode complementar o GPS fornecendo as informações de posição inicial e de velocidade após a perda de sinal do receptor. Mesmo quando a visibilidade dos

satélites é insuficiente, menos de quatro satélites visíveis, o INS pode fornecer informações contínuas de trajetória. O Sistema de sensores inerciais pode ser usado para interpolar as posições do GPS quando a recepção de um dado satélite for obstruída. Por outro lado, o bom desempenho do sistema GPS pode suplementar o sistema de navegação inercial por meio da estimação acurada do comportamento de seus erros ao longo do tempo. Estes erros estão relacionados ao problema de deriva das informações de trajetória inerente aos sistemas inerciais. Os erros internos do INS são modelados como desvios nos giros e deslocamentos nos acelerômetros, e são corrigidos com filtragem de *Kalman*, a partir das informações de posição provenientes do GPS (CRAMER; STALLMANN, 2001). A Tabela 1 apresenta as principais características entre os sistemas GPS e INS (RUY, 2004).

Tabela 1: Tabela comparativa entre os sistemas GPS e INS.
Fonte: (RUY, 2004).

	GPS	INS
Princípio de observação	Rádio-frequência	Leis físicas
Observações	Intervalos (atrasos de tempo)	Acelerações e variações angulares
Parâmetros de saída	Tempo-posição-velocidade	Posição-velocidade-atitude
Erros (comprimento de onda largo)	Baixo	Alto
Erros (comprimento de onda curto)	Alto	Baixo
Amostragem de frequência	1 Hz – 10 Hz	≥ 50 Hz
Operação	Depende de serviço externo	Autônomo
Disponibilidade	Visibilidade dos satélites	Todo tempo

3.6 Obtenção dos dados do GPS/INS

Os dados do GPS/INS são obtidos por meio de um registro de *log*, que neste trabalho foi definido para ser armazenado a uma taxa de 1 Hz. Esse *log* possibilita a obtenção da posição do INS, velocidade e atitude (ângulos de orientação). Na seção 5.2, mais especificamente na Figura 5.3 são ilustrados os equipamentos (aparelho GPS e Unidade de Medida Inercial (INS)) usados no Sistema Móvel de Mapeamento para registrar os dados obtidos em um levantamento. O arquivo de *log* usado neste trabalho é o INSPVAA (*INS Position, Velocity and Attitude*). A Tabela 2 apresenta a estrutura original do arquivo de *log* INSPVAA do GPS/INS obtida no manual do usuário OM-20000124 Rev 7 (SPAN-SE, 2011).

Tabela 2: Arquivo de *log* INSPVAA do GPS/INS.
Fonte: (SPAN-SE, 2011).

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84)	Double	8	H+12
5	Longitude	Longitude (WGS84)	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84)	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a negative value implies a southerly direction)	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a negative value implies a westerly direction)	Double	8	H+44
9	Up Velocity	Velocity in an up direction	Double	8	H+52
10	Roll	Right handed rotation from local level around Y-axis in degrees	Double	8	H+60
11	Pitch	Right handed rotation from local level around X-axis in degrees	Double	8	H+68
12	Azimuth	Left handed rotation around Z-axis Degrees clockwise from North	Double	8	H+76
13	Status	INS Status	Enum	4	H+84
14	Xxxx	32-bit CRC	Hex	5	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

A Figura 3.5 apresenta um exemplo de uma amostra (uma linha) do arquivo de *log* INSPVAA no formato ASCII. Apenas alguns campos são usados neste trabalho, são eles:

- (a): Seconds – segundos a partir do início da semana;
- (b): Latitude (WGS84);
- (c): Longitude (WGS84);
- (d): Height – Altura Elipsoidal (WGS84);
- (e): Roll – rotação de nível local ao redor do eixo y em graus (este ângulo é o φ das equações de colinearidade (seção 3.7));
- (f): Pitch – rotação de nível local ao redor do eixo x em graus (este ângulo é o ω das equações de colinearidade);
- (g): Azimuth – rotação ao redor do eixo z , κ graus à direita do Norte – sentido horário (esse ângulo κ é utilizado nas equações de colinearidade).

```

#INSPVAA,USB1,0,95.5,FINESTEERING,1645,(a)238542.600,00000000,
0000,187;1645,(b)238542.602500000,-22.135257511,(c)-51.404862379,
468.317551454,(d)-0.004616479,-0.006297944,0.000059920,
3.516160125,(e)1.458639033,(f)206.988606062,(g)
INS_ALIGNMENT_COMPLETE*b8ebf726

```

Figura 3.5: Exemplo de uma amostra do arquivo de *log* INSPVAA.

O sistema GPS/IMU fornece a posição (X,Y,Z) da antena do receptor e a orientação do sistema (ω,φ,κ) . Para realizar a interseção fotogramétrica e obter a posição (X,Y,Z) da placa reconhecida é necessário determinar a posição e orientação do centro perspectivo (CP) de cada uma das câmaras. Por construção, a geometria do sistema é mostrada na Figura 3.6.

Figura 3.6: Translação da posição da antena GPS para o centro perspectivo de cada imagem do estéreo par.

Fonte: (SILVA et al., 2003).

As duas coordenadas dos centros perspectivos de cada imagem do estéreo par (CP_1 , CP_2) são determinadas a partir da posição da antena GPS (P_1) e um ponto calculado P_2 . Este ponto é calculado em função da distância de P_1 ao centro do eixo dos centros perspectivos CP_1 e CP_2 (*stereobase*) com ângulo κ (*Azimuth*), extraído a partir do arquivo de *log*

INSPVAA do GPS/INS (Tabela 2 da seção 3.6). As equações a seguir são usadas para calcular CP_1 e CP_2 .

$$\begin{aligned}\alpha &= \tan^{-1}(\text{sterobase}/2/S_{P_1P_2}) \\ Az_{P_1P_2} &= \text{Azimuth} \\ Az_{P_1CP_1} &= Az_{P_1P_2} - \alpha \\ Az_{P_1CP_2} &= Az_{P_1P_2} + \alpha\end{aligned}\tag{26}$$

$$S_{P_1CP_1} = S_{P_1CP_2} = \sqrt{(\text{sterobase}/2)^2 + (S_{P_1P_2})^2}\tag{27}$$

$$\begin{bmatrix} X_{CP_1} \\ Y_{CP_1} \\ X_{CP_2} \\ Y_{CP_2} \end{bmatrix} = \begin{bmatrix} \sin(Az_{P_1CP_1}) & 0 \\ \cos(Az_{P_1CP_1}) & 0 \\ 0 & \sin(Az_{P_1CP_2}) \\ 0 & \cos(Az_{P_1CP_2}) \end{bmatrix} \begin{bmatrix} S_{P_1CP_1} \\ S_{P_1CP_2} \end{bmatrix} + \begin{bmatrix} X_{P_1} \\ Y_{P_1} \\ X_{P_1} \\ Y_{P_1} \end{bmatrix}\tag{28}$$

Dessa forma, a posição (X,Y) de cada câmara é determinada e a altura (Z) é a coordenada h obtida de maneira direta com o GPS (as duas câmaras estão na mesma altura).

3.7 Interseção fotogramétrica

Com o conhecimento da posição e orientação de cada uma das câmaras, é realizada a interseção fotogramétrica para se obter as coordenadas (X,Y,Z) da placa de sinalização detectada e reconhecida.

Com a posição do centro de massa do retângulo mínimo que contém a placa detectada e reconhecida, essa posição em pixels é convertida para fotocoordenadas em milímetros. Para isso, é necessária uma transformação geométrica linear do sistema de coordenadas da imagem (dadas em pixels) (x_p, y_p) para um sistema de fotocoordenadas (x_f, y_f) de uma imagem digital dado em milímetros (Figura 3.7).

Figura 3.7: Sistemas de fotocoordenadas (x_f, y_f) e da imagem (x_p, y_p) .

O sistema de fotocoordenadas de uma imagem digital tem o eixo x_f paralelo ao eixo x_p e o eixo y_f refletido em relação ao eixo y_p , cuja transformação é:

$$\begin{bmatrix} x_f \\ y_f \end{bmatrix} = \begin{bmatrix} t_{px} & 0 \\ 0 & t_{py} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_p - x_c - 0,5 \\ y_p - y_c - 0,5 \end{bmatrix} \quad (29)$$

onde t_{px} é o tamanho do pixel na horizontal e t_{py} é o tamanho do pixel na vertical, x_c e y_c o centro da imagem em coordenadas de pixel.

As equações de colinearidade usadas foram (WOLF, 1983):

$$\begin{cases} x_f = x_0 - f \frac{N_x}{D} \\ y_f = y_0 - f \frac{N_y}{D} \end{cases} \quad (30)$$

onde:

(x_f, y_f) : fotocoordenadas (mm)

(x_0, y_0) : coordenadas do ponto principal em relação ao centro da imagem (mm)

f : distância focal (mm)

$$N_x = r_{11}(X - X_c) + r_{12}(Y - Y_c) + r_{13}(Z - Z_c)$$

$$N_y = r_{21}(X - X_c) + r_{22}(Y - Y_c) + r_{23}(Z - Z_c)$$

$$D = r_{31}(X - X_c) + r_{32}(Y - Y_c) + r_{33}(Z - Z_c)$$

(X, Y, Z) : coordenadas de pontos no terreno (m)

(X_c, Y_c, Z_c) : coordenadas do centro perspectivo (m)

onde r_{ij} são os elementos da matriz de rotação R (Equação 31):

$$R = R_\omega R_\phi R_\kappa = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (31)$$

em relação aos ângulos ω , ϕ e κ :

$$R = \begin{bmatrix} \cos \phi \cos \kappa & \sin \omega \sin \phi \cos \kappa + \cos \omega \sin \kappa & -\cos \omega \sin \phi \cos \kappa + \sin \omega \sin \kappa \\ -\cos \phi \sin \kappa & -\sin \omega \sin \phi \sin \kappa + \cos \omega \cos \kappa & \cos \omega \sin \phi \sin \kappa + \sin \omega \cos \kappa \\ \sin \phi & -\sin \omega \cos \phi & \cos \omega \cos \phi \end{bmatrix} \quad (32)$$

A geometria da condição de colinearidade no caso terrestre com os ângulos de rotação para uma imagem é ilustrada na Figura 3.8.

Figura 3.8: Geometria da condição de colinearidade para o caso terrestre.

Fonte: Adaptado de Wolf (1983).

No caso de um par de imagens com sobreposição, a geometria da condição de colinearidade é ilustrada na Figura 3.9.

Figura 3.9: Geometria da condição de colinearidade para o caso terrestre com um par de imagens.
Fonte: (WOLF, 1983).

O fator de escala foi calculado como:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda_e R_e^{-1} \begin{bmatrix} x_e \\ y_e \\ -f_e \end{bmatrix} + \begin{bmatrix} X_e^C \\ Y_e^C \\ Z_e^C \end{bmatrix} \quad (33)$$

Para a câmara da direita:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda_d R_d^{-1} \begin{bmatrix} x_d \\ y_d \\ -f_d \end{bmatrix} + \begin{bmatrix} X_d^C \\ Y_d^C \\ Z_d^C \end{bmatrix} \quad (34)$$

Como a matriz de rotação é ortogonal, a inversa é igual à transposta, e as equações podem ser escritas, para as fotocoordenadas da esquerda:

$$X = \lambda_e (r_{11}^e x_e + r_{21}^e y_e - r_{31}^e f_e) + X_e^C = \lambda_e N_{x_e} + X_e^C \quad (35)$$

$$Y = \lambda_e (r_{12}^e x_e + r_{22}^e y_e - r_{32}^e f_e) + Y_e^C = \lambda_e N_{y_e} + Y_e^C \quad (36)$$

$$Z = \lambda_e (r_{13}^e x_e + r_{23}^e y_e - r_{33}^e f_e) + Z_e^C = \lambda_e D_e + Z_e^C \quad (37)$$

e para as fotocoordenadas da direita:

$$X = \lambda_d (r_{11}^d x_d + r_{21}^d y_d - r_{31}^d f_d) + X_d^C = \lambda_d N_{x_d} + X_d^C \quad (38)$$

$$Y = \lambda_d (r_{12}^d x_d + r_{22}^d y_d - r_{32}^d f_d) + Y_d^C = \lambda_d N_{y_d} + Y_d^C \quad (39)$$

$$Z = \lambda_d (r_{13}^d x_d + r_{23}^d y_d - r_{33}^d f_d) + Z_d^C = \lambda_d D_d + Z_d^C \quad (40)$$

Como o ponto no terreno é único as equações podem ser igualadas:

$$\lambda_e N_{x_e} + X_e^C = \lambda_d N_{x_d} + X_d^C \quad (41)$$

$$\lambda_e N_{y_e} + Y_e^C = \lambda_d N_{y_d} + Y_d^C \quad (42)$$

$$\lambda_e D_e + Z_e^C = \lambda_d D_d + Z_d^C \quad (43)$$

e para isolar o fator de escala, serão utilizadas apenas duas equações, pois em seguida, os valores aproximados pelo fator de escala são ajustados pelo método dos mínimos quadrados.

Fazendo:

$$\begin{aligned} \Delta X &= X_d^C - X_e^C = \lambda_e N_{x_e} - \lambda_d N_{x_d} \\ \Delta Y &= Y_d^C - Y_e^C = \lambda_e N_{y_e} - \lambda_d N_{y_d} \end{aligned} \quad (44)$$

$$\begin{aligned} \lambda_d &= \frac{\lambda_e N_{x_e} - \Delta X}{N_{x_d}} \\ \Delta Y &= \lambda_e N_{y_e} - \lambda_d N_{y_d} = \lambda_e N_{y_e} - \frac{\lambda_e N_{x_e} - \Delta X}{N_{x_d}} N_{y_d} = \\ &\lambda_e N_{y_e} N_{x_d} - \lambda_e N_{x_e} N_{y_d} + \Delta X N_{y_d} = \Delta Y N_{x_d} \\ \lambda_e &= \frac{\Delta Y N_{x_d} - \Delta X N_{y_d}}{N_{y_e} N_{x_d} - N_{x_e} N_{y_d}} \end{aligned} \quad (45)$$

O fator de escala da direita pode ser obtido de maneira análoga:

$$\lambda_d = \frac{\Delta Y N_{x_e} - \Delta X N_{y_e}}{N_{y_d} N_{x_e} - N_{x_d} N_{y_e}} \quad (46)$$

As coordenadas do terreno são aproximadas inicialmente pela média entre o cálculo das Equações (35 e 38), equações (36 e 39) e equações (37 e 40). Com essa aproximação inicial para as coordenadas (X, Y, Z) da placa é feito um ajustamento com o método

paramétrico pelo critério dos Mínimos Quadrados. O modelo matemático para o ajustamento utiliza as equações de colinearidade (Equação 47) que relacionam as fotocoordenadas de um ponto na placa, os parâmetros de orientação exterior (posição do CP e ângulos eulerianos) das imagens, parâmetros de orientação interior (x_0, y_0, f, k_1) de cada imagem e coordenadas (X, Y, Z) do ponto no terreno:

$$L_a = F(X_a) \quad (47)$$

onde:

L_a : vetor das observações ajustadas

X_a : vetor dos parâmetros ajustados

com:

$$X_a = X_0 + X \quad (48)$$

$$L_a = L_b + V \quad (49)$$

onde:

X_0 : vetor dos parâmetros aproximados inicialmente pelos fatores de escala da direita e da esquerda

X : vetor correção dos parâmetros

$X_a = X_0$: vetor das fotocoordenadas ajustadas

L_b : vetor das fotocoordenadas observadas

V : vetor dos resíduos das fotocoordenadas

Fazendo a linearização pela fórmula de Taylor, tem-se

$$L_b + V = F(X_0 + X) = F(X_0) + \left. \frac{\partial F}{\partial X_a} \right|_{X_a=X_0} X \quad (50)$$

Tomando a função dos parâmetros aproximados por:

$$L_0 = F(X_0) \quad (51)$$

tem-se:

$$L_b + V = L_0 + AX \quad (52)$$

com:

$$A = \left. \frac{\partial F}{\partial X_a} \right|_{X_a = X_0} \quad (53)$$

e

$$V = AX + L_0 - L_b = AX + L \quad (54)$$

com $L = L_0 - L_b$, esse é o modelo matemático linearizado que é minimizado para se encontrar a solução única, que é dada por:

$$X = -(A^t A)^{-1} A^t L \quad (55)$$

Como o modelo matemático não é linear, a solução da Equação 55 requer iterações. A matriz das derivadas parciais para esse modelo matemático é dada por:

$$A = \begin{bmatrix} -\frac{f_e}{D_e^2}(r_{11}^e D_e - r_{31}^e N_{x_e}) & -\frac{f_e}{D_e^2}(r_{12}^e D_e - r_{32}^e N_{x_e}) & -\frac{f_e}{D_e^2}(r_{13}^e D_e - r_{33}^e N_{x_e}) \\ -\frac{f_e}{D_e^2}(r_{21}^e D_e - r_{31}^e N_{y_e}) & -\frac{f_e}{D_e^2}(r_{22}^e D_e - r_{32}^e N_{y_e}) & -\frac{f_e}{D_e^2}(r_{23}^e D_e - r_{33}^e N_{y_e}) \\ -\frac{f_d}{D_d^2}(r_{11}^d D_d - r_{31}^d N_{x_d}) & -\frac{f_d}{D_d^2}(r_{12}^d D_d - r_{32}^d N_{x_d}) & -\frac{f_d}{D_d^2}(r_{13}^d D_d - r_{33}^d N_{x_d}) \\ -\frac{f_d}{D_d^2}(r_{21}^d D_d - r_{31}^d N_{y_d}) & -\frac{f_d}{D_d^2}(r_{22}^d D_d - r_{32}^d N_{y_d}) & -\frac{f_d}{D_d^2}(r_{23}^d D_d - r_{33}^d N_{y_d}) \end{bmatrix} \quad (56)$$

O resultado do ajustamento é a posição da placa que pode ser visualizada na base cartográfica utilizando as coordenadas E (*East*), N (*North*). Para que a interseção fotogramétrica possa ser realizada é necessário conhecer os parâmetros de orientação interior da câmara digital utilizada na tomada das imagens (ANDRADE, 1998). Os parâmetros utilizados neste trabalho são as coordenadas do ponto principal (x_0, y_0) , a distância focal calibrada (f) e o parâmetro da distorção k_1 . Como foi utilizada uma câmara não métrica, o processo de calibração utilizado foi a calibração *on the job*⁶ realizado no Campo de Calibração da FCT/Unesp (GALO, 2011). A Figura 3.10 mostra o SIMMOV (Sistema Móvel de Mapeamento) no Campo de Calibração para a obtenção dos dados usados no processo de calibração das câmaras.

⁶ Calibração *on-the-job*: é o processo de determinar os parâmetros do sistema de lentes e fazer a calibração nas mesmas situações ou condições do trabalho real (ATKINSON, 1996).

Figura 3.10: SIMMOV no Campo de Calibração.

A Figura 3.11 mostra o Campo de Calibração da FCT/UNESP, com os pontos usados no processo de calibração.

Figura 3.11: Pontos do Campo de Calibração de Câmaras da FCT/Unesp.
Fonte: (GALO, 2011).

A calibração das câmaras foi realizada com a posição do CP (centros perspectivos) das câmaras determinada com o sistema GPS/INS e a Equação 28.

Os valores iniciais para as coordenadas do ponto principal (x_0, y_0) foi (0,0), a focal de 20 mm conforme o manual da câmara e o parâmetro de distorção radial k_1 arbitrado como zero. O método convergiu na segunda interação e os resultados estão na Figura 3.12.

Foto	Omega	Phi	Kappa	Xc	YC	ZC
Esq	0,02005258	-0,03102346	4,02565376	457979,48000178	7554028,73395846	415,04828980
Dir	0,01950173	-0,03310251	4,02570076	457978,88391447	7554029,46129423	415,04828980
Foto	x ₀	y ₀	f	k1		
Esq	0,16363214	-0,11483896	19,98941042	0,03600066		
Dir	0,16809576	-0,12458146	19,98886032	0,03758206		
Orientação relativa:						
Omega : -0,00055085						
Phi : -0,00207905						
Kappa : 0,00004700						
Xc : -0,59608731						
Yc : 0,72733577						
Zc : 0,00000000						
Ajustamento da base:						
Base : 0,94039216						

Figura 3.12: Resultados do processo de calibração das câmaras.

Esses valores ajustados dos parâmetros de orientação exterior foram utilizados para a determinação das coordenadas no terreno das placas detectadas e reconhecidas.

Capítulo 4

Trabalhos Relacionados

4.1 Considerações Iniciais

Neste capítulo, resumem-se os resultados das investigações realizadas para obter conhecimento de trabalhos relacionados ao apresentado nesta tese, envolvendo inclusive abordagens diferentes. Esta busca objetivou a aquisição de conhecimento das técnicas, algoritmos e metodologias para a detecção, classificação e reconhecimento de placas de sinalização de trânsito em imagens digitais e sequências de vídeo, com processamento *off-line* ou em tempo real. Com base na investigação realizada, são apresentados neste capítulo, trabalhos que utilizam o algoritmo SIFT para realizar a detecção das placas de sinalização, que estão mais próximos da metodologia apresentada neste trabalho de doutorado, mas são também apresentados trabalhos recentes que utilizam técnicas para detecção, classificação e reconhecimento baseadas em cor, formas, detecção de bordas, operadores morfológicos, métodos de aprendizagem SVM e algoritmos construídos com técnicas de Inteligência Artificial, como redes neurais e lógica *fuzzy*.

4.2 Trabalhos Relacionados com relação à detecção, classificação e reconhecimento de placas de sinalização de trânsito

Devido à crescente demanda por sistemas inteligentes de auxílio a usuários, reconhecimento de sinais de trânsito tornou-se um tema de grande interesse. Com a possibilidade da detecção e do reconhecimento de sinais, muitas aplicações tornam-se viáveis, como por exemplo, sistemas que fornecem informações para motoristas, sistemas para pessoas cegas auxiliando-as na navegação nas ruas, ou até mesmo sistemas de orientação

turística que faz uso de informações dos sinais das placas. Detectar e principalmente reconhecer sinais de trânsito é uma tarefa desafiadora em Visão Computacional devido às variações de captura que as imagens podem sofrer. Essas variações podem ser em relação a diferentes pontos de vista, diferentes condições de iluminação e variações climáticas do tempo (ensolarado, nublado, nebuloso, chuvoso etc.). A oclusão parcial dos sinais de tráfegos causada por árvores ou por algum objeto, por exemplo, pode tornar ainda mais dificultoso a detecção e reconhecimento de sinais nas imagens (NGUYEN; ANDERSEN, 2008).

4.2.1 Abordagem baseada no algoritmo SIFT

Nesta subseção são apresentados alguns trabalhos que utilizam o algoritmo SIFT para a detecção das placas de sinalização, algoritmo este empregado na metodologia apresentada neste trabalho de doutorado para a extração dos descritores dos pontos chave.

No trabalho de Farag e Abdel-Hakim (2004), o enfoque de detecção e reconhecimento de sinal de trânsito (*Road Sign Detection and Recognition – RSR*), utiliza um classificador de *Bayes* para detectar os sinais de trânsito na imagem capturada, baseado em seu conteúdo de cor. O classificador de *Bayes* não apenas rotula a imagem capturada, mas também classifica os rótulos para a categoria apropriada para os sinais de trânsito. A partir dos resultados obtidos pelo classificador de *Bayes*, o algoritmo SIFT (LOWE, 2004) é empregado para extrair um conjunto de descritores invariantes para os sinais de trânsito detectados. Usando o SIFT para o processo de correspondência, alcançam-se algumas vantagens em relação a trabalhos prévios, como por exemplo, melhoria no processo de correspondência com o *template*, não havendo a necessidade de se ter um número grande de imagens reais de sinais (*templates*) para treinar o sistema, como acontece com trabalhos com abordagens baseadas em redes neurais.

Segundo os autores, a vantagem de se usar o classificador de *Bayes* é a aceleração de extração de descritores e operações de correspondência do algoritmo SIFT, pois reduz a área para realizar a correspondência na imagem. Também, limita o subespaço de procura do SIFT pela determinação da categoria de cor do sinal detectado. Segundo os autores, o enfoque proposto é para ser empregado em um projeto integrado de navegação de veículo inteligente.

O Algoritmo 3 a seguir, relaciona com mais detalhes os passos do enfoque de detecção e reconhecimento de sinal de trânsito utilizado no trabalho.

Algoritmo 3 – Abordagem RSR proposta por Farag e Abdel-Hakim (2004)

- 1: Extrair os descritores SIFT dos *templates* dos sinais de trânsito (*off-line*).
 - 2: Adquirir as imagens das placas dos sinais de trânsito.
 - 3: Rotular cada imagem adquirida usando o classificador de *Bayes* com relação ao conteúdo de cor.
 - 4: Extrair os descritores SIFT para cada rótulo.
 - 5: Para cada rótulo, realizar a correspondência de seus descritores com os descritores dos sinais padrão na mesma categoria de cor.
 - 6: Para cada rótulo, encontrar o sinal padrão com o máximo valor objetivo ($f_{i_{\max}}$).
 - 7: Para cada rótulo, se ($f_{i_{\max}}$) é inferior a um determinado limiar, então não há nenhum sinal detectado para o rótulo.
 - 8: Caso contrário, o sinal padrão com ($f_{i_{\max}}$) é detectado na imagem adquirida, para aquele rótulo.
-

No trabalho de Kus et al. (2008) foi proposta uma técnica de detecção e reconhecimento de sinal de trânsito adicionando ao algoritmo SIFT novos descritores relacionados à cor de regiões locais. Esses descritores são utilizados para verificar a precisão das correspondências que são encontradas pelo SIFT. O método de classificação de cor proposto utiliza algumas regras de classificação e realiza a inspeção de cor avaliando os valores de um pixel no espaço de cor HSV e o coloca em uma classe de cor de acordo com as regras que foram desenvolvidas. Segundo os autores, a técnica proposta funciona para as imagens que contêm sinais de trânsito rotacionados, assim como também para as imagens que sofreram transformações, distorções, foram danificadas, oclusas, ofuscadas, apresentam alteração na cor, obtidas em diferentes condições climáticas e diferentes condições de iluminação. Os autores realizaram três experimentos com 150 imagens de teste, sendo que no primeiro experimento, usaram um software de demonstração provido por David Lowe (2004) e obtiveram 74% de precisão no reconhecimento. No segundo experimento, os autores adicionaram informação de cor, e com isso aumentaram a precisão para 84,67%, enquanto que no terceiro experimento, os autores também adicionaram informações de orientação dos sinais, conseguindo com isso aumentar ainda mais a precisão para 99,33%.

No trabalho proposto por Höferlin e Zimmermann (2009) é apresentada uma arquitetura robusta de sistema para o reconhecimento de sinais de trânsito circulares baseada nas três fases comuns de sistemas recentes de TSR (*Traffic Sign Recognition*) que são:

detecção, classificação e localização. Esse sistema emprega abordagens complementares para as diferentes fases de sistemas TSR atuais. Isto introduz a aplicação de descritores SIFT locais (LOWE, 2004) para detecção de sinal de trânsito baseado em conteúdo, juntamente com abordagens baseadas em formas, amplamente aplicadas. Os autores adicionaram uma quarta fase, denominada de fase de refinamento, onde, segundo eles, foi possível melhorar a localização dos candidatos de sinal de trânsito detectados, usando uma versão rápida do algoritmo de CCD (*Contracting Curve Density*). Adicionalmente, os autores introduziram um módulo de detecção baseado no SIFT e um módulo de reconhecimento no fundamento dos descritores SIFT e SURF. Para a fase de reconhecimento os autores utilizaram uma rede neural MLP (*Multi-Layer Perceptron*) para distinguir os sinais entre várias classes de sinais de trânsito. A camada de entrada da rede neural consiste de 128 entradas, refletindo a dimensionalidade do vetor de descritores dos pontos chave extraídos com o algoritmo SIFT. O sistema foi avaliado usando uma sequência de testes de 30 minutos de duração com 94 sinais que são relevantes aos motoristas. Todos esses sinais foram extraídos pelo módulo de detecção, dos quais 96,4% foram identificados corretamente.

O trabalho proposto por Reiterer et al. (2009) apresenta um método para reconhecimento automático dos sinais de trânsito a partir de sequências de imagens obtidas por um sistema móvel de mapeamento (VISATTM como foi denominado). A técnica apresentada é baseada em algoritmos existentes, como o algoritmo SIFT e o algoritmo de agrupamento *k-means*. O procedimento desenvolvido utiliza uma rotina de correspondência convencional combinada com um banco de dados para detectar sinais de trânsito em sequências de imagens. Foi dividido nas seguintes etapas (Figura 4.1): construção do banco de dados de sinais de trânsito (composto de modelos/*templates* de sinais de trânsito); extração dos descritores SIFT das imagens dos modelos do banco de dados; extração dos descritores SIFT das imagens obtidas pelo sistema móvel de mapeamento; correspondência entre os descritores das imagens com todos os descritores no banco de dados; agrupamento das combinações resultantes para se obter os sinais de trânsito separados (quando existem mais de um sinal de trânsito em uma mesma imagem); e cálculo das coordenadas 3D dos sinais de trânsito.

Figura 4.1: Procedimento de detecção.
Fonte: (REITERER et al., 2009)

Os componentes de hardware do sistema móvel de mapeamento VISATTM são os seguintes: um sistema de navegação inercial (SINS – *Strap Down Inertial Navigation System*), um receptor GNSS-*navigation system* de dupla frequência, e um conjunto de câmaras digitais coloridas. O sistema integra também um instrumento de medição de distâncias (DMI – *Distance Measuring Instrument*), usado para disparar a aquisição das imagens das câmaras em intervalos de distâncias constantes. O conjunto de câmaras consiste em dois compartimentos de câmaras, um ao lado do outro (cada compartimento de câmaras é o espelho exato do outro e formam um campo de visão de 330°). O sistema desenvolvido é baseado no uso de duas câmaras com a face voltada para frente, que é a posição mais comum para capturar a maioria dos sinais de trânsito. Para encontrar os sinais de trânsito nas imagens obtidas, foi realizado o processo de correspondência dos descritores das imagens baseado no processo proposto por Lowe (2004), que objetiva encontrar uma correspondência entre os descritores de cada imagem e os descritores dos *templates* pela avaliação da distância Euclidiana. De acordo com o procedimento do vizinho mais próximo, a correspondência de cada descritor F_1^i de uma imagem é procurada em cada modelo de descritores F_2^j no banco de dados. Quando a distância euclidiana entre os descritores F_1^i e F_2^j está abaixo de um limiar, o conjunto $M(F_1^i, F_2^j)$ é rotulado como positivo, formando uma lista de pontos classificados como pontos

dentro de um sinal de trânsito. É realizado o agrupamento dos resultados das correspondências para obter os sinais de trânsito separadamente quando existem mais de um sinal em uma mesma imagem, e para isto é utilizado o algoritmo *k-means* (HARTIGAN; WONG, 1979) (ideal para agrupar objetos baseado em atributos). O agrupamento é processado para minimizar a soma dos quadrados das distâncias entre os dados e o centróide do grupo correspondente. Toda a sequência da técnica foi desenvolvida em um protótipo utilizando o MATLAB. Os experimentos foram realizados utilizando um grande número de exemplos de imagens de qualidades diferentes (aproximadamente 400 pares de imagens), e segundo os autores, o sistema apresentou uma taxa de detecção de aproximadamente 74%. Em condições ideais de iluminação (por exemplo, iluminação constante) as taxas de detecção atingiram mais de 80%.

O trabalho de Ihara et al. (2009) apresenta um método de reconhecimento de sinais de trânsito baseado na classificação de pontos chave por AdaBoost usando descritores PCA-SIFT (*Principal Component Analysis Scale Invariant Feature Transform*). Os autores propuseram uma técnica para melhorar a velocidade de comparação e precisão usando descritores PCA-SIFT, projetados em diferentes subespaços de descritores, e AdaBoost é usado para classificar se eles pertencem ou não ao sinal de trânsito. O descritor PCA-SIFT (KE; SUKTHANKAR, 2004) descreve os gradientes horizontais e verticais de um *patch* 41 x 41 (uma região com uma determinada escala e rotação para alinhar sua orientação dominante a uma direção canônica), centrado nos pontos chave detectados pelo SIFT. O descritor PCA-SIFT tem um vetor de descritores com $2 \times (41 - 2)^2 = 3042$ elementos. O fluxo do método é descrito no Algoritmo 4 de forma resumida.

Algoritmo 4 – Método de reconhecimento de sinais de trânsito baseado na classificação de pontos chave por AdaBoost usando descritores PCA-SIFT

Passos:

- 1: Detectar pontos chave em imagem de entrada usando SIFT.
 - 2: Extrair descritores de gradiente horizontais e verticais centradas nos pontos chave.
 - 3: Projetar os gradientes horizontais e verticais em diferentes subespaços de descritores.
 - 4: Combinar os descritores projetados.
 - 5: Discriminar os pontos chave usando o classificador AdaBoost Real a partir do aprendizado dos descritores combinados.
 - 6: Realizar a correspondência somente dos pontos chave que foram discriminados como pontos sobre o sinal de trânsito.
-

Os resultados experimentais, segundo os autores, mostram que o custo computacional para realizar a correspondência de pontos chave pode ser reduzido para cerca da metade do tempo se comparado com o método convencional.

O trabalho proposto por Cai et al. (2010) apresenta um método de reconhecimento para sinais de trânsito triangulares baseado nos descritores SIFT. Os descritores SIFT são primeiramente extraídos dos sinais de trânsito para então ser realizada a correspondência usando a distância Euclidiana. Por fim, uma árvore de decisão é adotada para classificar os sinais de trânsito. Para o experimento realizado foram aplicadas mudanças de escala e de rotação para 45 sinais de trânsito de advertência, sendo dezesseis imagens para cada sinal de trânsito, constituindo um total de 720 imagens utilizadas para o conjunto de treinamento. Segundo os autores a precisão de reconhecimento para o método proposto é de 96,81%. Ainda segundo eles, os resultados experimentais indicam que o método proposto pode resolver os problemas da transformação de escala, rotação e distorção e tem excelente capacidade para tratar imagens com ruído e imagens que apresentam oclusão do objeto de interesse.

No trabalho de Höferlin e Heidemann (2010) é apresentada uma discussão de como características locais discriminativas são aplicadas a TSR (*Traffic Sign Recognition*) e quais problemas surgem neste domínio específico. Desde que TSR tem que lidar com objetos altamente estruturados e simétricos, que são capturados frequentemente em baixa resolução, apenas um número pequeno de descritores podem ser correspondentes. Para minimizar estes problemas, os autores proveram uma abordagem para a seleção de características discriminativas e robustas a fim de aumentar o desempenho e a precisão da correspondência. Adicionalmente os autores abordam a questão de sensibilidade de descritores e estabilidade posicional em mudanças de escala, bem como sua confiabilidade às variações de ponto de vista.

Os autores focaram seu trabalho especialmente nos descritores locais invariantes à escala e discriminativos como o SIFT (*Scale Invariant Feature Transform*) (LOWE, 2004). Tais descritores locais são comumente calculados em dois estágios. Primeiramente, um detector de pontos de interesse é usado para encontrar regiões da imagem salientes às suas características de escala; posteriormente um descritor robusto é extraído para cada região.

Usando descritores locais, duas imagens podem ser comparadas de forma muito rápida, já que apenas algumas regiões salientes são consideradas, ao invés das imagens inteiras. Além disto, descritores locais como os extraídos com o uso do SIFT são mais

robustos para várias transformações de imagem (mudanças de escala, rotação e iluminação). Em contraste com métodos globais, descritores locais podem inerentemente lidar com oclusões parciais na imagem. Descritores locais podem ser aplicados a todos os três estágios do TSR: detecção do sinal de trânsito candidato, classificação do candidato para a classe de sinal de trânsito (isto é, para o tipo de sinal de trânsito, como por exemplo, limite de velocidade) e rastrear o candidato em uma sequência de vídeo.

Os autores apresentaram um método baseado no algoritmo SIFT para selecionar um conjunto pequeno de descritores locais discriminativos com excelentes propriedades de correspondência de descritores de duas imagens. Para se chegar à metodologia adotada, os autores realizaram estudos avaliando abordagens dos seguintes descritores locais: SIFT (LOWE, 2004), GLOH (*Gradient Location and Orientation Histogram*) (MIKOLAJCZYK; SCHMID, 2005) e algumas variantes de SURF (BAY et al., 2006). Segundo os autores, na avaliação de sua abordagem eles se restringiram a utilizar o SIFT, uma vez que este método, comparado com os algoritmos testados, foi identificado como sendo o mais adequado para TSR. Os autores tomaram esta decisão baseado na avaliação realizada por eles e também na avaliação de descritores de característica provida por Mikolajczyk e Schmid (2005), na qual o SIFT é apontado como sendo o mais forte descritor local em termos de sensibilidade e precisão em relação às abordagens testadas. Os autores utilizaram dois conjuntos de teste para avaliação do método proposto, sendo o primeiro contendo 12 imagens com resolução de 640x480 pixels apresentando nove classes de sinais, enquanto que o segundo conjunto estava composto por 30 imagens com resolução de 360x270 pixels de três classes de sinais.

4.2.2 Abordagem baseada em propriedades de cor

Na literatura, podem ser encontradas duas abordagens principais para resolver o problema do reconhecimento de sinal de trânsito: 1) baseado em cor e 2) baseada em escala de cinza. Reconhecimento de sinal baseado em cores faz uso da cor para reduzir os resultados falso-positivos no processo de reconhecimento (HSU; HUANG, 2001), ao passo que o método de escala de cinza concentra-se na geometria do objeto (LOY; ZELINSKY, 2003). Trabalhos recentes usam a combinação de ambas as abordagens para melhorar as taxas de detecção. Por exemplo, em Maldonado-Bascon et al. (2007), um limiar (*threshold*) é aplicado

sobre a matiz (*hue*), saturação e representação do valor da imagem para encontrar regiões com alta probabilidade de ter um sinal de trânsito.

No trabalho de Nguyen e Andersen (2008) foi desenvolvida uma técnica que detecta a localização dos sinais de trânsito, considerando sua aparência em cores. É utilizado um domínio de localização de sinais de trânsito abrangendo os sinais de instruções, advertência, informações ou alerta às pessoas, incluindo não apenas os sinais de trânsito, mas também sinais de nomes de rua, descrição textual em sinais de trânsito etc. Segundo os autores, a segmentação usando cor é uma abordagem mais adequada para detectar regiões candidatas que podem conter sinais, enquanto que as abordagens apenas baseadas em formas não produzem soluções satisfatórias. Os sinais geralmente podem ser distintamente notados pela sua cor, possuem características comuns, contém fundo de cores conhecidas e contêm textos ou figuras com diferentes cores.

Foi realizada uma segmentação usando detecção de borda cromática, na qual o brilho indica a intensidade de borda. A maioria dos sinais contém uma borda com cor diferente da cor do fundo, por exemplo, alguns sinais de trânsito de alerta e sinais de proibição têm uma borda de cor vermelha, com fundo branco e com textos ou figuras de cor preta. A partir dessa propriedade de distribuição de sinais de cor, os autores propuseram uma abordagem para localizar as regiões dos sinais com êxito, usando um modelo invariante a cor para a segmentação baseado no trabalho de Geusbroek et al. (2001). Para realizar os experimentos, os autores usaram dois diferentes conjuntos de imagens, sendo o primeiro constituído de 48 imagens de sinais de trânsito capturados em cenas complexas contendo dois diferentes sinais, e o segundo constituído de 60 imagens. Com o primeiro conjunto de imagens, os autores conseguiram 60% de precisão no reconhecimento dos sinais de trânsito, e com o segundo conjunto, eles conseguiram 71%.

No trabalho de Creusen et al. (2010), é realizada a detecção de sinal de trânsito em imagens panorâmicas obtidas a partir de um veículo, onde são utilizadas duas câmaras fotográficas com lentes de olho de peixe. O núcleo do processo é baseado no algoritmo *Histogram of Oriented Gradients* (HOG) que é estendido incorporando informação de cor no vetor de descritores. O algoritmo HOG foi originalmente criado por Dalal e Triggs (2005) e aplica uma descrição fortemente baseada no algoritmo SIFT (LOWE, 2004). A escolha do espaço de cor tem uma grande influência no desempenho, onde segundo os autores, os espaços de cores CIELab e YCbCr apresentam os melhores resultados e o uso de cores

melhora significativamente o desempenho de detecção. É usada uma SVM linear (*Support Vector Machine*) para melhorar a eficiência computacional, pelo fato dos autores utilizarem um amplo banco de dados constituído de 5000 imagens com resolução de 4800x2400 pixels cada. Os autores avaliaram a detecção de três diferentes classes de sinais de trânsito: circular azul, circular vermelho e triangular. O sistema foi treinado, respectivamente, com 170, 74 e 53 amostras de cada classe. Os experimentos foram realizados usando os seguintes espaços de cores: RGB, HSV, CIELab e YCbCr. Os melhores resultados na detecção dos sinais de trânsito, segundo os autores, para cada classe foram: 56% para sinais da classe circular vermelho usando o espaço de cor CIELab, 85% para sinais da classe circular azul usando o espaço de cor CIELab e 74,6% para sinais da classe triangular usando o espaço de cor YCbCr.

4.2.3 Abordagem baseada em propriedades de forma

Em Barnes e Zelinsky (2004), uma plataforma baseada em visão veicular é usada para detectar sinais de trânsito, onde o objetivo principal está nos sinais circulares de limites de velocidade. Os autores utilizaram um detector rápido de simetria radial adaptado para trabalhar com frames de imagem de uma câmara de vídeo, onde a classificação dos sinais foi realizada por uma correlação cruzada normalizada, ideal para sinais com formas circulares. Para demonstrar que o operador de simetria radial é capaz de ser usado eficientemente com a correlação cruzada padrão, os autores realizaram em seus experimentos a classificação de sinais com limites de velocidade 40 e 60 em imagens com resolução de 320x240 pixels, sendo classificados 96% dos 126 candidatos válidos. Já para os sinais com limites de velocidade 25 e 40, apenas 75% foram classificados corretamente.

No trabalho de Baró e Vitria (2004) é descrito um *framework*⁷ para detecção de sinais de trânsito para imagens em tons de cinza. Os sinais de trânsito são detectados usando o método proposto por Viola e Jones (2004) adaptado para detecção de sinais de trânsito ao invés de face, como o original. Esse método para detecção de objetos é constituído por classificadores formados por uma cascata de características de retângulos seguida por filtros específicos para cada forma. Foram definidos dois tipos de filtros: o primeiro é baseado em

⁷ Um *framework* é uma coleção de vários componentes totalmente ou parcialmente implementados com padrões de cooperação em grande parte pré-definidos entre eles (FAYAD; SCHMIDT, 1997).

mudanças locais da direção de gradiente e o segundo é baseado na ideia de simetria radial e encontra o centro de formas circulares. Os filtros usam a posição e a razão de aspecto das características descobertas para diferenciar entre objetos reais e falsos positivos. Por fim, é utilizada a análise de componentes principais (PCA) para eliminar os falsos positivos restantes. Os autores usaram um conjunto formado por três classes para treinamento, onde cada classe tem uma cascata de características de retângulos, e uma PCA treinada especificamente para cada grupo. As cascatas de características de retângulos são treinadas com 7.000 exemplos positivos e 1.200 exemplos negativos, já a PCA é treinada usando 5.000 exemplos positivos.

O algoritmo de reconhecimento, proposto no trabalho de Paulo e Correia (2008), analisa os contornos exteriores de pictogramas (símbolo ou desenho da representação do sinal de trânsito). O enfoque proposto visa reconhecer um sinal de trânsito analisando a curvatura da representação do sinal. Um primeiro estágio é realizado para segmentar o pictograma contido no sinal. Um pictograma deve aparecer como uma ou como várias regiões desconectadas. Em outro estágio é feita a representação do contorno e correspondência, e uma curvatura de representação do espaço escala (*Curvature Scale Space* – CSS) (MOKHTARIAN, 1995) é selecionada para representar o contorno do pictograma. O reconhecimento de Pictograma é feito pela combinação das representações CSS.

Quando o pictograma do sinal estiver composto por várias regiões desconectadas, os seus contornos exteriores não podem ser transformados diretamente em uma representação de CSS. Nesse caso, um contorno exterior que inclui a composição de todas as regiões é encontrado usando contornos ativos (*Snakes*) que, iniciam a partir da borda do pictograma, deformando até contornar aos limites exteriores do pictograma.

O algoritmo de reconhecimento explora o contorno exterior do pictograma de sinal, usando uma representação de CSS que é robusto à escala e a mudanças de orientação que podem ser observadas em imagens de sinais reais. Segundo os autores, são alcançados bons resultados de reconhecimento quando o pictograma é capturado com detalhes suficientes. Nos experimentos, os autores utilizaram um conjunto de 579 imagens de sinais de trânsito, contendo 218 imagens com baixa luminosidade, 344 com boa luminosidade e 17 com excelente qualidade de iluminação. Segundo os autores, dependendo da classe de sinal (perigo, proibição, obrigação, informação etc) utilizado, os resultados variam, podendo alcançar uma taxa de reconhecimento média de 80%.

No trabalho de Md-Sallah et al. (2010), foi proposto um algoritmo de detecção e reconhecimento de sinal de trânsito para uma aplicação em sistemas embarcados. O algoritmo proposto foi implementado e testado em MATLAB. A fase de detecção do sinal de trânsito do algoritmo implementado baseou-se no método da Transformada de *Hough* (GONZALEZ; WOODS, 2002) para detectar linhas, a fim de identificar e determinar a forma do sinal de trânsito. O número de cúes e linhas indica as formas do sinal de trânsito, como triângulos, losangos, retângulos, quadrados, círculos e hexágonos. Medições de forma são empregadas para identificar as relações de área e perímetro do sinal de trânsito. As variáveis representando as medidas de forma são comparadas com uma biblioteca de *templates*, desenvolvida pelos autores. A relação de área e perímetro é determinada para reconhecer sinais de trânsito tais como, um sinal de cruzamento, um sinal de parada, entre outros. Para avaliar a precisão, o algoritmo proposto foi testado na maioria dos sinais de trânsito da Malásia, e segundo os autores a maioria dos sinais de trânsito pode ser corretamente detectada e reconhecida pelo método proposto, com uma taxa de detecção e reconhecimento de 83,6%.

4.2.4 Abordagem baseada em propriedades de cor e forma

Em Senekal (2008) é apresentada uma técnica baseada na análise de cor e forma de uma imagem para a detecção e classificação de sinais de trânsito triangulares, como advertência e prioridade. O algoritmo desenvolvido oferece capacidade de reconhecimento robusto, sob condições normais da luz do dia, na ausência de oclusões. Os sinais de trânsito desejados para reconhecimento têm uma moldura triangular vermelha que normalmente envolve uma representação de um objeto escuro em um fundo branco. A técnica é aplicada a um conjunto de imagens obtidas a partir de uma sequência de vídeo, usando uma câmara montada em um veículo em movimento. Cada imagem é convertida do espaço de cores RGB para o espaço de cores HSV, ao qual um limiar inicial é aplicado para determinar regiões vermelhas (possíveis sinais de trânsito). Em seguida, os interiores e exteriores (fundo) de possíveis sinais são marcados. As bordas são então extraídas da imagem, onde nos interiores estão os possíveis sinais. Um algoritmo é aplicado para determinar segmentos de extremidade, e pequenos segmentos de borda que são prováveis de ser ruído são descartados. São testados segmentos de bordas separados para determinar se eles provêm um bom ajuste para um triângulo. Se tal ajuste for estabelecido, os três vértices do triângulo são anotados. Usando

esses vértices, uma interpolação com base em coordenadas baricêntricas é aplicada para mapear o triângulo na imagem original sobre um novo triângulo normalizado, com escala e rotação fixas. Esta normalização visa também reduzir o efeito de distorção de perspectiva. A classificação é obtida fazendo a correspondência desses triângulos normalizados com um conjunto de modelos de referência (*templates*). Foram usados 87 *templates* de sinais de alerta e dois *templates* de sinais *yield* (sinais que significam que o motorista deve diminuir o ritmo e estar pronto para parar, se necessário, para dar passagem a qualquer veículo, ciclista ou pedestre antes de prosseguir). O algoritmo foi testado em imagens com resolução de 640x480 pixels extraídas a partir de sequências de vídeo, e segundo o autor, baseado nos resultados dos experimentos, apresentou precisão no reconhecimento acima de 85% nos testes realizados.

Arlicot et al. (2009) propuseram um algoritmo que consiste em três passos principais. O primeiro passo utiliza propriedades de cor de sinais e executa uma pré-deteção usando cores, que provê um conjunto de ROIs (Regiões de Interesse) na imagem. Em seguida, um algoritmo de detecção de elipse é aplicado para detectar sinais de formato circular dentro dos ROIs. As formas detectadas são consideradas como hipóteses de sinal de trânsito. O passo final consiste em validação ou rejeição das hipóteses. Isto é feito comparando as hipóteses detectadas com um conjunto de sinais circulares padrão da mesma cor. Segundo os autores, o espaço de cor HSV (*Hue*, *Saturation*, *Value*) foi utilizado no trabalho por ser robusto às condições de variação de luminosidade. Os parâmetros do limiar adaptado de saturação (*Saturation*) e matiz (*Hue*) foram escolhidos a partir de um conjunto de amostras de sinais de trânsito em diferentes condições de iluminação. Os autores avaliaram o algoritmo proposto usando um conjunto de 1370 imagens adquiridas em uma área urbana com condições de tráfego real. Os resultados experimentais apresentados pelos autores mostraram uma precisão de reconhecimento de 67%

Le et al. (2010) apresentam uma nova abordagem para recuperar regiões candidatas de sinais de trânsito em sequências de imagens, baseada na detecção de cores e na segmentação por SVM (*Support Vector Machine*). Ao invés de processar cada pixel, esta abordagem utiliza um bloco de pixels como um vetor de entrada de SVM para classificação de cor, onde a dimensão de cada vetor pode ser estendida por um grupo de pixels vizinhos, localizados em cada posição da imagem. Segundo os autores, o método de aprendizagem *Support Vector Machine* (SVM) é eficiente para o problema de detecção de sinal de trânsito.

Saturnino et al. (2007), e Kiran et al. (2009), aplicaram este método para a fase de classificação de forma, assim como para a fase de reconhecimento.

No trabalho de Le et al. (2010), os autores aplicaram a transformada de *Hough* para detectar sinais de trânsito circulares. No caso de sinais de trânsito triangulares, foi utilizado o algoritmo de detecção de borda *Canny*, com múltiplos limiares para obter regiões candidatas, e a partir destas, extrair propriedades geométricas como o número de extremidades e o tamanho de ângulos entre duas linhas para verificação de triângulo.

Segundo os autores, a complexidade da fase de classificação depende do número de regiões candidatas e, como a área das regiões candidatas utilizada é pequena, a complexidade é pequena o suficiente para processar em tempo real. Os autores realizaram experimentos com imagens de 640x480 pixels de resolução, apresentando uma taxa de detecção de 92,91% no reconhecimento.

4.2.5 Outras abordagens

O trabalho de Martinović et al. (2001) lida com métodos para localização e classificação automatizada de sinais de trânsito. O algoritmo de detecção é treinado para detectar apenas sinais de trânsitos circulares. Uma fase de classificação adicional é necessária para separar estes sinais dos outros. Para que o sistema possa trabalhar em tempo real, os autores empregaram o algoritmo de detecção Viola-Jones (VIOLA; JONES, 2001), o qual trabalha deslizando uma janela de detecção em uma imagem e é baseado em uma cascata de características de Haar. Em cada posição, o classificador toma uma decisão se houver um objeto desejado dentro da janela. Após um sinal ter sido detectado com sucesso em uma imagem, o processo de classificação inicia-se para determinar o tipo do sinal. Para treinar o classificador, os autores usaram 757 imagens de sinais de trânsito normalizadas para um tamanho de 24x24 pixels, e 3000 imagens como falso positivo. Para classificar o sinal corretamente, é exigida uma normalização de tamanho, tendo sido usado um sinal padrão de 10x10 pixels. O procedimento de redimensionamento da imagem é realizado usando um algoritmo de interpolação bilinear. Uma operação de recorte é necessária, de tal forma que apenas a parte central do sinal permanece. Após a normalização de tamanho, a informação de cor é descartada, e a imagem é convertida em imagem binária. Foi utilizada uma rede neural

Multilayer Perceptron (MLP) com treinamento *Back Propagation* (BP) para classificação dos sinais de trânsito.

O sistema consiste em duas aplicações, sendo uma usada para detectar e reconhecer sinais de trânsito em imagens estáticas, e a segunda usada para detectar sinais de trânsito em vídeos em tempo real. Ambos os sistemas usam o mesmo núcleo, com a adição de um subsistema de rastreamento de objetos na segunda aplicação.

O sistema desenvolvido foi testado com um conjunto de teste de 146 imagens estáticas (para a primeira aplicação) apresentando uma taxa de reconhecimento de 72,6% e uma sequência de vídeo de 98 minutos contendo 128 sinais de trânsito com taxa de reconhecimento de 82,8%. Segundo os autores, os resultados obtidos mostram que o sistema é aplicável para processamento de vídeo em tempo real, apresentando melhores resultados quando usado em uma sequência de vídeo, em comparação com o método padrão de detecção de sinais de trânsito em imagens estáticas.

O trabalho de Fleyeh (2005) apresenta um algoritmo para detecção e segmentação de sinais de trânsito baseado em cores e conjuntos *fuzzy*. As imagens utilizadas neste trabalho foram obtidas por uma máquina fotográfica digital instalada em um carro. As imagens RGB são convertidas para o espaço de cor HSV, e segmentadas por meio de um conjunto de regras *fuzzy*, dependendo dos valores de matiz e saturação de cada pixel no espaço de cor HSV. O autor aplicou sete regras *fuzzy* para realizar a segmentação e extrair as cores dos sinais de trânsito. Foram usadas a matiz e a saturação normalizada, como um conhecimento a priori do sistema de inferência *fuzzy*, para especificar o intervalo de cada cor no sistema. O método foi testado em imagens obtidas ao ar livre, sob diferentes condições de luz, e foi testado em imagens coloridas de diferentes países europeus, e segundo o autor mostrou ter alta robustez.

O trabalho de Lorsakul e Suthakorn (2007) apresenta um estudo para o reconhecimento de sinais de trânsito usando técnicas de redes neurais. As imagens são pré-processadas em estágios, com técnicas de processamento de imagens, tais como, técnicas de *threshold* (limiarização), filtro Gaussiano, e detector de bordas *Canny*, contornos e elipses. O objetivo principal é reduzir o espaço de busca e indicar somente regiões potenciais para aumentar a eficácia e velocidade do sistema.

Na fase de detecção, a imagem é pré-processada, realçada, e segmentada de acordo com as propriedades de cor e forma do sinal. As imagens de sinal de trânsito são investigadas para detectar potenciais regiões de pixel que poderiam ser reconhecidas como possíveis sinais

de trânsito. Os objetos potenciais são normalizados a um tamanho especificado para serem usados como entrada para a fase de reconhecimento. Este estudo apenas investiga objetos de formas circulares e hexagonais, que são formas comuns em muitos tipos de sinais de trânsito.

O sistema é treinado e validado utilizando redes neurais para reconhecer os sinais de trânsito padrões. A principal razão para escolher este método, segundo os autores, é reduzir o custo computacional, a fim de facilitar a implementação em tempo real. A primeira estratégia consiste em reduzir o número de entradas MLP (*Multilayer Perceptron*) pelo pré-processamento das imagens com sinais de trânsito, e a segunda estratégia consiste em buscar por uma melhor arquitetura de rede com complexidade reduzida pela seleção de um critério de erro desejável por treinamento. Os autores apresentaram os experimentos mostrando resultados consistentes com classificações precisas de sinais de trânsito padrões em imagens com fundo complexo. O tempo de processamento em cada quadro (*frame*) de imagem foi considerado pelos autores satisfatório para aplicações de tempo real.

O trabalho de Wanniarachchi et al. (2008) apresenta os resultados da tentativa de localizar e extrair sinais de trânsito baseados em técnicas de processamento de imagens. Os sinais de trânsito são categorizados em três classes de acordo com suas formas: sinais em forma de losango com fundo amarelo (sinais de advertência); sinais em círculos de cor vermelha (sinais de controle) e sinais de formas retangulares com fundo azul (outros sinais úteis aos motoristas). Entretanto, o trabalho focou somente no reconhecimento de sinais de advertência, com borda na forma de losango, e sinal na cor preta com fundo amarelo. Para isto, foi realizado um pré-processamento inicial para identificar as regiões amarelas (o espaço de cores RGB foi convertido para o espaço de cor CIExyz). O próximo passo foi encontrar as bordas das regiões identificadas. A detecção de bordas é um processo para a detecção de descontinuidades nos valores de intensidade. Tais descontinuidades podem ser detectadas usando a primeira ou segunda ordem padrão do operador de detecção de bordas *Canny* (CANNY, 1986). Este operador encontra bordas, procurando locais máximos no gradiente da imagem.

Depois de extrair as bordas, foi aplicada a operação morfológica “dilatação” às imagens para localizar o local da placa. Dilatação é uma operação que “cresce” ou “engrossa” objetos em uma imagem binária. O modo específico e a medida do espessamento são controlados por um parâmetro de forma, referido como um elemento estruturante. A morfologia matemática é uma ferramenta para extração de componentes de imagem que são úteis na representação e descrição das formas dos contornos de regiões. Segundo os autores, o

algoritmo foi testado para uma variedade de imagens e apresentou 98% de precisão na localização e extração de sinais de trânsito.

O trabalho de Medici et al. (2008) descreve a fase final de um sistema de reconhecimento de sinal de trânsito, capaz de classificar em tempo real cerca de 200 diferentes sinais de trânsito italianos. Os sinais foram subdivididos em seis classes, sob diferentes condições de luz e diferentes condições climáticas. Para a obtenção das imagens, foi utilizada uma máquina fotográfica colorida montada em um veículo em movimento.

O algoritmo de detecção é baseado em análise de cor e forma, e gera uma lista de regiões candidatas onde podem ser encontrados potenciais sinais de trânsito. Para ser independente de condições de iluminação, uma fase de equalização de cor é realizada na imagem de entrada, a imagem de RGB resultante é convertida para o espaço de cor HSV, que foi escolhido pelo fato do componente de matiz (*Hue*) ser invariante às mudanças de intensidade de iluminação. Um limiar na matiz (*Hue*) e saturação (*Saturation*) é usado para obter imagens binárias Vermelhas, Azuis, Amarelas e Brancas.

As imagens coloridas candidatas são redimensionadas e convertidas para tons de cinza, onde é aplicada para cada imagem uma técnica de alongamento de contraste. Uma rede neural *Perceptron* multicamada é então usada para classificação de diferentes modelos de sinal de trânsito. Segundo os autores, o sistema pode classificar mais de 200 diferentes sinais de trânsito italianos em tempo real, com uma taxa de reconhecimento de 80% a 90%.

Belaroussi e Tarel (2009) apresentam uma abordagem para detecção de sinais de trânsito circulares e poligonais em tempo real em imagens fotográficas, independentemente da sua orientação e pose. A detecção foi realizada usando uma transformação de simetria, denominada de *Bilateral Chinese Transform* (BCT), baseada em transformação da orientação de pares de gradientes capaz de detectar círculos e polígonos indistintamente. Essa transformação decompõe um objeto em um conjunto de contornos paralelos com gradientes opostos, e modela essa simetria de campo gradiente usando um acúmulo de evidências de simetria radial. Em um teste realizado com a técnica, usando um banco de teste de 89 imagens de 640x480 pixels de resolução contendo 92 sinais de trânsito com diferentes cores e formas, a abordagem proposta apresentou uma taxa de detecção de 86%, com 25 falsos positivos. Os autores também mencionam a rapidez do processo de detecção, onde foram gastos cerca de 30 milissegundos por imagem.

Em Baró et al. (2009c), é apresentada uma abordagem para a detecção e classificação de sinais de trânsito. Os autores caracterizam seu trabalho em duas partes, primeiramente, durante a fase de detecção, para discriminar um conjunto de sinais de trânsito, foi proposta uma nova classificação binária através de uma versão evolutiva de *Adaboost* (um algoritmo de aprendizagem de máquina proposto por Freund e Schapire (1997)). É uma meta-heurística, que evita as limitações do algoritmo original em termos de dimensionalidade do espaço de características.

Em segundo lugar, o processo de classificação é definido como um problema de classificação multiclasse, e para distinguir entre um grande conjunto de classes, foi proposta uma técnica de aprendizagem. A abordagem é baseada na incorporação de uma floresta de estruturas de árvores ótima, no âmbito do *framework Error-Correcting Output Code* (ECOC), permitindo compartilhamento de características (por exemplo, nodos da árvore, classificadores base, e dicotomias) entre classes de uma maneira muito robusta. Finalmente, foi desenvolvido um sistema de reconhecimento de sinal de trânsito a partir de um processo de mapeamento móvel, em que foi validada a robustez da abordagem. De acordo com os autores, o sistema oferece alto desempenho e uma melhor precisão do que as estratégias do estado da arte, além de ser robusto a uma alta variabilidade da aparência do sinal (ruído, deformação afim, oclusões parciais e iluminação reduzida).

Um trabalho similar ao de Baró et al. (2009c) é encontrado em Madeira et al. (2005), onde um sistema móvel de mapeamento processa de forma automática os sinais de trânsito. Neste trabalho, a precisão do reconhecimento é de mais de 80% em um conjunto reduzido de tipos de sinal de trânsito.

4.3 Trabalhos com aplicações de informações georreferenciadas

Diversos trabalhos podem ser encontrados na literatura com aplicações de banco de imagens georreferenciadas, onde as informações desses bancos foram obtidas a partir de mapeamento móvel, muitas vezes com uso de veículos terrestres.

Amlacher e Paletta (2008) propuseram o uso de imagens georreferenciadas para reconhecer um conjunto de pontos turísticos por meio de serviços móveis de visão em cenários urbanos. Outros trabalhos têm utilizado as informações de GPS para complementar a informação visual para o reconhecimento de eventos em sequências de vídeo (YUAN et al., 2008), ou para definir a indexação de imagem de lugares (CAI; ZHENG, 2008), que podem ser aplicados para orientar a aquisição de imagens para a percepção espacial, para ajudar a configuração de câmara de vigilância, ou para a indexação de imagens em excursões virtuais.

O trabalho de Baró et al. (2009a) está relacionado ao reconhecimento de objetos em bancos de dados de imagem urbanas. Foi idealizado em virtude do crescimento exponencial do conteúdo da multimídia na internet, onde mecanismos de mineração de dados e processos de recuperação são cada vez mais necessários. Os autores afirmam que a interação de mapas *on-line* é uma das aplicações emergentes, onde estas tecnologias podem ser implantadas para acesso a dados multimídia associados à lugares físicos. Esta fonte de conhecimento pode ser útil para usuários, a fim de computar automaticamente a rota de uma viagem ou procurar por lugares de entretenimento e hotéis. Inicialmente foi utilizado um sistema móvel de mapeamento (ALAMÚS et al., 2005) para criar um inventário das ruas de Barcelona, e para capturar fachadas dos edifícios. No total, 3.800 ruas (o que corresponde a mais de 1.290 km) foram digitalizadas com seis câmaras coloridas com diferentes orientações, todas sincronizadas com um sistema de GPS/INS (*Global Positioning System/Inertial Navigation System*), adquirindo uma imagem a cada cinco metros, totalizando cerca de 500.000 imagens georreferenciadas. A arquitetura do projeto de Barcelona é ilustrada na Figura 4.2.

Figura 4.2: Esquema utilizado no projeto de Barcelona.
 Fonte: (BARÓ et al., 2009a).

As imagens capturadas foram armazenadas usando o formato *LizardTech MrSID* (www.lizardtech.com) com resolução de 1.200x1.600 pixels. Em cada imagem foi aplicado o detector de região *Hessian-Laplace* (MIKOLAJCZYK; SCHMID, 2005). Depois de normalizar as elipses detectadas, o descritor SURF (BAY et al., 2008) é computado para cada região detectada. Foram obtidas mais de 1.000 regiões por imagem, o que corresponde a um total de 500.000.000 descrições no banco de dados. Em virtude da grande quantidade de dados, foi gerada uma representação binária compacta para cada descritor, que é usada como um código *hash*, com intuito de ter um melhor desempenho na busca. O sistema é composto de duas aplicações, uma no lado do servidor e outra no lado do cliente. A do servidor foi implementada utilizando a tecnologia Java (*Servlet*) para desenvolvimento de aplicações Web. Esta aplicação é responsável pela tarefa de recuperação de objetos e geração do código para mostrar os resultados no lado do cliente. O processo de recuperação consiste de duas

operações básicas, que são usadas em diferentes passos: a correspondência “Ponto a Ponto”, que consiste em encontrar pontos em uma cena os quais são similares a cada ponto do modelo; e estimação de Homografia. Quando um conjunto de pontos do modelo e um conjunto de pontos da cena são ligados, essas correspondências são usadas a fim de estimar a homografia H , que permite transformar o modelo de pontos x para x' da cena correspondente. O resultado é a matriz de transformação H , onde $x' = Hx$. A estimação de H é realizada usando o algoritmo RANSAC (FISCHLER; BOLLES, 1981), que permite descartar *outliers* e por essa razão é robusta a erros no processo de correspondência ponto a ponto. O aplicativo cliente faz uso da API *Google Maps* (GOOGLE APIs, 2010), para organizar as informações geográficas em diferentes camadas, cada uma com um conteúdo específico, tais como, imagens aéreas ou de satélite, nomes das ruas, informações de tráfego, empresas etc. Foi proposta a incorporação de uma nova camada de Conteúdo Visual, que permite consultas baseadas nessa camada. O aplicativo cliente permite ao usuário escolher entre um conjunto de objetos visuais predefinidos, representados como um conjunto de características de ponto. O objeto selecionado é enviado para a aplicação do lado do servidor, onde é processado. Uma vez que o processo de recuperação é finalizado, a aplicação cliente mostra os pontos onde as instâncias candidatas para o objeto selecionado são encontradas, com a imagem do exemplo e suas coordenadas geográficas. A fim de validar ou complementar os resultados, os autores utilizaram o *Google Street View* para mostrar a área correspondente à recuperação. Em experimentos realizados, as taxas de acerto sempre permaneceram acima de 60%. Segundo os autores, a maior parte das falhas nos testes é produzida por sinais com dimensões pequenas, onde o número de pontos de interesse detectados é reduzido e o processo de *matching* não encontra correspondências suficientes.

Em outro trabalho de Baró et al. (2009b), foi proposta a construção de uma camada de conteúdo visual para reconhecimento de objetos em um banco de dados de imagens urbanas, que descreve a aparência visual das localizações geográficas na cidade de Barcelona. Mapas padrões são definidos como uma fusão de camadas, onde cada uma contém dados específicos para localização, por exemplo, de uma empresa em particular. Com este tipo de sistema, os usuários podem calcular automaticamente a rota ideal para uma viagem, procurar locais de entretenimento ou hotéis próximos de sua posição atual. As imagens georreferenciadas das ruas formam uma base de dados que pode ser potencialmente útil para diversas aplicações, tais como a procura de empresas por meio de logotipos, manutenção de mobiliário urbano, ou para obter qualquer tipo de categoria de objeto visual. Nesse trabalho,

os autores utilizam o mesmo banco de dados de imagens do trabalho de Baró et al. (2009a). Um programa *JavaScript* no navegador Web do cliente controla a interação com o mapa, enviando consultas a um servidor que une os dados e a aplicação com o *Google Maps*. Finalmente, os dados recebidos são mostrados em uma camada de Conteúdo Visual no cliente, indicando sua localização geográfica correspondente. O enfoque *Candid covariance-free incremental principal component analysis* (CCPA) (WENG et al., 2003) é aplicado para projetar as 64 dimensões originais do espaço de características SURF dentro de um espaço de características de 31 dimensões. Esta estratégia permite uma representação compacta dos descritores, computando os componentes principais de forma incremental, sem estimar a matriz de covariância, e assim, ser capaz de processar rapidamente a enorme quantidade de dados. Isto é feito mantendo a escala das observações e computando a média das observações de forma incremental. Uma vez que os componentes principais são computados, os primeiros 31 componentes mais significativos são selecionados, mantendo mais de 95% da variância. Baró (2009b) cita o trabalho de Quack et al. (2008), no qual o autor apresenta um sistema com complexidade reduzida, considerando somente as imagens em um mosaico geográfico, que é muito útil para localizar determinados pontos turísticos ou edifícios de uma cidade. No entanto, os autores propõem um método interessante de ligação entre imagens e outras fontes de informação como *Wikipédia*. Este método, segundo Baró et al. (2009b) é possível de ser introduzido em seu sistema.

4.4 Trabalhos Relacionados com relação ao reconhecimento de caracteres

Apesar da área de reconhecimento de caracteres já ser uma área de pesquisa consolidada, com um número considerável de trabalhos realizados, nesta seção são apresentados alguns trabalhos investigados mais recentes, que realizam o reconhecimento de caracteres em alguns domínios de problemas.

O trabalho de Jain e Ko (2008) apresenta um algoritmo de classificação para reconhecer dígitos numéricos manuscritos (0-9), sendo utilizada uma implementação de uma PCA (*Principal Component Analysis*), combinada com o algoritmo 1-*Nearest Neighbor* (Primeiro Vizinho mais Próximo) para reconhecer os dígitos numéricos. Os resultados

experimentais mostram que o algoritmo é capaz de obter uma acurácia máxima de 78,4% no reconhecimento.

A contribuição do trabalho de Pereira et al. (2010) foi melhorar a taxa de precisão de reconhecimento de caracteres manuscritos usando um novo método de extração de características, também baseado em Análise de Componentes Principais (PCA). Neste caso, aplica-se uma nova técnica que visa combinar os melhores aspectos de uma Análise de Componentes Principais Modular (MPCA) e uma Análise de Componentes Principais de Imagem (IMPCA). IMPCA também conhecida como Análise de Componentes Principais Duas Dimensões, é um enfoque baseado em PCA que representa cada imagem como uma matriz bidimensional, em vez de um vetor de uma dimensão. Com o uso desta técnica, a matriz da imagem original não precisa ser convertida em um vetor. Desta forma, o custo computacional da computação da matriz de covariância e extração dos autovetores é consideravelmente menor que o exigido por técnicas padrão de PCA. Os autores realizaram experimentos usando um banco de dígitos numéricos (também utilizado por Lecun et al. (1998)) contendo 70.000 imagens de dígitos numéricos manuscritos (*handwritten*), normalizadas para uma resolução de 28x28 pixels. O banco foi dividido em três partes: 50.000 amostras para treinamento, 10.000 amostras para validação e 10.000 amostras para testes. Segundo os autores, os resultados com a técnica foram: 96,98% de precisão usando uma PCA com somente oito componentes principais, e 97,01% de precisão usando trinta componentes principais.

No trabalho de Trentini et al. (2010), imagens de placas de licenciamento veicular são capturadas por câmaras de vídeo convencionais e são pré-processadas para melhorar a qualidade (remoção de ruídos) e tratar os efeitos das perspectivas geométricas das placas nas imagens. A partir da imagem segmentada de uma placa, é realizada uma varredura em cada coluna da imagem e são contadas as quantidades de pixels pretos, representando a densidade correspondente a cada coluna. Para a segmentação dos caracteres, ou seja, separar os caracteres em relação ao fundo da placa são observados os picos, vales e pontos de mínimo por meio de uma função de análise de máximos ou mínimos locais. Para o reconhecimento dos caracteres é utilizado o algoritmo *Random Trees*, também chamado de *Random Forests*, o qual é um classificador baseado em árvores de decisão e pode reconhecer os padrões de várias classes ao mesmo tempo. Na fase de treinamento do classificador, os autores utilizaram três vídeos, onde obtiveram 2.102 amostras de letras e 2.011 amostras de dígitos numéricos. Os

resultados obtidos pelo classificador na fase de reconhecimento foram de 93,7% para o reconhecimento de letras e de 86,9% para o reconhecimento de números.

O trabalho de Shrivastava e Gharde (2010) é utilizado para reconhecimento de números *Devanagari* manuscritos. *Devanagari* é um manuscrito alfabético usado por vários idiomas na Índia. Para a realização do trabalho os autores utilizaram a técnica de aprendizagem de máquina *Support Vector Machines* (SVM). Foram coletadas 2.000 amostras de 20 diferentes pessoas que tem variação no estilo de escrita. Para cada imagem contendo os números escritos pelas pessoas, foram extraídos 18 descritores (baseados em Momentos Invariantes) para serem usados na SVM. Segundo os autores, os experimentos realizados apresentaram uma taxa média de 99,48% de precisão no reconhecimento.

4.5 Considerações Finais

Os trabalhos investigados mostram que, o SIFT é considerado um forte descritor de características locais em termos de sensibilidade e precisão segundo a avaliação provida por Mikolajczyk e Schmid (2005), e é apontando com sendo o mais adequado para TSR segundo Höferlin e Heidemann (2010). Segundo Farag e Abdel-Hakim (2004), o SIFT alcança algumas vantagens para o processo de correspondência, pois não há a necessidade de se ter um número grande de imagens reais de sinais (*templates*) para treinar o sistema, como acontece, por exemplo, com abordagens baseadas em redes neurais, SVM etc.

A revisão bibliográfica evidenciou a possibilidade de se utilizar o algoritmo SIFT para a detecção e o reconhecimento de placas de sinalização de trânsito, pelo fato dos descritores SIFT serem robustos a mudanças na escala e rotação, mesmo que o processamento tenha sido feito de forma *off-line* em alguns trabalhos. Segundo Ihara et al. (2009), o custo computacional da extração e correspondência SIFT é alto, o que torna o processamento em tempo real uma tarefa difícil de ser alcançada.

As imagens utilizadas no presente trabalho são obtidas a partir de um veículo em movimento, que capturam imagens com cenas com fundo complexo em uma alta resolução, representando uma alta complexidade para os algoritmos definidos e implementados, o que justifica o processamento ser realizado de forma *off-line*.

Os trabalhos de Reiterer et al., (2009) e Baró et al. (2009a) apresentam metodologia semelhante, embora comum para sistema que usam descritores de pontos chave, mas que serviu de inspiração para a metodologia proposta neste trabalho de doutorado. Ambos os trabalhos apresentam a construção de um banco de dados de imagens modelos (*templates*); extração dos descritores dos pontos de interesse das imagens dos *templates* do banco de dados; extração dos descritores das imagens obtidas pelo sistema móvel de mapeamento; correspondência entre os descritores das imagens com todos os descritores armazenados no banco de dados; reconhecimento dos sinais; e por fim, cálculo das coordenadas 3D.

Em alguns trabalhos, como por exemplo, o de Martinović et al. (2001), tanto na fase de treinamento quanto na fase de classificação, o sinal deve ser normalizado para um tamanho padrão. Isto pode ter a desvantagem da necessidade de usar imagens capturadas de forma controlada, onde o sinal na imagem tem que apresentar um tamanho padronizado, para que se possam obter bons resultados no reconhecimento.

Dos trabalhos investigados, apenas em Baró et al. (2009a) foi utilizado o algoritmo RANSAC (FISCHLER; BOLLES, 1981) para a extração dos *inliers* na fase de reconhecimento, entretanto, o trabalho dos autores é destinado ao reconhecimento de fachadas, que muitas vezes são maiores e apresentam mais detalhes que as placas de sinalização de trânsito. Mesmo assim, os autores, demonstraram ter problemas com o número pequeno de correspondências. Segundo os autores, a maior parte das falhas no reconhecimento é produzida por sinais com tamanhos pequenos, devido ao número de pontos de interesse detectado ser reduzido, o que faz com que o processo de *matching* não encontre correspondências suficientes. Essa dificuldade aparece com mais evidência quando os objetos de reconhecimento são as placas de sinalização de trânsito. Segundo Höferlin e Heidemann (2010) os sinais de trânsito são objetos altamente estruturados e simétricos, que podem ser capturados em baixa resolução, o que faz com que apenas um número pequeno de descritores seja correspondente, dificultando o reconhecimento. Todas essas evidências inspiraram a definição e a implementação de um algoritmo para a extração dos *inliers* e ser utilizado, neste trabalho, para o estágio de reconhecimento das placas de sinalização.

Na metodologia definida neste trabalho, durante o estágio de reconhecimento, quando o sinal de trânsito for de limite de velocidade, faz-se necessário o reconhecimento dos dígitos numéricos, sendo muito importante para evitar falsos positivos. Embora existam muitos trabalhos na literatura para essa finalidade, foi definido e implementado um algoritmo estritamente rápido para o reconhecimento dos dígitos numéricos das placas de velocidade (SILVA et al., 2011).

Capítulo 5

Metodologia da detecção, reconhecimento e determinação das coordenadas das placas de sinalização

5.1 Considerações Iniciais

Neste capítulo, é descrita a metodologia utilizada neste trabalho de doutorado para se realizar a detecção, reconhecimento das placas de sinalização e o georreferenciamento das mesmas. A implementação dessa metodologia deu origem a um Sistema denominado de *AutoTSR (Automatic Traffic Sign Recognition)*. Inicialmente são elucidados os Materiais e Métodos abordando resumidamente ao final o fluxograma da metodologia utilizada. Posteriormente, é realizada uma descrição técnica do Sistema construído e das classes de objetos implementadas a partir da modelagem em UML (*Unified Modeling Language*), assim como também do Banco de Dados construído para armazenar as informações de *templates*, de levantamentos para coleta de dados e de resultados produzidas pelo Sistema. As estruturas de dados empregadas para minimizar acesso ao banco e para armazenar temporariamente os resultados no estágio de detecção e reconhecimento das placas também são descritas. Na sequência, são explanados os algoritmos definidos e implementados para a detecção e reconhecimento das placas de sinalização. Um deles é o algoritmo para a extração dos *inliers*, que é executado após a extração dos pontos chave pelo algoritmo SIFT (LOWE, 2004) e após a correspondência entre os pontos chave das imagens de entrada (provenientes da sequência de imagens provida por levantamento) com os pontos chave do *template* correspondente. Além deste, é apresentado o algoritmo também definido e implementado neste trabalho para realizar o reconhecimento dos dígitos numéricos das placas com sinais de limites de velocidade. Finalmente, neste capítulo, é apresentado o processo para a determinação das coordenadas do terreno das placas de sinalização.

5.2 Materiais e Métodos

Para a implementação do Sistema *AutoTSR* de detecção, reconhecimento e georreferenciamento de placas de sinalização, foram desenvolvidas as seguintes etapas: criação de um banco de dados contendo os *templates* das placas de sinalização que foram utilizadas, as informações das imagens capturadas, e os resultados do reconhecimento e georreferenciamento; aquisição dos pares de imagens das placas, dos dados de navegação e posicionamento usando o Sistema Móvel de Mapeamento; extração dos descritores das imagens (imagens de entrada e *templates*); reconhecimento das placas e georreferenciamento das mesmas. Cada uma dessas etapas é detalhada a seguir.

As imagens das placas de sinalização foram obtidas com o SIMMOV (Sistema Móvel de Mapeamento) que consiste de um veículo Peugeot Partner, com um par de câmaras instaladas no teto (Figura 5.1).

Figura 5.1: Sistema Móvel de Mapeamento.

O par de câmaras NIKON D300S (Figura 5.2) tem 12 *Megapixels* de resolução geométrica, e são posicionadas no teto do veículo com o eixo óptico paralelo ao deslocamento. O sistema foi montado de modo que os pares de imagens adquiridas tivessem uma sobreposição longitudinal mínima de 60%.

Figura 5.2: Câmera Nikon D300S.

Antes do levantamento das imagens e dos dados de posicionamento e navegação é necessário calibrar o sistema (seção 3.7), para determinação dos parâmetros de orientação interior (x_0, y_0, f, k_1) (coordenadas do ponto principal em relação ao centro da imagem, distância focal e distorção radial simétrica) de cada câmera. Nesse processo de calibração, também são determinados os deslocamentos entre a posição da antena do GPS (modelo PROPAK-V3-424-RT2j) (Figura 5.3 (a)) e a posição da Unidade de Medida Inercial (*Inertial Measure Unit*) (Figura 5.3 (b)) (modelo IMAR FSAS-EI-SN).

(a)

(b)

Figura 5.3: a) Aparelho GPS. b) Unidade de Medida Inercial – IMU.

Esses deslocamentos são importantes, pois com eles, são obtidos os parâmetros de posicionamento e atitudes (ângulos de orientação) de cada câmara ($X_c, Y_c, Z_c, \omega, \phi, \kappa$) (MONICO, 2000) para cada imagem tomada em uma sequência (SILVA et al., 2003).

Em uma sequência de pares de imagens, as placas são “buscadas” em regiões de interesse, onde se localizam as placas de sinalização. Em geral, as placas de sinalização encontram-se dos lados esquerdo e direito das ruas e as imagens da direita da sequência são utilizadas para a realização da detecção das feições das placas de sinalização.

Antes do estágio de detecção e reconhecimento, são definidos os *templates* (sinais de trânsito de interesse), os quais podem ser obtidos a partir das imagens adquiridas nos levantamentos. Os seguintes dados dos *templates* são armazenados no banco de dados para uso posterior pelo algoritmo de reconhecimento: imagem do sinal de trânsito, informações dos descritores SIFT (LOWE, 2004) e informações a respeito do tipo de placa. O tamanho dos *templates* pode variar, não sendo necessário definir um tamanho padrão fixo. A vantagem disso é que não há a necessidade de redimensionar as imagens dos *templates*. Como consequência a metodologia desenvolvida neste trabalho pode ser aplicada não apenas para o reconhecimento de sinais de trânsito, mas também para o reconhecimento de fachadas de edifícios comerciais ou para outro tipo de reconhecimento de objetos. O detalhamento do banco de dados é apresentado na seção 5.3.2.

Como já descrito no capítulo 4, existe uma grande quantidade de algoritmos para a extração de descritores (*features*) (GONZALEZ; WOODS, 2002) e um dos algoritmos que tem se destacado na literatura e é utilizado nesse trabalho é o algoritmo SIFT – *Scale Invariant Feature Transform* (LOWE, 2004), que é razoavelmente invariante à mudança de iluminação, ruído, rotação, escala e pequenas mudanças no ponto de vista. A seção 2.3 descreve com detalhes todos os estágios da computação do algoritmo SIFT.

Os descritores extraídos com o algoritmo SIFT nas imagens da direita são casados (*matching* ou correspondência) com os descritores dos *templates*, que já foram anteriormente armazenados em um banco de dados com as placas de sinalização utilizadas. O *matching* entre os descritores extraídos e os armazenados no banco de dados deve ser o de menor distância Euclidiana entre o vetor extraído na sequência e o vetor do *template* (REITERER et al., 2009). Valores abaixo de um limiar, determinado de maneira empírica, é considerado como um *matching* positivo. A partir dos pontos que são correspondentes entre as duas

imagens ocorre a extração dos pontos que condizem com o modelo do *template* da placa (*inliers*).

Nessa etapa, as placas de sinalização estão detectadas e reconhecidas com as imagens da direita da sequência. Com o conhecimento da montagem das câmaras no SIMMOV, é possível determinar em qual região da imagem da esquerda a placa será buscada, pois essa região se torna conhecida após se estimar suas coordenadas, não necessitando realizar a busca pela placa em toda a imagem. Em seguida extraem-se os descritores de pontos chave da imagem da esquerda para realizar a correspondência e o reconhecimento da mesma placa da imagem da direita. O reconhecimento é necessário na imagem da esquerda, pois podem ocorrer oclusões por conta da mudança do ponto de vista da câmara.

Para determinar o posicionamento da placa no terreno são utilizados os respectivos centros geométricos das placas detectadas nas imagens da direita e da esquerda. Esses centros geométricos são transformados em fotocoordenadas (milímetros) para a realização da interseção fotogramétrica (seção 3.7), cujos resultados são as coordenadas (X , Y , Z) da placa no terreno, ou seja, o seu georreferenciamento.

A partir do levantamento realizado, as placas de sinalização são reconhecidas e georreferenciadas, e a partir desses resultados, é confeccionada uma carta topográfica no Sistema de Referência WGS84 (MONICO, 2000) (IBGE, 2011) utilizando uma base cartográfica de Presidente Prudente/SP.

O Microsoft Visual Studio 2010 foi escolhido como ambiente de programação e a linguagem de programação C# como base para o desenvolvimento de um sistema que implemente todos os algoritmos e processos para a detecção, o reconhecimento e o georreferenciamento das placas de sinalização de maneira automática. A visualização da carta topográfica contendo os resultados é feita com as funcionalidades da API do *MapWindow* para a linguagem de programação C# (MAPWINDOW, 2010).

A Figura 5.4 ilustra um fluxograma da metodologia utilizada para detecção, reconhecimento e georreferenciamento das placas de sinalização.

Figura 5.4: Fluxograma da metodologia utilizada para detecção, reconhecimento e georreferenciamento das placas de sinalização.

A Figura 5.5 apresenta um exemplo de um par de imagens adquirido a partir de um levantamento com o SIMMOV. Em (a) tem-se a imagem capturada com a câmara da esquerda e em (b) a imagem capturada com a câmara da direita. Os retângulos tracejados representam a região de sobreposição dessas duas imagens.

Figura 5.5: Par de imagens esquerda (a) e direita (b) capturadas pelo SIMMOV.

5.3 Descrição técnica do Sistema *AutoTSR*

O Sistema *AutoTSR* (*Automatic Traffic Sign Recognition*) como um todo é composto por diversas funcionalidades, todas elas destinadas a contribuir para a proposta deste trabalho. As principais funcionalidades do Sistema *AutoTSR* são relacionadas a seguir:

- Cadastro de *Templates*: a partir desta funcionalidade são cadastrados os *templates* das placas de sinalização, constituídos de uma descrição do sinal de trânsito, do limite de velocidade quando for placa de velocidade, dos pontos chave e seus descritores extraídos a partir das imagens de placas utilizando o algoritmo SIFT (LOWE, 2004), e o número mínimo de pontos chave necessário para o reconhecimento. É importante ressaltar, que o algoritmo para extração dos *inliers* (seção 5.4.1) definido e implementado neste trabalho, necessita de um número mínimo de 3 pontos chave, onde esse valor (3) já foi definido empiricamente no Sistema. No entanto, para cada *template*, pode-se definir o número mínimo de pontos chave desejado, sendo que um maior número implica em um reconhecimento mais acurado, mas por outro lado, pode descartar resultados de reconhecimentos corretos de objetos de interesse que apresentaram um menor número de pontos;
- Cadastro de Projetos: esta funcionalidade é utilizada para cadastrar os dados de projetos, sendo que cada projeto possui um ou mais levantamentos;

- Configuração das Câmaras – Orientação Interior (OI): os dados das configurações das câmaras utilizadas nos levantamentos são cadastrados por essa funcionalidade;
- Levantamento de Placas: é nesta funcionalidade que, a partir da entrada de informações, tais como, diretórios com as imagens da Esquerda e Direita, descrição do levantamento, o projeto ao qual o levantamento pertence, e as configurações das câmaras utilizadas, é realizado todo o processo de detecção e reconhecimento das placas de sinalização (as seções 5.4 e 5.5 descrevem todo esse processo). Os resultados desse processo são armazenados no banco de dados (seção 5.3.2);
- Georreferenciamento das Placas de Sinalização: nesta funcionalidade é realizado o processo de georreferenciamento das placas de sinalização com os dados obtidos com o processo de detecção e reconhecimento, ou a partir de coletas manuais utilizando a funcionalidade Coleta Pontos. Os dados providos do GPS/INS a partir dos levantamentos realizados são sincronizados com a sequência dos pares de imagens também providos pelos levantamentos (seção 3.6);
- Coleta Pontos: a partir desta funcionalidade, pontos são coletados manualmente e são utilizados para se fazer experimentos no que se refere a cálculos das coordenadas do terreno (georreferenciamento). Os resultados podem ser visualizados com o uso do *MapWindow* (MAPWINDOW, 2010) através dos arquivos *ShapeFiles* (camada para uma base cartográfica georreferenciada para ser visualizada na aplicação) que podem ser gerados na função anterior (Georreferenciamento das Placas de Sinalização);
- Consulta Levantamento(s) Realizado(s): Nesta funcionalidade, os levantamentos são encontrados no banco de dados por meio de uma busca. A partir dessa localização, nesta funcionalidade, podem-se acionar as seguintes funcionalidades: Georreferenciamento das Placas de Sinalização; Visualização do Relatório contendo os resultados da detecção, reconhecimento e georreferenciamento das placas de sinalização; Geração dos *ShapeFiles* contendo os resultados do georreferenciamento automático das placas de sinalização; e a funcionalidade Coleta Pontos;

Nas subseções seguintes é dada ênfase para a modelagem e a descrição das classes do Sistema *AutoTSR*, para o Banco de Dados definido e construído e para a Estrutura de

Dados criada para armazenar em memória os dados dos *templates* das placas de sinalização recuperados do Banco de Dados.

5.3.1 Modelagem e descrição das classes do Sistema *AutoTSR* em UML

Esta subseção tem como objetivo mostrar a modelagem das classes que foram definidas e implementadas para as funcionalidades do Sistema *AutoTSR* por meio de diagramas de classes simplificados. Segundo Booch et al. (2006), os diagramas de classes são os diagramas encontrados com maior frequência na modelagem de sistemas orientados a objetos. Um diagrama de classes mostra um conjunto de classes, interfaces, colaborações e seus relacionamentos. São importantes para a visualização, a especificação e documentação de modelos estruturais. Estas classes foram modeladas através da Linguagem de Modelagem Unificada (UML – *Unified Modeling Language*) e implementadas utilizando a linguagem de programação orientada a objetos C#. Também nesta subseção, são descritas as finalidades dessas classes para a construção do Sistema.

No processo de desenvolvimento de sistemas orientados a objetos, é importante se ter uma notação de especificação padronizada e realmente eficaz nas fases de análise de requisitos, análise de sistemas e de projeto. Especificar significa construir modelos precisos, sem ambiguidades e completos. Em particular, a UML dá suporte às necessidades geradas pela análise, projeto e implementação, que devem ser tomadas para o desenvolvimento e implementação de sistemas complexos de software.

A UML é uma linguagem padrão para a elaboração da estrutura de projetos de software. É empregada para a visualização, a especificação, a construção e a documentação de artefatos de um sistema, podendo ser utilizada com todos os processos ao longo do ciclo de desenvolvimento e através de diferentes tecnologias de implementação. A UML é adequada para a modelagem de sistemas, cuja abrangência poderá incluir sistemas de informação corporativos e até sistemas complexos embutidos de tempo real. É uma linguagem muito expressiva, abrangendo todas as visões necessárias ao desenvolvimento e implantação desses sistemas (BOOCH et al., 2006). A notação UML passou por um processo de padronização pelo OMG (*Object Management Group*) e se tornou um padrão do OMG (UML SPECIFICATION, 2011).

As classes do presente trabalho foram modeladas tendo por base o padrão proposto pela linguagem UML. Nestas classes estão codificadas todas as funcionalidades do sistema. Para uma melhor organização, as classes foram agrupadas segundo suas funcionalidades em pacotes (*packages*) para cada parte do Sistema *AutoTSR*. Na UML, um pacote é um mecanismo de propósito geral para a organização de elementos da modelagem em grupos (BOOCH et al., 2006). Na prática, esses pacotes são pastas (diretórios) dentro da pasta do projeto do Sistema *AutoTSR* onde estão subpastas (pacotes internos) e os arquivos com as implementações das classes. Os pacotes de nível mais externo criados para o desenvolvimento do Sistema deste trabalho de doutorado são: “*CamadaApresentacao*”, “*CamadaLogica*”, “*CamadaNegocio*”, “*Cartografia*”, “*ColetaPontos*”, “*ExifMetadata*”, “*Lowe*”, “*OpenCV*”, “*ReconheceCaracteres*” e “*ReconhecePlacas*”. Na sequência, cada pacote com suas classes e relações é mostrado graficamente em um diagrama simplificado juntamente com suas relações.

Primeiramente, na Figura 5.6, é apresentado um diagrama contendo um modelo simplificado das classes do pacote “*CamadaApresentacao*” e suas relações. As classes desse pacote são classes construídas para as interfaces de tela do sistema, possuem métodos para manipular os objetos de tela, instanciam objetos das classes do pacote “*CamadaNegocio*” que fazem acesso ao banco de dados, e também instanciam objetos de classes de outros pacotes para realizar as operações de detecção, reconhecimento e georreferenciamento das placas de sinalização.

Figura 5.6: Classes do pacote “*CamadaApresentacao*” e suas relações.

A Figura 5.7 apresenta um diagrama contendo as classes do pacote “*CamadaLogica*” e as classes do pacote “*CamadaNegocio*” com suas relações.

As classes do pacote “*CamadaLogica*” implementam estruturas de dados para serem usadas durante o estágio de detecção e reconhecimento das placas de sinalização, como é o caso das classes “*ListaTemplate*” e “*NoListaTemplate*” (mostrada mais adiante na Figura 5.15), que foram criadas para implementar uma estrutura de dados (uma lista encadeada de objetos) utilizada para armazenar em memória os dados dos *templates* das placas de sinalização. As classes “*ListaImagens*” e “*NoImagens*” também implementam uma lista encadeada de objetos, porém para guardar os dados das placas detectadas e reconhecidas.

O pacote “*CamadaNegocio*” possui classes responsáveis por acessar as tabelas do banco de dados do Sistema *AutoTSR* (a subseção 5.3.2 apresenta detalhes do banco de dados). Todas essas classes relacionam-se com a classe “*BancoDados*” do pacote “*CamadaLogica*” e possuem métodos com instruções SQL (*Structured Query Language*) para realizar a inserção, consulta, alteração e exclusão de dados no banco de dados. O acesso efetivo ao banco de dados propriamente dito é feito pelos métodos da classe “*BancoDados*”, que são responsáveis por realizar a conexão com o banco, por realizar inserções de dados, executar instruções SQL dentro de uma Transação⁸ e consultas.

Figura 5.7: Classes do pacote “*CamadaLogica*” e do pacote “*CamadaNegocio*” com suas relações.

⁸ Uma transação é uma sequência de operações executadas como uma única unidade lógica de trabalho onde sua integridade depende de quatro propriedades (Atomicidade, Consistência, Isolamento e Durabilidade).

A Figura 5.8 apresenta um diagrama das classes do pacote “*Cartografia*”, que são responsáveis por realizar o estágio do processo de georreferenciamento das placas de sinalização. Neste estágio, são calculadas as fotocoordenadas das placas detectadas e reconhecidas. Também, este pacote possui classes com métodos para fazer conversões do tempo do GPS para UTC (*Universal Time Coordinated*), calcular posições das câmaras a partir das coordenadas da antena do GPS e da orientação do sistema inercial INS, ler os arquivos de *log* do GPS (a seção 3.6 mostra mais detalhes) e sincronizar os dados do GPS com os pares de imagens obtidas por levantamento.

No pacote interno “*LatLongUTM*” estão classes com métodos para a conversão das coordenadas no sistema de coordenadas latitude, longitude e altitude para as coordenadas no sistema de coordenadas UTM (Universal Transversa de Mercator) (*E (East)*, *N (North)*, *H (Height)*) e vice-versa.

Figura 5.8: Classes do pacote “*Cartografia*”.

A Figura 5.9 apresenta um diagrama das classes do pacote “*ColetaPontos*”. Essas classes são utilizadas pela funcionalidade Coleta Pontos, onde os pontos são coletados manualmente e utilizados para se fazer experimentos nos cálculos das coordenadas do terreno.

Figura 5.9: Classes do pacote “*ColetaPontos*”.

A Figura 5.10 apresenta um diagrama contendo as classes do pacote “*ExifMetadata*” com suas relações. Essas classes possuem métodos para obter informações das imagens a partir de *tags* (campos ou etiquetas) de *metadados*, que são armazenados nas imagens pelas câmaras fotográficas. Esses *metadados* são baseados numa especificação denominada Exif (*Exchangeable image file format*), que é seguida por fabricantes de câmaras digitais, que determina a forma de gravar informações sobre as condições técnicas de captura e configurações da câmara junto ao arquivo da imagem. Essa especificação é um padrão criado pela *Japan Electronics and Information Technology Industries Association* (JEIDA). A especificação do Exif pode ser obtida no site “http://www.jeita.or.jp/cgi-bin/standard_e/list.cgi?cateid=1&subcateid=4”.

Embora a especificação do Exif seja bem ampla, neste trabalho foram utilizadas apenas as etiquetas de *metadados* para se obter a data e a hora de captura da imagem (“*DateTimeOriginal*”), e para se obter as dimensões da imagem (“*PixelXDimension*” e “*PixelYDimension*”). As classes do pacote “*ExifMetadata*” não foram criadas neste trabalho, tendo sido obtidas no site “<http://www.codeproject.com/KB/graphics/exiftagcol.aspx>” e utilizadas sem nenhuma modificação.

Figura 5.10: Classes do pacote “*ExifMetadata*”.

A Figura 5.11 apresenta um diagrama contendo as classes do pacote “*Lowe*” e as classe do pacote “*DeteccaoReconhecimento*” com suas relações.

As classes do pacote “*Lowe*” implementam os códigos do algoritmo SIFT, do algoritmo que faz a correspondência dos descritores dos pontos chave (*matching*) e do algoritmo RANSAC, todos convertidos originalmente da linguagem de programação C para a linguagem de programação C#, do paradigma estruturado para o paradigma orientado a objetos.

As classes do pacote “*ReconhecePlacas*” possuem métodos que implementam o algoritmo de detecção e reconhecimento das placas de velocidade descrito na seção 5.4 e em suas subseções

Figura 5.11: Classes do pacote “*Lowe*” e do pacote “*ReconhecePlacas*”.

A Figura 5.12 apresenta um diagrama de classes que compõem o pacote “*OpenCV*” e seus pacotes internos “*cv*”, “*cxcore*”, “*highgui*” e “*tools*”. Essas classes implementam métodos para acessar as funções da biblioteca de Visão Computacional OpenCV (OPENCV, 2010) a partir de seus arquivos .DLL⁹ (*Dynamic Link Library*) usando a linguagem de programação C#. Isso é feito, pois a biblioteca OpenCV implementa originalmente apenas recursos para serem usados diretamente nas linguagens de programação C, C++, Python e mais recentemente na linguagem Java, para o Sistema Operacional Android nos dispositivos móveis da Google. Essas classes não foram criadas neste trabalho de doutorado, tendo sido obtidas no site “<http://www.codeproject.com/KB/cs/>

⁹ Um arquivo .DLL possui o mesmo formato dos arquivos executáveis para o Sistema Operacional Windows, podendo conter códigos, dados e recursos. É um padrão criado pela Microsoft para o conceito de bibliotecas compartilhadas, onde mais de um programa pode fazer uso da mesma biblioteca.

Intel_OpenCV.aspx” . Algumas funções utilizadas foram modificadas, enquanto outras foram criadas para funcionar com a versão do OpenCV2.2 no Visual Studio 2010 da Microsoft.

Figura 5.12: Classes do pacote “*OpenCV*”.

A Figura 5.13 apresenta um diagrama da classe do pacote “*ReconheceCaracteres*” e das classes com suas relações do pacote interno “*RecCaracteresTransicaoPixels*”. As classes desses pacotes possuem métodos que implementam o algoritmo de Reconhecimento dos dígitos numéricos das placas de velocidade descrito na seção 5.5 e em suas subseções.

Figura 5.13: Classes do pacote “*ReconheceCaracteres*”.

No decorrer do texto, as classes que são utilizadas na detecção e reconhecimento das placas de sinalização são detalhadas com figuras mostrando seus atributos, construtores e métodos, bem como explicações das funções que elas exercem. São utilizados diagramas de sequência em alguns momentos das explicações, para demonstrar o funcionamento dos algoritmos desenvolvidos. Um diagrama de sequência é um diagrama de interação que dá ênfase à ordenação temporal de mensagens. Um diagrama de sequência mostra um conjunto de papéis e as mensagens enviadas e recebidas pelas instâncias que representam os papéis. São usados para ilustrar a visão dinâmica de um sistema (BOOCH et al., 2006).

5.3.2 Banco de Dados

Foi construído um banco de dados utilizando o Microsoft SQLServer 2008, para armazenar os dados provenientes dos levantamentos realizados para obtenção das imagens, informações de navegação e posicionamento, configurações das câmaras, dados dos *templates* com seus descritores SIFT, bem como os resultados do reconhecimento das placas de sinalização e do georreferenciamento. A seguir são apresentadas as tabelas, suas descrições e utilizações:

- “*projetos*”: nesta tabela são armazenados dados como descrição, data de início e fim de projetos. Um projeto pode ter um ou mais levantamentos;
- “*levantamentos*”: os dados de um levantamento compreendem os diretórios onde estão armazenadas as imagens capturadas com as câmaras acopladas ao SIMMOV (Sistema Móvel de Mapeamento), o arquivo de *log* do GPS/INS (seção 3.6), descrição, data e a qual projeto este levantamento pertence;
- “*configuracao_camaras*”: são armazenadas nesta tabela as configurações das câmaras, podendo elas ser fotográficas e filmadoras. Vale ressaltar que neste trabalho foram utilizadas apenas câmaras fotográficas. Os dados que podem ser armazenados são: descrição da câmara, tamanho do pixel (tpx , tpy), coordenadas do ponto principal em relação ao centro da imagem – eixo óptico da câmara em milímetros (x_0, y_0), distância focal em milímetros (f) e k_1 , distorção radial simétrica (mm^{-2});

- “*levantamentos_configuracao_camaras*”: registra as câmaras utilizadas no levantamento, juntamente com a posição no veículo. Para este trabalho são utilizadas duas câmaras frontais, uma à esquerda e outra à direita no veículo;
- “*levantamentos_imagens*”: os dados de cada par de imagens (esquerda e direita) são armazenados nesta tabela. Estes dados são obtidos a partir do *log* do arquivo do GPS/INS (seção 3.6) e são dados referentes aos valores da latitude, longitude e altitude, e aos ângulos ω (*ômega*), φ (*phi*), κ (*kapa*). As coordenadas de latitude, longitude e altitude são convertidas para o sistema de coordenadas UTM – *E (East)*, *N (North)*, *H (Height)* – e armazenadas respectivamente nos campos X_c , Y_c , e Z_c . Também nesta tabela é armazenada uma imagem (direita da sequência de imagens) reduzida, contendo as placas detectadas e reconhecidas demarcadas por um retângulo ao redor, e um campo para guardar uma informação quando nenhuma placa tenha sido detectada e reconhecida na imagem da direita;
- “*levantamentos_templates*”: são armazenados nesta tabela os campos X_t , Y_t , Z_t provenientes da determinação das coordenadas do terreno das placas de sinalização após detecção e reconhecimento das mesmas. Além destes campos, são armazenadas nesta tabela as coordenadas em *pixels* das placas detectadas e reconhecidas, tanto para a imagem da direita quanto para a imagem da esquerda. Uma imagem recortada da imagem da direita referente à placa detectada e reconhecida é também armazenada nesta tabela, assim como a quantidade de pontos chave presente na imagem da placa em correspondência com o *template* desta placa. Todos esses dados são exibidos em um relatório para se verificar o resultado da detecção, reconhecimento e determinação das coordenadas do terreno das placas de sinalização a partir dos dados obtidos de um levantamento. Também são usados para a construção de *ShapeFiles* (*layer* – camada para um mapa georreferenciado da área na qual foi realizado o levantamento) para serem visualizados na aplicação construída com a API do *MapWindow* (MAPWINDOW, 2010);
- “*templates*”: nesta tabela são armazenados a descrição do *template*, a imagem utilizada do *template*, a quantidade de pontos chave extraídos a partir do algoritmo SIFT (LOWE, 2004) e o número de descritores por ponto chave (é utilizado o valor 128, que corresponde à quantidade definida por Lowe (2004)). Também é armazenada nesta tabela uma informação indicando se o *template* é de

uma placa de velocidade, e caso for, é armazenado o limite de velocidade correspondente. Por último, é armazenado o número mínimo de pontos chave necessário para reconhecer a placa de sinalização. Como mencionado na introdução da seção 5.3 o valor mínimo definido é 3. Este valor inicialmente pode ser alterado de acordo com a complexidade de formas da placa, e após a realização de experimentos, pode ser modificado para um valor mínimo limite;

- “*features*”: esta tabela é utilizada para armazenar os dados referentes aos pontos chave de cada *template*. Esses dados são compostos pelas coordenadas do ponto chave (x, y), pela escala e orientação, e pelos 128 descritores extraídos com o uso do algoritmo SIFT (LOWE, 2004). Os descritores do ponto chave são armazenados em um único campo do tipo *varchar* (sequência de caracteres);
- “*pontos_homologos*”: para cada par de imagens de um levantamento, cujos dados são armazenados na tabela “*levantamentos_imagens*”, pontos que são coletados na função de Coleta Pontos (descrito no início da seção 5.3) são armazenados nesta tabela. Estes pontos coletados manualmente são utilizados para se fazer experimentos, e para se calcular posteriormente as coordenadas do terreno e realizar a calibração das câmaras.

A Figura 5.14 ilustra o Modelo Entidade Relacionamento do banco de dados modelado para o Sistema *AutoTSR*.

Figura 5.14: Modelo Entidade Relacionamento do banco de dados do Sistema *AutoTSR*.

5.3.3 Estrutura de Dados contendo os *templates* recuperados do banco de dados

Como já mencionado na seção 5.3.2, os *templates* (imagens das placas usadas e descritores de pontos chave extraídos pelo algoritmo SIFT) são armazenadas no banco de dados, para serem usados durante o estágio de detecção e reconhecimento das placas de

sinalização. Neste estágio, cada imagem de entrada é comparada (*matching*) com todos os *templates*, a fim de se encontrar uma correspondência entre os descritores de pontos chave, e para que não seja necessário recuperar os dados dos *templates* armazenados no banco toda vez que são utilizados, foi construída uma lista encadeada de objetos.

As classes “*ListaTemplate*” e “*NoListaTemplate*” (Figura 5.15) do pacote “*CamadaLogica*” (Figura 5.7) são utilizadas para se construir a lista encadeada de objetos contendo os dados de todos os *templates* recuperados via consulta SQL do Banco de Dados.

Figura 5.15: Classes “*ListaTemplate*” e “*NoListaTemplate*”.

Um esboço de um nodo referente ao objeto controlador da lista encadeada instanciado a partir da classe “*ListaTemplate*” é mostrado na Figura 5.16. Os campos “*inicio*”, “*fim*” e “*atual*” são respectivamente ponteiros para o início da lista, para o final da lista e para o nodo atual em utilização. O campo “*qtde*” armazena a quantidade de *templates* de placas de sinalização. A finalidade do valor deste campo é para ser utilizado no cálculo de estimação do progresso da atividade de detecção e reconhecimento automático das placas de sinalização a partir de uma sequência de pares de imagens. Para cada imagem da direita na sequência de imagens, seus descritores de pontos chave extraídos com o algoritmo SIFT (LOWE, 2004) serão comparados com os descritores de pontos chave dos *templates* (a seção 2.4 trata da correspondência dos descritores SIFT).

Figura 5.16: Nó controlador da lista encadeada.

A Figura 5.17 ilustra um esboço de um nó (objeto) da lista encadeada instanciada a partir da classe “*NoListaTemplate*”. Esse nó é responsável por armazenar em memória todas as informações referentes a um *template* de uma placa de sinalização. Os campos “*templ_codigo*”, “*templ_descricao*”, “*templ_placa_velocidade*”, “*templ_km_hora*” e “*templ_num_minimo_pts*” são campos para os dados recuperados da tabela “*templates*” do banco de dados (Figura 5.14). O campo “*f*” é um vetor de objetos da classe “*feature*” de tamanho “*n*” contendo os pontos chave com seus descritores do *template*. Estes dados foram recuperados via consulta SQL da tabela “*features*” do banco de dados (Figura 5.14).

Figura 5.17: Nó da lista, usado para armazenar as informações de um *template* de uma placa de sinalização.

5.4 Detecção e reconhecimento das placas de sinalização

Nesta seção e em suas subseções, são descritos o comportamento e os estágios da computação do algoritmo de detecção e reconhecimento das placas de sinalização. Inicialmente, na Figura 5.18, é apresentado um diagrama de sequência do algoritmo desenvolvido, para ilustrar a sequência do comportamento de maneira completa de todos os estágios da computação, porém global, procurando expressar da melhor forma seu comportamento.

Figura 5.18: Diagrama de Sequência do algoritmo de Detecção e Reconhecimento das placas.

O algoritmo representado no diagrama de sequência da Figura 5.18 contempla todos os estágios da computação descritos de forma resumida na sequência:

- construir a estrutura de dados contendo os dados dos *templates* recuperados do banco de dados (lista encadeada de objetos dos *templates* descrita na subseção 5.3.3);
- percorrer todos pares da sequência de imagens, e para cada imagem da direita executar os estágios seguintes;
- executar o algoritmo SIFT (LOWE, 2004) para extrair os descritores de pontos chave da imagem da direita da sequência. Considerada-se neste trabalho que pode haver a presença de placas de sinalização em ambos os lados das vias. Para tornar o algoritmo mais eficiente, a imagem é recortada em duas partes, de forma a haver uma parte com sobreposição na outra, precavendo a existência de uma placa de sinalização no centro horizontal desta imagem;
- percorrer toda a lista encadeada de objetos dos *templates*, e para cada *template* executar os estágios seguintes;
- montar uma árvore *k-d tree* (subseção 2.4.1) contendo os descritores dos pontos chave do *template*;
- executar o algoritmo para computar a correspondência (*matching*) entre os descritores dos pontos chave da imagem da direita da sequência com os descritores dos pontos chave do *template*;
- eliminar os pontos chave cujos descritores combinam entre si, mas que possuem orientações que não condizem com o modelo da placa (*template*). Considera-se neste trabalho que as placas de sinalização possam ter apenas uma pequena rotação (mais ou menos 10 graus) em relação à sua posição correta. Esse algoritmo representa a primeira parte do algoritmo desenvolvido neste trabalho para extração dos *inliers* após a computação do *matching* (estágio anterior). Esse algoritmo está descrito com detalhes na subseção 5.4.1;
- executar o algoritmo para encontrar uma região aproximada da placa de sinalização na imagem da direita, em função das coordenadas espaciais dos pontos chave do *template*, e após isto, os pontos chave que não pertencem a esta região são eliminados do conjunto de pontos prováveis a serem considerados *inliers*;
- após eliminar os *outliers* (pontos que não correspondem com o modelo do *template*), o retângulo da placa onde o objeto de interesse está inserido é

encontrado e a imagem desta região é recortada para posteriormente, ao ser reconhecida, ser armazenada no banco de dados juntamente com suas coordenadas em pixels na imagem;

- quando a placa de sinalização é detectada a partir de um *template*, este é utilizado para a detecção na imagem da esquerda da sequência, porém, apenas uma região aproximada é utilizada para calcular os descritores dos pontos chave a partir do SIFT e realizar a correspondência. Não há a necessidade de se procurar a placa na imagem inteira, já que a posição da placa na imagem da direita é conhecida na imagem da esquerda;
- quando se tratar de uma placa de sinalização de limite de velocidade, os dígitos referentes a esse limite, são segmentados da imagem da placa detectada usando um *threshold* (OTSU, 1979), e posteriormente é utilizado um algoritmo, descrito na seção 5.5, para o reconhecimento dos números;
- após todos esses estágios da computação, o *template* da placa de sinalização é reconhecido ou é considerado não reconhecido;
- a partir da detecção e reconhecimento da placa na imagem da direita e da esquerda, seus dados (recorte da imagem reconhecida na imagem da direita, coordenadas nas imagens esquerda e direita, quantidade de pontos que combinaram na atividade de correspondência) são armazenados em uma lista duplamente encadeada de objetos para, posteriormente serem armazenados no banco de dados. Os dados referentes às coordenadas em pixels das placas nas imagens serão utilizados posteriormente no estágio de georreferenciamento das placas (cálculo das coordenadas do terreno das placas de sinalização).

A lista duplamente encadeada de objetos construída para armazenar em memória os dados obtidos com a detecção e reconhecimento das placas de sinalização é esboçada na Figura 5.19. Nesta figura está uma representação de um nodo referente ao objeto controlador da lista duplamente encadeada, instanciado a partir da classe “*ListaImagens*” (Figura 5.22). Os campos “*inicio*” e “*fim*” são respectivamente ponteiros para o início da lista e para o final da lista.

Figura 5.19: Nodo controlador da lista duplamente encadeada.

A Figura 5.20 ilustra um esboço de um nodo (objeto) da lista encadeada instanciada a partir da classe “*NoImagens*” (Figura 5.22). Esse nodo é responsável por armazenar em memória as informações referentes à detecção e reconhecimento de uma placa de sinalização.

Figura 5.20: Nodo da lista, usado para armazenar os resultados da detecção e reconhecimento de uma placa de sinalização.

Na Figura 5.20 o campo “*codigo*” refere-se ao código dos dados do *template* que foi reconhecido e está armazenado no banco de dados, “*img*” é o recorte da imagem da direita da sequência de imagens obtidas a partir de levantamento onde foi encontrado o *template*, “*pts_chave*” é a quantidade de pontos chave que combinaram na atividade de correspondência, “*str_placa_det*” é uma *string* que contém a descrição da placa, “*pt_xy_esq*” é um vetor com as coordenadas da placa reconhecida na imagem da esquerda, “*pt_xy_dir*” é um vetor com as coordenadas da placa reconhecida na imagem da direita, “*flag_remove*” indica que a placa reconhecida deverá ser removida posteriormente da lista antes de seus dados serem gravados no banco de dados. Isto acontece, pois há a possibilidade de se cadastrar placas com o mesmo significado, mas que possuem formatos ligeiramente diferentes. Um exemplo pode ser visualizado na Figura 5.21 onde se pode observar que ambas as placas tem o mesmo significado para os motoristas, mas suas formas possuem algumas diferenças, principalmente no que se refere à altura das letras da palavra “PARE”. Quando isto acontece, o melhor resultado é permanecido, ou seja, aquele que tiver mais pontos chave (*inliers*) correspondentes. Por fim, os campos “*ant*” e “*prox*” guardam os endereços dos objetos vizinhos dos nodos da lista encadeada.

Figura 5.21: Placas de sinalização com o mesmo significado, porém com formas ligeiramente diferentes.

A Figura 5.22 apresenta o diagrama das classes “*ListaImagens*” e “*NoImagens*” pertencentes ao pacote “*CamadaLogica*” (Figura 5.7) desenvolvidas para a construção da lista duplamente encadeada que é responsável por armazenar os resultados do reconhecimento das placas de sinalização na sequência de pares de imagens.

Figura 5.22: Classes “*ListaImagens*” e “*NoImagens*”.

Conforme já mencionado, foi criado um algoritmo para a extração dos *inliers*, cujos descritores correspondem com os descritores dos pontos chave dos *templates*. Esse algoritmo foi criado em substituição ao algoritmo RANSAC (FISCHLER; BOLLES, 1981) devido ao fato de muitas placas de sinalização não apresentarem um número de pontos chave (*inliers*) suficiente para o reconhecimento, conduzindo a resultados enganosos ou ao não

reconhecimento. Na subseção 2.5.3, com o uso de alguns exemplos, são relatadas as dificuldades encontradas para o reconhecimento de placas de sinalização com a utilização do algoritmo RANSAC.

A subseção 5.4.1 a seguir, descreve detalhadamente o comportamento do algoritmo definido e implementado neste trabalho para a extração dos *inliers*.

5.4.1 Algoritmo de extração dos *inliers* para detecção e reconhecimento das placas de sinalização

Primeiramente são realizadas a computação do algoritmo SIFT (LOWE, 2004) e a computação do estágio da correspondência dos descritores dos pontos chave extraídos das imagens (*template* e imagem de entrada). Em seguida, é executado o algoritmo implementado e descrito nessa subseção para identificar os pontos chave considerados *inliers*, o que é feito em dois estágios: 1) comparando os ângulos de orientação dos gradientes locais; 2) buscando uma região onde se concentra a maior parte dos *inliers* e calculando a correlação entre as coordenadas espaciais dos pontos chave na imagem de entrada e no *template*. Os pontos chave que não forem classificados como *inliers*, são eliminados do conjunto de pontos (são classificados como *outliers*).

Para um melhor entendimento do algoritmo desenvolvido para extração dos *inliers*, algumas figuras serão mostradas a cada passo da computação. Inicialmente, a Figura 5.23 apresenta o resultado do estágio da computação da correspondência entre os descritores de pontos chave das duas imagens utilizadas no exemplo, em (a) tem-se um *template* de uma placa de sinalização de limite de velocidade e em (b) tem-se uma imagem de entrada.

Figura 5.23: Correspondência entre os pontos chave após avaliação dos descritores do SIFT.

No primeiro estágio do algoritmo de extração de *inliers*, é realizada a comparação das direções dos gradientes locais de cada ponto chave correspondente entre as duas imagens, calculadas pelo algoritmo SIFT (LOWE, 2004). Essas direções dos gradientes locais definem o ângulo de orientação de cada ponto chave e são obtidas a partir de picos (acima de um determinado limiar) no histograma com algoritmo SIFT (Equação 17 na subseção 2.3.3). Como o histograma pode ter até 36 regiões (*bins*) cobrindo os 360 graus de orientações, cada região equivale a 10 graus.

A computação para se verificar se os ângulos das orientações dos gradientes dos pontos chave do *template* são semelhantes aos ângulos das orientações dos gradientes dos pontos chave correspondentes na imagem de entrada, é realizada pelo método da classe “*DeteccaoReconhecimento*” (Figura 5.30) denominado “*elimina_pts_out_ori(...)*”. Apenas é considerado neste trabalho placas de sinalização com até 10 graus de diferença na rotação,

quando comparadas com os *templates*. Sendo assim, não é interesse o reconhecimento de placas em posições muito diferentes de onde deveriam estar originalmente, o que se fosse o caso, poderia ser feito rotacionando o modelo do *template* de 10 em 10 graus, até atingir os 360 graus de rotação e fazer as devidas comparações.

O processo proposto para a extração de *inliers* compara as direções dos gradientes locais θ , tomando n pontos chave K_i^j ($j=1,...,n$) na imagem de entrada e seus correspondentes K_i no *template*, e os considera candidatos a *inliers* se as orientações $K_i\theta$ e $K_i^j\theta$, respectivamente, satisfazem $|K_i\theta - K_i^j\theta| \leq 2\pi/36$. Caso contrário, eles são classificados imediatamente como *outliers* e são removidos do conjunto de candidatos a *inliers*.

Os pontos chave com as coordenadas (x, y) e direções dos gradientes locais θ , resultantes da correspondência dos descritores SIFT (*matching*) entre os pontos chave das duas imagens utilizadas no exemplo da Figura 5.23 são mostrados na Tabela 3. Na metade esquerda da tabela, estão os pontos chave do *template*, enquanto que na metade direita da tabela estão os pontos da imagem entrada. As linhas da tabela (todas) representam os pontos chave correspondentes obtidos após a computação da correspondência dos descritores SIFT (a seção 2.4 trata do algoritmo para encontrar a correspondência (*matching*) entre duas imagens, ou seja, encontrar pontos em comum entre as duas). As linhas escuras da tabela representam os pontos chave (*inliers*) que foram extraídos pelo algoritmo proposto, por meio da avaliação das orientações dos gradientes dos pontos chave entre as duas imagens. Pode-se observar nesta tabela, que um ponto chave de um *template* pode corresponder com mais de um ponto chave da imagem de entrada, como é o caso do ponto 3 do *template* que corresponde com os pontos 3 e 4 da imagem de entrada, com o ponto 4 que corresponde com os pontos 5 e 6, e com o ponto 13 que corresponde com os pontos de 15 a 44, respectivamente.

Tabela 3: Pontos chave com as coordenadas (x , y) e orientação (θ) após a computação da correspondência dos descritores SIFT (*matching*) e comparação das orientações θ .

Pontos chave do <i>template</i>				Pontos chave da imagem			
#	x	y	θ	#	x	y	θ
1	49	128	2,7962014	1	342	177	2,9036152
2	60	80	-0,5123180	2	346	162	-0,5471121
3	63	83	1,6073940	3	388	201	1,5912648
	63	83	1,6073940	4	491	182	1,6128330
4	66	106	-1,6473240	5	316	89	-2,5279552
	66	106	-1,6473240	6	507	76	1,3538668
5	68	56	0,1608393	7	349	153	0,0989794
6	74	94	-1,9623687	8	351	166	-1,9441054
7	78	125	3,0956446	9	352	177	3,1016327
8	78	125	1,6176384	10	352	177	1,5687513
9	82	101	-0,0278290	11	464	147	-1,7645825
10	96	106	1,3994543	12	358	170	1,3714497
11	96	106	-2,3956214	13	358	170	-2,4691109
12	102	133	-2,9755010	14	360	179	-3,0606964
13	111	196	-1,4028393	15	31	168	1,4599882
	111	196	-1,4028393	16	85	54	1,2602546
	111	196	-1,4028393	17	98	97	1,2946232
	111	196	-1,4028393	18	131	82	1,1247921
	111	196	-1,4028393	19	141	78	1,0994135
	111	196	-1,4028393	20	145	68	1,1958128
	111	196	-1,4028393	21	171	89	1,5213536
	111	196	-1,4028393	22	193	89	1,7193546
	111	196	-1,4028393	23	197	113	1,1769931
	111	196	-1,4028393	24	223	117	1,0714568
	111	196	-1,4028393	25	227	107	1,0544834
	111	196	-1,4028393	26	242	131	0,8634531
	111	196	-1,4028393	27	262	137	0,7685542
	111	196	-1,4028393	28	266	159	0,7793816
	111	196	-1,4028393	29	270	150	0,9450644
	111	196	-1,4028393	30	324	79	2,3476329
	111	196	-1,4028393	31	360	30	0,0088697
	111	196	-1,4028393	32	426	75	1,6051449
	111	196	-1,4028393	33	564	155	-3,1151710
	111	196	-1,4028393	34	647	360	-3,1055538
	111	196	-1,4028393	35	649	227	-3,0451288
	111	196	-1,4028393	36	650	267	-2,8369527
	111	196	-1,4028393	37	650	333	-3,1029577
	111	196	-1,4028393	38	651	228	-3,0337183
	111	196	-1,4028393	39	652	290	-3,0321450
	111	196	-1,4028393	40	653	285	-3,0501339
	111	196	-1,4028393	41	653	295	-3,0298980
	111	196	-1,4028393	42	655	363	3,1047102
	111	196	-1,4028393	43	658	368	2,9833593
	111	196	-1,4028393	44	659	376	-2,9785271
14	119	115	-1,9541516	45	365	174	-1,9615910
15	130	65	-0,0984942	46	370	157	-0,1244778
16	130	65	-3,0604850	47	370	157	-3,0601810
17	130	84	3,0427167	48	369	163	3,0120502
18	130	84	0,0655612	49	369	163	0,0813810
19	136	143	-3,0692313	50	352	183	-3,0881594
20	143	112	1,0830422	51	374	173	1,1741271
21	143	112	-1,0060023	52	374	173	-1,1233201
22	148	125	0,4344592	53	31	236	-3,0559504
23	161	133	2,6796017	54	379	181	2,6382332

A Figura 5.24 ilustra os pontos numerados correspondentes a todas as linhas da Tabela 3 (metade esquerda) representando o conjunto de pontos chave do *template* cujos descritores correspondem aos descritores de pontos chave da imagem de entrada.

Figura 5.24: Conjunto de pontos chave do *template* cujos descritores correspondem aos descritores da imagem de entrada.

Os pontos chave numerados da imagem de entrada, cujos descritores correspondem aos descritores dos pontos chave do *template*, e que ocupam a parte direita da Tabela 3, são mostrados na Figura 5.25.

Figura 5.25: Conjunto de pontos chave da imagem de entrada cujos descritores correspondem aos descritores dos pontos chave do *template*.

Após a execução do estágio da avaliação dos ângulos de orientação dos gradientes locais, obteve-se um conjunto com 19 pontos candidatos a *inliers* (representados pelas linhas escuras na Tabela 3). A Figura 5.26 ilustra os pontos chave que permaneceram após a eliminação dos pares de pontos chave com diferença das orientações do gradiente superior a 10 graus.

Figura 5.26: Pontos chave candidatos a *inliers* após a avaliação dos ângulos de orientação dos gradientes locais.

Neste momento, boa parte dos pontos chave do *template* estão ligados pelas linhas retas aos pontos chave da imagem de entrada na região do sinal de trânsito correspondente ao *template*, porém, ainda podem persistir alguns pontos chave dispersos pela imagem de entrada, que devem ser considerados *outliers*.

Para remover estes pontos, é executado o segundo estágio do algoritmo proposto, onde se busca a região com a mais alta concentração de pontos chave na imagem de entrada, levando em conta também as coordenadas espaciais dos pontos chave do *template*. É importante observar que, para ser caracterizada uma correspondência razoável entre um dos *templates* e uma imagem de entrada, é necessário e suficiente que seja encontrado uma quantidade mínima de *inliers*, que empiricamente foi definido como três *inliers* ou que pode ser definida ao se cadastrar o *template* no banco de dados (subseção 5.3.2). Para esse estágio,

é executada a função “*encontrar_regiao_aprox(...)*”, a função “*eliminar_pts_out_reg(...)*” e a função “*eliminar_pts_out_menor_maior(...)*” da classe “*DeteccaoReconhecimento*” (Figura 5.30), que após seu término é encontrada uma região aproximada da placa na imagem de entrada, sendo que os pontos chave que estão fora são classificados como *outliers*, mesmo com diferença de orientação de gradiente abaixo do limiar definido. O processo do segundo estágio do algoritmo proposto é demonstrado no Algoritmo 5 a seguir.

Algoritmo 5 – Busca de uma região aproximada de mais alta concentração de *inliers*

Objetivo: Encontrar uma região onde se concentra a maior parte dos *inliers* e calcular a correlação entre as coordenadas espaciais dos pontos chave na imagem de entrada e no *template*, e com isso, classificar os pontos fora dessa região como *outliers*.

Passos:

- 1: Ordene as coordenadas x e y dos pontos chave candidatos a *inliers* separadamente.
 - 2: Encontre a região mais densa, na horizontal (x) e na vertical (y), buscando a menor soma da diferença do conjunto de coordenadas. Esta região equivale à região de mais alta concentração de pontos chave na imagem de entrada.
 - 3: Calcular uma região aproximada da placa (*template*) na imagem de entrada a partir da região encontrada no passo anterior. Isto é calculado por meio da correlação entre as coordenadas espaciais dos pontos chave na imagem de entrada e no *template*.
 - 4: Eliminar do conjunto os pontos chave candidatos a *inliers* que estiverem fora da região calculada da placa na imagem de entrada.
-

Após a execução do Algoritmo 5, todos os pontos que estiverem fora da região aproximada da placa de sinalização são eliminados do conjunto de pontos candidatos a *inliers*. Nas imagens utilizadas no exemplo, apenas um ponto foi eliminado desse conjunto (classificado como *outlier*). A Figura 5.27 mostra esse ponto chave classificado com *outlier* (ponto de número 3 do *template* em (a) correspondente com ponto de número 4 da imagem de entrada em (b)).

Figura 5.27: Ponto chave candidato a *inlier* classificado como *outlier*.

Após a execução do algoritmo proposto, os pontos chave correspondentes classificados como *inliers* são mostrados na Figura 5.28.

Figura 5.28: Conjunto de pontos chave classificados como *inliers*.

Por fim, o sinal de trânsito é detectado, porém para sinais de limite de velocidade ainda falta reconhecer os dígitos numéricos para completar o processo, o que não ocorre com os demais sinais de trânsito, que, somente necessitam ter um número mínimo de *inliers* extraídos para serem considerados reconhecidos.

A necessidade de se reconhecer os dígitos numéricos das placas de velocidade se deve ao fato de que os números impressos nas placas geralmente apresentam poucos pontos chave correspondentes. Isso faz com que a informação “Km/h”, também impressa na placa, seja predominante para o algoritmo de reconhecimento, o que torna, por exemplo, as placas “40 Km/h” e “60 Km/h” muito parecidas para o algoritmo, ocasionando erros no reconhecimento, caso não se conheça os números referentes à velocidade. Na seção 5.5.1 é apresentado o algoritmo definido e implementado para realizar o reconhecimento dos dígitos numéricos referentes ao limite de velocidade, e na subseção 5.5.1.4 encontra-se um experimento para demonstrar esse algoritmo.

A Figura 5.29 ilustra o sinal de trânsito utilizado no exemplo, com o conjunto de pontos chave classificados como *inliers*. Em (a) tem-se o *template* utilizado como base para a detecção e reconhecimento, e em (b) tem-se uma imagem recortada e ampliada proveniente da placa reconhecida na imagem de entrada. Nesta figura, pode-se visualizar além dos pontos chave e suas correspondências (representado pelas linhas ligando cada ponto chave entre as duas imagens), setas saindo dos pontos representando as direções dos gradientes locais de cada ponto chave.

Figura 5.29: Pontos chave (*inliers*) com as direções dos gradientes representadas pelas setas saindo dos pontos.

A classe denominada “*DeteccaoReconhecimento*”, apresentada na Figura 5.30, pertence ao pacote “*ReconhecePlacas*”, e foi definida e implementada contendo métodos para realizar a computação da detecção e reconhecimento das placas de sinalização.

DeteccaoReconhecimento
-kdtree : Kdtree -xform : Xform -imgfeat : Imgfeatures -KDTREE_BBF_MAX_NN_CHKS : int = 200 -NN_SQ_DIST_RATIO_THR : double = 0.49
+DeteccaoReconhecimento() +executa_matching(kd_root : Kd_node, stacked : IplImage, img1_esqdir : IplImage, feat1_esqdir : feature [], n1_esqdir : int, feat2 : feature [], n2 : int, arrfeat1_esqdir : ArrayList, passo_a_passo : bool) : void +calcula_angulo(feature feat) : double +elimina_pts_out_ori(feat2 : feature [], n2 : int, SIFT_ORI_HIST_BINS : int, tl : int) : void +encontrar_regiao_aprox(feat2 : feature [], n2 : int, stacked : IplImage, pts_reg_apr : CvPoint2D64f [], width_t : int, height_t : int, width : int, height : int, passo_a_passo : int) : void +eliminar_pts_out_reg(feat2 : feature [], n2 : int, pts_reg_apr : CvPoint2D64f []) : void +elimina_maior_menor_fwdx(feat2 : feature [], n2 : int, lado : char) : void +elimina_maior_menor_fwdy(feat2 : feature [], n2 : int, lado : char) : void +eliminar_pts_out_menor_maior(feat2 : feature [], n2 : int, num_medio_pts : int) : bool

Figura 5.30: Classe “*DeteccaoReconhecimento*” pertencente ao pacote “*ReconhecePlacas*”.

5.4.2 Resultados com o algoritmo desenvolvido para extração dos *inliers*

Os experimentos apresentados nesta subseção para demonstração do algoritmo desenvolvido foram baseados nos mesmos exemplos que não tiveram resultados bons no reconhecimento de placas de sinalização apresentados na subseção 2.5.3 utilizando o algoritmo RANSAC. Assim, como já mencionado, quando a quantidade de pontos for pequena ou suas disposições muito próximas entre si, a atividade de reconhecimento será prejudicada.

A Figura 5.31 apresenta um resultado positivo na detecção e reconhecimento de uma placa de sinalização “Proibido virar à direita”, com o algoritmo implementado. Após a execução do algoritmo e eliminação dos *outliers*, a placa foi reconhecida com 5 pontos chave (*inliers*). Na Figura 5.32 a placa de sinalização “Siga em frente ou à direita” foi detectada e reconhecida com apenas 3 pontos chave. Por fim, na Figura 5.33 a placa “Estacionamento regulamentado” foi detectada e reconhecida com 6 pontos chave.

Figura 5.31: Placa “Proibido virar à direita” detectada e reconhecida com 5 pontos chave.

Figura 5.32: Placa “Siga em frente ou à direita” detectada e reconhecida com 3 pontos chave.

Figura 5.33: Placa “Estacionamento regulamentado” detectada e reconhecida com 6 pontos chave.

5.5 Reconhecimento dos dígitos numéricos das placas de limite de velocidade

Reconhecimento de caracteres (letra ou número) é outra importante área de reconhecimento de padrões, com grandes implicações em automação e tratamento de informação (THEODORIDIS; KOUTROUMBAS, 2009).

Nesta seção e em suas subseções é apresentado o algoritmo desenvolvido e utilizado neste trabalho de doutorado para reconhecer os dígitos numéricos das placas de sinalização de velocidade, como por exemplo: 10 km/h, 20 km/h, 40 km/h etc. Um artigo científico a respeito deste algoritmo foi publicado e apresentado no VII Workshop de Visão Computacional – WVC 2011 (SILVA et al., 2011).

5.5.1 Algoritmo para o reconhecimento de caracteres

O algoritmo desenvolvido para o reconhecimento de caracteres é baseado na classificação supervisionada dos caracteres. A classificação supervisionada busca obter regras que permitam prever, com precisão, a classe de cada amostra. Para isto, utilizam-se medições realizadas em caracteres, cujas classes são conhecidas. Quando a classificação é realizada de forma não supervisionada, esta é chamada de agrupamento e, neste caso, procura colocar os registros em classes, que apresentam grande similaridade entre seus integrantes e baixa similaridade entre os elementos de classes distintas.

Em Berkhin (2002) é mencionado que, embora tenha sido proposta uma grande variedade de algoritmos para realizar o agrupamento de dados, o número elevado de atributos, que deveria ser útil para uma melhor caracterização dos agrupamentos, geralmente compromete a execução da tarefa. Este problema é conhecido na literatura como a “maldição da alta dimensionalidade”, por causa da incapacidade que a maioria dos algoritmos de agrupamento apresenta para tratar dados com muitos atributos.

O algoritmo desenvolvido neste trabalho apresenta uma estratégia simples e rápida para modelar o comportamento dos caracteres, usando apenas as transições que ocorrem entre os níveis de pixels adjacentes que formam os caracteres.

5.5.1.1 Modelagem do Comportamento das Sequências de Pixels

A estratégia proposta neste algoritmo sugere descrever os caracteres, enquadrando-os em uma malha com dimensões definidas previamente e, em seguida, observar as transições entre os níveis de cinza (0 e 1 – imagens binárias) dos pixels adjacentes. Deste modo, uma imagem com dimensões $m \times n$ gera um vetor de transições (atributos) com $m \times n$ posições, sendo os pixels percorridos na sequência indicada na Figura 5.34. Assim, determina-se o comportamento das sequências de pixels em cada classe e, então, este conhecimento pode ser usado para classificar as poligonais de outros registros, para os quais não se conhece a classe.

Figura 5.34: Sequência de pixels na imagem usada.

Nesta estratégia, é construída uma lista com as transições permitidas para cada caractere, anotando as transições que ocorrem entre os atributos adjacentes em cada uma das classes ou, então, as transições que não ocorrem em cada classe. Como as imagens usadas são binárias, as transições possíveis entre dois pixels que formam um caractere são: 00, 01, 10 e 11. A Figura 5.35 apresenta um exemplo em que as poligonais da classe 1 possuem apenas a transição 10, não ocorrendo as transições 01, 11 e 00 entre os atributos a_1 e a_2 . Quanto à classe 2, nota-se que não ocorrem as transições 10 e 00. Assim, para cada classe são determinadas todas as transições que não ocorrem. Em seguida, a partir desta informação, os caracteres a serem reconhecidos são incluídos na classe que apresenta o menor número de inconsistências em relação às transições características anotadas em cada classe. A Figura 5.35 (a) exibe a visualização em coordenadas paralelas, do conjunto de dados apresentado em (b), destacando as transições.

Figura 5.35: a) Exibição dos registros da classe 1 em preto e da classe 2 em cinza; b) conjunto de dados contendo 4 registros com treze atributos em duas classes; c) Exibição dos registros da classe 1 e d) classe 2.

Em seguida, a classificação dos caracteres pode então ser conduzida usando as três regras de classificação propostas na Figura 5.36. A última regra (R_3) prevê um limiar definido pelo usuário para determinar quando o registro deve ser classificado como ruído.

- R_1 – Reconhecer o caractere na classe cujas transições não são violadas pelas transições do registro;
- R_2 – Havendo mais de uma classe que satisfaz esta condição, a classe que possuir mais restrições (classe mais restritiva) deverá ser a escolhida;
- R_3 – Quando o registro não atende todas as restrições de nenhuma classe, deverá ser inserido na classe menos violada, ou ser classificado como ruído em casos extremos.

Figura 5.36: Regras usadas para fazer a descrição dos caracteres.

As restrições das duas classes do conjunto de dados apresentado na Figura 5.35 (b) são apresentadas na Tabela 4. No caso, as duas classes possuem quantidades iguais de restrições, ou seja, não ocorrem as trinta transições na classe 1 e as trinta transições na classe 2, indicadas na Tabela 4.

Tabela 4: Transições que não ocorrem entre atributos adjacentes no conjunto de dados ilustrado na Figura 5.35 (b).

	Atributos adjacentes											
	a_1-a_2	a_2-a_3	a_3-a_4	a_4-a_5	a_5-a_6	a_6-a_7	a_7-a_8	a_8-a_9	a_9-a_{10}	$a_{10}-a_{11}$	$a_{11}-a_{12}$	$a_{12}-a_{13}$
classe 1	00	10	00	00	10	01	00	00	00	00	00	00
	01	11	10	01	11	11	10	01	01	10	01	01
	11			11			11	10			10	11
classe 2	00	00	00	00	00	10	01	01	00	00	00	00
	10	01	10	01	01	11	11	10	10	01	10	01
		11	11	10	11			11	11			

Embora esta abordagem tenha sido proposta para operar com dados binários (com cardinalidade igual a dois), dados de maior cardinalidade também podem ser processados de duas formas diferentes. Na primeira, os valores no conjunto de dados devem ser convertidos para binário, o que conduz a um aumento no número de atributos, melhorando o processo, pois aumenta o número de transições. A segunda possibilidade consiste em discretizar os valores dos atributos em número finito de níveis e então definir as transições para todos os níveis. Assim, dados dois atributos a_i e a_j , com cardinalidade c_i e c_j , respectivamente, o número de transições entre eles é dado pelo produto $c_i \cdot c_j$. O número total de transições T para um conjunto de dados contendo n atributos é dado pela soma de todas as transições entre os atributos adjacentes c_i e c_{i+1} indicada pela Equação (57).

$$T = \sum_{i=1}^{n-1} c_i \cdot c_{i+1} \quad (57)$$

5.5.1.2 Descrição das classes definidas e implementadas para o algoritmo de reconhecimento de caracteres

O diagrama de classes da Figura 5.37 apresenta as classes definidas e implementadas para o algoritmo de reconhecimento de caracteres. Nesse diagrama, a classe “*ClassificacaoCaracteres*” é a principal, pois é nela que estão implementados os métodos necessários para realizar tanto o estágio do treinamento do algoritmo quanto o estágio do reconhecimento dos dígitos numéricos das placas de limite de velocidade. As classes “*Classe*” e “*NaoOcorre*” são utilizadas para guardar os dados das restrições das classes do conjunto de dados, contemplando as transições que não ocorrem para cada caractere. A classe “*UtilPlaca*” possui métodos para recortar os dígitos das imagens, seja no estágio de treinamento, seja no estágio de reconhecimento do caractere.

Vale ressaltar que o algoritmo de reconhecimento de caracteres numéricos não apenas pode ser aplicado para o reconhecimento de caracteres numéricos, como é o caso das velocidades impressas nas placas de limite de velocidade, mas também para caracteres alfanuméricos. Pode-se visualizar na classe “*ClassificacaoCaracteres*” da Figura 5.37 métodos para se realizar o treinamento de conjunto de dados numéricos e alfanuméricos. As subseções 5.5.1.3 e 5.5.1.4 posteriores a esta, mostram respectivamente, os experimentos realizados com o algoritmo desenvolvido. Inicialmente com dois conjuntos de dados de caracteres numéricos (0 a 9), um constituído de caracteres manuscritos e outro constituído de caracteres escritos com fontes de letra diferentes, e na sequência, utilizando um conjunto de dados de caracteres numéricos, a partir de uma imagem contendo números de 0 a 9, com fonte de letra semelhante à utilizada na confecção das placas.

Figura 5.37: Diagrama de classes do algoritmo de Reconhecimento de Caracteres.

A Figura 5.38 apresenta um diagrama de sequência do estágio de treinamento do algoritmo de Reconhecimento de Caracteres. Esse diagrama apresenta a sequência dos passos realizados para se realizar o treinamento a partir do conjunto de dados (sequência dos pixels dos caracteres recortados da imagem de treinamento). No final do processo do treinamento, um arquivo texto (padrão ASCII) é construído, e nele estão os registros das classes (sequências de pixels) do conjunto de treinamento.

Inicialmente ao estágio da computação de detecção e reconhecimento das placas de sinalização, o classificador do algoritmo de reconhecimento de caracteres é inicializado, e neste momento, o arquivo texto contendo a sequência de pixels é aberto e as restrições (sequência de pixels que não ocorrem) são lidas do arquivo e são armazenadas na memória. Quando uma placa detectada for de limite de velocidade, o estágio de reconhecimento de caracteres é realizado para se obter os valores numéricos referentes ao limite de velocidade, e com isso comprovar se os números impressos na placa equivalem aos números referentes ao *template*.

Figura 5.38: Diagrama de Sequência do estágio de treinamento do algoritmo de Reconhecimento de Caracteres.

A Figura 5.39 apresenta um diagrama de sequência do algoritmo de Reconhecimento de Caracteres no estágio de reconhecimento dos dígitos numéricos das placas de limite de velocidade. Esse estágio da computação é executado durante a detecção e reconhecimento das placas de sinalização, quando se deseja comprovar o valor referente ao número da velocidade impresso na placa.

Figura 5.39: Diagrama de Sequência do algoritmo de Reconhecimento de Caracteres no estágio de reconhecimento dos dígitos numéricos.

5.5.1.3 Experimentos iniciais realizados com o algoritmo para reconhecimento de caracteres

Durante a fase de concepção do algoritmo para reconhecimento de caracteres, foram realizados dois experimentos.

No primeiro experimento foi utilizado um conjunto de dados de caracteres manuscritos (números: 0, 1, ..., 9) – conjunto de dados *binaryalphadigs* (UCI – *Machine Learning*) (FRANK; ASUNCION, 2010). Este conjunto de dados é formado por 390 registros, obtidos a partir das imagens de 39 caracteres, conforme ilustra a Figura 5.40.

Figura 5.40: Trinta e nove caracteres (números: 0, 1, ..., 9) escritos à mão.

Fonte: (FRANK; ASUNCION, 2010).

No experimento realizado, cada caractere foi digitalizado em uma grade de 16 x 20 pixels, usando apenas as cores preta e branca. Assim, cada caractere é representado neste conjunto através de 320 atributos que podem assumir os valores zero (preto) ou um (branco).

A Figura 5.41 ilustra as transições entre os 320 atributos. A visualização dos registros em algumas de suas classes revela um comportamento particular para cada classe, evidenciando diferentes conjuntos de restrições para cada classe.

Figura 5.41: Visualização obtida usando: a) todas as transições de todos os caracteres nas dez classes; b) apenas os registros da classe do caractere “0”; c) apenas os registros da classe do caractere “1”; d) apenas os registros da classe do caractere “2”; e) apenas os registros da classe do caractere “9”.

As quantidades de restrições nas classes zero até nove são, respectivamente, 228, 315, 155, 197, 117, 188, 196, 277, 88 e 272. Assim, a classe dos caracteres “1” é a mais exigente (possui um número maior de restrições), enquanto que a classe dos caracteres “8”

possui o menor número de restrições. Como a cardinalidade de todos os atributos é igual a dois, e o conjunto tem 320 atributos, o número total de transições entre os atributos do conjunto, obtido usando a Equação 57, é dado por 1.276.

A matriz de confusão mostrada na Figura 5.42 apresenta a eficácia da estratégia para classificar os caracteres de acordo com as regras estabelecidas na subseção 5.5.1.1, ou seja, o registro na classe cujas transições não são violadas pelas transições do registro. Nos casos em que se encontra mais de uma classe satisfazendo esta condição, o registro é inserido na classe que possui mais restrições.

	0	1	2	3	4	5	6	7	8	9
0	39	0	0	0	0	0	0	0	0	0
1	0	39	0	0	0	0	0	0	0	0
2	0	0	39	0	0	0	0	0	0	0
3	0	0	0	39	0	0	0	0	0	0
4	0	0	0	0	39	0	0	0	0	0
5	0	0	0	1	0	38	0	0	0	0
6	0	0	0	0	0	0	39	0	0	0
7	0	0	0	0	0	0	0	39	0	0
8	0	0	0	0	0	0	0	0	39	0
9	0	0	0	0	0	0	0	0	0	39

Figura 5.42: Matriz de confusão obtida com a classificação usando as transições entre os atributos adjacentes.

Apesar da dificuldade com este conjunto de caracteres (Figura 5.40), a matriz de confusão mostra que a maior parte dos registros foi colocada em suas devidas classes, exceto um registro da classe 5 (caractere “5”), que foi colocado na classe 3 (caractere “3”). Trata-se do 23º caractere “5” na Figura 5.40, que atende todas as restrições das classes 3 e 5 e, pela regra R_2 (Figura 5.36), foi reconhecido como da classe 3, porque ela possui um número maior de restrições que a classe 5.

Para resolver este problema e melhorar a qualidade das classificações, duas alternativas podem ser usadas: 1) aumentar as dimensões da malha usada para representar os caracteres; 2) Anotar as transições entre três pixels no lugar das transições entre dois pixels, propostas inicialmente. Assim, as transições a serem verificadas seriam: 000, 001, 010, 011, 100, 101, 110 e 111. Com estas duas estratégias, aumenta-se a quantidade de restrições usadas e, conseqüentemente, aumentam-se as chances de se classificar os caracteres corretamente. Os resultados a partir dessas estratégias mostram que o algoritmo consegue identificar caracteres em um tempo relativamente baixo, mesmo com o custo computacional maior em virtude do aumento do número de pixels e de verificações de transições. Os resultados mostram que a

identificação ocorre em um tempo de aproximadamente 0,062 segundos para reconhecer todos os 390 caracteres da Figura 5.40, em uma máquina com processador Intel Core i3 M330 2.13 GHz, o que é bastante razoável para tarefas que precisam realizar identificações de caracteres em tempo real (SILVA et al., 2011).

No segundo experimento realizado foi usado o conjunto de imagens de caracteres exibido na Figura 5.43. Neste experimento, o treinamento foi realizado usando apenas os caracteres em (a), enquanto que o teste foi feito usando apenas os caracteres em (b), de fontes diferentes.

Figura 5.43: a) Conjunto de caracteres usados no treinamento do classificador (30 fontes diferentes); b) Conjunto usado no teste (outras 10 fontes diferentes).

Os resultados destes dois experimentos bem como a análise de desempenho do algoritmo são apresentados em Silva et al. (2011).

5.5.1.4 Reconhecimento dos dígitos numéricos das placas de velocidade

Para utilizar o algoritmo de reconhecimento de caracteres no reconhecimento dos dígitos numéricos das placas de velocidade, foi utilizado um conjunto de dados de caracteres

numéricos (0 a 9) com fonte mais semelhante possível da utilizada na confecção das placas. Esse conjunto de dados utilizado no treinamento do algoritmo para a realização dos experimentos é formado por 10 registros, obtidos pelos caracteres numéricos de 0 a 9, conforme ilustra a Figura 5.44.

Figura 5.44: Conjunto de caracteres numéricos usados no treinamento do classificador para o reconhecimento dos dígitos numéricos das placas de velocidade.

Nesse experimento, para a formação do conjunto de dados de treinamento para o algoritmo, cada caractere numérico foi segmentado da imagem da Figura 5.44 em uma grade de tamanho variável capaz de enquadrá-los, isto porque os números não apresentam tamanhos uniformes. Pode-se observar na Figura 5.44 que o número “1” é menor em largura (número de pixels) que todos os outros números. Cada caractere foi redimensionado para uma malha de 64 x 80 pixels, usando apenas a cor branca (1) para o fundo da imagem, e preta (0) para os caracteres. Cada caractere é representado neste conjunto por meio de 5120 atributos, que representam o comportamento das sequências de pixels em cada classe. Na Figura 5.45 podem ser visualizados os trechos dos registros das classes (sequências de pixels) do conjunto de treinamento, onde é possível observar um comportamento particular para cada classe, evidenciando a possibilidade de formação de diferentes conjuntos de restrições. Foram anotadas as transições entre três pixels, de maneira a se ter uma maior quantidade de restrições usadas e, dessa forma, melhorar as possibilidades de se classificar os caracteres corretamente.

[illegible]

Figura 5.45: Parte do conjunto de 5120 atributos, representando a sequência de pixels para cada classe.

Para um exemplo da aplicação do algoritmo de reconhecimento de caracteres, a Figura 5.46 apresenta uma imagem contendo uma placa de sinalização de velocidade

detectada em (a), e o *template* utilizado em (b). Os pontos chave em ambas as imagens (*template* e imagem de entrada), foram identificados a partir da execução do algoritmo de detecção e reconhecimento descrito na seção 5.4 e em suas subseções.

Figura 5.46: Exemplo de uma imagem contendo uma placa de sinalização de velocidade detectada e reconhecida.

Quando a placa de trânsito for de limite de velocidade, faz-se necessário segmentar os dígitos para realizar o reconhecimento dos caracteres numéricos, onde é utilizado um *threshold* (OTSU, 1979) para a fase de pré-processamento. Conforme apresentando pela Figura 5.47, a placa é recortada da imagem e o dígito “4” é contornado por um retângulo, ficando também visíveis os pontos chave pertencentes ao conjunto de *inliers*.

Figura 5.47: Exemplo de uma imagem segmentada de uma placa de sinalização de velocidade detectada e reconhecida.

5.6 Determinação das coordenadas do terreno das placas de sinalização

Após a realização dos estágios de detecção e reconhecimento das placas de sinalização, a posição do centro de massa das regiões das placas reconhecidas nas imagens da direita e nas correspondentes imagens da esquerda, juntamente com a posição e orientação das câmaras, são utilizadas para realizar o georreferenciamento. São calculadas as coordenadas da placa no terreno usando a interseção fotogramétrica (seção 3.7) para posterior apresentação em uma base cartográfica. A Figura 5.48 apresenta um diagrama de sequência para ilustrar o estágio de georreferenciamento das placas de sinalização.

Figura 5.48: Diagrama de Sequência do estágio de Georreferenciamento das placas de sinalização.

Capítulo 6

Resultados e Discussões

Este capítulo apresenta os resultados de um experimento realizado usando 60 pares de imagens com 3.1 *Megapixels* de resolução (2144x1424 pixels), obtidas por meio de levantamento com o SIMMOV (Sistema Móvel de Mapeamento) (seção 5.2) em movimento, em condições normais de tráfego. Essas imagens apresentam cenas complexas contendo placas de sinalização e também diversos outros objetos, tais como: árvores, pessoas, veículos, fachadas de edificações, painéis de propaganda etc. Neste experimento, foi aplicada a metodologia proposta, assim como também o algoritmo RANSAC (FISCHLER; BOLLES, 1981) (seção 2.5) para a extração do *inliers*, no estágio de detecção e reconhecimento das placas de sinalização. Além das imagens, no experimento aplicando a metodologia proposta, é utilizado o arquivo de *log* provido pelo GPS/INS (seção 3.6) no estágio da computação do georreferenciamento, contendo dados referentes ao posicionamento e atitude (ângulos de orientação) do veículo que fez o levantamento para coleta de dados.

Na Figura 6.1, estão relacionados os sinais de trânsito (*templates*) cadastrados no Sistema *AutoTSR*: (a) Velocidade máxima permitida – 20 Km/h; (b) Velocidade máxima permitida – 40 Km/h; (c) Estacionamento regulamentado; (d) Siga em frente ou à direita; (e) Proibido estacionar; (f) Proibido parar e estacionar; (g) Proibido retornar à esquerda; (h) Proibido virar à direita; (i) Proibido virar à esquerda; (j) Passagem sinalizada de pedestres; (k) Interseção em círculo; (l) Semáforo à frente; (m) Altura máxima permitida 5,4 m; (n) Parada obrigatória; (o) Passagem sinalizada de escolares; (p) Travessia de pedestres; (q) Estacionamento regulamentado – Área de emergência. Nesta figura, é possível observar os diferentes tamanhos das imagens de sinais de trânsito utilizadas como *template*, sendo que uma parte dessas imagens foi recortada das imagens adquiridas por meio do levantamento realizado com o SIMMOV e utilizado no experimento, e a outra parte obtida a partir de outro conjunto de imagens.

Figura 6.1: Placas de sinalização de trânsito usadas como *templates*.

A Tabela 5 apresenta os resultados obtidos aplicando a metodologia proposta neste trabalho e também aplicando o algoritmo RANSAC, para detecção e reconhecimento das placas de sinalização, sendo a primeira coluna referente aos índices alfabéticos existentes na Figura 6.1.

Tabela 5: Resultados no reconhecimento das placas de sinalização.

<i>Templates</i>	Metodologia deste trabalho			Algoritmo RANSAC		
	Acertos	Erros	Não detectadas	Acertos	Erros	Não detectadas
(a) Velocidade máxima permitida – 20 Km/h	4	0	0	1	0	3
(b) Velocidade máxima permitida – 40 Km/h	3	0	0	3	0	0
(c) Estacionamento regulamentado	14	0	4	6	0	12
(d) Siga em frente ou à direita	0	0	1	0	0	1
(e) Proibido estacionar	4	0	0	0	0	4
(f) Proibido parar e estacionar	5	0	0	0	0	5
(g) Proibido retornar à esquerda	7	0	3	0	0	10
(h) Proibido virar à direita	8	0	0	1	0	7
(i) Proibido virar à esquerda	4	0	3	1	0	6
(j) Passagem sinalizada de pedestres	1	0	0	1	0	0
(k) Interseção em círculo	2	0	0	1	0	1
(l) Semáforo à frente	7	0	0	0	0	7
(m) Altura máxima permitida 5,4 m	1	0	0	1	0	0
(n) Parada obrigatória	3	0	1	3	0	1
(o) Passagem sinalizada de escolares	2	0	0	2	0	0
(p) Travessia de pedestres	2	0	0	2	0	0
(q) Estacionamento regulamentado – Área de emergência	6	6(*)	0	6	6(*)	0
Total	73	6	12	28	6	57

Nos 60 pares de imagens usadas neste experimento existem 91 placas de sinalização. Conforme apresentado na Tabela 5, usando a metodologia proposta, 73 placas foram reconhecidas corretamente, obtendo-se uma taxa de acerto de 80,22%. Ocorreram apenas seis erros de identificação (6,59% de erros), entretanto, todos eles contêm o sinal “Estacionamento regulamentado”, conforme apresentados na Figura 6.2. Se esses erros forem considerados reconhecimentos válidos, devido ao fato dos sinais (c) e (q) conterem semelhanças, a taxa de acerto sobe para 86,81%. Das 91 placas, 12 placas não foram detectadas (13,19%). Já usando o algoritmo RANSAC (seção 2.5) para a extração dos *inliers*, na fase de detecção e reconhecimento, apenas 28 placas foram reconhecidas corretamente, representando 30,77% de taxa de acerto. Os mesmos seis erros de identificação também ocorreram com o uso do RANSAC, e das 91 placas, 57 placas não foram detectadas (62,64%). O experimento foi realizado em uma máquina com processador Intel Core i3 M330 2.13 GHz (memória RAM de

4 *Gigabytes*), onde o tempo médio de processamento para cada par de imagens é de 110 segundos, isso para cada abordagem (metodologia deste trabalho e algoritmo RANSAC).

Figura 6.2: Erro no reconhecimento das placas contendo o sinal “Estacionamento regulamentado”.

A Figura 6.3 apresenta alguns exemplos de *template* com os seus respectivos recortes na imagem de entrada, em que o reconhecimento, da metodologia proposta, foi realizado com sucesso, apesar de alguns fatores complicadores, tais como: obstáculos à frente da placa, como galhos e folhas; placa enferrujada ou danificada; placa inclinada etc.

Figura 6.3: Exemplos de reconhecimento apesar de fatores complicadores utilizando a metodologia proposta.

6.1 Avaliação dos Resultados

Foi realizada uma avaliação qualitativa para determinar se as placas são detectadas e reconhecidas de maneira correta e quais os problemas práticos ocorridos. Foi estabelecida uma área na cidade de Presidente Prudente, interior de São Paulo, para um levantamento experimental. Foi percorrido um total de 6,5 km com o SIMMOV em algumas das principais vias da cidade para coleta de dados (pares de imagens, dados de navegação e posicionamento). As atividades de detecção, reconhecimento e georreferenciamento a partir dos dados coletados no levantamento foram realizadas de forma *off-line*, porém automática, utilizando o Sistema AutoTSR. A partir do número de placas reconhecidas obtido com os resultados do experimento realizado, e das quantidades contadas manualmente nas imagens, foi construída a Tabela 5. Dessa maneira, tem-se a proporção de detecção correta das placas.

O mapa da área na qual foi realizado o levantamento contendo as placas detectadas e reconhecidas é apresentado no Apêndice A. Os resultados produzidos pelo Sistema AutoTSR na forma de um relatório, referentes à detecção, reconhecimento e cálculo das coordenadas das placas no terreno, são apresentados no Apêndice B. No Apêndice C são mostradas as placas não detectadas pela metodologia proposta, aplicada nas imagens de entrada, as quais foram obtidas por meio de levantamento.

6.2 Discussões

Na etapa da construção do banco de dados, os *templates* desejados para reconhecimento são cadastrados em uma funcionalidade do Sistema *AutoTSR* desenvolvido (seção 5.3). Nela, o algoritmo SIFT (LOWE, 2004) é aplicado em cada imagem de *template*, e seus descritores, juntamente com as informações do sinal de trânsito, são armazenados no banco de dados. As imagens dos *templates* são recortes das imagens provenientes dos levantamentos ou mesmo de outras imagens contendo os sinais de trânsito desejados, e independem de tamanho padrão, não necessitando ser redimensionados em nenhum momento. Isso não acontece em trabalhos quando se usa redes neurais, como no trabalho de Martinović et al. (2001), em que as imagens dos sinais de tráfegos necessitam ser normalizadas, tanto na fase de treinamento quanto na fase de classificação.

Outra característica importante da metodologia proposta neste trabalho de doutorado baseada no algoritmo SIFT, é que não há a necessidade de se ter um número grande de imagens de sinais de trânsito (*templates*) para treinar o sistema, como acontece com abordagens baseadas em redes neurais, segundo Farag e Abdel-Hakim (2004).

Os resultados da metodologia proposta apresentada na Tabela 5 foram significantes, mostrando a relevância do trabalho na detecção e reconhecimento das placas de sinalização de trânsito, mesmo utilizando imagens de entrada com alta resolução, capturadas em um ambiente real, sem nenhum controle, repleta de objetos, onde a placa muitas vezes não se destaca perante aos outros objetos. Observa-se também que os erros de identificação apresentados na Figura 6.2, ocorreram em virtude da semelhança em parte dos sinais detectados e reconhecidos com o sinal “Estacionamento regulamentado”. Desconsiderando-se esses problemas, não ocorreram outros erros no reconhecimento. No entanto, 12 placas não foram detectadas, o que poderia ser minimizado adicionando ao banco de dados outros *templates* referentes a esses sinais provenientes de imagens diferentes. Na Figura 6.1 pode-se observar que para alguns sinais, o cadastro de mais de um *template* já havia sido feito, como pode ser observado, nessa figura, nos *templates* (c), (d), (g), (h), (i), (k), (l) e (n). Isto pode melhorar a precisão no reconhecimento, porém pode aumentar o tempo de processamento, pois haverá mais *templates* a serem buscados nas imagens de entrada.

Os tempos de processamento são relativamente altos (tempo médio de processamento para cada par de imagens é de 110 segundos) em função do tamanho das imagens de entrada utilizadas e do número de pontos chave que são identificados em virtude da riqueza de detalhes das cenas. Essa informação vem comprovar os relatos de Ihara et al. (2009), que afirma que o custo computacional da extração e correspondência dos descritores SIFT é alto, o que torna o processamento em tempo real uma tarefa difícil de ser alcançada.

Capítulo 7

Conclusões e Sugestões

Este trabalho apresentou a proposta de uma metodologia para realizar a detecção, o reconhecimento e o georreferenciamento de placas de sinalização de trânsito de forma automática, usando pares de imagens (esquerda e direita) obtidas com um Sistema Móvel de Mapeamento (SIMMOV) em vias de uma área urbana em condições normais de tráfego. As imagens capturadas, além de conter os objetos de interesse, apresentam uma grande quantidade de detalhes em cenas com fundo complexo, com grande variação de iluminação, uma vez que não foram capturadas em ambientes controlados. Além da sequência dos pares de imagens capturadas através de câmaras fotográficas digitais instaladas no teto do veículo, para a computação da determinação das posições de feições dos sinais de trânsito no terreno durante a atividade do georreferenciamento, foram utilizados dados de navegação (Sistema Inercial – seção 3.5) e posicionamento (GPS – seção 3.4).

O Sistema *AutoTSR* (*Automatic Traffic Sign Recognition*) desenvolvido nesse trabalho, que contempla todos os estágios da computação para a metodologia proposta, foi idealizado para trabalhar de forma automática, não sendo necessária qualquer intervenção do usuário durante a computação. Há, porém, a necessidade do usuário fornecer parâmetros para que o sistema possa processar os dados coletados pelo SIMMOV a partir de levantamentos em campo. Esses parâmetros são os de orientação interior de cada câmara (coordenadas do ponto principal em relação ao centro da imagem, distância focal e distorção radial simétrica) que são determinados por meio de calibração (a seção 3.7 apresenta informações a respeito) durante uma fase inicial à atividade de detecção e reconhecimento. Após a atividade de detecção e reconhecimento ter sido realizada, para a atividade de georreferenciamento, faz-se necessário o fornecimento dos parâmetros de orientação exterior (posição dos centros perspectivos de cada câmara, calculados em função dos dados de posicionamento e ângulos de navegação).

O Sistema implementado baseou-se no algoritmo SIFT (LOWE, 2004) para a extração dos descritores de pontos chave das placas de sinalização. O uso do algoritmo SIFT, que consegue fornecer os pontos de interesse nas imagens, com uma descrição muito

satisfatória, e que favorece a correspondência entre os pontos de interesse de duas imagens, é um dos diferenciais dessa metodologia, contribuindo decisivamente para a qualidade dos resultados. Assim como afirmado e constatado por Mikolajczyk e Schmid (2005), o SIFT apresenta um desempenho superior em qualidade da descrição dos pontos chave em relação a alguns algoritmos com a mesma finalidade. O algoritmo SIFT foi utilizado em duas etapas no trabalho, durante a criação de um banco de dados contendo os *templates* das placas de sinalização desejadas para o reconhecimento e posteriormente durante o estágio da detecção e reconhecimento das placas de sinalização. É durante esse estágio da computação que é efetuada a correspondência entre duas imagens (a imagem de entrada e outra contendo o sinal de trânsito desejado, representando o *template* de busca), baseando-se nos descritores de pontos chave obtidos com o algoritmo SIFT.

O processo de eliminação dos pontos chave que não condizem com o *template*, resultando apenas nos pontos chave considerados *inliers*, que é realizado durante o estágio de reconhecimento dos sinais de trânsito, representa um desafio que motivou uma das contribuições científicas deste trabalho. Este algoritmo para extração dos *inliers* (seção 5.4.1) foi definido e implementado na tentativa de maximizar os resultados no reconhecimento de objetos que apresentam poucos detalhes, como acontece com boa parte das placas contendo sinais de trânsito, e minimizar os falsos positivos. Com poucos detalhes, muitos dos sinais de trânsito apresentam um número pequeno de correspondências, o que pode acarretar em muitas falhas na detecção e no reconhecimento. Na literatura encontra-se com muita frequência o algoritmo RANSAC (FISCHLER; BOLLES, 1981) sendo empregado em diversos trabalhos de reconhecimento de objetos e confecção de imagens panorâmicas, que fazem uso da correspondência de pontos chave. Porém, este algoritmo não apresentou bons resultados com os dados trabalhados. No trabalho de Baró et al. (2009a) essa experiência pode ser comprovada, onde os autores relatam ter problemas com o número pequeno de correspondências usando o algoritmo RANSAC, onde a maior parte das falhas no reconhecimento é produzida por objetos de tamanhos pequenos. Os autores Höferlin e Heidemann (2010) também afirmam a dificuldade no reconhecimento devido à correspondência pobre de descritores, podendo até mesmo, os sinais serem capturados em baixa resolução.

O algoritmo para extração dos *inliers* definido e implementado neste trabalho necessita de um número mínimo de três pontos chave, definido empiricamente, o que pode ser aumentado para se ter um reconhecimento mais acurado quando o sinal de trânsito apresenta um número de detalhes grande. Em contrapartida, um aumento demasiado do número mínimo

de pontos pode descartar resultados de reconhecimentos corretos de objetos de interesse, por apresentarem um menor número de pontos em relação ao definido.

Quando o sinal de trânsito for de limite de velocidade, foi constatada a necessidade de se reconhecer os dígitos numéricos que indicam o valor da velocidade máxima da placa. Embora existam muitos trabalhos na literatura para o reconhecimento de caracteres, neste trabalho, foi definido e implementado um algoritmo (SILVA et al., 2011) para ser aplicado durante o estágio de reconhecimento dos números referentes à velocidade. Esse algoritmo apresenta uma abordagem diferente dos trabalhos encontrados na literatura, que além de poder ser aplicado para atividades que demandam alto desempenho, constitui também uma contribuição científica para esta tese.

Foi realizado um experimento com 60 pares de imagens, contendo 91 placas de sinalização. Desse total, 73 placas foram reconhecidas corretamente, obtendo-se uma taxa de acerto de 80,22% no reconhecimento. Ocorreram seis erros de identificação (6,59%), e 12 placas não foram detectadas (13,19%).

Como resultado de todos os algoritmos utilizados e definidos e implementados, a metodologia proposta neste trabalho de doutorado pode ser aplicada não apenas para a detecção, reconhecimento e georreferenciamento de sinais de trânsito, mas também para diversos outros objetos, como fachadas de edifícios comerciais, placas e painéis de propaganda etc.

7.1 Sugestões de Trabalhos Futuros

É de grande interesse que a detecção e reconhecimento das placas de sinalização de trânsito apresentem um melhor desempenho e uma melhor precisão nos resultados. Mas para que isso ocorra, se ainda for considerada a captura de imagens em ambientes não controlados, muitas vezes com deficiências de iluminação, ou com iluminação excessiva em virtude da reflexão da luz do Sol, um novo enfoque de pré-processamento das imagens de entrada deverá ser desenvolvido para equilibrar não apenas as condições de iluminação, mas também diminuir a complexidade das cenas. A eliminação de alguns objetos que sejam ricos em detalhes, como é o caso das árvores e de algumas fachadas, melhoraria o desempenho do algoritmo SIFT. Uma redução do número de detalhes poderia significar um aumento de desempenho, tanto na identificação e descrição dos pontos chave, que seriam em menor

quantidade, como também no processo de correspondência. A redução das dimensões espaciais das imagens de entrada não deve ser descartada, mas para que isso possa ocorrer, os sinais de trânsito devem aparecer em destaque nessas imagens, apresentando um tamanho considerável, caso contrário, se já se apresentassem em tamanhos pequenos, poderiam quase que desaparecer.

A investigação de uma alternativa mais eficiente em termos de tempo de processamento, para a extração de pontos chave, seria de grande valia para aprimorar a metodologia proposta e apresentada nesta tese. Na literatura podem ser encontrados alguns trabalhos com essa preocupação, a de encontrar uma metodologia para se ter um descritor de pontos chave eficiente com baixo tempo de processamento, como o trabalho proposto por Azad et al. (2009). Nesse trabalho, os autores fazem a combinação do detector de cantos de Harris (HARRIS; STEPHENS, 1988) com o algoritmo SIFT (LOWE, 2004), que determina descritores com alta repetibilidade e boas propriedades de correspondência com um tempo de processamento menor em relação ao da execução do algoritmo SIFT original. Segundo os autores, o SIFT consome um tempo de 500 a 600 milissegundos para calcular os descritores SIFT de uma imagem com resolução de 640x480 pixels em uma CPU de 3 GHz, enquanto que usando a abordagem proposta por eles, a combinação do detector de cantos de Harris com o algoritmo SIFT, consome um tempo de processamento de apenas 20 milissegundos. Para esse ganho de tempo, os autores substituíram o estágio mais custoso computacionalmente do algoritmo SIFT, a detecção de pontos chave estáveis no espaço escala, pelo detector de cantos de Harris. As experimentações consistiriam em utilizar essa abordagem proposta por Azad et al. (2009) em substituição ao algoritmo SIFT original e verificar se os pontos chave extraídos seriam adequados para a detecção e reconhecimento de sinais de trânsito.

Um trabalho de grande interesse, também, seria o aprimoramento do algoritmo de extração de *inliers*, para que ele seja invariante à rotação, diferente da forma a qual foi concebido para resolver o problema do reconhecimento de placas de sinalização de trânsito, onde se espera encontrar as placas em posições razoavelmente corretas. Esse aprimoramento poderia significar a aplicação do algoritmo no reconhecimento de diversos tipos de objetos em várias posições.

Uma avaliação da qualidade na determinação das coordenadas das placas poderia ser feita utilizando uma quantidade (n) de placas, escolhidas de forma arbitrária em um percurso e coletar as coordenadas *E* (*East*) e *N* (*North*) das mesmas, com o uso de um GPS. Essas coordenadas seriam consideradas como controle e seria calculada a raiz quadrada do erro

médio quadrático (RQEMQ), entre as coordenadas E e N desses pontos de controle e as coordenadas E e N computadas pelo Sistema *AutoTSR* de forma automática.

7.2 Artigos Publicados e Submetidos

7.2.1 Artigos Publicados

Silva, F. A.; Artero, A. O.; Paiva, M. S. V.; Barbosa, R. L. Um Algoritmo Rápido para o Reconhecimento de Caracteres. In: VII WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2011). **Anais...** Curitiba: UFPR, 2011. p. 149-154.

Silva, F. A.; Artero, A. O.; Paiva, M. S. V.; Barbosa, R. L. Uma Metodologia para Detectar e Reconhecer Placas de Sinalização de Trânsito. In: VIII WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2012). **Anais...** Goiás: UFG, 2012.

Silva, F. A.; Hiraga, A. K.; Artero, A. O.; Paiva, M. S. V.; Barbosa, R. L. Um novo Algoritmo para a Construção de Imagens Panorâmicas usando os Algoritmos SIFT e RANSAC. In: III SIMPÓSIO BRASILEIRO DE GEOMÁTICA. **Anais...** Presidente Prudente: FCT/Unesp, 2012. p. 151-156.

7.2.2 Artigos Aceitos para Publicação

Silva, F. A.; Artero, A. O.; Paiva, M. S. V.; Barbosa, R. L. Reconhecimento de Caracteres Baseado em Regras de Transições entre Pixels Vizinhos. **Avanços em Visão Computacional**. Omnipax Editora Ltda, Curitiba, PR, 2011.

7.2.3 Artigos Submetidos

Silva, F. A.; Hiraga, A. K.; Artero, A. O.; Paiva, M. S. V. A. New Algorithm to Construct Panoramic Images. **IET Image Processing**, 2012.

Silva, F. A.; Artero, A. O.; Paiva, M. S. V.; Barbosa, R. L. TSRs - A New Approach for Traffic Sign Recognition using the SIFT Algorithm. **International Journal of Pattern Recognition and Artificial Intelligence**, 2012.

7.2.4 Participação como co-autor em Artigos Publicados

Tarallo, A. S.; Hiraga, A. K.; Paiva, M. S. V.; Jorge L. A. C.; Silva, F. A. Construção Automática de Mosaico de Imagens Agrícolas Aéreas Sequenciais. **Agricultura de Precisão: Um Novo Olhar**, Embrapa, São Carlos, 2011. p. 106-109.

Tarallo, A. S.; Silva, F. A.; Hiraga, A. K.; Paiva, M. S. V.; Jorge, L. A. C. Construção de Mosaicos de Imagens Aéreas Agrícolas e Comparação com Outras Metodologias. In: VIII WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2012). **Anais...** Goiás: UFG, 2012.

7.2.5 Participação como co-autor em Artigos Aceitos para Publicação

Tarallo, A. S.; Silva, F. A.; Hiraga, A. K.; Paiva, M. S. V.; Jorge, L. A. C. Mosaicos de Imagens Aéreas Sequenciais Construídos Automaticamente. **Avanços em Visão Computacional**. Omnipax Editora Ltda, Curitiba, PR, 2012.

Referências Bibliográficas

Alamús, R.; Baron, A.; Casacuberta, J.; Pla, M.; Sánchez, A.; Serra, A.; Talaya, J. GEOMÒBIL: ICC land based mobile mapping system for cartographic data capture. In: XXII INTERNATIONAL CARTOGRAPHIC CONFERENCE OF THE ICA. **Proceedings...** Coruña, 2005.

Amlacher, K.; Paletta, L. Geo-indexed object recognition for mobile vision tasks. In: 10th INTERNATIONAL CONFERENCE ON HUMAN COMPUTER INTERACTION WITH MOBILE DEVICES AND SERVICES. **Proceedings...** 2008. p. 371-374.

Andrade, J. B. **Fotogrametria**. Curitiba: SBEE, 1998, 246 p.

Arlicot, A.; Soheilian, B.; Paparoditis, N. Circular Road Sign Extraction from Street Level Images Using Colour, Shape and Texture Database Maps. In: Stilla U, Rottensteiner F, Paparoditis N (Eds) CMRT09. IAPRS, v. 28, Part 3/W4, Paris, France, 3-4 September, 2009.

Atkinson, K. B. **Close Range Photogrammetry and Machine Vision**. Scotland: Whittles Publishing, 1996. 371 p.

Azad, P.; Asfour, T.; Dillmann, R. Combining Harris Interest Points and the SIFT Descriptor for Fast Scale-Invariant Object Recognition. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, St. Lois, USA. October 2009.

Barbosa, R. L. **Caminhamento fotogramétrico utilizando o fluxo óptico filtrado**. 2006. 118 f. Tese (Doutorado em Ciências Cartográficas) – Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente, 2006.

Barnes, N.; Zelinsky, A. Real-time radial symmetry for speed sign detection. In: IEEE Intell. Veh. Symp. **Proceedings...** 2004. p. 566-571.

Baró, X.; Vitria, J. Fast traffic sign detection on greyscale images. **Recent Advances in Artificial Intelligence Research and Development**, p. 69-76, October 2004.

Baró, X.; Escalera, S.; Radeva, P.; Vitrià, J. Generic Object Recognition in Urban Image Databases. In: 12th INTERNATIONAL CONFERENCE OF THE CATALAN ASSOCIATION FOR ARTIFICIAL INTELLIGENCE. **Proceedings...** 2009a. p. 27-34.

Baró, X.; Escalera, S.; Radeva, P.; Vitrià, J. Visual content layer for scalable object recognition in urban image databases. In: 2009 IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO. **Proceedings...** 2009b. p. 1616-1619.

Baró, X.; Escalera, S.; Vitrià, J.; Pujol, O.; Radeva, P. Traffic Sign Recognition using Evolutionary Adaboost Detection and Forest-ECOC classification. In: IEEE TRANS. ON INTELLIGENT TRANSPORTATION SYSTEMS. 2009c. p. 113-126.

Bauer, J.; Sünderhauf, N.; Protzel, P. Comparing Several Implementations of Two Recently Published Feature Detectors. In: PROC. OF THE INTERNATIONAL CONFERENCE ON INTELLIGENT AND AUTONOMOUS SYSTEMS. IAV, Toulouse, France, 2007.

Bay, H.; Tuytelaars, T.; Gool, L. V. SURF: Speeded Up Robust Features, volume 3951 of Lecture Notes in Computer Science. **Chapter Computer Vision - ECCV 2006**. Springer Berlin / Heidelberg , p. 404-417, 2006.

Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L. V. SURF: Speeded Up Robust Features. **Computer Vision and Image Understanding (CVIU)**, v. 110, n. 3, p. 346-359, 2008.

Beis, J.; Lowe, D. G. Shape indexing using approximate nearest-neighbour search in highdimensional spaces. In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. Puerto Rico, p. 1000-1006, 1997.

Belaroussi, R.; Tarel, J.-P. A real-time road sign detection using Bilateral Chinese Transform. In: INTERNATIONAL SYMPOSIUM ON VISUAL COMPUTING, ISVC. **Proceedings...** 2009. p. 1161-1170.

Belo, F. A. W. **Desenvolvimento de Algoritmos de Exploração e Mapeamento Visual para Robôs Móveis de Baixo Custo**. Rio de Janeiro, 276 p. Dissertação (Mestrado). Pontifícia Universidade Católica do Rio de Janeiro, 2006.

Bentley, J. L. Multidimensional binary search trees used for associative searching. **Communications of the ACM**, 18(9): p. 509-517, 1975.

Berg, M.; Cheong, O.; Kreveld, M.; Overmars, M. Computational Geometry: Algorithms and Applications. **Springer-Verlag**. 3rd. ed. 2008. 386 p.

Berkhin, P. **Survey Of Clustering Data Mining Techniques**, Accrue Software, San Jose, CA, <http://citeseer.ist.psu.edu/berkhin02survey.html>, 2002.

Booch, G.; Rumbaugh, J. Jacobson, I. **UML: Guia do usuário**. Rio de Janeiro: Elsevier, 2nd. ed. 2006. 474 p.

Brown, M.; Lowe, D .G. Invariant features from interest point groups. In: BRITISH MACHINE VISION CONFERENCE. Cardiff, Wales, 2002, p. 656-665.

Brown, M.; Lowe, D. G. Automatic Panoramic Image Stitching using Invariant Features. **International Journal of Computer Vision**, v. 74, Issue 1, p. 59-73, 2007.

Brunetti, M. F. Um pouco mais sobre o Sistema UTM. Fator **GIS – A Revista do Geoprocessamento**, n. 5, p. 33, 1994.

Cai, H.; Zheng, J. Yu. Locating key views for image indexing of spaces. In: 1st ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA INFORMATION RETRIEVAL. **Proceeding...** 2008. p. 31-38.

Cai, N.; Liang, W.; Xu, S.; Li, F. Traffic Sign Recognition Based on SIFT Features. *Advanced Materials Research* (vol. 121 - 122). **Nanotechnology and Computer Engineering**, 2010, p. 596-599.

Câmara, G.; Davis, C.; Monteiro, A. M. V. **Introdução à ciência da geoinformação**. INPE-10506-RPQ/249. 2001. Disponível em: http://www.geolab.faed.udesc.br/paginaweb/Pagina%20da%20disciplina%20geop_files/intoducaao.pdf. Acesso em: 4 de out. de 2011.

Canny, J. A Computational Approach to Edge Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 8, n. 6, 1986, p. 679-714.

Cramer, M.; Stallmann, D. On the use of GPS/Inertial Exterior Orientation Parameters in Airborne Photogrammetry. In: PHOTOGRAMMETRIC WEEK 2001, Fritsch/Spiller (eds.), Wichmann Verlag, Heidelberg, Germany, 2001.

Creusen, I. M.; Wijnhoven, R. G. J.; Herbschleb, E.; With, P. H. N. Color Exploitation in Hog-Based Traffic Sign Detection. In IEEE 17th INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. **Proceedings...** Hong Kong, 2010. p. 2669-2672.

Dalal, N.; Triggs, B. Histogram of oriented gradients for human detection. In: IEEE COMPUTER VISION AND PATTERN RECOGNITION (CVPR), **Proceedings...** v. 1, June 2005. p. 886-893.

Fayad, M. E.; Schmidt, D. C. Object-oriented application frameworks. **Communications of the ACM**, v. 40, Issue 10, p. 32-38, 1997.

Farag, A. A.; Abdel-Hakim, A. E. Detection, Categorization and Recognition of Road Signs for Autonomous Navigation. In: **ADVANCED CONCEPTS FOR INTELLIGENT VISION SYSTEMS (ACIVS 2004). Proceedings...** Brussels, Belgium, Aug. 31-Sept. 3, 2004. p. 125-130.

Fischler, M. A.; Bolles, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. **Communications of the ACM**, v. 24, p. 381-395, 1981.

Fleyeh, H. Road and Traffic Sign Color Detection and Segmentation - A Fuzzy Approach. In: **CONFERENCE ON MACHINE VISION APPLICATIONS (MVA2005)**, Tsukuba Science City, Japan, May 16-18, 2005. p. 124-127.

Frank, A.; Asuncion, A. (2010). **UCI Machine Learning Repository**. Irvine, CA: University of California, School of Information and Computer Science. Disponível em: <http://archive.ics.uci.edu/ml>. Acesso em: 12 de nov. 2010.

Franke, U.; Gavrila, D.; Gorzig, S.; Lindner, F.; Paetzold, F. and Wohler, C. Autonomous driving goes downtown. **IEEE Intelligent Systems**, v. 13, n. 6, p. 40-48. 1998.

Freund, Y.; Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. **Journal of Computer and System Sciences**, p.119-139, 1997.

Friedman, J.H., Bentley, J.L.; Finkel, R.A. An algorithm for finding best matches in logarithmic expected time. **ACM Transactions on Mathematical Software**, v. 3, Issue 3, p. 209-226, 1977.

Gallis, R. B. A. **Integração de Imagens Vetoriais e Matriciais na Visualização dos levantamentos realizados com a Unidade Móvel de Mapeamento Digital**. 2002. 66p. Dissertação (Mestrado em Ciências Cartográficas) – Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente, 2002.

Gallis, R. B. A. **Extração Semi-Automática da Malha Viária em Imagens Aéreas Digitais de Áreas Rurais utilizando Otimização por Programação Dinâmica no Espaço Objeto**. Presidente Prudente, 2006. 167p. Tese (Doutorado em Ciências Cartográficas) – Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente, 2006.

Galo, M. Calibração de Câmaras Digitais. Implantação e adaptação do campo de calibração da FCT/UNESP. Disponível em: <http://www4.fct.unesp.br/docentes/cartogalo/www22/vrml/fct.htm>. Acesso em: 11 de Nov. 2011.

Geusbroek, J. M.; Boomgaard, R. V. D.; Smeulders, A. W. M. Color invariance. **IEEE Trans. on Pattern Analysis and Machine Intelligence**, v. 23, Issue 12, p. 1338-1350, 2001.

Gonzalez, R. C.; Woods, R. E. **Digital image processing**. New Jersey: Prentice-Hall, Inc., 2002, 716 p.

Google APIs (2010). Disponível em: <http://code.google.com/intl/en/apis/ajax/>. Acesso em: 10 maio de 2010.

Grauman, K.; Leibe, B. **Visual Object Recognition: Synthesis Lectures on Artificial Intelligence and Machine Learning**. Morgan & Claypool Publishers, 2011, 163 p.

Handmann, U.; Kalinke, T.; Tzomakas, C.; Werner, M.; Seelen W. V. An image processing system for driver assistance. In IEEE International Conference on Intelligent Vehicles, Stuttgart, Germany. **Proceedings...** 1998. p. 481-486.

Harris, C.; Stephens, M. A combined corner and edge detector. In: 4th ALVEY VISION CONFERENCE. **Proceedings...** Manchester, UK, 1988. p. 147-151.

Hartigan, J. A., Wong, M. A. A K-means clustering algorithm. **Applied Statistics** 28, p. 100-108, 1979.

Hartley, R.; Zisserman, A. **Multiple View Geometry in Computer Vision**. University Press, Cambridge, 2 ed., 2003.

Heckel, D. Y. **Método Híbrido de correspondência para pares estereoscópicos de imagens aéreas e de satélite de alta definição**. 2008. 115 f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

Höferlin, B.; Zimmermann, K. Towards reliable traffic sign recognition. In INTELLIGENT VEHICLES SYMPOSIUM, 2009 IEEE, p. 324-329, June 2009.

Höferlin, B; Heidemann, G. Selection of an Optimal Set of Discriminative and Robust Local Features with Application to Traffic Sign Recognition. In 18th INT. CONF. IN CENTRAL EUROPE ON COMPUTER GRAPHICS, VISUALIZATION AND COMPUTER VISION. Vaclav Skala, Editor, WSCG 2010. **Proceedings...** v. 18, 2010. p. 9-16.

Hsu, S.-H; Huang, C.-L. Road sign detection and recognition using matching pursuit method. **Image and Vision Computing**, v. 19, n. 3, p. 119-129, Feb. 2001.

IBGE. Sistemas de Referência, Instituto Brasileiro de Geografia e Estatística. Disponível em: ftp://geofp.ibge.gov.br/documentos/geodesia/sisref_2.pdf. Acesso em: 5 de out. de 2011.

Ihara, A.; Fujiyoshi, H.; Takagi, M.; Kumon, H. Tamatsu, Y. Improved Matching Accuracy in Traffic Sign Recognition by Using Different Feature Subspaces. **Machine Vision Applications (MVA2009)**, 3-26, p. 130-133, 2009.

Jain, G.; Ko, J. Handwritten Digits Recognition. Multimedia Systems, **Project Report**, University of Toronto, November 21, 2008, p. 1-3.

Ke, Y.; Sukthankar, R. PCA-SIFT: A more distinctive representation for local image descriptors. In IEEE CONF. ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), v. 2, 2004. p. 506-513.

Kiran, C. G.; Lekhesh, V.; Prabhu, A. R. V.; Rajeev, K. Traffic Sign Detection and Pattern Recognition Using Support Vector Machine. In: SEVENTH INTERNATIONAL CONFERENCE ON ADVANCES IN PATTERN RECOGNITION, 2009. p. 87-90.

Koenderink, J. J. The structure of images. **Biological Cybernetics**, Springer-Verlag, v. 50, 1984, p. 363-396.

Kus, M. C.; Gokmen, M.; Etaner-Uyar, S. Traffic sign recognition using Scale Invariant Feature Transform and color classification. In 23rd INTERNATIONAL SYMPOSIUM ON COMPUTER AND INFORMATION SCIENCES (ISCIS'08), 2008. p. 1-6.

Le, T. T.; Tran, S. T.; Mita, S.; Nguyen, T. D. Real Time Traffic Sign Detection Using Color and Shape-Based Features. In: ACIIDS'10 PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON INTELLIGENT INFORMATION AND DATABASE SYSTEMS: Part II. **Proceeding...** Springer-Verlag Berlin, Heidelberg, 2010.

Lecun, Y.; Bottou, L; Bengio, Y. and Haffner, P. Gradient-based learning applied to document recognition. IEEE. **Proceedings...** v. 86, n. 11, 1998. p. 2278-2334.

Lienhart, R.; Maydt, J. An extended set of Haar-like features for rapid object detection. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. **Proceedings...** 2002. p. 155-162.

Lindeberg, T. Scale-space theory: A basic tool for analyzing structures at different scales. **Journal of Applied Statistics**, 21(2): 1994, p. 224-270.

Lorsakul, A.; Suthakorn, J. Traffic Sign Recognition for Intelligent Vehicle/Driver Assistance System Using Neural Network on OpenCV. In: 4th INTERNATIONAL CONFERENCE ON UBIQUITOUS ROBOTS AND AMBIENT INTELLIGENCE (URAI 2007), **Proceedings...** POSTECH, PIRO, KOREA, Nov 22-24, 2007. p. 279-284.

Lowe, D. G. Object recognition from local scale-invariant features. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, Corfu, Greece, 1999. p. 1150-1157.

Lowe, D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, n.2, p. 91-110, 2004.

Loy, G.; Zelinsky, A. Fast radial symmetry for detecting points of interest. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 25, n. 8, p. 959-973, Aug. 2003.

Machado, W. R. S.; Louro, A. H. F.; Gonzaga, A.; Boaventura, M. Estudo e proposta de adaptação do algoritmo SIFT em relação ao problema de iluminação em imagens. In: IV WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2008). **Anais...** Bauru: Canal 6 Projetos Editoriais, v. 1, 2008.

Madeira, S.; Bastos, L. C.; Sousa, A. M.; Sobral, J. F.; Santos, L. P. Automatic Traffic Signs Inventory Using a Mobile Mapping System. In: INTERNATIONAL CONFERENCE AND EXHIBITION ON GEOGRAPHIC INFORMATION. GIS PLANET 2005. **Proceedings....** Lisboa, Portugal, 2005.

Maldonado-Bascon, S.; Lafuente-Arroyo, S.; Gil-Jimenez, P.; Gomez-Moreno, H.; Lopez-Ferreras, F. Road-Sign Detection and Recognition Based on Support Vector Machines. **IEEE Transactions on Intelligent Transportation System**, v. 8, n. 2, p. 264-278, Jun. 2007.

MapWindow GIS Open Source Project. Disponível em: <http://www.mapwindow.org/>. Acesso em: 09 nov. 2010.

Martinović, A.; Glavaš, G.; Juribašić, M.; Sutić, D.; Kalafatić, Z. Real-time Detection and Recognition of Traffic Signs. In: 33rd INTERNATIONAL CONVENTION MIPRO 2010, **Proceedings...** v. III, 2010. p. 247-252.

Medici, P.; Caraffi, C.; Cardarelli, E.; Porta, P. P.; Ghisio, G. Real Time Road Signs Classification. In: IEEE INTERNATIONAL CONFERENCE ON VEHICULAR ELECTRONICS AND SAFETY. **Proceedings...** Columbus, OH, USA. September 22-24, 2008. p. 253-258.

Md-Sallah, S. S.; Hussin, F. A.; Yusoff, M. Z. Shape-based Road Sign Detection and Recognition for Embedded Application Using MATLAB. In: INTERNATIONAL CONFERENCE ON INTELLIGENT AND ADVANCED SYSTEMS (ICIAS 2010), Kuala Lumpur, Malaysia, June, 2010, p. 15-17.

Mikolajczyk, K.; Schmid, C. A Performance Evaluation of Local Descriptors. **IEEE Trans. on Pattern Analysis and Machine Intelligence**. v. 27, Issue 10, p. 1615-1630, 2005.

Mikolajczyk, K.; Leibe, B.; Schiele, B. Local Features for Object Class Recognition. In: 10th IEEE INT. CONF. COMPUT. VISION (ICCV'05). **Proceedings...** 17-21 October, 2005. p. 1792-1799.

Mokhtarian, F. Silhouette-Based Isolated Object Recognition through Curvature Scale Space. **IEEE Trans. on Pattern Analysis and Machine Intelligence**, v. 17, n. 5, May 1995.

Monico, J. F. G. **NAVSTAR-GPS: descrição, fundamentos e aplicações**. Fundação Editora da Unesp, 2000. 287 p.

Nguyen, G. P.; Andersen, H. J. A fast sign localization system using discriminative color invariant segmentation. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION AND COMPUTER VISION. Bangkok, Thailand, 17-19 dec., 2008.

Oliveira, E. B. **Implementação do módulo de interseção fotogramétrica em um banco de imagens georreferenciadas**. Presidente Prudente, 2003. 112p. Dissertação (Mestrado em Ciências Cartográficas) – Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente, 2003.

OpenCV. Open Computer Vision Library (2010). Disponível em: <http://sourceforge.net/projects/opencvlibrary/>. Acesso em: maio de 2010.

Otsu, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**. v. 9, n. 1, p. 62–66, 1979.

Paulo, C. F.; Correia, P. L. Traffic Sign Recognition Based on Pictogram Contours. In: WORKSHOP ON IMAGE ANALYSIS FOR MULTIMEDIA INTERACTIVE SERVICES (WIAMIS'2008), Klagenfurt, Austria, May 2008.

Pereira, J. F.; Alves V. M. O.; Cavalcanti, G. D. C.; Ren, T. I. “Modular Image Principal Component Analysis for Handwritten Digits Recognition”. IWSSIP 2010 - 17th INT. CONFERENCE ON SYSTEMS, SIGNALS AND IMAGE PROCESSING, Center of Informatics Federal University of Pernambuco Recife, Brazil, 2010.

Quack, T., Leibe, B.; Van Gool, L. World-scale mining of objects and events from community photo collections. In: CIVR '08. **Proceedings...** 2008. p. 47-56.

Reiterer, A.; Hassan, T.; El-Sheimy, N. Robust extraction of traffic signs form georeferenced mobile mapping images. In: 6th INTERNATIONAL SYMPOSIUM ON MOBILE MAPPING TECHNOLOGY. Presidente Prudente, Brasil, 2009.

Ruy, R. S. **Geração Automática de Estereomodelos a partir de Imagens Digitais Georreferenciadas**. 2004. 133 f. Dissertação (Mestrado em Ciências Cartográficas) – Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente, 2004.

Saturnino, M. -B.; Lafuente-Arroyo, S.; Gil-Jimenez, P.; Gomez-Moreno, H.; Lopez-Ferreras, F. Road-sign detection and recognition based on support vector machines. **IEEE Transactions on Intelligent Transportation Systems**, 8(2), 2007, p. 264-278.

Senekal, F. P. Traffic Sign Detection and Classification using Colour and Shape Cues. In: 19th ANNUAL SYMPOSIUM OF THE PATTERN RECOGNITION ASSOCIATION OF SOUTH AFRICA (PRASA 2008), Cape Town, South Africa, 27-28 November 2008.

Shrivastava, S. K.; Gharde, S. S. Support Vector Machine for Handwritten Devanagari Numeral Recognition. **International Journal of Computer Applications** (0975 – 8887), v. 7, n.11, October 2010, p. 9-14.

Silpa-Anan, C.; Hartley, R. Optimised kd-trees for fast image descriptor matching. In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2008), p. 1-8. 2008.

Silva, J. F. C.; Oliveira, R. A. Caminhamento Fotogramétrico para Levantamentos Urbanos. In: XVIII CONGRESSO BRASILEIRO DE CARTOGRAFIA. Rio de Janeiro, 1997, 8 p.

Silva, J. F. C.; Camargo, P. O.; Gallis, R. B. A. Development of a low-cost mobile mapping system: a South American Experience. **Photogrammetric Record**, v. 18, n. 101, p. 5-26, 2003.

Silva, F. A.; Artero, A. O.; Paiva, M. S. V.; Barbosa, L. R. Um Algoritmo Rápido para o Reconhecimento de Caracteres. In: VII WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2011). **Anais...** Curitiba: UFPR, 2011. p. 149-154.

SPAN-SE User Manual. Disponível em: webone.novatel.ca/assets/Documents/Manuals/om-20000124.pdf. Acesso em: 30 de jul. de 2011.

Tao, C.; Chapman, M. A.; Chaplin, B. A. Automated processing of mobile mapping image sequences. **Journal of Photogrammetry & Remote Sensing**, v. 55, p. 330-346, 2001.

Tao, C. V.; Li, J. **Advances in Mobile Mapping Technology**. London: Taylor&Francis. (ISPRS Book Series). 2007.

Theodoridis, S.; Koutroumbas K. Pattern Recognition. **Elsevier Inc** Fourth Edition, 2009, 984 p.

Treiber, M. An Introduction to Object Recognition: Selected Algorithms for a Wide Variety of Applications. **Springer-Verlag London Limited**, 2010, 201 p.

Trentini, V. B.; Godoy, L. A. T.; Marana, A. N. Reconhecimento Automático de Placas de Veículos. In: VI WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2010). **Anais...** Presidente Prudente, 2010. p. 267-272.

UML Specification. Disponível em: <http://www.omg.org/spec/UML/>. Acesso em: 24 de set. de 2011.

Viola, P.; Jones, M. J. Robust real-time object detection. In: SECOND INTERNATIONAL WORKSHOP ON STATISTICAL AND COMPUTATIONAL THEORIES OF VISION – MODELING, LEARNING, COMPUTING AND SAMPLING, July 13, 2001.

Viola, P.; Jones, M. J. Robust real-time face detection. **International Journal of Computational Vision**, v. 57, n. 2, p. 137-154, 2004.

Wanniarachchi, W. K. I. L.; Sonnadara, D. U. J.; Jayananda, M. K. Detection and Extraction of Road Traffic Signs. **Proceedings of the Technical Sessions**, 24 (2008). Institute of Physics – Sri Lanka, 2008, p.13-19.

Weng, J.; Zhang, Y.; Hwang, Wey-Shiuan. Candid covariance-free incremental principal component analysis. **IEEE Transactions on PAMI**, v. 25, n.8, p. 1034-1040, Aug. 2003.

Witkin, A. P. Scale-space filtering. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, KARLSRUHE, GERMANY, 1983. p. 1019-1022.

Wolf, P. R. **Elements of photogrammetry, with air photo interpretation and remote sensing**. New York: McGraw-Hill, 1983. 628 p.

Yuan, J.; Luo, J.; Kautz, H.; Wu, Y. Mining GPS traces and visual words for event classification, In: 1st ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA INFORMATION RETRIEVAL. **Proceeding...** 2008. p. 2-9.

A Apêndice A – Mapa contendo as placas detectadas e reconhecidas

Neste Apêndice, são mostradas as placas detectadas e reconhecidas, posicionadas em um mapa da área na qual foi realizado o levantamento. As placas são representadas por um ponto no mapa (Figura A.1 e Figura A.2) e são numeradas sequencialmente. Os números rotulando as placas são úteis para identificar as placas nos resultados referentes à detecção, reconhecimento e cálculo das coordenadas das placas no terreno apresentado no Apêndice B.

Figura A.1: Mapa contendo o posicionamento e numeração sequencial das placas detectadas e reconhecidas (Parte 1/2).

Figura A.2: Mapa contendo o posicionamento e numeração sequencial das placas detectadas e reconhecidas (Parte 2/2).

Apêndice B – Resultados da Detecção, Reconhecimento e Georreferenciamento das placas de Sinalização

Neste Apêndice, são mostrados os resultados produzidos pelo Sistema *AutoTSR* na forma de um relatório, referentes à detecção, reconhecimento e cálculo das coordenadas das placas no terreno.

Na primeira página, são descritas algumas informações, tais como: data do levantamento, diretórios onde estão armazenadas as imagens (esquerda e direita), descrição do levantamento, o projeto ao qual pertence o levantamento e o arquivo de *log* do GPS/INS (seção 3.6) contendo os dados de navegação e posicionamento.

Na sequência, ao longo do Apêndice, são listadas as informações referentes aos resultados de cada par de imagens (esquerda e direita), sendo exibidas as informações nos seguintes campos:

- “*Imagem Esq*” e “*Imagem Dir*”: nomes das imagens da sequência capturada durante o levantamento;
- “*Lat_e*”, “*Long_e*”, “*Alt_e*”, “*Lat_d*”, “*Long_d*” e “*Alt_d*”: coordenadas geodésicas de posicionamento (latitude, longitude e altitude) referentes aos CP (centros perspectivos) das câmaras;
- “*w_e*”, “*phi_e*”, “*kp_e*”, “*w_d*”, “*phi_d*” e “*kp_d*”: ângulos de orientação (dados de navegação) em radianos calculados em função dos dados obtidos a partir do *log* do GPS/INS;
- “*Xc_e*”, “*Yc_e*”, “*Zc_e*”, “*Xc_d*”, “*Yc_d*” e “*Zc_d*”: coordenadas no sistema de coordenadas UTM – *E* (*East*), *N* (*North*), *H* (*Height*), com base no Sistema de Referência WGS84 (subseção 3.3.3). Esses campos são as coordenadas geodésicas convertidas para coordenadas cartesianas, usadas na etapa de georreferenciamento.

Também, para cada par de imagens é exibida uma imagem reduzida referente à imagem da direita da sequência do levantamento, e nela, cada um dos sinais reconhecidos são demarcados por um retângulo. Ainda para cada par de imagens, são exibidos os dados das placas detectadas e reconhecidas. Nesses resultados, é mostrada a imagem do *template*

utilizado, bem como o recorte da imagem do sinal reconhecido juntamente com um número sequencial envolto por um círculo, representando as placas numeradas no mapa apresentado no Apêndice A e o número de pontos (campo “*pts (match)*”). Também são exibidas as coordenadas no terreno em UTM (X_t , Y_t , Z_t) e em coordenadas geodésicas (latitude e longitude, nos campos “*Lat*” e “*Long*”, respectivamente) calculadas em função do processo de georreferenciamento usando a interseção fotogramétrica (seção 3.7).

Levantamento

Código: 1 **Data:** 13/03/2012

Diretório Img Esq: C:\Imagens Levantamento\Esq **Diretório Img Dir:** C:\Imagens Levantamento\Dir

Descrição Levantamento: Levantamento Doutorado

Projeto: 1 - Projeto Doutorado

Arquivo dados INS: D:\Levantamento 21_11_2011\levantamento.gps

Imagem Esq: 0001_ESQ.jpg

Lat_e: -22,116016269 **Long_e:** -51,401446084 **Alt_e:** 410,412350075

w_e: 0,0157943058 **phi_e:** 0,0724712783 **kp_e:** 1,6569301981

Xc_e: 458595,941304653 **Yc_e:** 7554276,970256010 **Zc_e:** 410,412350075

Imagem Dir: 0001_DIR.jpg

Lat_d: -22,116024908 **Long_d:** -51,401446909 **Alt_d:** 410,412350075

w_d: 0,0157943058 **phi_d:** 0,0724712783 **kp_d:** 1,6569301981

Xc_d: 458595,858718343 **Yc_d:** 7554276,013814950 **Zc_d:** 410,412350075

Código Template: 2 **Descrição:** Velocidade máxima permitida - 40 Km/h

Template:

Placa Reconhecida:

pts (match): 11

1

Xt: 458598,8549909

Yt: 7554272,9031645

Zt: 453,6901564

Lat: -22,116053081

Long: -51,401417938

Código Template: 17 **Descrição:** Semáforo à frente

Template:

Placa Reconhecida:

pts (match): 10

2

Xt: 458597,4688850

Yt: 7554277,7559653

Zt: 426,3599081

Lat: -22,116009207

Long: -51,401431253

Imagem Esq: 0002_ESQ.jpg

Lat_e: -22,116056722 **Long_e:** -51,400798899 **Alt_e:** 409,632910523

w_e: 0,0050853823 **phi_e:** 0,0439363895 **kp_e:** 1,6646244283

Xc_e: 458662,702732140 **Yc_e:** 7554272,668447010 **Zc_e:** 409,632910523

Imagem Dir: 0002_DIR.jpg

Lat_d: -22,116065355 **Long_d:** -51,400799796 **Alt_d:** 409,632910523

w_d: 0,0050853823 **phi_d:** 0,0439363895 **kp_d:** 1,6646244283

Xc_d: 458662,612789270 **Yc_d:** 7554271,712669690 **Zc_d:** 409,632910523

Código Template: 10 **Descrição:** Proibido retornar à esquerda

Template:

Placa Reconhecida:

pts (match): 5

3

Xt: 458664,0897305

Yt: 7554271,5530249

Zt: 440,3054088

Lat: -22,116066832

Long: -51,400785480

Imagem Esq: 0003_ESQ.jpg

Lat_e: -22,116089967 **Long_e:** -51,400525462 **Alt_e:** 409,920761159
w_e: 0,0205377539 **phi_e:** 0,0532787313 **kp_e:** 1,6410502051
Xc_e: 458690,914351680 **Yc_e:** 7554269,062822410 **Zc_e:** 409,920761159

Imagem Dir: 0003_DIR.jpg

Lat_d: -22,116098617 **Long_d:** -51,400526139 **Alt_d:** 409,920761159
w_d: 0,0205377539 **phi_d:** 0,0532787313 **kp_d:** 1,6410502051
Xc_d: 458690,846963423 **Yc_d:** 7554268,105190520 **Zc_d:** 409,920761159

Código Template: 14 **Descrição:** Passagem sinalizada de pedestres

Template:

Placa Reconhecida:

pts (match): 15

4

Xt: 458692,2676410
Yt: 7554269,1170662
Zt: 430,8414564
Lat: -22,116089509
Long: -51,400512339

Imagem Esq: 0004_ESQ.jpg

Lat_e: -22,116114400 **Long_e:** -51,400325561 **Alt_e:** 410,118235490
w_e: -0,0155163669 **phi_e:** 0,0707282665 **kp_e:** 1,6634392430
Xc_e: 458711,538943718 **Yc_e:** 7554266,412584580 **Zc_e:** 410,118235490

Imagem Dir: 0004_DIR.jpg

Lat_d: -22,116123034 **Long_d:** -51,400326446 **Alt_d:** 410,118235490
w_d: -0,0155163669 **phi_d:** 0,0707282665 **kp_d:** 1,6634392430
Xc_d: 458711,450133684 **Yc_d:** 7554265,456701330 **Zc_d:** 410,118235490

Código Template: 5 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 12

5

Xt: 458712,1333093
Yt: 7554265,4396705
Zt: 414,8161919
Lat: -22,116123204
Long: -51,400319823

Código Template: 8 **Descrição:** Proibido parar e estacionar

Template:

Placa Reconhecida:

pts (match): 15

6

Xt: 458713,6965082
Yt: 7554265,8546992
Zt: 441,0426123
Lat: -22,116119492
Long: -51,400304656

Imagem Esq: 0005_ESQ.jpg

Lat_e: -22,116249197 **Long_e:** -51,399170547 **Alt_e:** 410,067699646
w_e: 0,0116390089 **phi_e:** 0,0697895835 **kp_e:** 1,6424607386
Xc_e: 458830,704352918 **Yc_e:** 7554251,804946010 **Zc_e:** 410,067699646

Imagem Dir: 0005_DIR.jpg

Lat_d: -22,116257846 **Long_d:** -51,399171238 **Alt_d:** 410,067699646
w_d: 0,0116390089 **phi_d:** 0,0697895835 **kp_d:** 1,6424607386
Xc_d: 458830,635613956 **Yc_d:** 7554250,847410140 **Zc_d:** 410,067699646

Código Template: 8 **Descrição:** Proibido parar e estacionar

Template:

Placa Reconhecida:

pts (match): 7

7

Xt: 458831,6471620

Yt: 7554252,7441697

Zt: 419,1785226

Lat: -22,116240734

Long: -51,399161382

Imagem Esq: 0006_ESQ.jpg

Lat_e: -22,116305707 **Long_e:** -51,398649519 **Alt_e:** 410,027686037

w_e: -0,0202226597 **phi_e:** 0,0658821712 **kp_e:** 1,6518992666

Xc_e: 458884,458684604 **Yc_e:** 7554245,690685740 **Zc_e:** 410,027686037

Imagem Dir: 0006_DIR.jpg

Lat_d: -22,116314350 **Long_d:** -51,398650297 **Alt_d:** 410,027686037

w_d: -0,0202226597 **phi_d:** 0,0658821712 **kp_d:** 1,6518992666

Xc_d: 458884,380911109 **Yc_d:** 7554244,733841300 **Zc_d:** 410,027686037

Código Template: 8 **Descrição:** Proibido parar e estacionar

Template:

Placa Reconhecida:

pts (match): 8

8

Xt: 458885,6124456

Yt: 7554247,3169783

Zt: 422,1694883

Lat: -22,116291042

Long: -51,398638291

Imagem Esq: 0007_ESQ.jpg

Lat_e: -22,116512118 **Long_e:** -51,396820409 **Alt_e:** 411,489028580

w_e: 0,0172016277 **phi_e:** 0,0518151432 **kp_e:** 1,6978719771

Xc_e: 459073,169484191 **Yc_e:** 7554223,336240790 **Zc_e:** 411,489028580

Imagem Dir: 0007_DIR.jpg

Lat_d: -22,116520718 **Long_d:** -51,396821613 **Alt_d:** 411,489028580

w_d: 0,0172016277 **phi_d:** 0,0518151432 **kp_d:** 1,6978719771

Xc_d: 459073,047819629 **Yc_d:** 7554222,383981510 **Zc_d:** 411,489028580

Código Template: 11 **Descrição:** Proibido virar à direita

Template:

Placa Reconhecida:

pts (match): 6

9

Xt: 459074,3997026

Yt: 7554224,4913652

Zt: 448,4529382

Lat: -22,116501711

Long: -51,396808452

Imagem Esq: 0008_ESQ.jpg

Lat_e: -22,116771787 **Long_e:** -51,396116271 **Alt_e:** 413,864106523

w_e: 0,0499752901 **phi_e:** 0,0243466387 **kp_e:** 2,0001956716

Xc_e: 459145,867743189 **Yc_e:** 7554194,782799510 **Zc_e:** 413,864106523

Imagem Dir: 0008_DIR.jpg

Lat_d: -22,116779663 **Long_d:** -51,396120168 **Alt_d:** 413,864106523

w_d: 0,0499752901 **phi_d:** 0,0243466387 **kp_d:** 2,0001956716

Xc_d: 459145,468071427 **Yc_d:** 7554193,909952170 **Zc_d:** 413,864106523

Código Template: 10 **Descrição:** Proibido retornar à esquerda

Template:

Placa Reconhecida:

pts (match): 17

10

Xt: 459147,1373210
Yt: 7554190,9680787
Zt: 450,2270677
Lat: -22,116806280
Long: -51,396104058

Código Template: 13 **Descrição:** Proibido virar à esquerda

Template:

Placa Reconhecida:

pts (match): 9

11

Xt: 459146,3783988
Yt: 7554193,1545419
Zt: 441,7140694
Lat: -22,116786509
Long: -51,396111361

Imagem Esq: 0009_ESQ.jpg

Lat_e: -22,117535082 **Long_e:** -51,394503144 **Alt_e:** 426,406903852
w_e: 0,0550348987 **phi_e:** 0,0507231648 **kp_e:** 2,0179855147
Xc_e: 459312,461341605 **Yc_e:** 7554110,726235350 **Zc_e:** 426,406903852

Imagem Dir: 0009_DIR.jpg

Lat_d: -22,117542893 **Long_d:** -51,394507190 **Alt_d:** 426,406903852
w_d: 0,0550348987 **phi_d:** 0,0507231648 **kp_d:** 2,0179855147
Xc_d: 459312,046206086 **Yc_d:** 7554109,860635850 **Zc_d:** 426,406903852

Código Template: 7 **Descrição:** Proibido estacionar

Template:

Placa Reconhecida:

pts (match): 5

12

Xt: 459292,8655249
Yt: 7554128,8779269
Zt: 643,2806173
Lat: -22,117370637
Long: -51,394692683

Imagem Esq: 0010_ESQ.jpg

Lat_e: -22,117576728 **Long_e:** -51,394417823 **Alt_e:** 427,087934126
w_e: 0,0618266820 **phi_e:** 0,0402225755 **kp_e:** 2,0073046942
Xc_e: 459321,272984567 **Yc_e:** 7554106,139307680 **Zc_e:** 427,087934126

Imagem Dir: 0010_DIR.jpg

Lat_d: -22,117584578 **Long_d:** -51,394421780 **Alt_d:** 427,087934126
w_d: 0,0618266820 **phi_d:** 0,0402225755 **kp_d:** 2,0073046942
Xc_d: 459320,867117865 **Yc_d:** 7554105,269323650 **Zc_d:** 427,087934126

Código Template: 12 **Descrição:** Proibido virar à direita

Template:

Placa Reconhecida:

pts (match): 5

13

Xt: 459316,8893886
Yt: 7554109,6612937
Zt: 520,0778156
Lat: -22,117544807
Long: -51,394460237

Imagem Esq: 0011_ESQ.jpg

Lat_e: -22,117864325 Long_e: -51,393790262 Alt_e: 432,029617804
w_e: 0,0822712840 phi_e: 0,0433877818 kp_e: 2,0005644417
Xc_e: 459386,080341996 Yc_e: 7554074,472960800 Zc_e: 432,029617804

Imagem Dir: 0011_DIR.jpg

Lat_d: -22,117872200 Long_d: -51,393794162 Alt_d: 432,029617804
w_d: 0,0822712840 phi_d: 0,0433877818 kp_d: 2,0005644417
Xc_d: 459385,680348381 Yc_d: 7554073,600260900 Zc_d: 432,029617804

Código Template: 1 Descrição: Velocidade máxima permitida - 20 Km/h

Template:

Placa Reconhecida:

pts (match): 6

Xt: 459388,6849966
Yt: 7554067,5054294
Zt: 480,9756977
Lat: -22,117927332
Long: -51,393765183

14

Código Template: 21 Descrição: Passagem sinalizada de escolares

Template:

Placa Reconhecida:

pts (match): 6

Xt: 459388,1527445
Yt: 7554068,4628129
Zt: 475,6747025
Lat: -22,117918671
Long: -51,393770319

15

Imagem Esq: 0012_ESQ.jpg

Lat_e: -22,117905449 Long_e: -51,393702734 Alt_e: 432,789232112
w_e: 0,0602557083 phi_e: 0,0392403773 kp_e: 2,0049022188
Xc_e: 459395,119553725 Yc_e: 7554069,944347780 Zc_e: 432,789232112

Imagem Dir: 0012_DIR.jpg

Lat_d: -22,117913308 Long_d: -51,393706670 Alt_d: 432,789232112
w_d: 0,0602557083 phi_d: 0,0392403773 kp_d: 2,0049022188
Xc_d: 459394,715778307 Yc_d: 7554069,073391170 Zc_d: 432,789232112

Código Template: 1 Descrição: Velocidade máxima permitida - 20 Km/h

Template:

Placa Reconhecida:

pts (match): 13

Xt: 459397,6523249
Yt: 7554063,3877229
Zt: 479,0499010
Lat: -22,117964742
Long: -51,393678341

16

Código Template: 21 Descrição: Passagem sinalizada de escolares

Template:

Placa Reconhecida:

pts (match): 18

Xt: 459396,7223660
Yt: 7554065,5491765
Zt: 470,0826765
Lat: -22,117945194
Long: -51,393687303

17

Imagem Esq: 0013_ESQ.jpg

Lat_e: -22,118251434 **Long_e:** -51,392952336 **Alt_e:** 439,419460208
w_e: 0,0682286555 **phi_e:** 0,0424915935 **kp_e:** 2,0001805845
Xc_e: 459472,612279796 **Yc_e:** 7554031,847457770 **Zc_e:** 439,419460208

Imagem Dir: 0013_DIR.jpg

Lat_d: -22,118259311 **Long_d:** -51,392956232 **Alt_d:** 439,419460208
w_d: 0,0682286555 **phi_d:** 0,0424915935 **kp_d:** 2,0001805845
Xc_d: 459472,212621203 **Yc_d:** 7554030,974604400 **Zc_d:** 439,419460208

Código Template: 18 **Descrição:** Semáforo à frente

Template:

Placa Reconhecida:

pts (match): 8

18

Xt: 459474,6474855
Yt: 7554026,6045779
Zt: 482,8513315
Lat: -22,118298847
Long: -51,392932734

Imagem Esq: 0014_ESQ.jpg

Lat_e: -22,119120653 **Long_e:** -51,391067868 **Alt_e:** 454,371548767
w_e: 0,0386370518 **phi_e:** 0,0480283783 **kp_e:** 2,0022667003
Xc_e: 459667,217504843 **Yc_e:** 7553936,134863940 **Zc_e:** 454,371548767

Imagem Dir: 0014_DIR.jpg

Lat_d: -22,119128521 **Long_d:** -51,391071782 **Alt_d:** 454,371548767
w_d: 0,0386370518 **phi_d:** 0,0480283783 **kp_d:** 2,0022667003
Xc_d: 459666,816026247 **Yc_d:** 7553935,262846200 **Zc_d:** 454,371548767

Código Template: 9 **Descrição:** Proibido retornar à esquerda

Template:

Placa Reconhecida:

pts (match): 5

19

Xt: 459673,0900833
Yt: 7553925,8366170
Zt: 537,0012006
Lat: -22,119213826
Long: -51,391011185

Imagem Esq: 0015_ESQ.jpg

Lat_e: -22,119123091 **Long_e:** -51,391063588 **Alt_e:** 454,065254161
w_e: 0,0484833113 **phi_e:** 0,0498036958 **kp_e:** 2,0046750038
Xc_e: 459667,659555031 **Yc_e:** 7553935,866127980 **Zc_e:** 454,065254161

Imagem Dir: 0015_DIR.jpg

Lat_d: -22,119130951 **Long_d:** -51,391067523 **Alt_d:** 454,065254161
w_d: 0,0484833113 **phi_d:** 0,0498036958 **kp_d:** 2,0046750038
Xc_d: 459667,255977518 **Yc_d:** 7553934,995079650 **Zc_d:** 454,065254161

Código Template: 11 **Descrição:** Proibido virar à direita

Template:

Placa Reconhecida:

pts (match): 5

20

Xt: 459659,2600705
Yt: 7553949,0964730
Zt: 596,5788173
Lat: -22,119003370
Long: -51,391144699

Imagem Esq: 0016_ESQ.jpg

Lat_e: -22,120109339 Long_e: -51,388990540 Alt_e: 463,172468917
w_e: 0,0184610340 phi_e: 0,0335353316 kp_e: 2,0070654075
Xc_e: 459881,744853366 Yc_e: 7553827,246862050 Zc_e: 463,172468917

Imagem Dir: 0016_DIR.jpg

Lat_d: -22,120117190 Long_d: -51,388994495 Alt_d: 463,172468917
w_d: 0,0184610340 phi_d: 0,0335353316 kp_d: 2,0070654075
Xc_d: 459881,339194851 Yc_d: 7553826,376780920 Zc_d: 463,172468917

Código Template: 4 Descrição: Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 6

21

Xt: 459787,9176830
Yt: 7553911,4714310
Zt: 1640,8060568
Lat: -22,119346266
Long: -51,389898189

Imagem Esq: 0017_ESQ.jpg

Lat_e: -22,120231684 Long_e: -51,388737415 Alt_e: 464,225761489
w_e: 0,0237334980 phi_e: 0,0399295885 kp_e: 2,0007752125
Xc_e: 459907,885566102 Yc_e: 7553813,771274060 Zc_e: 464,225761489

Imagem Dir: 0017_DIR.jpg

Lat_d: -22,120239559 Long_d: -51,388741317 Alt_d: 464,225761489
w_d: 0,0237334980 phi_d: 0,0399295885 kp_d: 2,0007752125
Xc_d: 459907,485388556 Yc_d: 7553812,898658490 Zc_d: 464,225761489

Código Template: 6 Descrição: Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 4

22

Xt: 459902,8744915
Yt: 7553819,3249220
Zt: 530,9486181
Lat: -22,120181396
Long: -51,388785865

Imagem Esq: 0018_ESQ.jpg

Lat_e: -22,120782511 Long_e: -51,387571530 Alt_e: 464,785221103
w_e: -0,0131141557 phi_e: 0,0440657462 kp_e: 2,0335102924
Xc_e: 460028,284818223 Yc_e: 7553753,107360770 Zc_e: 464,785221103

Imagem Dir: 0018_DIR.jpg

Lat_d: -22,120790262 Long_d: -51,387575706 Alt_d: 464,785221103
w_d: -0,0131141557 phi_d: 0,0440657462 kp_d: 2,0335102924
Xc_d: 460027,856295031 Yc_d: 7553752,248310210 Zc_d: 464,785221103

Código Template: 6 Descrição: Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 7

23

Xt: 460040,6206713
Yt: 7553744,2752751
Zt: 740,9142570
Lat: -22,120862586
Long: -51,387452140

Código Template: 9 **Descrição:** Proibido retornar à esquerda

Template:

Placa Reconhecida:

pts (match): 5

(24)

Xt: 460035,3515173

Yt: 7553743,5914115

Zt: 537,4813393

Lat: -22,120868643

Long: -51,387503247

Imagem Esq: 0019_ESQ.jpg

Lat_e: -22,120782921 **Long_e:** -51,387570824 **Alt_e:** 464,799744978

w_e: -0,0114765673 **phi_e:** 0,0444312691 **kp_e:** 2,0337357965

Xc_e: 460028,357753634 **Yc_e:** 7553753,062172880 **Zc_e:** 464,799744978

Imagem Dir: 0019_DIR.jpg

Lat_d: -22,120790671 **Long_d:** -51,387575002 **Alt_d:** 464,799744978

w_d: -0,0114765673 **phi_d:** 0,0444312691 **kp_d:** 2,0337357965

Xc_d: 460027,929036734 **Yc_d:** 7553752,203218970 **Zc_d:** 464,799744978

Código Template: 6 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 6

(25)

Xt: 460029,0684918

Yt: 7553753,8482641

Zt: 506,2090692

Lat: -22,120775835

Long: -51,387563913

Imagem Esq: 0020_ESQ.jpg

Lat_e: -22,120842061 **Long_e:** -51,387498998 **Alt_e:** 464,918120317

w_e: 0,0777437879 **phi_e:** 0,0048204463 **kp_e:** 2,5207724312

Xc_e: 460035,782183243 **Yc_e:** 7553746,534821780 **Zc_e:** 464,918120317

Imagem Dir: 0020_DIR.jpg

Lat_d: -22,120847088 **Long_d:** -51,387506583 **Alt_d:** 464,918120317

w_d: 0,0777437879 **phi_d:** 0,0048204463 **kp_d:** 2,5207724312

Xc_d: 460035,001317702 **Yc_d:** 7553745,976387350 **Zc_d:** 464,918120317

Código Template: 5 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 13

(26)

Xt: 460036,7504488

Yt: 7553745,0021642

Zt: 477,8840233

Lat: -22,120855930

Long: -51,387489648

Imagem Esq: 0021_ESQ.jpg

Lat_e: -22,121196282 **Long_e:** -51,387646429 **Alt_e:** 468,515685540

w_e: 0,0632596344 **phi_e:** 0,0310084755 **kp_e:** 3,5879496258

Xc_e: 460020,676805506 **Yc_e:** 7553707,287547470 **Zc_e:** 468,515685540

Imagem Dir: 0021_DIR.jpg

Lat_d: -22,121192518 **Long_d:** -51,387654815 **Alt_d:** 468,515685540

w_d: 0,0632596344 **phi_d:** 0,0310084755 **kp_d:** 3,5879496258

Xc_d: 460019,810860820 **Yc_d:** 7553707,701962480 **Zc_d:** 468,515685540

Código Template: 4 **Descrição:** Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 10

27

Xt: 460018,1450508
Yt: 7553700,4859152
Zt: 527,3937010
Lat: -22,121257671
Long: -51,387671145

Código Template: 6 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 8

28

Xt: 460020,1560339
Yt: 7553703,9656246
Zt: 507,5729415
Lat: -22,121226281
Long: -51,387651561

Imagem Esq: 0022_ESQ.jpg

Lat_e: -22,121489658 **Long_e:** -51,387807831 **Alt_e:** 471,228798099
w_e: 0,0689662520 **phi_e:** 0,0425209068 **kp_e:** 3,5801981669
Xc_e: 460004,113451253 **Yc_e:** 7553674,771473860 **Zc_e:** 471,228798099

Imagem Dir: 0022_DIR.jpg

Lat_d: -22,121485955 **Long_d:** -51,387816248 **Alt_d:** 471,228798099
w_d: 0,0689662520 **phi_d:** 0,0425209068 **kp_d:** 3,5801981669
Xc_d: 460003,244320293 **Yc_d:** 7553675,179164150 **Zc_d:** 471,228798099

Código Template: 4 **Descrição:** Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 7

29

Xt: 460003,3195490
Yt: 7553669,6435459
Zt: 515,1505152
Lat: -22,121535967
Long: -51,387815655

Imagem Esq: 0023_ESQ.jpg

Lat_e: -22,122272067 **Long_e:** -51,388238297 **Alt_e:** 477,160962113
w_e: 0,0521935017 **phi_e:** 0,0268994957 **kp_e:** 3,5756218608
Xc_e: 459959,938559676 **Yc_e:** 7553588,053867540 **Zc_e:** 477,160962113

Imagem Dir: 0023_DIR.jpg

Lat_d: -22,122268400 **Long_d:** -51,388246732 **Alt_d:** 477,160962113
w_d: 0,0521935017 **phi_d:** 0,0268994957 **kp_d:** 3,5756218608
Xc_d: 459959,067572108 **Yc_d:** 7553588,457576170 **Zc_d:** 477,160962113

Código Template: 12 **Descrição:** Proibido virar à direita

Template:

Placa Reconhecida:

pts (match): 8

30

Xt: 459952,3735931
Yt: 7553576,6891632
Zt: 561,1372943
Lat: -22,122374564
Long: -51,388311929

Imagem Esq: 0024_ESQ.jpg

Lat_e: -22,122412513 Long_e: -51,388321205 Alt_e: 478,400821672
w_e: 0,0531336705 phi_e: 0,0129248981 kp_e: 3,5778002526
Xc_e: 459951,427633533 Yc_e: 7553572,486194600 Zc_e: 478,400821672

Imagem Dir: 0024_DIR.jpg

Lat_d: -22,122408828 Long_d: -51,388329632 Alt_d: 478,400821672
w_d: 0,0531336705 phi_d: 0,0129248981 kp_d: 3,5778002526
Xc_d: 459950,557527466 Yc_d: 7553572,891799620 Zc_d: 478,400821672

Código Template: 4 Descrição: Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 10

31

Xt: 459924,8185769
Yt: 7553521,1762073
Zt: 824,1769918
Lat: -22,122875445
Long: -51,388580480

Imagem Esq: 0025_ESQ.jpg

Lat_e: -22,122498317 Long_e: -51,388370035 Alt_e: 478,835750072
w_e: -0,0001636055 phi_e: 0,0114638816 kp_e: 3,5804169337
Xc_e: 459946,415853702 Yc_e: 7553562,975737490 Zc_e: 478,835750072

Imagem Dir: 0025_DIR.jpg

Lat_d: -22,122494612 Long_d: -51,388378451 Alt_d: 478,835750072
w_d: -0,0001636055 phi_d: 0,0114638816 kp_d: 3,5804169337
Xc_d: 459945,546811951 Yc_d: 7553563,383617910 Zc_d: 478,835750072

Código Template: 7 Descrição: Proibido estacionar

Template:

Placa Reconhecida:

pts (match): 12

32

Xt: 459936,8648369
Yt: 7553554,5309151
Zt: 559,0893002
Lat: -22,122574389
Long: -51,388462852

Imagem Esq: 0026_ESQ.jpg

Lat_e: -22,122655934 Long_e: -51,388460903 Alt_e: 478,944767072
w_e: -0,0021178288 phi_e: 0,0109884952 kp_e: 3,5865633744
Xc_e: 459937,088840720 Yc_e: 7553545,505218530 Zc_e: 478,944767072

Imagem Dir: 0026_DIR.jpg

Lat_d: -22,122652181 Long_d: -51,388469295 Alt_d: 478,944767072
w_d: -0,0021178288 phi_d: 0,0109884952 kp_d: 3,5865633744
Xc_d: 459936,222322382 Yc_d: 7553545,918432720 Zc_d: 478,944767072

Código Template: 4 Descrição: Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 9

33

Xt: 459931,1294537
Yt: 7553538,7347127
Zt: 582,4022163
Lat: -22,122716963
Long: -51,388518854

Imagem Esq: 0027_ESQ.jpg

Lat_e: -22,124781171 Long_e: -51,389615361 Alt_e: 469,328313850
w_e: -0,0775221399 phi_e: 0,0312370918 kp_e: 3,5815868446
Xc_e: 459818,627925556 Yc_e: 7553309,959404090 Zc_e: 469,328313850

Imagem Dir: 0027_DIR.jpg

Lat_d: -22,124777456 Long_d: -51,389623773 Alt_d: 469,328313850
w_d: -0,0775221399 phi_d: 0,0312370918 kp_d: 3,5815868446
Xc_d: 459817,759361584 Yc_d: 7553310,368300930 Zc_d: 469,328313850

Código Template: 17 Descrição: Semáforo à frente

Template:

Placa Reconhecida:

pts (match): 6

34

Xt: 459841,3736599
Yt: 7553323,1012940
Zt: 584,5519066
Lat: -22,124662970
Long: -51,389394487

Imagem Esq: 0028_ESQ.jpg

Lat_e: -22,125419657 Long_e: -51,389964541 Alt_e: 462,818921282
w_e: -0,1069522720 phi_e: 0,0172541841 kp_e: 3,5827259825
Xc_e: 459782,797517554 Yc_e: 7553239,193420620 Zc_e: 462,818921282

Imagem Dir: 0028_DIR.jpg

Lat_d: -22,125415934 Long_d: -51,389972948 Alt_d: 462,818921282
w_d: -0,1069522720 phi_d: 0,0172541841 kp_d: 3,5827259825
Xc_d: 459781,929419935 Yc_d: 7553239,603306610 Zc_d: 462,818921282

Código Template: 8 Descrição: Proibido parar e estacionar

Template:

Placa Reconhecida:

pts (match): 4

35

Xt: 459782,3693060
Yt: 7553242,3438175
Zt: 493,6510698
Lat: -22,125391185
Long: -51,389968614

Imagem Esq: 0029_ESQ.jpg

Lat_e: -22,126157382 Long_e: -51,390367725 Alt_e: 453,241842691
w_e: -0,1472442832 phi_e: 0,0190216541 kp_e: 3,5722376102
Xc_e: 459741,426023835 Yc_e: 7553157,428265530 Zc_e: 453,241842691

Imagem Dir: 0029_DIR.jpg

Lat_d: -22,126153742 Long_d: -51,390376174 Alt_d: 453,241842691
w_d: -0,1472442832 phi_d: 0,0190216541 kp_d: 3,5722376102
Xc_d: 459740,553675006 Yc_d: 7553157,829024220 Zc_d: 453,241842691

Código Template: 6 Descrição: Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 8

36

Xt: 459741,8031798
Yt: 7553160,2224745
Zt: 465,3161525
Lat: -22,126132148
Long: -51,390363999

Imagem Esq: 0030_ESQ.jpg

Lat_e: -22,127448452 Long_e: -51,391071419 Alt_e: 447,041703699
w_e: -0,0095440173 phi_e: 0,0383938319 kp_e: 3,5702891845
Xc_e: 459669,220788086 Yc_e: 7553014,333964780 Zc_e: 447,041703699

Imagem Dir: 0030_DIR.jpg

Lat_d: -22,127444826 Long_d: -51,391079875 Alt_d: 447,041703699
w_d: -0,0095440173 phi_d: 0,0383938319 kp_d: 3,5702891845
Xc_d: 459668,347660065 Yc_d: 7553014,733023000 Zc_d: 447,041703699

Código Template: 4 Descrição: Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 34

Xt: 459667,8608911
Yt: 7553012,6736278
Zt: 490,7278308
Lat: -22,127463420
Long: -51,391084646

37

Imagem Esq: 0031_ESQ.jpg

Lat_e: -22,128824056 Long_e: -51,391825018 Alt_e: 458,228392307
w_e: 0,0954707839 phi_e: 0,0642052218 kp_e: 3,5584723543
Xc_e: 459591,894280368 Yc_e: 7552861,868880930 Zc_e: 458,228392307

Imagem Dir: 0031_DIR.jpg

Lat_d: -22,128820524 Long_d: -51,391833520 Alt_d: 458,228392307
w_d: 0,0954707839 phi_d: 0,0642052218 kp_d: 3,5584723543
Xc_d: 459591,016497813 Yc_d: 7552862,257593920 Zc_d: 458,228392307

Código Template: 4 Descrição: Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 11

Xt: 459592,8115133
Yt: 7552856,2171388
Zt: 505,1825547
Lat: -22,128875137
Long: -51,391816265

38

Imagem Esq: 0032_ESQ.jpg

Lat_e: -22,128863461 Long_e: -51,391845785 Alt_e: 458,595614206
w_e: 0,0953146632 phi_e: 0,0572083818 kp_e: 3,5523434171
Xc_e: 459589,763767569 Yc_e: 7552857,501699110 Zc_e: 458,595614206

Imagem Dir: 0032_DIR.jpg

Lat_d: -22,128859977 Long_d: -51,391854310 Alt_d: 458,595614206
w_d: 0,0953146632 phi_d: 0,0572083818 kp_d: 3,5523434171
Xc_d: 459588,883619118 Yc_d: 7552857,885024960 Zc_d: 458,595614206

Código Template: 4 Descrição: Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 11

Xt: 459581,7169322
Yt: 7552830,5721271
Zt: 608,6389059
Lat: -22,129106561
Long: -51,391924485

39

Imagem Esq: 0033_ESQ.jpg

Lat_e: -22,129120840 **Long_e:** -51,391988839 **Alt_e:** 460,774141068
w_e: 0,0748696577 **phi_e:** 0,0452374519 **kp_e:** 3,5774922138
Xc_e: 459575,084212280 **Yc_e:** 7552828,974505310 **Zc_e:** 460,774141068

Imagem Dir: 0033_DIR.jpg

Lat_d: -22,129117157 **Long_d:** -51,391997267 **Alt_d:** 460,774141068
w_d: 0,0748696577 **phi_d:** 0,0452374519 **kp_d:** 3,5774922138
Xc_d: 459574,213981312 **Yc_d:** 7552829,379842280 **Zc_d:** 460,774141068

Código Template: 6 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 10

40

Xt: 459575,4950726
Yt: 7552823,3095790
Zt: 512,3730917
Lat: -22,129172027
Long: -51,391984997

Código Template: 10 **Descrição:** Proibido retornar à esquerda

Template:

Placa Reconhecida:

pts (match): 5

41

Xt: 459573,5800241
Yt: 7552828,3535350
Zt: 468,3355310
Lat: -22,129126415
Long: -51,392003440

Imagem Esq: 0034_ESQ.jpg

Lat_e: -22,129247831 **Long_e:** -51,392059912 **Alt_e:** 461,574411617
w_e: 0,0354069913 **phi_e:** 0,0803246014 **kp_e:** 3,5723292257
Xc_e: 459567,790747374 **Yc_e:** 7552814,898995790 **Zc_e:** 461,574411617

Imagem Dir: 0034_DIR.jpg

Lat_d: -22,129244189 **Long_d:** -51,392068361 **Alt_d:** 461,574411617
w_d: 0,0354069913 **phi_d:** 0,0803246014 **kp_d:** 3,5723292257
Xc_d: 459566,918435264 **Yc_d:** 7552815,299834390 **Zc_d:** 461,574411617

Código Template: 6 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 11

42

Xt: 459569,4208208
Yt: 7552814,2316417
Zt: 488,5036210
Lat: -22,129253898
Long: -51,392044123

Imagem Esq: 0035_ESQ.jpg

Lat_e: -22,129756812 **Long_e:** -51,392336163 **Alt_e:** 463,198879027
w_e: -0,0065767108 **phi_e:** 0,0382740713 **kp_e:** 3,5730155381
Xc_e: 459539,446651000 **Yc_e:** 7552758,486659800 **Zc_e:** 463,198879027

Imagem Dir: 0035_DIR.jpg

Lat_d: -22,129753165 **Long_d:** -51,392344609 **Alt_d:** 463,198879027
w_d: -0,0065767108 **phi_d:** 0,0382740713 **kp_d:** 3,5730155381
Xc_d: 459538,574614196 **Yc_d:** 7552758,888096980 **Zc_d:** 463,198879027

Código Template: 6 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 7

43

Xt: 459504,1301474
Yt: 7552714,4185976
Zt: 798,0115728
Lat: -22,130154108
Long: -51,392679716

Código Template: 22 **Descrição:** Travessia de pedestres

Template:

Placa Reconhecida:

pts (match): 20

44

Xt: 459540,3407001
Yt: 7552759,0069578
Zt: 493,2070118
Lat: -22,129752132
Long: -51,392327481

Imagem Esq: 0036_ESQ.jpg

Lat_e: -22,129888637 **Long_e:** -51,392408665 **Alt_e:** 463,189574048
w_e: -0,0216389248 **phi_e:** 0,0409053900 **kp_e:** 3,5769831838
Xc_e: 459532,007341142 **Yc_e:** 7552743,875637790 **Zc_e:** 463,189574048

Imagem Dir: 0036_DIR.jpg

Lat_d: -22,129884959 **Long_d:** -51,392417095 **Alt_d:** 463,189574048
w_d: -0,0216389248 **phi_d:** 0,0409053900 **kp_d:** 3,5769831838
Xc_d: 459531,136903958 **Yc_d:** 7552744,280531740 **Zc_d:** 463,189574048

Código Template: 22 **Descrição:** Travessia de pedestres

Template:

Placa Reconhecida:

pts (match): 55

45

Xt: 459534,9962627
Yt: 7552745,4297095
Zt: 499,8211795
Lat: -22,129874667
Long: -51,392379643

Imagem Esq: 0037_ESQ.jpg

Lat_e: -22,130776880 **Long_e:** -51,392895933 **Alt_e:** 459,474405744
w_e: -0,0833500572 **phi_e:** 0,0435233595 **kp_e:** 3,5823587196
Xc_e: 459482,010111474 **Yc_e:** 7552645,426681880 **Zc_e:** 459,474405744

Imagem Dir: 0037_DIR.jpg

Lat_d: -22,130773159 **Long_d:** -51,392904342 **Alt_d:** 459,474405744
w_d: -0,0833500572 **phi_d:** 0,0435233595 **kp_d:** 3,5823587196
Xc_d: 459481,141863378 **Yc_d:** 7552645,836249020 **Zc_d:** 459,474405744

Código Template: 6 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 7

46

Xt: 459482,4360779
Yt: 7552648,3251105
Zt: 526,9903287
Lat: -22,130750705
Long: -51,392891730

Imagem Esq: 0038_ESQ.jpg

Lat_e: -22,130802596 Long_e: -51,392910289 Alt_e: 459,252185579
w_e: -0,0793859967 phi_e: 0,0421666280 kp_e: 3,5850193903
Xc_e: 459480,536983486 Yc_e: 7552642,576289090 Zc_e: 459,252185579

Imagem Dir: 0038_DIR.jpg

Lat_d: -22,130798855 Long_d: -51,392918687 Alt_d: 459,252185579
w_d: -0,0793859967 phi_d: 0,0421666280 kp_d: 3,5850193903
Xc_d: 459479,669828185 Yc_d: 7552642,988164900 Zc_d: 459,252185579

Código Template: 4 Descrição: Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 16

47

Xt: 459465,6416339
Yt: 7552636,5807390
Zt: 684,9580862
Lat: -22,130856414
Long: -51,393054874

Imagem Esq: 0039_ESQ.jpg

Lat_e: -22,132068951 Long_e: -51,393600732 Alt_e: 448,500424875
w_e: -0,0895420856 phi_e: 0,0400565351 kp_e: 3,5786411309
Xc_e: 459409,696014384 Yc_e: 7552502,219890260 Zc_e: 448,500424875

Imagem Dir: 0039_DIR.jpg

Lat_d: -22,132065259 Long_d: -51,393609156 Alt_d: 448,500424875
w_d: -0,0895420856 phi_d: 0,0400565351 kp_d: 3,5786411309
Xc_d: 459408,826249689 Yc_d: 7552502,626226790 Zc_d: 448,500424875

Código Template: 2 Descrição: Velocidade máxima permitida - 40 Km/h

Template:

Placa Reconhecida:

pts (match): 24

48

Xt: 459409,5257911
Yt: 7552504,7366257
Zt: 487,5989921
Lat: -22,132046210
Long: -51,393602319

Código Template: 18 Descrição: Semáforo à frente

Template:

Placa Reconhecida:

pts (match): 5

49

Xt: 459384,6151463
Yt: 7552487,8373155
Zt: 1050,3599373
Lat: -22,132198299
Long: -51,393844296

Imagem Esq: 0040_ESQ.jpg

Lat_e: -22,132237862 Long_e: -51,393691618 Alt_e: 447,053144537
w_e: -0,0829401565 phi_e: 0,0333870068 kp_e: 3,5676627473
Xc_e: 459400,371631505 Yc_e: 7552483,498875430 Zc_e: 447,053144537

Imagem Dir: 0040_DIR.jpg

Lat_d: -22,132234257 Long_d: -51,393700085 Alt_d: 447,053144537
w_d: -0,0829401565 phi_d: 0,0333870068 kp_d: 3,5676627473
Xc_d: 459399,497458395 Yc_d: 7552483,895639060 Zc_d: 447,053144537

Código Template: 8 Descrição: Proibido parar e estacionar

Template:

Placa Reconhecida:

pts (match): 7

50

Xt: 459395,8106130
Yt: 7552481,1633134
Zt: 570,3981190
Lat: -22,132258855
Long: -51,393735904

Imagem Esq: 0041_ESQ.jpg

Lat_e: -22,133633559 Long_e: -51,394460613 Alt_e: 439,234849352
w_e: 0,0191590148 phi_e: 0,0512029442 kp_e: 3,5853655560
Xc_e: 459321,468490196 Yc_e: 7552328,804127710 Zc_e: 439,234849352

Imagem Dir: 0041_DIR.jpg

Lat_d: -22,133629815 Long_d: -51,394469010 Alt_d: 439,234849352
w_d: 0,0191590148 phi_d: 0,0512029442 kp_d: 3,5853655560
Xc_d: 459320,601477525 Yc_d: 7552329,216303680 Zc_d: 439,234849352

Código Template: 10 Descrição: Proibido retornar à esquerda

Template:

Placa Reconhecida:

pts (match): 7

51

Xt: 459319,9974024
Yt: 7552328,4169061
Zt: 457,3732381
Lat: -22,133637023
Long: -51,394474888

Código Template: 25 Descrição: Proibido virar à esquerda

Template:

Placa Reconhecida:

pts (match): 9

52

Xt: 459320,8465643
Yt: 7552328,3448393
Zt: 461,6232930
Lat: -22,133637694
Long: -51,394466655

Imagem Esq: 0042_ESQ.jpg

Lat_e: -22,136109534 Long_e: -51,396014279 Alt_e: 436,577333090
w_e: -0,0159800828 phi_e: 0,0610966865 kp_e: 3,9126292319
Xc_e: 459161,959457126 Yc_e: 7552054,322816430 Zc_e: 436,577333090

Imagem Dir: 0042_DIR.jpg

Lat_d: -22,136103473 Long_d: -51,396020938 Alt_d: 436,577333090
w_d: -0,0159800828 phi_d: 0,0610966865 kp_d: 3,9126292319
Xc_d: 459161,270955988 Yc_d: 7552054,991820310 Zc_d: 436,577333090

Código Template: 4 Descrição: Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 10

53

Xt: 459163,1278606
Yt: 7552056,1247720
Zt: 453,9609868
Lat: -22,136093282
Long: -51,396002903

Imagem Esq: 0043_ESQ.jpg

Lat_e: -22,136683152 Long_e: -51,396178843 Alt_e: 438,050398491

w_e: 0,0555605520 phi_e: 0,0098470503 kp_e: 3,2516736901

Xc_e: 459145,154456513 Yc_e: 7551990,784929960 Zc_e: 438,050398491

Imagem Dir: 0043_DIR.jpg

Lat_d: -22,136682177 Long_d: -51,396188093 Alt_d: 438,050398491

w_d: 0,0555605520 phi_d: 0,0098470503 kp_d: 3,2516736901

Xc_d: 459144,200267202 Yc_d: 7551990,890394450 Zc_d: 438,050398491

Código Template: 9 Descrição: Proibido retornar à esquerda

Template:

Placa Reconhecida:

pts (match): 8

54

Xt: 459143,5894531

Yt: 7551990,4843343

Zt: 448,8803481

Lat: -22,136685831

Long: -51,396194026

Código Template: 25 Descrição: Proibido virar à esquerda

Template:

Placa Reconhecida:

pts (match): 10

55

Xt: 459143,6606047

Yt: 7551990,4530559

Zt: 447,6029020

Lat: -22,136686115

Long: -51,396193337

Imagem Esq: 0044_ESQ.jpg

Lat_e: -22,136834510 Long_e: -51,396234348 Alt_e: 439,375560790

w_e: 0,0674598911 phi_e: 0,0186487427 kp_e: 3,5110183791

Xc_e: 459139,474261322 Yc_e: 7551974,016268520 Zc_e: 439,375560790

Imagem Dir: 0044_DIR.jpg

Lat_d: -22,136831357 Long_d: -51,396243020 Alt_d: 439,375560790

w_d: 0,0674598911 phi_d: 0,0186487427 kp_d: 3,5110183791

Xc_d: 459138,579027858 Yc_d: 7551974,362905290 Zc_d: 439,375560790

Código Template: 1 Descrição: Velocidade máxima permitida - 20 Km/h

Template:

Placa Reconhecida:

pts (match): 11

56

Xt: 459138,1210956

Yt: 7551973,5535715

Zt: 446,4122128

Lat: -22,136838658

Long: -51,396247481

Código Template: 17 Descrição: Semáforo à frente

Template:

Placa Reconhecida:

pts (match): 7

57

Xt: 459138,3091884

Yt: 7551973,7995100

Zt: 444,3395052

Lat: -22,136836441

Long: -51,396245651

Código Template: 18 **Descrição:** Semáforo à frente

Template:

Placa Reconhecida:

pts (match): 10

58

Xt: 459139,0409027
Yt: 7551969,5492850
Zt: 488,8330210
Lat: -22,136874855
Long: -51,396238663

Imagem Esq: 0045_ESQ.jpg

Lat_e: -22,137062526 **Long_e:** -51,396356288 **Alt_e:** 441,380370867
w_e: 0,0649538578 **phi_e:** 0,0179581064 **kp_e:** 3,5817039762
Xc_e: 459126,965181920 **Yc_e:** 7551948,744436090 **Zc_e:** 441,380370867

Imagem Dir: 0045_DIR.jpg

Lat_d: -22,137058811 **Long_d:** -51,396364699 **Alt_d:** 441,380370867
w_d: 0,0649538578 **phi_d:** 0,0179581064 **kp_d:** 3,5817039762
Xc_d: 459126,096665849 **Yc_d:** 7551949,153434660 **Zc_d:** 441,380370867

Código Template: 1 **Descrição:** Velocidade máxima permitida - 20 Km/h

Template:

Placa Reconhecida:

pts (match): 16

59

Xt: 459126,4838222
Yt: 7551946,6117063
Zt: 472,4592614
Lat: -22,137081782
Long: -51,396361009

Código Template: 18 **Descrição:** Semáforo à frente

Template:

Placa Reconhecida:

pts (match): 7

60

Xt: 459126,9158126
Yt: 7551947,2561418
Zt: 468,7081062
Lat: -22,137075970
Long: -51,396356804

Imagem Esq: 0046_ESQ.jpg

Lat_e: -22,138631944 **Long_e:** -51,397212636 **Alt_e:** 455,448186195
w_e: 0,0483089497 **phi_e:** 0,0430040827 **kp_e:** 3,5787405639
Xc_e: 459039,109662867 **Yc_e:** 7551774,795455390 **Zc_e:** 455,448186195

Imagem Dir: 0046_DIR.jpg

Lat_d: -22,138628252 **Long_d:** -51,397221060 **Alt_d:** 455,448186195
w_d: 0,0483089497 **phi_d:** 0,0430040827 **kp_d:** 3,5787405639
Xc_d: 459038,239938580 **Yc_d:** 7551775,201878400 **Zc_d:** 455,448186195

Código Template: 5 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 8

61

Xt: 459039,5022176
Yt: 7551771,4637762
Zt: 502,8603256
Lat: -22,138662053
Long: -51,397208914

Código Template: 11 **Descrição:** Proibido virar à direita

Template:

Placa Reconhecida:

pts (match): 4

62

Xt: 459029,0124861
Yt: 7551762,0282522
Zt: 542,6946034
Lat: -22,138747047
Long: -51,397310874

Código Template: 25 **Descrição:** Proibido virar à esquerda

Template:

Placa Reconhecida:

pts (match): 7

63

Xt: 459037,6901985
Yt: 7551774,0619060
Zt: 469,0231186
Lat: -22,138638538
Long: -51,397226420

Imagem Esq: 0047_ESQ.jpg

Lat_e: -22,141553191 **Long_e:** -51,398800499 **Alt_e:** 475,169177946
w_e: 0,0550435378 **phi_e:** 0,0394333177 **kp_e:** 3,5854973973
Xc_e: 458876,214277369 **Yc_e:** 7551451,014595120 **Zc_e:** 475,169177946

Imagem Dir: 0047_DIR.jpg

Lat_d: -22,141549445 **Long_d:** -51,398808896 **Alt_d:** 475,169177946
w_d: 0,0550435378 **phi_d:** 0,0394333177 **kp_d:** 3,5854973973
Xc_d: 458875,347319047 **Yc_d:** 7551451,426885390 **Zc_d:** 475,169177946

Código Template: 4 **Descrição:** Estacionamento regulamentado - Área de emergência

Template:

Placa Reconhecida:

pts (match): 7

64

Xt: 458876,5982113
Yt: 7551449,1230659
Zt: 506,9521688
Lat: -22,141570288
Long: -51,398796824

Imagem Esq: 0048_ESQ.jpg

Lat_e: -22,143396644 **Long_e:** -51,399815426 **Alt_e:** 470,141661153
w_e: -0,1175378574 **phi_e:** 0,0521266502 **kp_e:** 3,5831085867
Xc_e: 458772,091855117 **Yc_e:** 7551246,688106750 **Zc_e:** 470,141661153

Imagem Dir: 0048_DIR.jpg

Lat_d: -22,143392917 **Long_d:** -51,399823832 **Alt_d:** 470,141661153
w_d: -0,1175378574 **phi_d:** 0,0521266502 **kp_d:** 3,5831085867
Xc_d: 458771,223914386 **Yc_d:** 7551247,098324840 **Zc_d:** 470,141661153

Código Template: 23 **Descrição:** Parada obrigatória

Template:

Placa Reconhecida:

pts (match): 6

65

Xt: 458787,0851485
Yt: 7551257,3261256
Zt: 538,0256432
Lat: -22,143300894
Long: -51,399669757

Imagem Esq: 0049_ESQ.jpg

Lat_e: -22,144508417 **Long_e:** -51,402011689 **Alt_e:** 452,410760917
w_e: -0,0249339760 **phi_e:** 0,0205684497 **kp_e:** 5,1166876384
Xc_e: 458545,941765339 **Yc_e:** 7551123,028742670 **Zc_e:** 452,410760917

Imagem Dir: 0049_DIR.jpg

Lat_d: -22,144500452 **Long_d:** -51,402008004 **Alt_d:** 452,410760917
w_d: -0,0249339760 **phi_d:** 0,0205684497 **kp_d:** 5,1166876384
Xc_d: 458546,319404435 **Yc_d:** 7551123,911346040 **Zc_d:** 452,410760917

Código Template: 3 **Descrição:** Altura máxima permitida 5,4 m

Template:

Placa Reconhecida:

pts (match): 4

66

Xt: 458546,7084854
Yt: 7551122,4208188
Zt: 466,5495004
Lat: -22,144513927
Long: -51,402004269

Imagem Esq: 0050_ESQ.jpg

Lat_e: -22,142196165 **Long_e:** -51,405980025 **Alt_e:** 445,986066484
w_e: 0,0097115863 **phi_e:** 0,0636760583 **kp_e:** 5,2956899930
Xc_e: 458136,052150202 **Yc_e:** 7551377,884099560 **Zc_e:** 445,986066484

Imagem Dir: 0050_DIR.jpg

Lat_d: -22,142188939 **Long_d:** -51,405974877 **Alt_d:** 445,986066484
w_d: 0,0097115863 **phi_d:** 0,0636760583 **kp_d:** 5,2956899930
Xc_d: 458136,580901037 **Yc_d:** 7551378,685362660 **Zc_d:** 445,986066484

Código Template: 2 **Descrição:** Velocidade máxima permitida - 40 Km/h

Template:

Placa Reconhecida:

pts (match): 9

67

Xt: 458137,9097220
Yt: 7551377,1873796
Zt: 466,4259028
Lat: -22,142202504
Long: -51,405962030

Imagem Esq: 0051_ESQ.jpg

Lat_e: -22,142054190 **Long_e:** -51,406194613 **Alt_e:** 446,533354219
w_e: 0,0197187182 **phi_e:** 0,0629445330 **kp_e:** 5,2851774186
Xc_e: 458113,881964257 **Yc_e:** 7551393,540225150 **Zc_e:** 446,533354219

Imagem Dir: 0051_DIR.jpg

Lat_d: -22,142046914 **Long_d:** -51,406189547 **Alt_d:** 446,533354219
w_d: 0,0197187182 **phi_d:** 0,0629445330 **kp_d:** 5,2851774186
Xc_d: 458114,402262693 **Yc_d:** 7551394,347002400 **Zc_d:** 446,533354219

Código Template: 15 **Descrição:** Interseção em círculo

Template:

Placa Reconhecida:

pts (match): 6

68

Xt: 458115,2307943
Yt: 7551392,4450617
Zt: 456,1517738
Lat: -22,142064116
Long: -51,406181561

Imagem Esq: 0052_ESQ.jpg

Lat_e: -22,140916595 **Long_e:** -51,407937499 **Alt_e:** 444,574213265
w_e: -0,1331934301 **phi_e:** 0,0213951554 **kp_e:** 5,3449415911
Xc_e: 457933,818380440 **Yc_e:** 7551518,979195170 **Zc_e:** 444,574213265

Imagem Dir: 0052_DIR.jpg

Lat_d: -22,140909614 **Long_d:** -51,407931975 **Alt_d:** 444,574213265
w_d: -0,1331934301 **phi_d:** 0,0213951554 **kp_d:** 5,3449415911
Xc_d: 457934,385937640 **Yc_d:** 7551519,753455350 **Zc_d:** 444,574213265

Código Template: 23 **Descrição:** Parada obrigatória

Template:

Placa Reconhecida:

pts (match): 18

69

Xt: 457932,8707309
Yt: 7551522,5014988
Zt: 462,6088139
Lat: -22,140884751
Long: -51,407946597

Imagem Esq: 0053_ESQ.jpg

Lat_e: -22,138969419 **Long_e:** -51,406765263 **Alt_e:** 439,597708301
w_e: 0,0263893969 **phi_e:** 0,0386065404 **kp_e:** 0,6247595867
Xc_e: 458054,123218878 **Yc_e:** 7551734,835832230 **Zc_e:** 439,597708301

Imagem Dir: 0053_DIR.jpg

Lat_d: -22,138974510 **Long_d:** -51,406757726 **Alt_d:** 439,597708301
w_d: 0,0263893969 **phi_d:** 0,0386065404 **kp_d:** 0,6247595867
Xc_d: 458054,901878489 **Yc_d:** 7551734,274326030 **Zc_d:** 439,597708301

Código Template: 11 **Descrição:** Proibido virar à direita

Template:

Placa Reconhecida:

pts (match): 9

70

Xt: 458062,9553862
Yt: 7551739,9257137
Zt: 508,5318330
Lat: -22,138923649
Long: -51,406679483

Imagem Esq: 0054_ESQ.jpg

Lat_e: -22,138636333 **Long_e:** -51,406496567 **Alt_e:** 441,532403979
w_e: 0,0495177474 **phi_e:** 0,0350566069 **kp_e:** 0,5968740324
Xc_e: 458081,733062460 **Yc_e:** 7551771,779203520 **Zc_e:** 441,532403979

Imagem Dir: 0054_DIR.jpg

Lat_d: -22,138641227 **Long_d:** -51,406488881 **Alt_d:** 441,532403979
w_d: 0,0495177474 **phi_d:** 0,0350566069 **kp_d:** 0,5968740324
Xc_d: 458082,527075229 **Yc_d:** 7551771,239626160 **Zc_d:** 441,532403979

Código Template: 6 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 7

71

Xt: 458085,0651469
Yt: 7551771,4363836
Zt: 488,2253478
Lat: -22,138639510
Long: -51,406464264

Imagem Esq: 0055_ESQ.jpg

Lat_e: -22,137452780 **Long_e:** -51,405766647 **Alt_e:** 456,275307194
w_e: 0,0815093158 **phi_e:** 0,0352267024 **kp_e:** 0,3861349459
Xc_e: 458156,654257630 **Yc_e:** 7551902,987526300 **Zc_e:** 456,275307194

Imagem Dir: 0055_DIR.jpg

Lat_d: -22,137456068 **Long_d:** -51,405758033 **Alt_d:** 456,275307194
w_d: 0,0815093158 **phi_d:** 0,0352267024 **kp_d:** 0,3861349459
Xc_d: 458157,543574363 **Yc_d:** 7551902,625979970 **Zc_d:** 456,275307194

Código Template: 12 **Descrição:** Proibido virar à direita

Template:

Placa Reconhecida:

pts (match): 8

72

Xt: 458156,7684098
Yt: 7551900,9857243
Zt: 469,7162761
Lat: -22,137470868
Long: -51,405765592

Imagem Esq: 0056_ESQ.jpg

Lat_e: -22,135509251 **Long_e:** -51,404888501 **Alt_e:** 469,488420258
w_e: -0,0491233142 **phi_e:** -0,0159044067 **kp_e:** 0,3873084959
Xc_e: 458246,638398955 **Yc_e:** 7552118,357815450 **Zc_e:** 469,488420258

Imagem Dir: 0056_DIR.jpg

Lat_d: -22,135512548 **Long_d:** -51,404879891 **Alt_d:** 469,488420258
w_d: -0,0491233142 **phi_d:** -0,0159044067 **kp_d:** 0,3873084959
Xc_d: 458247,527290783 **Yc_d:** 7552117,995225710 **Zc_d:** 469,488420258

Código Template: 16 **Descrição:** Interseção em círculo

Template:

Placa Reconhecida:

pts (match): 15

73

Xt: 458245,9359390
Yt: 7552118,1759058
Zt: 487,5498191
Lat: -22,135510878
Long: -51,404895317

Imagem Esq: 0057_ESQ.jpg

Lat_e: -22,135333365 **Long_e:** -51,404802324 **Alt_e:** 468,639069025
w_e: -0,0654713173 **phi_e:** 0,0209221402 **kp_e:** 0,3762981111
Xc_e: 458255,473488467 **Yc_e:** 7552137,850263250 **Zc_e:** 468,639069025

Imagem Dir: 0057_DIR.jpg

Lat_d: -22,135336574 **Long_d:** -51,404793676 **Alt_d:** 468,639069025
w_d: -0,0654713173 **phi_d:** 0,0209221402 **kp_d:** 0,3762981111
Xc_d: 458256,366318588 **Yc_d:** 7552137,497482330 **Zc_d:** 468,639069025

Código Template: 12 **Descrição:** Proibido virar à direita

Template:

Placa Reconhecida:

pts (match): 5

74

Xt: 458256,6478270
Yt: 7552139,4050340
Zt: 495,6134943
Lat: -22,135319347
Long: -51,404790897

Imagem Esq: 0058_ESQ.jpg

Lat_e: -22,134600456 **Long_e:** -51,404475852 **Alt_e:** 464,308923367
w_e: -0,0652946177 **phi_e:** 0,0058132646 **kp_e:** 0,3496761402
Xc_e: 458288,924968883 **Yc_e:** 7552219,065450200 **Zc_e:** 464,308923367

Imagem Dir: 0058_DIR.jpg

Lat_d: -22,134603449 **Long_d:** -51,404467115 **Alt_d:** 464,308923367
w_d: -0,0652946177 **phi_d:** 0,0058132646 **kp_d:** 0,3496761402
Xc_d: 458289,826873249 **Yc_d:** 7552218,736560380 **Zc_d:** 464,308923367

Código Template: 23 **Descrição:** Parada obrigatória

Template:

Placa Reconhecida:

pts (match): 9

75

Xt: 458290,1681731
Yt: 7552219,2373798
Zt: 469,2377877
Lat: -22,134598932
Long: -51,404463792

Imagem Esq: 0059_ESQ.jpg

Lat_e: -22,133599113 **Long_e:** -51,404008579 **Alt_e:** 459,166159183
w_e: -0,0500887640 **phi_e:** 0,0622940610 **kp_e:** 0,3502759252
Xc_e: 458336,818056230 **Yc_e:** 7552330,031972130 **Zc_e:** 459,166159183

Imagem Dir: 0059_DIR.jpg

Lat_d: -22,133602111 **Long_d:** -51,403999844 **Alt_d:** 459,166159183
w_d: -0,0500887640 **phi_d:** 0,0622940610 **kp_d:** 0,3502759252
Xc_d: 458337,719763171 **Yc_d:** 7552329,702541420 **Zc_d:** 459,166159183

Código Template: 5 **Descrição:** Estacionamento regulamentado

Template:

Placa Reconhecida:

pts (match): 9

76

Xt: 458341,0599696
Yt: 7552332,7289768
Zt: 494,2704945
Lat: -22,133574849
Long: -51,403967376

Código Template: 7 **Descrição:** Proibido estacionar

Template:

Placa Reconhecida:

pts (match): 7

77

Xt: 458340,4876593
Yt: 7552332,3249055
Zt: 492,8591812
Lat: -22,133578486
Long: -51,403972936

Imagem Esq: 0060_ESQ.jpg

Lat_e: -22,133538361 **Long_e:** -51,403979318 **Alt_e:** 458,855829557
w_e: -0,0562725852 **phi_e:** 0,0539845248 **kp_e:** 0,4224786579
Xc_e: 458339,817703278 **Yc_e:** 7552336,764634390 **Zc_e:** 458,855829557

Imagem Dir: 0060_DIR.jpg

Lat_d: -22,133541938 **Long_d:** -51,403970838 **Alt_d:** 458,855829557
w_d: -0,0562725852 **phi_d:** 0,0539845248 **kp_d:** 0,4224786579
Xc_d: 458340,693295699 **Yc_d:** 7552336,371012850 **Zc_d:** 458,855829557

Código Template: 7 **Descrição:** Proibido estacionar

Template:

Placa Reconhecida:

pts (match): 10

78

Xt: 458343,3798947
Yt: 7552339,4201799
Zt: 492,4628509
Lat: -22,133514455
Long: -51,403944708

Total de Placas Reconhecidas: 78

Apêndice C – Imagens de entrada com as placas não detectadas pela metodologia proposta

Neste Apêndice, são mostradas as imagens de entrada (lado direito da sequência), obtidas por meio de levantamento com o SIMMOV (Sistema Móvel de Mapeamento) (seção 5.2), que possuem placas não detectadas pela metodologia proposta. Nas imagens, as placas não detectadas estão demarcadas por um retângulo. No experimento realizado (Capítulo 6), usando a metodologia proposta, das 91 placas de sinalização existentes nas imagens usadas no experimento, 12 placas não foram detectadas. As placas não detectadas são: “Proibido virar à esquerda” (Figura C.1, Figura C.3 e Figura C.4), “Proibido retornar à esquerda” (Figura C.2, Figura C.3 e Figura C.10), “Estacionamento regulamentado” (Figura C.5, Figura C.6, Figura C.7 e Figura C.8), “Siga em frente ou à direita” (Figura C.9), “Parada obrigatória” (Figura C.11).

Figura C.1: Imagem com a placa “Proibido virar à esquerda” não detectada.

Figura C.2: Imagem com a placa “Proibido retornar à esquerda” não detectada.

Figura C.3: Imagem com as placas “Proibido virar à esquerda” e “Proibido retornar à esquerda” não detectadas.

Figura C.4: Imagem com a placa “Proibido virar à esquerda” não detectada.

Figura C.5: Imagem com a placa “Estacionamento regulamentado” não detectada.

Figura C.6: Imagem com a placa “Estacionamento regulamentado” não detectada.

Figura C.7: Imagem com a placa “Estacionamento regulamentado” não detectada.

Figura C.8: Imagem com a placa “Estacionamento regulamentado” não detectada.

Figura C.9: Imagem com a placa “Siga em frente ou à direita” não detectada.

Figura C.10: Imagem com a placa “Proibido retornar à esquerda” não detectada.

Figura C.11: Imagem com a placa “Parada obrigatória” não detectada.