



Tiago Filipe Crispim  
Pereira

Localização do ATLASCAR2 com recurso a  
imagens de satélite ou aéreas





**Tiago Filipe Crispim  
Pereira**

**Localização do ATLASCAR2 com recurso a  
imagens de satélite ou aéreas**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia de Automação Industrial, realizada sob orientação científica de Miguel Armando Riem de Oliveira, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.



**O júri / The jury**

Presidente / President

**Prof. Doutor Paulo Bacelar Reis Pedreiras**  
Professor Auxiliar da Universidade de Aveiro

Vogais / Committee

**Prof. Doutor Miguel Armando Riem de Oliveira**  
Professor Auxiliar da Universidade de Aveiro (orientador)

**Prof. Doutor Paulo Miguel de Jesus Dias**  
Professor Auxiliar da Universidade de Aveiro



## Palavras-chave

Condução Autónoma; Inverse Perspective Mapping; Geolocalização; Template Matching; Chamfer Matching

## Resumo

Os sistemas de condução autónoma têm sido fortemente explorados devido aos seus potenciais benefícios. De modo a prosseguir nesse sentido, esta dissertação pretende contribuir para o desenvolvimento desses sistemas, mais especificamente na área da navegação. Numa primeira fase, focou-se no desenvolvimento de um algoritmo, juntamente com as suas variantes, que converte imagens com perspetiva para imagens sem perspetiva. No contexto que se pretende aplicar o referido algoritmo, as imagens com perspetivas são referentes a imagens adquiridas de câmaras montadas no ATLASCAR2 que normalmente ilustram uma estrada, por outro lado as imagens sem perspetiva referem-se às mesmas estradas agora aparentemente vistas de cima. Este algoritmo é denominado por *Inverse Perspective Mapping* (IPM) e considerou-se, ao longo do seu desenvolvimento, a sua aplicabilidade em tempo real. Esta investigação propõe utilizar a imagem resultante como *input* no algoritmo de geolocalização. Numa segunda fase foi realizada uma investigação sobre os métodos de geolocalização já existentes. Eventualmente construíram-se os métodos *Template Matching* e *Chamfer Matching* que pretendem localizar uma imagem IPM numa imagem de satélite. Validaram-se os algoritmos construídos, em primeiro lugar, em condições simuladas, de modo a analisar melhor o comportamento face à variação de determinadas propriedades. Finalmente, testaram-se os algoritmos em condições reais, onde se pretende localizar uma porção de estrada adquirida localmente numa imagem de satélite. Verificou-se, por observação dos resultados obtidos, que os algoritmos de geolocalização desenvolvidos nesta investigação não são adequados para a aplicação em condições reais.





**Keywords**

Autonomous Driving; Inverse Perspective Mapping; Geolocation; Template Matching; Chamfer Matching

**Abstract**

Nowadays, autonomous driving systems have been heavily explored for its potential benefits. In order to proceed accordingly, this dissertation intends to contribute to the development of those type of systems, more specifically in the navigation thematics. In the first phase, a great effort was put in developing an algorithm with its variations that converts images with perspective to images without said perspective. In the context in which the algorithm will be applied the images with perspective refer to images acquired directly from cameras mounted on the ATLASCAR2 that show portions of the road. Images without perspective refer to the same portions of road as before but that are now apparently visualized from above. This algorithm is named Inverse Perspective Mapping (IPM) and during development its usage in Real Time applications was always kept in mind. It was intended since the beginning to use these images as inputs for the geolocation algorithm. In the second phase of this dissertation it was investigated already developed geolocation methods. Eventually Template Matching and Chamfer Matching were constructed for the purpose of locating an IPM image on a satellite image. These algorithms were firstly validated on simulated conditions in order to observe the behaviour of the resulting values according to the variation of certain parameters. Lastly, the constructed algorithms were tested in real conditions. In these conditions it was intended to locate an IPM image, that was acquired locally, in a satellite image. By observation of the results returned by the developed geolocation methods in this investigation it can be inferred that these are not adequate for real world applications.



# Lista de Siglas e Acrónimos

<b>CCM</b>	<i>Classic Chamfer Matching</i>
<b>CM</b>	<i>Chamfer Matching</i>
<b>CNN</b>	<i>Convolutional Neural Network</i>
<b>CTM</b>	<i>Classic Template Matching</i>
<b>DEM</b>	Departamento de Engenharia Mecânica
<b>DGPS</b>	<i>Differential Global Positioning System</i>
<b>DT</b>	<i>Distance Function</i>
<b>DVL</b>	<i>Doppler Velocity Log</i>
<b>FFT</b>	<i>Fourier Frequency Transform</i>
<b>GCM</b>	<i>Gaussian Chamfer Matching</i>
<b>GOLD</b>	<i>Generic Obstacle and Lane Detection</i>
<b>GPS</b>	<i>Global Positioning System</i>
<b>GTM</b>	<i>Gaussian Template Matching</i>
<b>INS</b>	<i>Inertial Navigation System</i>
<b>IPM</b>	<i>Inverse Perspective Mapping</i>
<b>LRF</b>	<i>Laser Range Finder</i>
<b>MM-IPM</b>	<i>Multi-Modal Inverse Perspective Mapping</i>
<b>MWLD</b>	<i>Modified Weber's Local Descriptor</i>
<b>NCC</b>	<i>Normalized Cross-Correlation</i>
<b>PAPRICA</b>	<i>Parallel Processor for Image Checking and Analysis</i>
<b>PA</b>	Passos de Ângulos
<b>PP</b>	Passos de Píxeis
<b>ROI</b>	<i>Region of Interest</i>

**ROS** *Robot Operating System*

**RSF** *Relative Scale Factor*

**SNN** *Siamese Neural Network*

**SV** *Street View*

**TM** *Template Matching*

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Estrutura da tese . . . . .	3
<b>2</b>	<b>Estado de arte</b>	<b>5</b>
2.1	Inverse Perspective Mapping . . . . .	5
2.2	Multi-Modal Inverse Perspective Mapping . . . . .	13
2.3	Pré-processamentos e otimizações do método Inverse Perspective Mapping	16
2.4	Aplicações do método Inverse Perspective Matching . . . . .	21
2.4.1	Deteção de estrada . . . . .	21
2.4.2	Análise de tráfego . . . . .	23
2.5	Geolocalização . . . . .	24
2.5.1	Template Matching e variantes . . . . .	24
2.5.2	Geolocalização por comparação de imagens de diferentes perspectivas . . . . .	27
2.6	Apreciação crítica . . . . .	30
<b>3</b>	<b>Metodologia</b>	<b>31</b>
3.1	Inverse Perspective Mapping . . . . .	31
3.2	Geolocalização . . . . .	41
3.2.1	Imagem Satélite . . . . .	41
3.2.2	Template Matching . . . . .	42
3.2.3	Chamfer Matching . . . . .	44
<b>4</b>	<b>Resultados e Discussão</b>	<b>49</b>
4.1	Performance do método Inverse Perspective Mapping . . . . .	49
4.1.1	Variação da resolução . . . . .	51
4.1.2	Variação da polígono de percepção . . . . .	55
4.1.3	Variação de RSF . . . . .	58
4.1.4	Escolha do melhor método IPM . . . . .	60
4.2	Geolocalização . . . . .	61
4.2.1	Classic Template Matching e Gaussian Template Matching . . . . .	62
4.2.2	Classic Chamfer Matching e Gaussian Chamfer Matching . . . . .	77
4.2.3	Análise dos algoritmos em condições reais . . . . .	81
<b>5</b>	<b>Conclusão</b>	<b>89</b>



# Lista de Tabelas

3.1	Modo como cada método IPM foi construída. . . . .	38
4.1	Tempos de execução dos algoritmos GTM e GCM, com a variação do parâmetro PP. . . . .	80
5.1	Evolução da média e desvio padrão dos tempos de execução, na construção da máscara representativa de área sem obstáculos, consoante a variação da resolução da imagem original. Dados utilizados dos métodos IPM C, IPM D e IPM E. . . . .	95
5.2	Variação dos tempos de execução de cada fase do método IPM A, consoante a variação da resolução da imagem a ser utilizada. . . . .	95
5.3	Variação dos tempos de execução de cada fase do método IPM B, consoante a variação da resolução da imagem a ser utilizada. . . . .	96
5.4	Variação dos tempos de execução de cada fase do método IPM C, consoante a variação da resolução da imagem a ser utilizada. . . . .	96
5.5	Variação dos tempos de execução de cada fase do método IPM D, consoante a variação da resolução da imagem a ser utilizada. . . . .	97
5.6	Variação dos tempos de execução de cada fase do método IPM E, consoante a variação da resolução da imagem a ser utilizada. . . . .	97
5.7	Variação dos tempos de execução de cada fase do método IPM F, consoante a variação da resolução da imagem a ser utilizada. . . . .	98
5.8	Variação dos tempos de execução de cada fase do método IPM G, consoante a variação da resolução da imagem a ser utilizada. . . . .	98
5.9	Variação dos tempos de execução de cada fase do método IPM A, consoante a variação da área do polígono da imagem a ser utilizada. . . . .	99
5.10	Variação dos tempos de execução de cada fase do método IPM B, consoante a variação da área do polígono da imagem a ser utilizada. . . . .	99
5.11	Variação dos tempos de execução de cada fase do método IPM C, consoante a variação da área do polígono da imagem a ser utilizada. . . . .	100
5.12	Variação dos tempos de execução de cada fase do método IPM D, consoante a variação da área do polígono da imagem a ser utilizada. . . . .	100
5.13	Variação dos tempos de execução de cada fase do método IPM E, consoante a variação da área do polígono da imagem a ser utilizada. . . . .	101
5.14	Variação dos tempos de execução de cada fase do método IPM F, consoante a variação da área do polígono da imagem a ser utilizada. . . . .	101
5.15	Variação dos tempos de execução de cada fase do método IPM G, consoante a variação da área do polígono da imagem a ser utilizada. . . . .	102

5.16	Variação dos tempos de execução de cada fase do método IPM A, consoante a variação da escala RSF. . . . .	102
5.17	Variação dos tempos de execução de cada fase do método IPM B, consoante a variação da escala RSF. . . . .	103
5.18	Variação dos tempos de execução de cada fase do método IPM C, consoante a variação da escala RSF. . . . .	103
5.19	Variação dos tempos de execução de cada fase do método IPM D, consoante a variação da escala RSF. . . . .	104
5.20	Variação dos tempos de execução de cada fase do método IPM E, consoante a variação da escala RSF. . . . .	104
5.21	Variação dos tempos de execução de cada fase do método IPM F, consoante a variação da escala RSF. . . . .	105
5.22	Variação dos tempos de execução de cada fase do método IPM G, consoante a variação da escala RSF. . . . .	105
5.23	Tempos de execução de cada fase de cada método. . . . .	105



# Lista de Figuras

2.1	Posição e orientação da câmara no sistema de coordenadas do mundo. . . . .	6
2.2	Relação entre os diferentes referenciais através do <i>Pinhole Camera Model</i> . . . . .	7
2.3	Funcionamento do método IPM. . . . .	9
2.4	Representação simbólica das várias zonas de uma imagem IPM. . . . .	10
2.5	Processo de agregação de duas imagens IPM. . . . .	12
2.6	Análise da precisão da imagem resultante do método IPM. . . . .	13
2.7	Resultados e comparação do método proposto com o método IPM clássico. . . . .	14
2.8	Representação do método de deteção de obstáculo do sistema GOLD. . . . .	15
2.9	Exemplificação de um cenário de uma estrada com vários carros. . . . .	15
2.10	Diagrama representativo do sistema GOLD. . . . .	17
2.11	Esquema demonstrativo de como a imagem é segmentada de maneira a reduzir o custo computacional. . . . .	17
2.12	Erro causado por variações de inclinações e rotações. . . . .	18
2.13	Resultados de filtros de <i>Gabor</i> . . . . .	19
2.14	Demonstração do processo, faseado, para a determinação do ponto de convergência. . . . .	20
2.15	Resultados intermédios de um “Temporal Blur”. . . . .	22
2.16	Filtragem e binarização de imagem. . . . .	22
2.17	Esquema de deteção de marcadores rodoviários segmentado nas várias fases. . . . .	23
2.18	Deteção de tráfego. . . . .	24
2.19	Pseudocódigo de uma função de distância. . . . .	27
2.20	Aplicação do DT Chamfer 3-4. . . . .	27
2.21	Processo de segmentação de marcadores de faixas. . . . .	28
2.22	Processo de estimação da localização do veículo. . . . .	29
3.1	Exemplo de um painel axadrezado adequado para a calibração da câmara. . . . .	32
3.2	Exemplo de um comando para iniciar a calibração da câmara. . . . .	32
3.3	Comportamento da interface a novas poses do painel axadrezado. . . . .	33
3.4	Referenciais definidos no sistema ATLASCAR2. . . . .	33
3.5	Vista frontal do ATLASCAR2. . . . .	35
3.6	Sobreposição do polígono de área sem obstáculos na imagem IPM final. . . . .	35
3.7	Conversão do polígono de área sem obstáculos para o referencial da imagem. . . . .	36
3.8	Máscaras criadas pela fase “Masking”. . . . .	36
3.9	Resultados da fase “Interpolation/Smoothing”. . . . .	37
3.10	Estrutura do método IPM A. . . . .	39

3.11	Estrutura do método IPM B. . . . .	39
3.12	Estrutura dos métodos IPM C, IPM D, IPM E e IPM G. . . . .	40
3.13	Estrutura dos métodos IPM F. . . . .	40
3.14	Processo de criação de uma imagem no formato “tiff”. . . . .	42
3.15	Rotação do <i>template</i> . . . . .	43
3.16	Exemplo de matrizes inerentes ao método <i>Template Matching</i> . . . . .	44
3.17	Procedimento de adição de matriz de <i>padding</i> de dimensões adequadas. . . . .	46
3.18	Exemplo de matrizes intrínsecas ao método <i>Chamfer Matching</i> . . . . .	47
4.1	Imagem adquirida do sistema ATLASCAR2 quando este está estacionado na oficina do DEM. . . . .	50
4.2	Evolução da média e desvio padrão dos tempos de execução, na construção da máscara representativa da área sem obstáculos, consoante a variação da resolução da imagem original. . . . .	51
4.3	Evolução dos tempos de execução da fase “Masking” entre todos os métodos, face à variação da resolução da imagem original. . . . .	52
4.4	Evolução dos tempos de execução da fase “Interpolation/Smoothing” entre todos os métodos, face à variação da resolução da imagem original. . . . .	53
4.5	Evolução dos tempos de execução da fase “Indexing” entre todos os métodos, face à variação da resolução da imagem original. . . . .	54
4.6	Evolução dos tempos de execução da fase “Mapping” entre todos os métodos, face à variação da resolução da imagem original. . . . .	55
4.7	Evolução dos tempos de execução da fase “Interpolation/Smoothing” de todos os métodos, face à variação da área do polígono de percepção. . . . .	56
4.8	Evolução dos tempos de execução da fase “Masking” de todos os métodos, face à variação da área do polígono de percepção. . . . .	56
4.9	Evolução dos tempos de execução da fase “Indexing” de todos os métodos, face à variação da área do polígono de percepção. . . . .	57
4.10	Evolução dos tempos de execução da fase “Mapping” de todos os métodos, face à variação da área do polígono de percepção . . . . .	57
4.11	Evolução dos tempos de execução da fase “Masking” de todos os métodos, face à variação do RSF. . . . .	58
4.12	Evolução dos tempos de execução da fase “Indexing” de todos os métodos, face à variação do RSF. . . . .	59
4.13	Evolução do tempo de execução da fase “Mapping” de todos os métodos, face à variação do RSF. . . . .	59
4.14	Evolução dos tempos de execução da fase “Interpolation” de todos os métodos, face à variação do RSF. . . . .	60
4.15	Tempos de execução de cada fase de cada método IPM desenvolvido. . . . .	61
4.16	Tipos de <i>templates</i> utilizados na validação dos algoritmos de localização. . . . .	62
4.17	Distâncias euclidianas entre a posição calculada pelos algoritmos CTM e GTM e a posição <i>ground truth</i> , face à variação do ângulo a que o <i>template</i> é rodado. Com os parâmetros PP=1 e PA=1. . . . .	64
4.18	Desvio padrão das distâncias euclidianas entre a posição calculada pelos algoritmos CTM e GTM e a posição <i>ground truth</i> , face à variação do ângulo a que o <i>template</i> é rodado. Com os parâmetros PP=1 e PA=1. . . . .	65

4.19	Valores de correlação de CTM e GTM, face à variação do ângulo $\alpha$ que o <i>template</i> é rodado. Com os parâmetros PP=1 e PA=1. . . . .	65
4.20	Distância euclidiana entre a posição calculada pelos algoritmos CTM e GTM e a posição <i>ground truth</i> , face à variação do ângulo $\alpha$ que o <i>template</i> é rodado. Com os parâmetros PP=5 e PA=4. . . . .	66
4.21	Desvio padrão das distâncias euclidianas entre a posição calculada pelos algoritmos CTM e GTM e a posição <i>ground truth</i> , face à variação do ângulo $\alpha$ que o <i>template</i> é rodado. Com os parâmetros PP=5 e PA=4. . . . .	67
4.22	Valores de correlação de CTM e GTM, face à variação do ângulo $\alpha$ que o <i>template</i> é rodado. Com os parâmetros PP=5 e PA=4. . . . .	67
4.23	Distâncias euclidianas do algoritmo CTM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado. . . . .	68
4.24	Distâncias euclidianas do algoritmo GTM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado. . . . .	69
4.25	Desvio padrão das distâncias euclidianas do algoritmo CTM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado. . . . .	69
4.26	Desvio padrão das distâncias euclidianas do algoritmo GTM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado. . . . .	70
4.27	Desvio padrão do algoritmo CTM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado (Zoom sobre a figura 4.24). . . . .	70
4.28	Desvio padrão do algoritmo GTM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado (Zoom sobre a figura 4.25). . . . .	71
4.29	Distâncias euclidianas do algoritmo GTM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado (Zoom sobre a figura 4.26). . . . .	71
4.30	Comparação dos valores de correlação para valores de ângulos centrado ou desfasado 180 graus do ângulo <i>ground truth</i> (Algoritmo CTM e PP=1). . . . .	73
4.31	Comparação dos valores de correlação para valores de ângulos centrado ou desfasado 180 graus do ângulo <i>ground truth</i> (Algoritmo CTM e PP=5). . . . .	73
4.32	Comparação dos valores de correlação para valores de ângulos centrado ou desfasado 180 graus do ângulo <i>ground truth</i> (Algoritmo GTM e PP=1). . . . .	74
4.33	Comparação dos valores de correlação para valores de ângulos centrado ou desfasado 180 graus do ângulo <i>ground truth</i> (Algoritmo GTM e PP=5). . . . .	74
4.34	Distâncias euclidianas retornadas dos algoritmos GTM e CTM, face à variação dos parâmetros PP e PA . . . . .	75
4.35	Desvio padrão das distâncias euclidianas retornadas dos algoritmos GTM e CTM, face à variação dos parâmetros PP e PA . . . . .	76
4.36	Distâncias euclidianas do algoritmo CCM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado . . . . .	77
4.37	Distâncias euclidianas do algoritmo GCM, face à variação do parâmetro PP e do ângulo $\alpha$ que o <i>template</i> é rodado. . . . .	78
4.38	<i>Templates</i> que geram situações diferentes nos algoritmos <i>Chamfer Matching</i> . . . . .	79
4.39	Distâncias euclidianas do algoritmo CCM, considerando apenas <i>templates</i> adequados, face à variação do ângulo $\alpha$ que este é rodado. . . . .	79
4.40	Distâncias euclidianas do algoritmo GCM, considerando apenas <i>templates</i> adequados, face à variação do ângulo $\alpha$ que este é rodado. . . . .	80
4.41	<i>Templates</i> utilizados na validação dos algoritmos em condições reais. . . . .	81

4.42	Distâncias euclidianas obtidas do algoritmo CTM sobre cada imagem de teste, face à variação do ângulo a que o <i>template</i> é rodado. . . . .	82
4.43	Distâncias euclidianas obtidas do algoritmo GTM sobre cada imagem de teste, face à variação do ângulo a que o <i>template</i> é rodado. . . . .	83
4.44	Valores de correlação obtidas do algoritmo CTM sobre cada imagem de teste, face à variação do ângulo a que o <i>template</i> é rodado. . . . .	83
4.45	Valores de correlação obtidas do algoritmo GTM sobre cada imagem de teste, face à variação do ângulo a que o <i>template</i> é rodado. . . . .	84
4.46	Distâncias euclidianas obtidas do algoritmo CCM sobre cada imagem de teste, face à variação do ângulo a que o <i>template</i> é rodado. . . . .	85
4.47	Distâncias euclidianas obtidas do algoritmo GCM sobre cada imagem de teste, face à variação do ângulo a que o <i>template</i> é rodado. . . . .	86
4.48	Valores de correlação obtidas do algoritmo CCM sobre cada imagem de teste, face à variação do ângulo a que o <i>template</i> é rodado. . . . .	86
4.49	Valores de correlação obtidas do algoritmo GCM sobre cada imagem de teste, face à variação do ângulo a que o <i>template</i> é rodado. . . . .	87

# Capítulo 1

## Introdução

A presente dissertação foi elaborada no âmbito do mestrado em Engenharia de Automação Industrial. A condução autónoma será o tópico fulcral desta investigação, particularmente ao nível da navegação, dada a importância da mobilidade quotidiana numa sociedade em crescimento. Atualmente tem-se verificado um maior foco de interesse sobre sistemas de condução autónoma. A eliminação do fator humano, responsável primário pelos acidentes rodoviários, contribui para o aumento da eficiência dos meios de transporte automóveis e segurança dos passageiros e cargas.

### 1.1 Motivação

A condução autónoma possui um controlo mais eficiente de tempo e energia empregue no transporte, apresentando assim uma vantagem relativamente à condução autónoma. Em termos de segurança, as principais causas de acidentes provêm de erros humanos [7]. Apesar de existirem sistemas de deteção de distração do condutor, [8], estes necessitam ainda de um aumento na precisão para serem funcionais em situações reais. Mas indubitavelmente, o melhor método para evitar falhas humanas é não permitir que humanos conduzam. Tendo isto em conta, pretende-se com este estudo contribuir para o avanço do conhecimento neste sentido.

No contexto de condução, os sistemas de navegação atuais baseiam-se principalmente no *Global Positioning System* (GPS) e as melhorias relacionadas, como por exemplo o *Differential Global Positioning System* (DGPS), o *Inertial Navigation System* (INS), o *Doppler Velocity Log* (DVL). Contudo, estes métodos também possuem falhas, como certas situações em que o sinal de GPS é fraco ou ausente. Estas ocorrências são normalmente associadas à ofuscação do sinal de GPS devido a prédios altos, árvores ou terrenos montanhosos. Foram conduzidos vários estudos, como o de Xiong et. al. [9], com o intuito de mitigar estas fragilidades. Contudo nas situações mais agravadas não é possível uma condução absolutamente automática, surgindo então a necessidade de outros sistemas que substituam ou complementam o existente atualmente. Propõem-se, portanto, a utilização de técnicas de comparação entre imagens para geolocalizar o veículo.

Para uma comparação entre imagens a uma escala global, é necessária a transformação das imagens adquiridas pelo veículo para imagens “Birdview”, das quais estão isentas do efeito de perspetiva. A remoção deste efeito implica a ilustração da mesma

informação (e.g., uma porção de estrada) numa aparente vista de cima. Assim, métodos de comparação entre imagens podem ser utilizados como simples métodos de geolocalização. Outras abordagens de geolocalização que utilizam imagens com diferentes perspectivas entre si são complexas e normalmente não adequadas para serem aplicadas em grande escala. Um exemplo disto são as *Convolutional Neural Networks* (CNN's), em que requerem o treino do sistema, algo impossível em maior escala, dado o facto de se ser exigido a atribuição manual das localizações às imagens. Este tipo de imagens e técnicas serão exploradas em maior detalhe no Capítulo 2 (“Estado de Arte”).

Adicionalmente, a deteção e remoção de obstáculos verticais (e.g., pedestres e veículos) da imagem “Birdview” poderá melhorar o processo de comparação, ao eliminar objetos que poderão ser inexistentes nas imagens de satélite e/ou áreas. Por outro lado, estas últimas imagens poderão possuir obstáculos inexistentes na imagem “Birdview”. De modo a obter este efeito, existem outras técnicas que recorrem a visão estéreo ou à identificação de descritores na imagem que a segmentam em solo, componentes verticais, e céu, como demonstrado por Hoiem et. al. [10].

Na presente investigação explorou-se a extração destes obstáculos através da integração de outros sensores para além de câmaras, sendo que a inclusão deste modo de navegação com os existentes é um passo no sentido de uma condução 100% autónoma.

## 1.2 Objetivos

O objetivo principal deste estudo centra-se no desenvolvimento de uma técnica de localização do ATLASCAR2 que poderá, inclusive, auxiliar outros sistemas de navegação existentes no veículo na hipotética falha ou imprecisão dos mesmos.

Para tal, decidiu-se dividir a presente investigação em várias fases, com objetivos específicos para cada uma destas. A primeira fase centrou-se na análise do estado de arte relativo às técnicas *Inverse Perspective Mapping* (IPM) e *Multi-Modal Inverse Perspective Mapping* (MM-IPM), e suas variantes. Adicionalmente, foram também analisadas outras técnicas de localização como a comparação com imagens de satélite. Durante esta revisão, foram destacadas as limitações e inovações de cada técnica e o seu enquadramento no presente estudo.

A segunda fase compreendeu o desenvolvimento dos algoritmos IPM e MM-IPM, juntamente com as suas respetivas validações, através da análise dos tempos de execução. Tendo em conta que, numa fase inicial, o algoritmo seria adotado como complementar à navegação no sistema ATLASCAR2, seria fulcral respeitar o processamento do mesmo em tempo real. Neste contexto, a frequência a que o algoritmo deverá processar terá que ser equivalente aos *frames per second* (fps) da câmara. Esta propriedade é dependente da câmara a ser utilizada, portanto escolheu-se uma frequência arbitrária alvo da qual o algoritmo deverá ser processado, sendo esta 50 Hz.

A fase final consiste em aplicar vários métodos de localização de imagens “Birdview” em imagens de satélite e/ou aéreas. A validação destes métodos implica a possibilidade de utilizar os mesmos em contexto de geolocalização. O teste dos métodos desenvolvidos é realizado em ambientes simulados, ou seja, as porções de estradas que se pretendem localizar são extraídas das próprias imagens satélite.

### 1.3 Estrutura da tese

Com o propósito de apresentar o trabalho realizado, o presente documento foi organizado nos seguintes capítulos: Introdução, Estado de arte, Metodologia, Resultados e Discussão.

No presente capítulo (Capítulo 1) são apresentados e descritos os objetivos propostos a serem atingidos, assim como a motivação que levou à proposição deste tema.

O Capítulo 2, “Estado de Arte”, tem como função a análise de estudos anteriores focados no tema, de modo a explicitar inovações e limitações da área, além do seu enquadramento com o presente estudo.

Seguidamente, o Capítulo 3, “Metodologia”, descreve detalhadamente os métodos desenvolvidos para a construção da técnica final.

O Capítulo 4, “Resultados e Discussão”, complementa o anterior ao comparar os resultados da performance dos métodos desenvolvidos, tanto em termos de tempo de processamento como em obtenção de resultados viáveis.

Finalmente o Capítulo 5, “Conclusão”, que consiste na discussão dos objetivos atingidos, assim como os obstáculos que emergiram durante o decorrer da investigação.





## Capítulo 2

# Estado de arte

O presente capítulo tem como intuito explorar os vários temas integrantes desta dissertação, nomeadamente o *Inverse Perspective Mapping* (IPM), o *Multi-Modal Inverse Perspective Mapping* (MM-IPM) e geolocalização por comparação de imagens do tipo “Birdview” com imagens de satélite e/ou aéreas. O destaque das limitações, vantagens e desvantagens destas técnicas permite um melhor enquadramento do estudo realizado, cuja revisão de literatura permitiu delinear o projeto e realizar uma tomada de decisão informada durante o processo metodológico. Esta revisão focou-se em estudos realizados no contexto de condução autónoma, aplicáveis também noutros contextos.

A imagem resultante deste método será referida no resto desta dissertação por imagem “Birdview” ou imagem IPM.

### 2.1 Inverse Perspective Mapping

Atualmente, qualquer aplicação que envolva a análise da estrada (e.g., deteção de estrada, deteção de faixas, etc...) é facilitada ao se recorrer previamente ao método IPM ou às suas variantes. Visto este método ser *standard* no contexto de condução autónoma, considera-se pertinente elaborar detalhadamente o seu funcionamento, i.e., explorar a fundamentação matemática subjacente. Como tal, analisou-se todo o processo para a realização deste método, e deduções associadas.

Antes de se avançar para a técnica IPM, considera-se relevante rever o processo inverso, denominado por *Perspective Mapping* (PM), que é aplicável em todas as câmaras existentes. Este processo consiste na conversão do sistema de coordenadas do mundo ( $Wc$ ) para o sistema de coordenadas da imagem em píxeis ( $Xpixel, Ypixel$ ). Apesar de existirem vários modelos que relacionam ambos os sistemas, optou-se pelo *Pinhole Camera Model* [11].

Como referido anteriormente, o método IPM recebe como *input* uma imagem adquirida por uma câmara, e retorna como *output* uma versão da imagem original sem o efeito de perspetiva. Por outras palavras, este método realiza a conversão do sistema de coordenadas da imagem ( $Ic$ ) para um sistema de coordenadas do mundo ( $Wc$ ). De modo a escolher a abordagem ao método IPM a aplicar nesta investigação, analisaram-se duas abordagens alternativas, as quais, em teoria, produzem resultados idênticos. Independentemente da abordagem a ser utilizada a aplicação da mesma requiere três elementos: 1) matriz intrínseca da câmara; 2) matriz extrínseca, entre a câmara e um

referencial arbitrário no mundo; 3) as coordenadas de píxeis no referencial da imagem. Relativamente à primeira abordagem, que será sucintamente explicitada, o método IPM considera relações geométricas entre os dois sistemas de coordenadas. Recorre-se aos ângulos e distâncias da câmara face ao referencial do mundo ( $Wc$ ), como representado pelas seguintes equações:

$$x = h \cdot \text{ctg} \left( \frac{2 \cdot \alpha}{R_y - 1} \cdot u - \alpha + \theta \right) \cdot \sin \left( \frac{2 \cdot \alpha}{R_x - 1} \cdot v - \alpha + \gamma \right) + w, \quad (2.1)$$

$$y = h \cdot \text{ctg} \left( \frac{2 \cdot \alpha}{R_y - 1} \cdot u - \alpha + \theta \right) \cdot \cos \left( \frac{2 \cdot \alpha}{R_x - 1} \cdot v - \alpha + \gamma \right) + l, \quad (2.2)$$

onde  $R_x$  e  $R_y$  são a resolução da imagem (em píxeis) nas direções verticais e horizontais;  $x$  e  $y$  são duas das coordenadas no referencial  $Wc$ ;  $u$  e  $v$  são coordenadas no referencial  $Ic$ ;  $w$ ,  $l$  e  $h$  são as coordenadas da câmara no referencial  $Wc$ ;  $\theta$  é o ângulo de desfazamento entre a norma da lente da câmara e o plano  $y$  e  $z = h$ ;  $2 \cdot \alpha$  é o ângulo do campo de visão nas direções verticais e horizontais;  $\gamma$  é o ângulo entre a linha de projeção do eixo ótico no plano  $z = 0$  e eixo  $y$ .

O modo como as variáveis se relacionam entre si é ilustrado na figura 2.1.

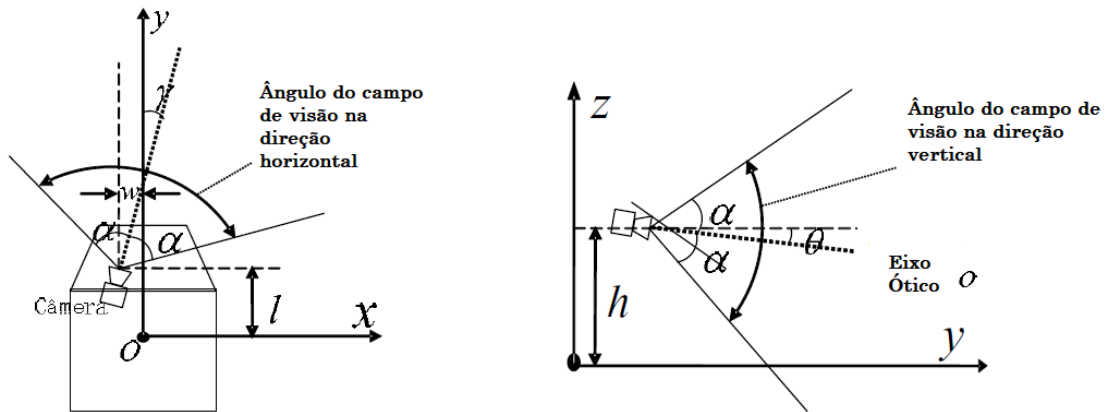


Figura 2.1: Posição e orientação da câmara no sistema de coordenadas do mundo no plano x-y (esquerda) e no plano y-z (direita) [12].

No que diz respeito à segunda abordagem, selecionada para ser aplicada no algoritmo final, esta recorre ao uso de matrizes de transformação que convertem o referencial <sup>1</sup> da imagem, para o referencial da câmara e, finalmente, para o referencial no mundo. A figura 2.2 demonstra como estes referenciais se relacionam entre si. Esta relação é traduzida pela equação 2.3:

<sup>1</sup>Neste contexto “referencial” deverá ser entendido como “sistema de coordenadas”.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K} \cdot {}^C\mathbf{T}_W \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.3)$$

onde  $u$ ,  $v$  e  $w$  são coordenadas no referencial  $Ic$ ;  $X$ ,  $Y$  e  $Z$  são coordenadas no referencial  $Wc$ ;  ${}^C\mathbf{T}_W$  é a matriz extrínseca entre a pose do referencial  $Wc$  e a pose da câmara;  $\mathbf{K}$  é a matriz intrínseca da câmara.

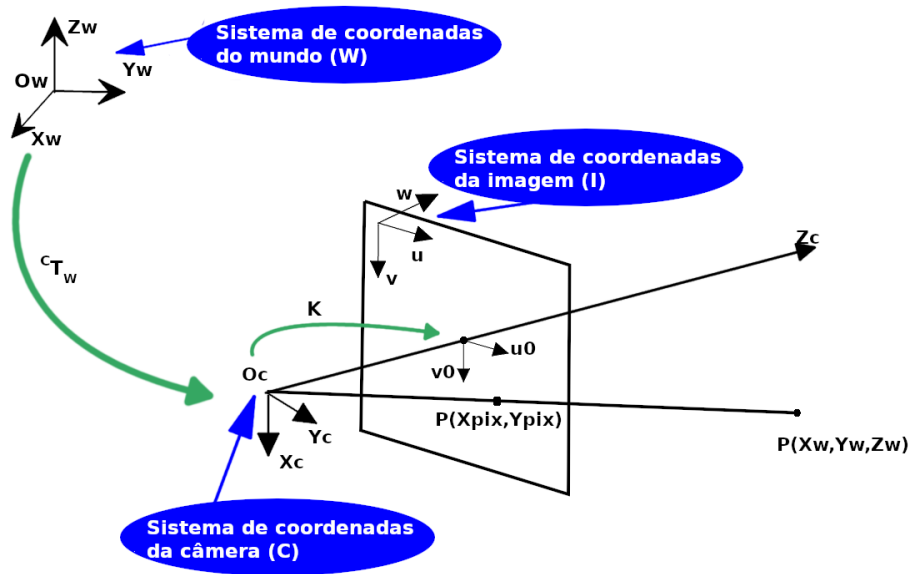


Figura 2.2: Relação entre os diferentes referenciais através do *Pinhole Camera Model*. A relação extrínseca entre os referenciais mundo,  $Wc$ , e câmara,  $C$ , é representada através da matriz extrínseca  ${}^C\mathbf{T}_W$ . De forma similar, a relação intrínseca entre os referenciais câmara e imagem,  $Ic$ , são representados através da matriz intrínseca  $\mathbf{K}$ .

Como referido anteriormente e agora evidenciado pela equação 2.3, a conversão das coordenadas do referencial  $Ic$  para o referencial  $Wc$  (*Inverse Perspective Mapping*) requer três elementos: 1) matriz intrínseca da câmara,  $\mathbf{K}$ ; 2) matriz extrínseca entre a pose da câmara e a pose do referencial do mundo,  ${}^C\mathbf{T}_W$ ; 3) coordenadas dos píxeis no referencial  $Ic$ ,  $(Xpix, Ypix)$ . A matriz intrínseca, inerente às especificações da câmara utilizada, é geralmente definida por

$$\mathbf{K} = \begin{bmatrix} \alpha_x & 0 & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (2.4)$$

onde  $\alpha_x$  e  $\alpha_y$  são as distâncias focais em píxeis;  $x_0$  e  $y_0$  representam o ponto central da imagem. É de notar que, devido a eventuais distorções por parte das lentes da câmara, esta matriz poderá ser modificada após a aplicação de calibrações sobre o modelo da câmara.

A matriz extrínseca é igualmente fundamental para a realização do método IPM e a mesma define-se, neste contexto, como a relação geométrica entre o referencial da câmara e um referencial definido arbitrariamente no mundo. Este último deverá ser colocado paralelamente à estrada de modo a facilitar determinadas simplificações, que serão explicadas de seguida. Uma vez conhecida a pose entre esses dois referenciais é determinado a matriz de transformação (matriz extrínseca).

Finalmente, resta analisar as coordenadas de cada pixel nas coordenadas do referencial  $Ic$ . Cada imagem adquirida possui apenas as coordenadas de cada pixel, que se relacionam com o referencial  $Ic$  da seguinte forma:

$$x_{pixel} = \frac{u}{w}, \quad (2.5)$$

$$y_{pixel} = \frac{v}{w}. \quad (2.6)$$

As equações 2.5 e 2.6 formam um sistema impossível de se resolver, visto que existem duas variáveis conhecidas ( $x_{pixel}$ ,  $y_{pixel}$ ) e três incógnitas ( $u, v, w$ ). A colocação das incógnitas em evidência origina as seguintes equações:

$$u = x_{pixel} \cdot w, \quad (2.7)$$

$$v = y_{pixel} \cdot w, \quad (2.8)$$

$$w = w. \quad (2.9)$$

Igualando as variáveis  $u$ ,  $v$  e  $w$  das equações 2.7, 2.8 e 2.9 com a equação 2.3 obtêm-se a seguinte equação:

$$\begin{bmatrix} x_{pixel} \\ x_{pixel} \\ 1 \end{bmatrix} \cdot w = K \cdot {}^C T_W \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (2.10)$$

A equação matricial 2.10 possui quatro incógnitas ( $X, Y, Z, w$ ) para três equações, sendo portanto um sistema impossível de se resolver. Neste caso é imprescindível a suposição do valor de uma das incógnitas, de modo a tornar o sistema possível e determinável. A estrada que se pretende representar na imagem IPM é normalmente plana e disposta sempre do mesmo modo em relação ao veículo. Portanto o referencial do mundo ( $Wc$ ) é posicionado (em relação à câmara) onde a estrada estaria posicionada e, de modo análogo, o plano  $z = 0$  é disposto da mesma forma que o plano da estrada estaria. Com a variável  $z$  definida a equação matricial 2.10 torna-se num sistema possível e determinável.

Seguidamente, as matrizes intrínseca e extrínseca são multiplicadas, de modo a obter a matriz de transformação de perspetiva ( $P$ ), simplificando as deduções posteriores, como demonstrado de seguida:

$$P = K \cdot {}^C T_W = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}. \quad (2.11)$$

Substituindo  $K \cdot {}^C T_W$  pela variável  $P$  na equação 2.3 obtém-se a seguinte equação:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = P \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (2.12)$$

Com base na equação 2.12, de modo a colocar as variáveis  $X$  e  $Y$  em evidência, recorreu-se à dedução feita por Oliveira e Santos [13]:

$$X = \frac{X_{pixel} \cdot (p_{22} \cdot p_{34} - p_{24} \cdot p_{32}) + Y_{pixel} \cdot (p_{14} \cdot p_{32} - p_{12} \cdot p_{34}) - p_{14} \cdot p_{22} + p_{12} \cdot p_{24}}{D}, \quad (2.13)$$

$$Y = \frac{X_{pixel} \cdot (p_{21} \cdot p_{34} - p_{24} \cdot p_{31}) + Y_{pixel} \cdot (p_{14} \cdot p_{31} - p_{11} \cdot p_{34}) - p_{11} \cdot p_{24} + p_{14} \cdot p_{21}}{D}, \quad (2.14)$$

onde,

$$D = X_{pixel} \cdot (p_{21} \cdot p_{32} - p_{24} \cdot p_{31}) + Y_{pixel} \cdot (p_{31} \cdot p_{12} - p_{32} \cdot p_{11}) - p_{11} \cdot p_{22} + p_{12} \cdot p_{21}. \quad (2.15)$$

Recorrendo às equações 2.13 e 2.14, é possível obter as coordenadas de cada pixel no referencial mundo. O método IPM é uma função injetiva, mas não sobrejetiva, i.e, é uma função que nunca aponta elementos distintos do seu domínio para o mesmo elemento do seu contradomínio, mas nem todos os elementos do contradomínio possuem correspondência com elementos do domínio. Portanto o método IPM gera imagens esparsas, ou seja, imagens com píxeis que possuem informação pertinente (originados da imagem da câmara) e píxeis sem informação, geralmente representados a preto (Figura 2.3).

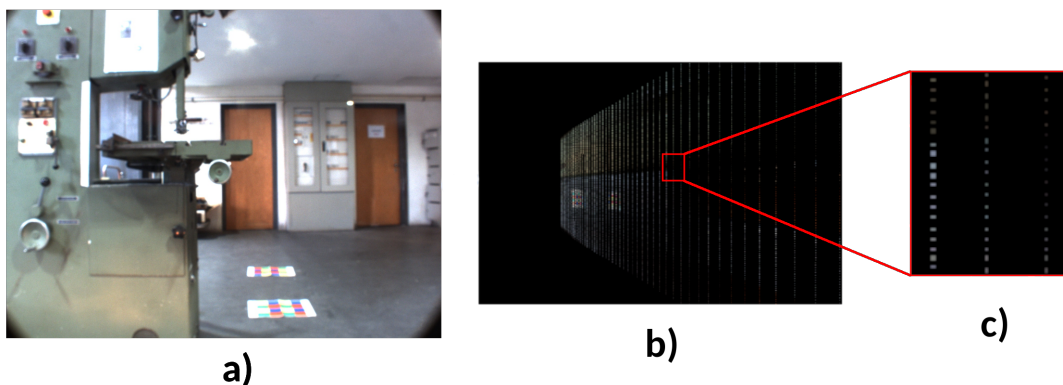


Figura 2.3: Método IPM. a) Imagem original; b) Imagem resultante do método IPM; c) Zoom numa porção da imagem IPM, sendo visível píxeis pertinentes por entre píxeis pretos.

De modo a organizar e entender de que modo alguns parâmetros influenciam a dimensão da imagem IPM visualiza-se o mesmo como uma interseção de vários polígonos, sendo cada um destes representativo de um tipo de informação diferente. Uma imagem IPM simples é constituída por dois polígonos: o de percepção ( $\Psi_{perception}$ ) e o de projeção ( $\Psi_{projection}$ ). O primeiro polígono delimita a área do referencial mundo que será ilustrada na imagem final. Por outro lado, o segundo polígono é representativo da área com informação obtida através do método IPM. A criação de um polígono de percepção de tamanho suficiente para conter informação pertinente, apesar de ser reduzido de modo a descartar informação inútil, costuma, ainda assim, resultar numa IPM de dimensões consideráveis.

O método IPM é computacionalmente dispendioso, visto ser necessário recalculer a coordenada de cada pixel para o novo referencial. Na presente investigação, a aplicação deste método no contexto da condução autónoma torna imprescindível a sua adequação para aplicação em tempo real. De seguida, serão exploradas duas ferramentas que, potencialmente, aceleram o método IPM: utilização de uma *Region Of Interest* (ROI) da imagem original e diminuição da sua resolução.

A definição de um ROI apresenta-se de grande utilidade, visto existirem píxeis pertencentes a elementos (e.g., céu, árvores, prédios) não situados no plano da estrada, possuindo assim uma componente  $z$  diferente de zero. Para além de existirem píxeis associados a elementos no mundo com a coordenada  $z$  diferente de zero, verifica-se também a ocorrência do fenómeno denominado por *half-space*, ilustrado na figura 2.4.

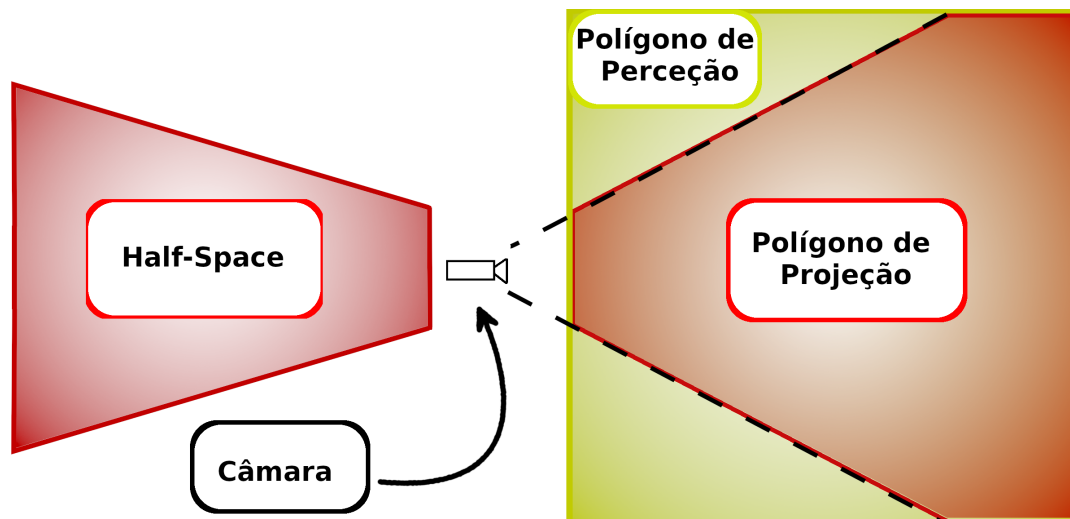


Figura 2.4: Representação simbólica das várias zonas de uma imagem IPM, que inclui o polígono de projeção, polígono de percepção e *Half-Space*

Para além de um ROI remover píxeis não pertinentes, diminui também a dimensão do polígono de projeção. Hipoteticamente, caso se queira representar todos os píxeis da imagem original sobre a sua imagem resultante, esta última será excessivamente grande. Tal deve-se ao facto de, quanto mais próximos do horizonte os píxeis se en-

contram na imagem original, maior será o afastamento destes em relação à origem no referencial mundo.

Uma abordagem alternativa para acelerar o método de IPM reside na diminuição da resolução da imagem original e, conseqüentemente, do número de píxeis, para os quais é necessário recalcular as suas coordenadas no novo referencial. Dependendo da finalidade da imagem, uma grande resolução da imagem original poderá não ser pertinente, pelo que a perda de qualidade da mesma poderá compensar a diminuição do custo computacional.

O rescalamento da imagem IPM obtida não se representa como uma solução adequada, visto que a informação esparsa nela contida é representada por píxeis individuais, geralmente rodeados por píxeis sem informação. Considerando que o pixel é a menor unidade presente numa imagem, o seu rescalamento iria danificar informação necessária. Uma solução a esta problemática é a utilização de um *Relative Scale Factor* (RSF). De um modo resumido, a variável RSF irá multiplicar alguns elementos da matriz extrínseca (i.e. elementos associados à translação), simulando deste modo a mudança de posição da câmara em relação ao referencial mundo. A variável RSF multiplica também as dimensões do polígono de percepção. Deste modo a dimensão de cada pixel representará maiores dimensões (em milímetros) do referencial mundo, conseqüentemente a imagem IPM possuirá a informação mais condensada, ou seja, com menos píxeis sem informação por entre os píxeis com informação.

Contudo, em alguns casos, a existência de píxeis esparsos poderá ser inevitável, interferindo negativamente com processamentos da imagem a serem aplicados futuramente. Por exemplo, métodos de deteção de faixas que utilizam a intensidade dos píxeis para efetuar o procedimento, irão obter resultados erróneos devido aos píxeis sem informação, representados a preto. De modo a mitigar esta fragilidade, Oliveira e Santos [13] sugeriram a interpolação linear sobre os píxeis pertinentes, seguido por um processo de “suavização”.

O sistema monocular possui limitação no campo de visão e por isso, muitas vezes, não possui a capacidade de visualizar toda a área da estrada, o que torna algumas operações (e.g. deteção de faixas) inefetivas. Frequentemente este tipo de sistema não visualiza ambas as marcações laterais delimitativas da estrada. Isto pode-se relacionar com a dimensão do veículo, o tipo de câmara, o seu posicionamento ou as lentes. Uma solução óbvia e bastante comum consiste na instalação de várias câmaras, capturando, assim, toda a área da estrada. Oliveira e Santos [13] exploram este conceito e introduzem um sistema móvel constituído por múltiplas câmaras. Neste sistema as posições das câmaras são conhecidas, tal como a matriz intrínseca de cada uma, em relação a um mesmo referencial. Como se sabe as relações geométricas entre as câmaras e o referencial no mundo cada imagem IPM gerada partilhará o mesmo sistema de coordenadas. Se cada imagem IPM partilha o mesmo sistema de coordenadas é possível agregar as imagens IPM, pois trata-se, de certa maneira, uma atribuição do pixel à correspondente coordenada. A figura 2.5 demonstra o resultado da agregação de duas imagens IPM.

Independentemente de o método IPM assentar sobre um sistema com uma ou várias câmaras, a sua importância é perceptível. Contudo não existem muitos estudos que se focam na medição da precisão dos resultados obtidos. Oliveira, Santos e Sappa [1] sugerem a aquisição de uma imagem IPM que captura um plano preenchido com quadrados coloridos (figura 2.6). De maneira a determinar a precisão do método verifica-se a percentagem de quadrados coloridos corretamente representados pela equação:

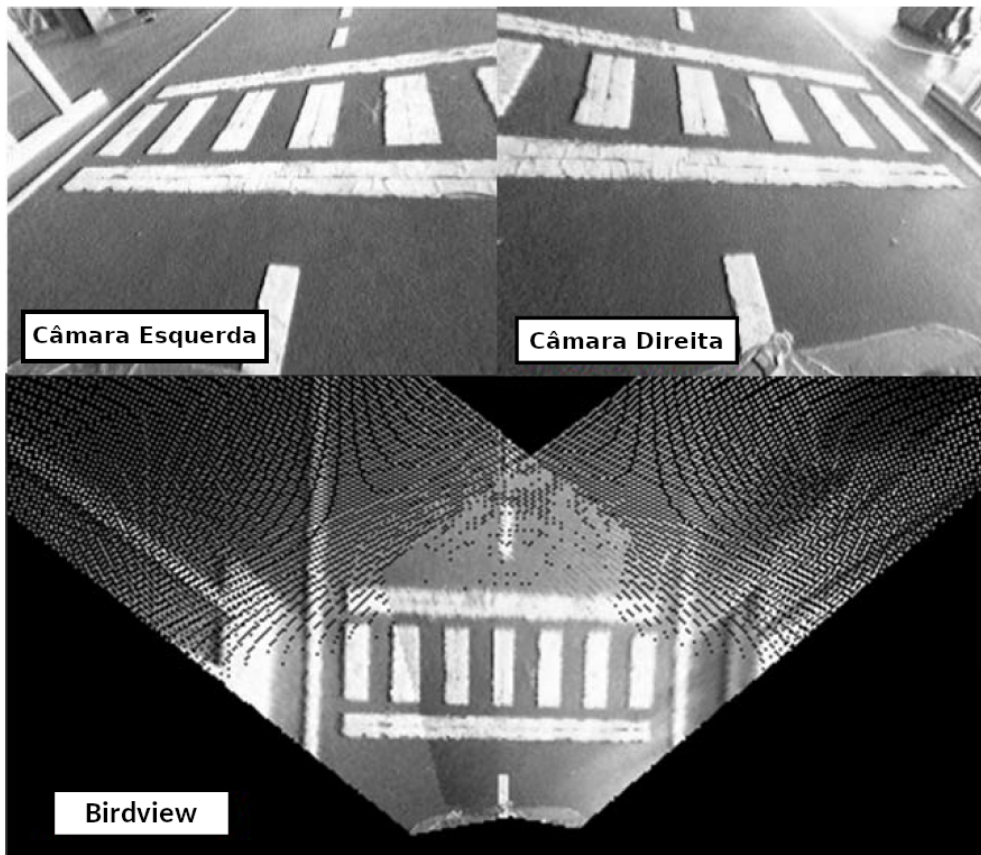


Figura 2.5: Processo de agregação de duas imagens IPM. Em cima, estão o *input* (imagens adquiridas pelas câmaras), em baixo está o resultado final [1] (figura adaptada).

$$\eta = \frac{N_{\text{pixelscorretos}}}{N_{\text{pixels}}}. \quad (2.16)$$



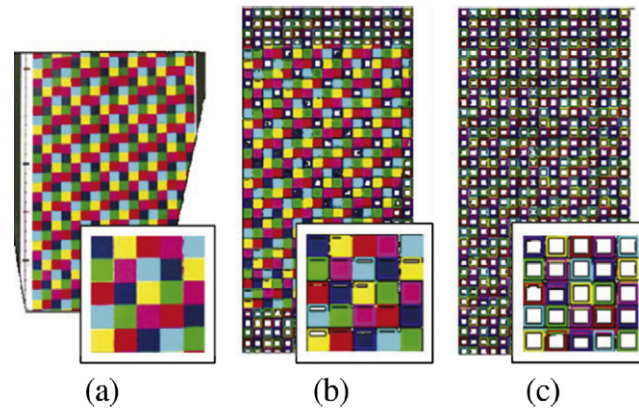


Figura 2.6: Análise da precisão da imagem resultante do método IPM. Em baixo de cada figura é representado uma região ampliada dos píxeis [1]. (a) Representação correta do plano que possui os quadrados coloridos; (b) Píxeis que foram corretamente projetados; (c) Píxeis que estão em falta.

## 2.2 Multi-Modal Inverse Perspective Mapping

No contexto de condução autónoma, as imagens IPM representam a estrada com bastante fidelidade, no entanto a introdução de obstáculos originará uma representação incorreta da realidade, por estes serem distorcidos. Isto deve-se ao facto de obstáculos não serem restringidos ao plano  $z=0$ , isto é, possuem uma dimensão em 3D e não apenas em 2D. Por esta razão será explorada nesta secção a remoção destes obstáculos nas imagens, com principal foco em imagens MM-IPM. Para fins demonstrativos a figura 2.7 representa imagens IPM com e sem obstáculos.

O método MM-IPM é uma versão do método IPM que integra outros sensores para além das câmaras. Apesar desta secção se focar na integração de sensores para a deteção de obstáculos, considera-se pertinente referir outros métodos para efetuar a mesma tarefa.

Desde há muito que se recorre à visão estéreo para a deteção de obstáculos. Isto porque a visão estéreo permite a extração de informação 3D, com recurso a operações de disparidade binocular. Normalmente esta operação realiza-se em imagens diretamente adquiridas da câmara, mas Bertozzi e Broggi [2] realizam esta operação em imagens IPM. Ao realizar a operação de disparidade sobre as imagens IPM obtêm-se uma representação do obstáculo (caso exista). A figura 2.8 exemplifica uma situação onde em frente do sistema de câmaras existe um obstáculo. É possível inferir características com base na forma que a representação do obstáculo toma, sendo estas a altura, a largura e a distância.

O uso de visão estéreo (ou similares) possui algum custo computacional associado, pois requer cálculos em cada píxel individual nas duas (ou mais) imagens. Outra alternativa de deteção de obstáculos com menor custo computacional é a integração de outros sensores para além das câmaras. Apesar de aumentar o custo monetário e a complexidade do sistema montado, comparativamente à visão estéreo, este método de deteção de obstáculos não apresenta muito ruído. Oliveira et. al. [1] exemplificam

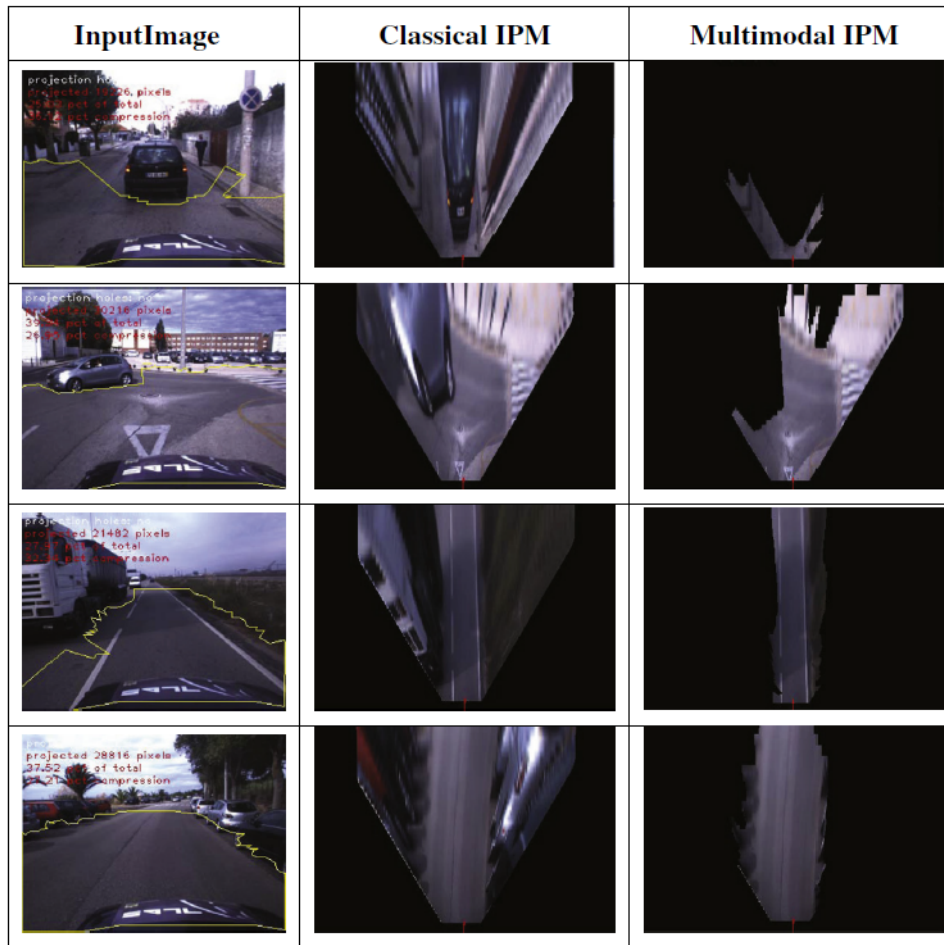


Figura 2.7: Resultados e comparação do método proposto com o método IPM clássico [1].

um sistema deste tipo que constrói uma imagem IPM sem obstáculos representados, ou seja, uma imagem MM-IPM.

No artigo a ser discutido considera-se que as coordenadas dos pontos gerados pelo *laser range finder* (LRF) partilham o mesmo sistema de coordenadas que a imagem IPM, pois sabe-se a posição do sensor laser. Considerando isto, é possível gerar um polígono de “espaço sem obstáculos” recorrendo à nuvem de pontos do laser. Tendo este polígono basta modificar o algoritmo IPM para que este efetue as operações de transformação dos píxeis que estão dentro da área do polígono. Os autores deste sistema interpretam isto como uma interseção de vários polígonos existentes no sistema (figura 2.9).

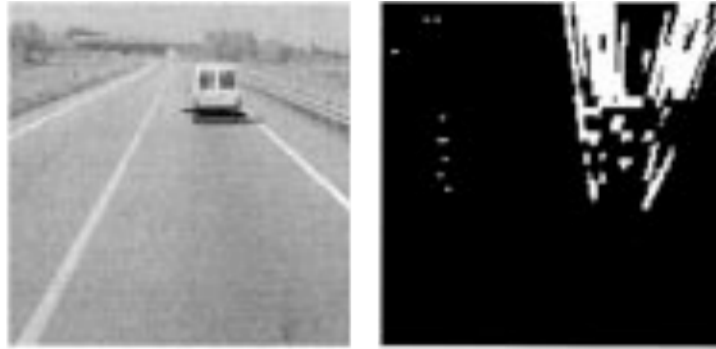


Figura 2.8: Representação do método de detecção de obstáculo do sistema GOLD. É fornecido como *input* a imagem à direita e o método retorna o *output* à esquerda [2].

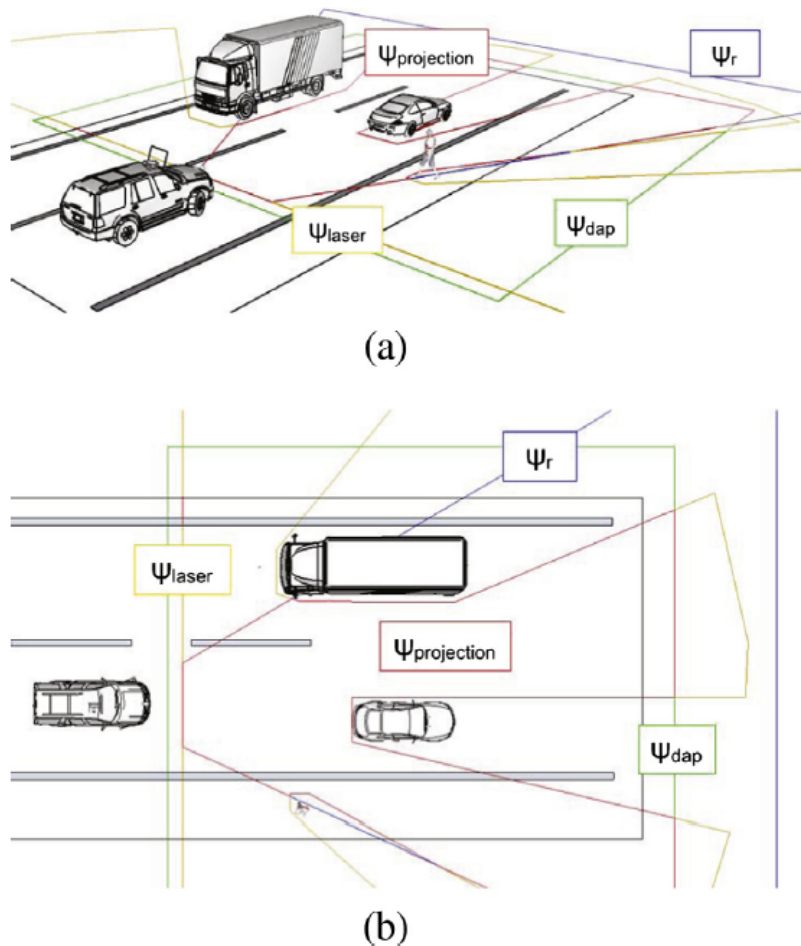


Figura 2.9: Exemplificação de um cenário de uma estrada com vários carros: (a) vista isométrica; (b) vista de cima. É possível a visualização dos vários polígonos: 1) polígono de projeção  $\psi_{Projection}$  (a vermelho); 2) polígono de percepção  $\psi_{dap}$  (a verde); 3) polígono de área sem obstáculos  $\psi_{laser}$  (a amarelo), [1].

### 2.3 Pré-processamentos e otimizações do método Inverse Perspective Mapping

Poderá haver situações em que se deseja executar o método IPM a grandes frequências e por isso é típico a aplicação de otimizações no sentido de diminuir o custo computacional. Atualmente é possibilitado a decisão entre o uso de hardware com grande poder computacional ou o uso de tempo (ou recursos) no desenvolvimento de otimizações. Mas há relativamente pouco tempo, era fundamental uma ponderação longa sobre como construir um sistema (tanto em termos de hardware como em software) capaz de executar algoritmos computacionalmente custosos. A título de exemplo refere-se Bertozzi e Broggi [2] que construíram, em meados de 1998, um sistema denominado por GOLD (*Generic Obstacle and Lane Detection*). Tal como o nome do sistema indica este foca-se na detetação de faixas de estradas juntamente com a deteção de obstáculos. Na construção deste sistema houve um grande foco na otimização do hardware e criou-se uma configuração denominada por PAPRICA (Parallel Processor for image Checking and Analysis). Esta configuração suporta computação paralela e pyramidal, sendo esta ideal para o processamento de várias imagens em simultâneo. A figura 2.10 demonstra o esquema do sistema GOLD. É de notar que apesar de tanto foco na otimização este sistema realiza a deteção de obstáculos e faixas com uma frequência de 10 Hz, o que não é ideal para uma aplicação no mundo real. Em contrapartida esta dissertação pretende testar e validar os algoritmos desenvolvidos sobre o sistema ATLASCAR2 e este não possui grandes otimizações em termos de hardware.

Apesar do grande avanço tecnológico ao longo dos anos, existe ainda a necessidade de realizar algumas otimizações em termos de software. Por exemplo, no contexto do método IPM, é comum a utilização de uma área de interesse, i.e., um ROI. Isto porque se o método IPM consiste na representação da estrada num plano, então não faz sentido existir na imagem elementos acima do horizonte, como o céu e prédios altos. Se a imagem original é reduzida a uma área de interesse, então há menos píxeis para processar. Vários estudos usam o ROI em simultâneo com o método IPM, por exemplo Borkar et. al. [5], Oliveira et. al. [1] e Aly e Mohamed [14]. A figura 2.11 demonstra através de um esquema o modo como o ROI reduz a imagem a ser analisada.

Considera-se pertinente discutir no resto desta secção otimizações direcionadas para a obtenção de resultados mais viáveis. Mais especificamente, será discutido daqui adiante correções ao ângulo da câmara em relação à estrada, ou seja, correções da matriz extrínseca.

O método IPM considera nos cálculos a relação entre a posição da câmara e um referencial arbitrário. Este referencial é suposto ser um plano representativo da estrada, ou seja, de um modo simplificado, deverá ser um plano paralelo ao veículo. No entanto, o veículo desloca-se por estradas que podem, ou não, possuir inclinações. Devido a isto o plano pressuposto da estrada diverge do verdadeiro (plano da estrada). Por outras palavras, o ângulo (e consequentemente a posição) da câmara em relação á estrada, definido à priori, é agora diferente. Com a figura 2.12 é possível visualizar como as várias inclinações da estrada influenciam a imagem IPM resultante. Os ângulos representativos da câmara são definidos como os ângulos de euler, sendo estes o *Roll*, o *Pitch* e o *Yaw*.

Dado que a imagem IPM é distorcida aquando da existência de inclinações da estrada é pertinente explorar métodos desenvolvidos para corrigir esta representação er-

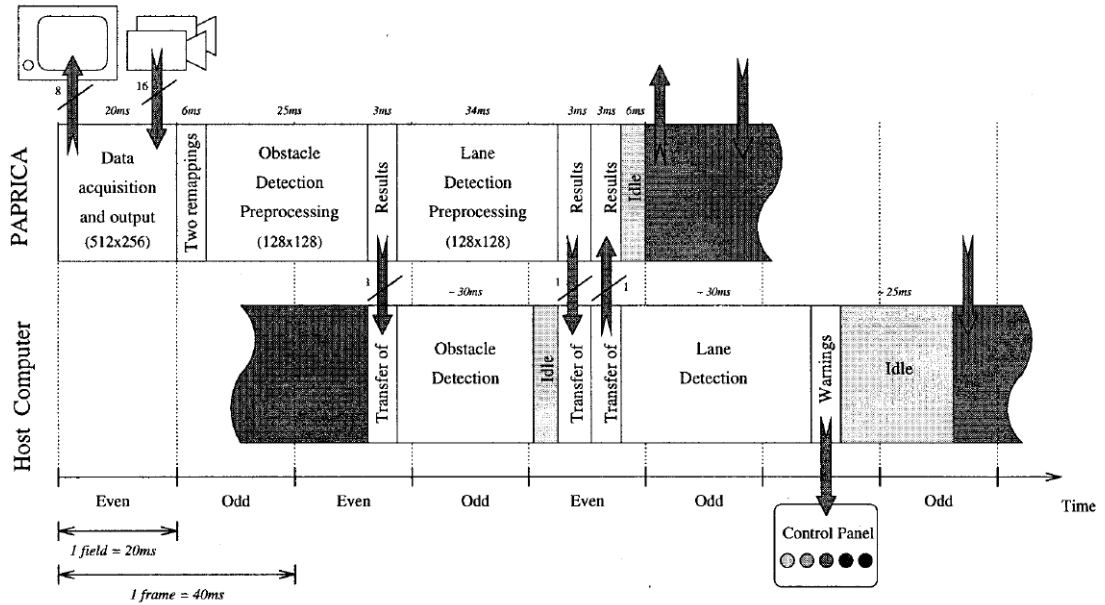


Figura 2.10: Diagrama representativo do sistema GOLD [2].

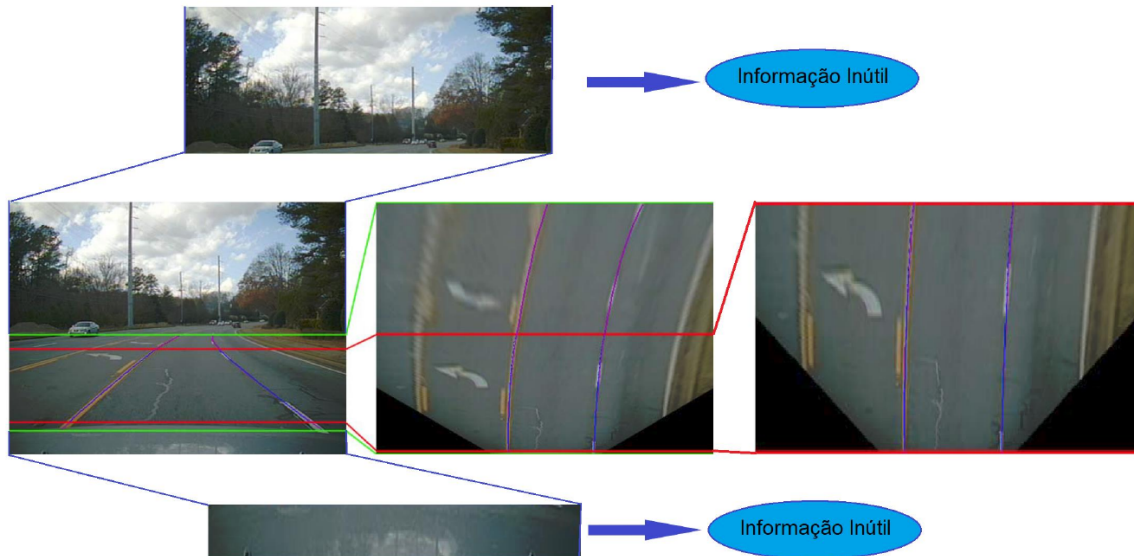


Figura 2.11: Esquema demonstrativo de como a imagem é segmentada de maneira a reduzir o custo computacional [3] (figura adaptada).

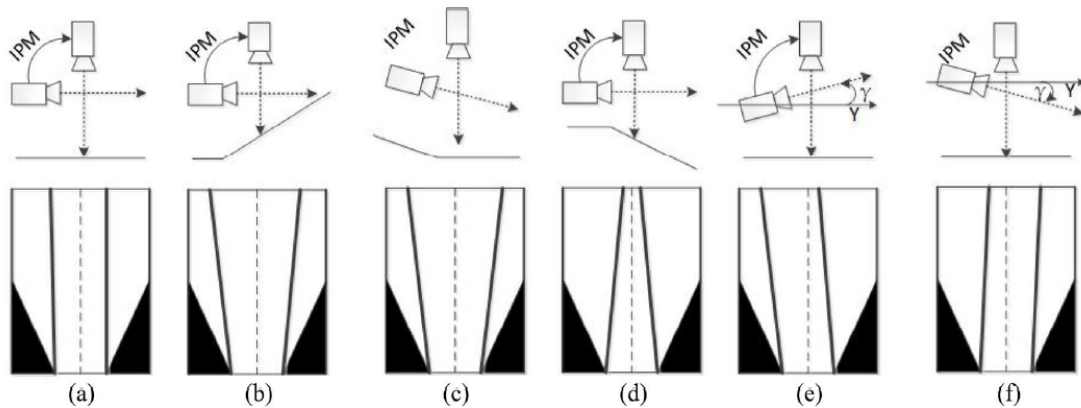


Figura 2.12: Erro causado por variações de inclinações e rotações: a-d) demonstra os efeitos da variação da inclinação na imagem IPM sem correção do *pitch*; e-f) demonstra o efeito da variação de rotações na imagem IPM sem correção do *yaw* [4].

rónea da estrada.

Jeong e Kim [15] estudam uma abordagem que corrige uma imagem IPM após a passagem sobre uma lombada. É verificado que uma pequena variação do *pitch* (e.g., 2,1 graus) é suficiente para produzir resultados inexatos. De um modo resumido, a abordagem desenvolvida identifica pontos de interesse com o algoritmo “mono visual odometry”, que recorre ao auxílio das câmaras. Os pontos na imagem são escolhidos consistentemente considerando características da imagem. Estes pontos permitem identificar a odometria do carro, ou seja, é possível determinar a posição da câmara, que inclui o *pitch*, corrigindo, assim, a imagem IPM. Apesar disso a abordagem sugerida não é robusta em ambientes com poucas características, i.e., ambientes que não possuem “objetos” que se destacam, como vegetação ou prédios. Para além disso, é apenas corrigido o *pitch*. Em situação de curva a imagem IPM será também distorcida, desta vez devido ao *yaw* incorreto.

Para uma correção robusta de ambos o *pitch* e o *yaw* muito tipicamente recorre-se ao chamado “*Vanishing Point*” (ponto de convergência). Este é definido como o ponto onde as linhas numa imagem tendem a convergir. Por exemplo, se uma estrada se encontra numa colina o ponto de convergência estará mais acima do que estaria se estivesse numa estrada sem inclinação. Ao localizar o ponto de convergência e sabendo onde este deveria estar, de modo a produzir uma correta imagem IPM, é possível corrigir os ângulos *pitch* e *yaw*. O verdadeiro desafio é a correta determinação do ponto de convergência.

Yang, Fang e Tang [4] apresentam um método de identificação do ponto de convergência que se divide em duas fases: obtenção da textura saliente e votação do ponto de convergência. A primeira fase consiste em realizar sobre a imagem original uma operação com o intuito de acentuar possíveis saliências. Para este efeito implementa-se o algoritmo que desenvolveram, denominado por *Modified Weber’s Local Descriptor* (MWLD). É argumentado que a imagem resultante simula como as pessoas percebem o ambiente e que por isso possui grande capacidade representacional. Possui também como vantagens a deteção de arestas e robustez face ao ruído e à variação da

iluminação.

Tendo a imagem, agora transformada, prossegue-se para a criação de vetores com as orientações dominantes de píxeis. Isto é feito com o uso de filtros de Gabor. Estes filtros possuem a capacidade de filtrar de uma imagem as orientações de grupos de píxeis. A figura 2.13 demonstra o funcionamento destes filtros. Para se obter com alta precisão as várias orientações existentes na imagem é aplicado sobre a imagem 36 filtros de Gabor com orientações intervaladas por  $5^\circ$ .

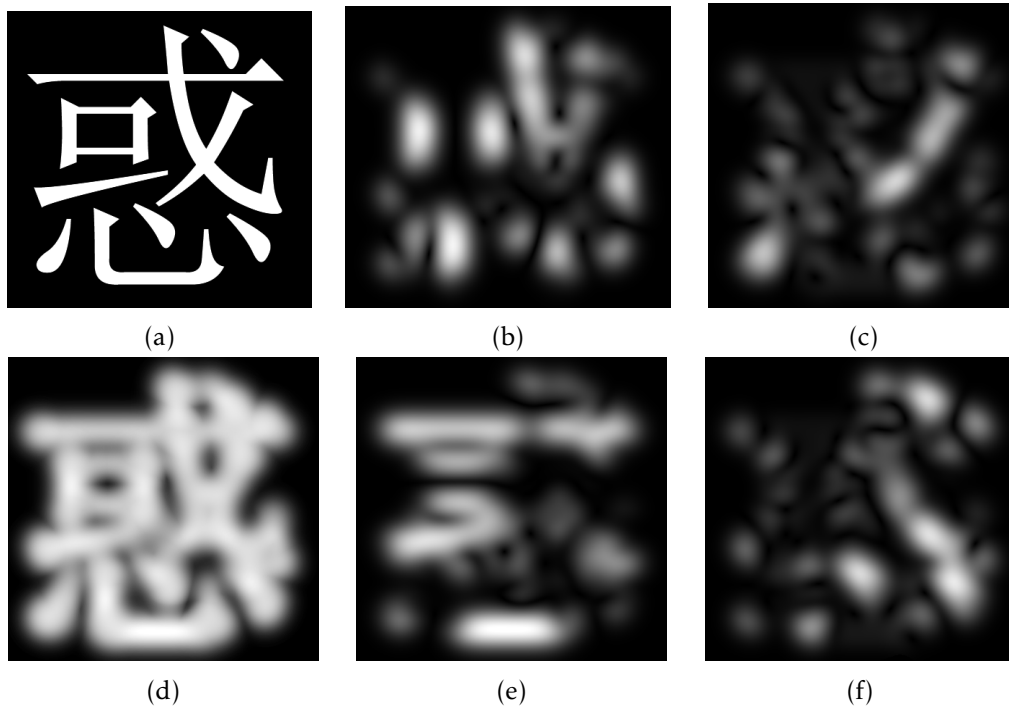


Figura 2.13: Resultados de filtros de *Gabor*: (a) imagem original; (b) resultado da aplicação de um filtro de *Gabor* com  $0^\circ$  de orientação; (c) resultado da aplicação de um filtro de *Gabor* com  $45^\circ$  de orientação; (d) junção das quatro imagens resultantes; (e) resultado da aplicação de um filtro de *Gabor* com  $90^\circ$  de orientação; (f) resultado da aplicação de um filtro de *Gabor* com  $135^\circ$  de orientação.

A última fase consiste num sistema de votação que possui como intuito o cálculo do ponto da imagem por onde passa um maior número de vetores. O ponto com a maior votação será o ponto de convergência da imagem.

A abordagem a ser analisada obtém bons resultados em ambientes estruturados, principalmente por transformar a imagem original numa com as arestas salientadas. No entanto, é previsto que a sua precisão diminua quando for testado em ambientes com pouca estruturação, como exemplo, estradas de terra. Todo este sistema agora descrito pode ser demonstrado pela figura 2.14.

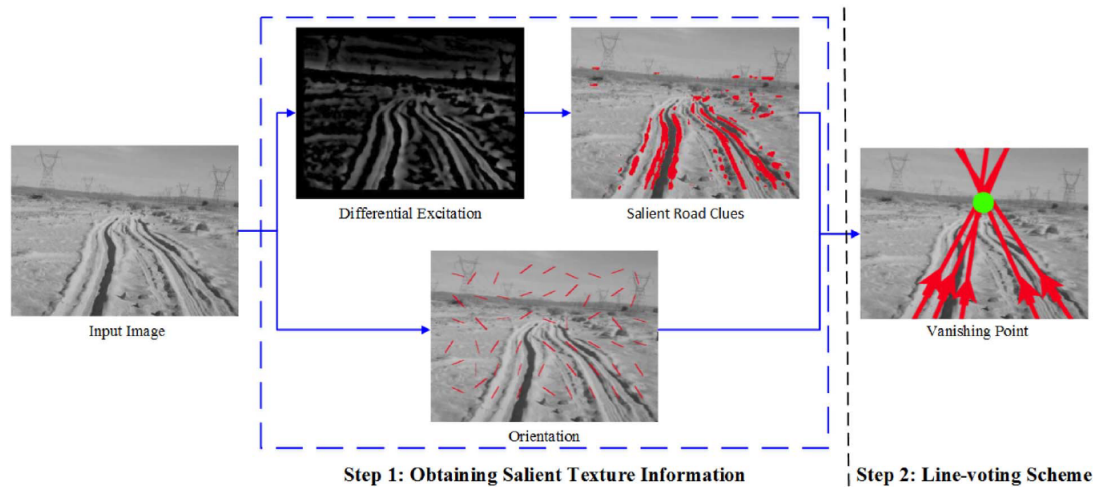


Figura 2.14: Demonstração do processo, faseado, para a determinação do ponto de convergência [4].

Moghadam, Starzyk e Wijesoma [16] apresentam uma abordagem similar de detecção de pontos de convergência, mas agora em ambientes não estruturados. A imagem a ser analisada não sofre qualquer transformação ao contrário do que acontecia na abordagem anterior. Adicionalmente o sistema de votação é mais elaborado e o cálculo das orientações é realizado de maneira diferente.

Em termos do cálculo de orientações dominantes, em vez de se usar 36 filtros de Gabor, são usados apenas 4 filtros de orientações distintas. Por cada pixel determina-se as duas orientações com maior dominância, soma-se os vetores dessas orientações e obtêm-se um novo vetor. Desta forma a determinação das orientações dominantes é realizada a um menor custo computacional e com melhor precisão, pois anteriormente a precisão era limitada a  $5^\circ$ .

Na votação, vetores provenientes de qualquer ponto da imagem normalmente possuem um mesmo valor na votação, o que se traduz numa incorreta determinação do ponto de convergência. Vetores criados nas zonas superiores ou laterais não contribuirão de uma forma construtiva para a determinação do ponto de convergência, pois nessas zonas normalmente existe elementos não relacionados com a estrada. Para mitigar este problema atribui-se um certo “peso” com recurso a uma função sinusoidal, que suprime orientações quasi-horizontais. Com esta distribuição de “peso”, píxeis mais acima na imagem recebem mais votos, que consequentemente se traduz novamente numa escolha errónea do ponto de convergência. De modo a resolver este problema é atribuído um outro “peso” ao longo de cada vetor, ou seja, atribui-se um “peso” baseado na norma do vetor.

Até este instante exploraram-se correções apenas do *pitch* e do *yaw*. Para veículos de quatro rodas o *roll* varia muito pouco e por isso não se considera pertinente corrigir. Mas no caso de motos o *roll* varia bastante e de forma a ser gerado uma correta imagem IPM é fundamental a correção do mesmo. Damon et. al. [17] exploram um método de correção do *roll* que consiste numa verificação iterativa da imagem IPM. Resumidamente, considera-se, no início, um valor de *roll* arbitrário e gera-se uma imagem



IPM. Verifica-se com a imagem criada se as marcações da estrada possuem as propriedades próprias de uma estrada, ou seja, existência marcadores laterais equidistantes e paralelos entre si. Dependendo do erro obtido, corrige-se, ou não, o *roll*.

Em síntese, existem bastantes pré-processos que auxiliam a criação de uma imagem IPM. Podendo estes terem como finalidade mitigar o custo computacional (e.g., implementação do ROI) ou o aumento de fidelidade nos resultados (e.g., correção do *pitch*, *yaw* e *roll*).

## 2.4 Aplicações do método Inverse Perspective Matching

Em contexto de condução autónoma, imagens IPM são ideais para realizar vários tipos de análise à estrada, sendo estas, especificamente, a deteção de estrada (que inclui a identificação de marcadores rodoviários) e análise do tráfego. Estes dois tópicos serão foco de discussão nas seguintes subsecções, onde serão exemplificados vários estudos relacionados.

### 2.4.1 Deteção de estrada

A identificação de marcadores rodoviários (que consequentemente permite identificar faixas de estrada) é um dos objetivos da condução autónoma. É uma tarefa que pode ser dificultada pelo desgaste dos marcadores rodoviários, condições de iluminação, existência de objetos que ofuscam os marcadores e, finalmente, por existir um efeito de perspectiva, que atribui uma maior nível de complexidade na deteção das mesmas.

Para fins demonstrativos, refere-se Borkar, Hayes, Smith e Pankanti [5], que exploram a identificação de marcadores rodoviários à noite, recorrendo a imagens sem o método IPM. Nesta abordagem, para além de se implementar o ROI, aplica-se também o chamado “*Temporal Blur*”. Esta última técnica consiste na realização de uma média entre imagens separadas temporalmente. É bastante útil para uma melhor visualização de marcadores quando estes são apenas parcialmente visíveis ou quando apresentam elevado desgaste. A figura 2.15 exemplifica o resultado de um “*Temporal Blur*”. Imediatamente a seguir aplica-se uma “binarização adaptativa”, que é uma técnica extremamente conhecida em visão computacional. Ao contrário de uma binarização comum esta binariza segmentos menores de dentro da própria imagem, deste modo a iluminação não afetará tanto a operação de binarização.

Após o tratamento inicial da imagem (muito comum neste contexto) procede-se para a identificação dos marcadores. Para esta finalidade utiliza-se a transformada de Hough, que é uma técnica de identificação de formas geométricas específicas, neste caso estaria ajustado para localizar linhas retas. Finalmente, resta validar os resultados obtidos. Divide-se a imagem em zonas de interesse e aplica-se ao longo das retas candidatas filtros gaussianos, de um forma discretizada, e adquire-se pontos que possuem maior intensidade. Finalmente aplica-se uma transformada de Hough nos pontos obtidos, adquirindo os marcadores rodoviários.

Constata-se que nesta abordagem a fase de identificação dos marcadores de estrada tornou-se desnecessariamente mais complexa do que seria com a utilização de uma imagem IPM. Numa imagem IPM os marcadores estariam paralelos entre si, e consequentemente a espessura dos mesmos seria constante ao longo da estrada. Devido a esta

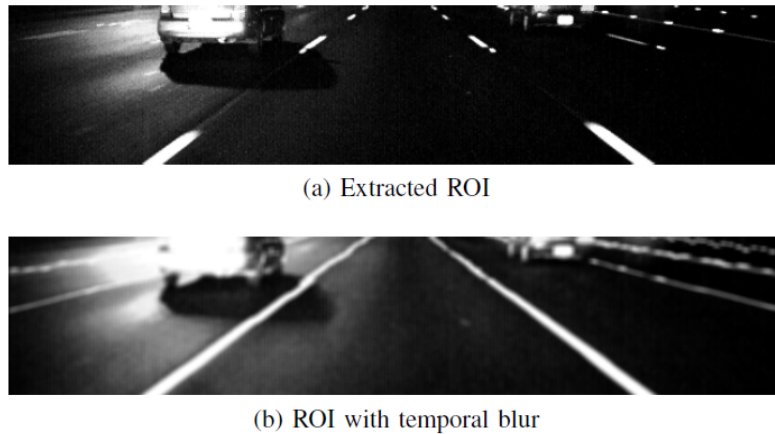


Figura 2.15: Resultados intermédios de um “Temporal Blur”. [5]

simplificação vários estudam adotam o método IPM.

Existem inúmeras abordagens para detetar marcadores rodoviários em imagens IPM que serão referidas a seguir. Ao longo das seguintes descrições não serão referidos os pré-processamentos, pois serão muito similares. Começando por uma abordagem muito simplista, o estudo feito por Bertozzi e Broggi [2] recorre ao algoritmo “Geodesic Morphological Dilation”, que destaca as marcações rodoviárias. Com a representação das marcações é possível inferir a localização das faixas.

Numa outra abordagem, estudada por Aly [14], aplica inicialmente sobre a imagem IPM dois tipos de filtros gaussianos com o intuito de se obter o chamado “mapa de evidências”, que possui objetos binários representativos de potenciais marcadores. A figura 2.16 exemplifica o resultado desta operação. Poderão estar no “mapa de evidências” outras linhas que não são representativas de marcadores rodoviários. Existe, ainda, necessidade de acrescentar mais um passo de maneira a aumentar a robustez do sistema, sendo este a implementação do método “RANSAC *Spline Fitting*”. De um modo simplificado, este método introduz vários pontos nas linhas candidatas e verifica se a linha formada por estas se assemelha o suficiente a uma linha reta para ser considerada um marcador.

Para além do uso de técnicas focadas em formas geométricas das linhas detetadas, existe um outro grupo de técnicas que recorre a filtros de partículas (ou método Se-

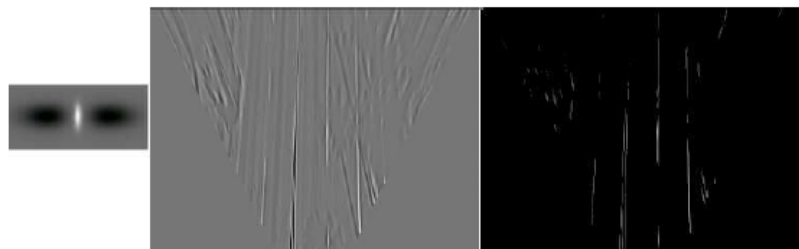


Figura 2.16: Filtragem e binarização de imagem: (Esquerda) núcleo para filtragem; (Centro) imagem após filtragem; (Direita) imagem após binarização [14].

quencial de Monte Carlo). Estes consistem na definição de um conjunto de partículas que, como um todo, definem o local provável do “objeto” de interesse. Cada partícula representa um possível pixel associado a um marcador rodoviário. Portanto uma distribuição espaçada de partículas representa uma grande incerteza, enquanto que uma distribuição concentrada de partículas representa uma menor incerteza.

Sehestedt, Kodagoda, Alempijevic e Dissanayake [18] exploram uma abordagem que consiste em dispersar um certo número de partículas por locais prováveis de serem marcadores. A técnica a ser estudada foi “especializada” para a um certo modelo de estrada. Querendo isto dizer que, para o posicionamento das partículas, se considerou o posicionamento típico dos marcadores, as curvas da estrada e a odometria do carro. Se nenhum marcador for detetado o aglomerado de partículas simplesmente espalhar-se-á.

Outra abordagem que emprega um filtro de partículas é estudada por Berriel, de Aguiar, de Souza Filho e Oliveira-Santos [3]. Este é muito similar ao anterior. Nesta abordagem as partículas distribuem-se de uma forma gaussiana por locais prováveis de haver marcadores. Tal como no estudo anterior, considera-se ainda onde as partículas se posicionaram na imagem anterior. Com base na observação do “mapa de evidências” gerado realiza-se uma reamostragem até sobrar apenas um grupo de partículas que representa o marcador com maior precisão. A figura 2.17 demonstra os vários processos desta abordagem.

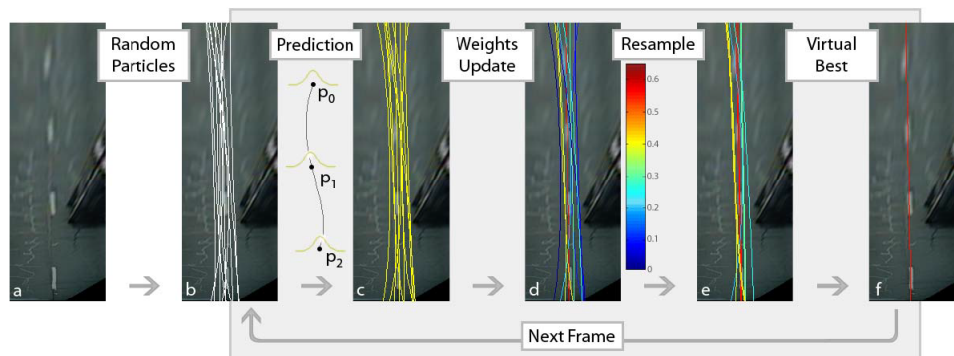


Figura 2.17: Esquema de detecção de marcadores rodoviários segmentado nas várias fases: (a) imagem original, (b) inicialização das partículas, (c) modificação da distribuição das partículas, considerando distribuições anteriores, (d) atualização do “peso” das partículas, (e) reamostragem e (f) resultado final [3]

#### 2.4.2 Análise de tráfego

A análise de tráfego é outra aplicação comum das imagens IPM. Wang e Shi [19] exemplificam um sistema com a capacidade de quantificar o tráfego. Numa primeira instância adquire-se uma imagem da porção de estrada que é posteriormente processada com o método IPM. Com uma imagem de referência (estrada sem carros) é possível segmentar os carros na nova imagem com recurso a técnicas morfológicas. Com a segmentação realizada, é agora possível determinar a quantidade de tráfego com recurso a duas possíveis abordagens, ambas envolvendo a determinação do contorno da área com

carros. Como indicado pela figura 2.18, o contorno utilizado pode ser determinado consoante a área do conjunto de carros ou determinado com base no comprimento da fila de carros em cada faixa.

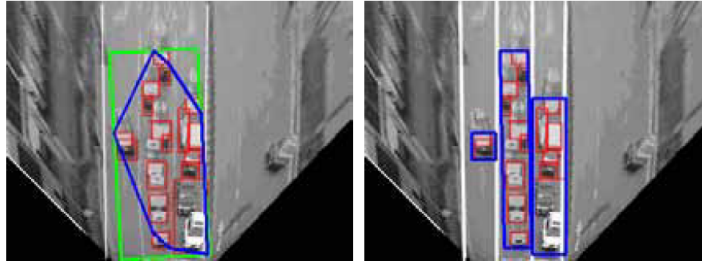


Figura 2.18: Detecção de tráfego: (Esquerda) utilização da área do conjunto de carros; (Direita) utilização da fila de carros em cada faixa [19].

## 2.5 Geolocalização

Esta última secção visa a explorar vários métodos de localização em coordenadas de latitude e longitude, ou seja, geolocalização. As seguintes subsecções demonstram várias técnicas de geolocalização que recorrem tanto a imagens diretamente provenientes da câmara como a imagens “transformadas” em imagens IPM. Em cada uma das seguintes técnicas será avaliado o custo computacional, robustez dos resultados e viabilidade na implementação.

### 2.5.1 Template Matching e variantes

Nesta subsecção pretende-se explorar alguns algoritmos de reconhecimento de objetos, mais especificamente algoritmos de comparação de imagens. Este foco deve-se ao facto de ser este o tipo de algoritmos que serão utilizados na fase final desta investigação. Entre estes serão principalmente explorados o *Template Matching* e *Chamfer Matching*.

*Template Matching*, tal como o nome indica, consiste em identificar e localizar uma imagem, apelidada como *template*, numa imagem maior. Tradicionalmente percorre-se pixel-a-pixel a imagem em estudo e extrai-se, em cada instância, um segmento de tamanho igual ao do *template* que será posteriormente comparada com o mesmo. A operação de comparação poderá ser efetuada com base em cálculo de “distâncias” ou correlação entre as duas imagens. Em ambas as operações de comparações, poderão advir resultados erróneos provenientes da distorção do *template*.

Existem vários modos de calcular a “distância” entre duas imagens, entre estes pode-se utilizar a soma das diferenças absolutas,

$$d(r,s) = \sum_{(i,j) \in R} |I(r+i, s+j) - R(i,j)|, \quad (2.17)$$

ou a soma das diferenças quadráticas,

$$d(r, s) = \sqrt{\sum_{(i,j) \in R} [I(r+i, s+j) - R(i, j)]^2}, \quad (2.18)$$

onde  $d(\cdot)$  é a “distância” entre as duas imagens;  $I(\cdot)$  é a imagem alvo;  $R(\cdot)$  é o *template*;  $i$  e  $j$  são coordenadas do *template*;  $r$  e  $s$  são as coordenadas da imagem alvo. O cálculo da correlação entre imagens poderá ser-se efetuada, entre muitas outras maneiras, com uma simples correlação cruzada,

$$d(r, s) = \frac{\sum_{(i,j) \in R} I(r+i, s+j) \cdot R(i, j)}{\sqrt{\sum_{(i,j) \in R} I^2(r+i, s+j)} \cdot \sqrt{\sum_{(i,j) \in R} R^2(i, j)}}, \quad (2.19)$$

ou uma correlação cruzada normalizada (NCC),

$$d(r, s) = \frac{\sum_{(i,j) \in R} [I(r+i, s+j) - \bar{I}(r, s)] \cdot [R(i, j) - \bar{R}]}{\sqrt{\sum_{(i,j) \in R} [I(r+i, s+j) - \bar{I}(r, s)]^2} \cdot \sqrt{\sum_{(i,j) \in R} [R(i, j) - \bar{R}]^2}}. \quad (2.20)$$

Este último considera diferenças de luminosidade do *template* que torna a comparação entre imagens mais robusto. Apesar de o método de correlação cruzada normalizada aparentar ser um bom candidato para identificar e localizar porções de estrada, extraídos localmente, em imagens de satélite este é computacionalmente custoso. Isto porque para além de se iterar, na imagem satélite, pixel-a-pixel é também realizado um conjunto de operações de um certo segmento de área centrado em cada um dos referidos píxeis.

Durante esta investigação foram observados vários métodos com o potencial de acelerar o método de NCC. Entre estes refere-se os que empregam a equação de NCC, mas que ao invés de recorrer ao domínio de espaço (a abordagem tradicional) efetua-se as operações no domínio de frequência. Ainda que o algoritmo final não explore o NCC no domínio de frequência considera-se, ainda assim, pertinente explorar de seguida um exemplo desta natureza.

Lewis [9] explora uma abordagem onde na equação 2.20 a expressão do numerador é transformada de um domínio espacial para um domínio de frequência através de FFT (*Fourier Frequency Transform*). A vantagem de se realizar esta operação no domínio de frequência é a redução do custo computacional.

Ainda no mesmo documento, o autor considera mais eficiente otimizar o denominador da equação 2.20 com o uso de imagens integrais, em vez de se usar o método convencional ou novamente a transformada de Fourier. Isto porque o modo menos eficiente de se calcular a equação

$$den(r, s) = \sqrt{\sum_{(i,j) \in R} [I(r+i, s+j) - \bar{I}(r, s)]^2} \quad (2.21)$$

é a realização pixel-a-pixel do cálculo da média e energia local, da porção de imagem “debaixo” do *template*, num dado instante.

As imagens integrais são uma estrutura de dados que permitem determinar a soma de valores de uma certa região de uma imagem em apenas quatro operações matemáticas em vez de se ter que efetuar a soma pixel-a-pixel. Considerando isto tanto a média

e a energia local de uma certa região de uma imagem serão também realizadas de modo mais eficiente.

Kaso [20] explora também a realização do método NCC no domínio de frequência, mas desta vez transforma tanto o numerador como o denominador da equação 2.20 para o domínio de frequência. Ainda no mesmo documento é feito uma comparação dos vários métodos referidos, em termos de custos computacionais. De acordo com Kaso, o método de Lewis [9] é mais eficiente do que o método NCC no domínio espacial, no entanto o método desenvolvido por Kaso é menos eficiente do que o método de Lewis.

Para além de se acelerar o método *template matching* ao modificar a própria equação relacionada outra abordagem será a introdução de operações de pré-processamento. A título de exemplo, Liu, Shu e Wu [21] desenvolvem um método que exclui certas regiões de uma imagem de serem aplicadas em algoritmos de comparação de imagens (e.g *template matching*). Esta exclusão de regiões é realizada com o cálculo de determinadas características, sendo estas neste caso a média e gradientes de Prewitt. Ao comparar os valores destas características com os obtidos no *template* é possível determinar a similaridade entre os dois. Se as referidas regiões da imagem diferirem muito do modelo de referência (*template*) é possível excluir as mesmas de serem utilizadas em algoritmos de comparação, acelerando o processo.

Apesar de existir certos mecanismos que aceleram o método *Template Matching* este continua a ser computacionalmente custoso e por isso investigou-se outra alternativa, sendo esta *Chamfer Matching*. Este método foi pela primeira vez introduzido por Barrow et al. [22]. Em primeiro lugar, aplica-se tanto na imagem de referência como no *template* algoritmos de deteção de arestas e aplica-se na imagem de referência uma função de distância. Funções de distância têm como finalidade a conversão de imagens, que possuem pontos de referência, em estruturas de dados cujos os elementos existentes nestes indicam a distância a que se situam do ponto de referência mais próximo. No caso a ser estudado os pontos de referência serão as arestas da imagem original. O ideal seria utilizar a distância euclidiana, mas apesar de este ser o mais preciso é também o mais computacionalmente custoso. Portanto as aproximações utilizadas em DT são de números inteiros. Estas funções de distância, que serão designadas por DT, recorrem a uma matriz relativamente pequena (3x3, 5x5 ou 7x7) que percorre a imagem alvo duas vezes. Em cada iteração por cada elemento da estrutura de dados é realizado uma operação no sentido de fornecer a este a distância ao ponto de referência. A figura 2.19 ilustra o pseudocódigo referente à operação em causa.

Borgefors [23] realizou um estudo sobre a precisão dos DT atuais, sendo estes, mais especificamente: *City Block*, *Chess*, *Octagonal*, *Chamfer 3-4* e *Chamfer 5-7-11*. Em termos de matrizes de dimensão 3x3 o DT Chamfer 3-4 é o mais eficiente, ou seja, é o algoritmo cuja distância produzida é a mais similar com a distância euclidiana. Apesar de o DT Chamfer 5-7-11 ser mais eficiente que o DT Chamfer 3-4 o supramencionada utiliza uma matriz de 5x5 o que envolveria maior custo computacional. Com o intuito de especificar de que maneira o DT Chamfer 3-4 transforma a imagem original ilustra-se a figura 2.20. Para finalizar, itera-se o *template*, que neste caso é uma imagem binária com as arestas salientadas, pela imagem resultante da função DT e multiplica-se o mesmo. O elemento com menor valor será supostamente o *match* do *template*.

Em teoria este método deverá ser mais robusto e menos computacionalmente custoso que o método *Template Matching*, mas esta assunção será comprovada no capítulo “Resultados e Discussão”.

```

#Primeira passagem pela imagem alvo, da esquerda para a direita e de cima
para baixo

for i in range(height):
    for j in range(width):
        F[i, j]=Minimum(F[i, j], F[i-1, j]+3, F[i+1, j]+3, F[i, j+1]+3, F[i-1, j+1]+4, F[i
+1, j+1]+4, F[i, j-1]+3, F[i-1, j-1]+4, F[i+1, j-1]+4)

#Segunda passagem pela imagem alvo, da direita para a esquerda e de baixo
para cima

for i in range(height-1, -1, -1):
    for j in range(width-1, -1, -1):
        F[i, j]=Minimum(F[i, j], F[i-1, j]+3, F[i+1, j]+3, F[i, j+1]+3, F[i-1, j+1]+4, F[i
+1, j+1]+4, F[i, j-1]+3, F[i-1, j-1]+4, F[i+1, j-1]+4)

```

Figura 2.19: Pseudocódigo da função de uma função de distância.

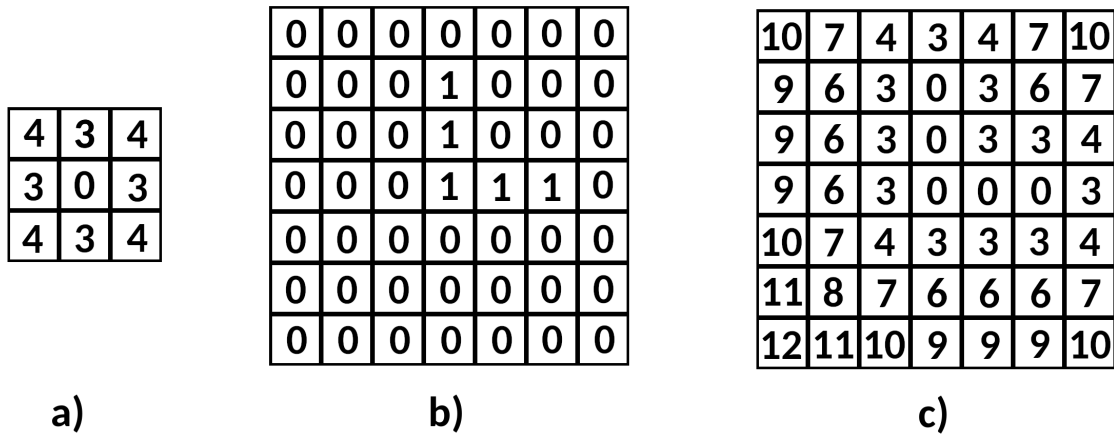


Figura 2.20: Aplicação do DT Chamfer 3-4: (a) Representação do Chamfer 3-4 numa matriz 3x3; (b) Imagem binária a ser empregue o DT; (c) Resultado final da aplicação do DT Chamfer 3-4 sobre a imagem binária b).

## 2.5.2 Geolocalização por comparação de imagens de diferentes perspetivas

Será explorado nesta subsecção a geolocalização com recurso a imagens contendo perspetivas diferentes entre si. Como será exposto de seguida, esta geolocalização poderá ser alcançada com a identificação de relações extrínsecas entre as fontes das imagens ou poderá ser efetuada com alguma ferramenta de inteligência artificial que associa as imagens a determinadas localizações.

Considera-se pertinente referir algumas técnicas que envolvem a segmentação de estradas em imagens de satélite. Ishino e Saji [24] apresentam uma abordagem que segmenta e identifica dois tipos de marcadores rodoviários: passadeiras e marcadores de faixas. Em ambos os casos, primeiramente, segmenta-se uma área de interesse (troço de estrada) (figura 2.21 (a)). De seguida, extrai-se elementos com base nos identificadores de arestas (figura 2.21 (c)). Cria-se um conjunto de retas candidatas (figura 2.21 (d)) e faz-se uma contagem do número de píxeis em cada coluna da imagem (figura 2.21 (b)).

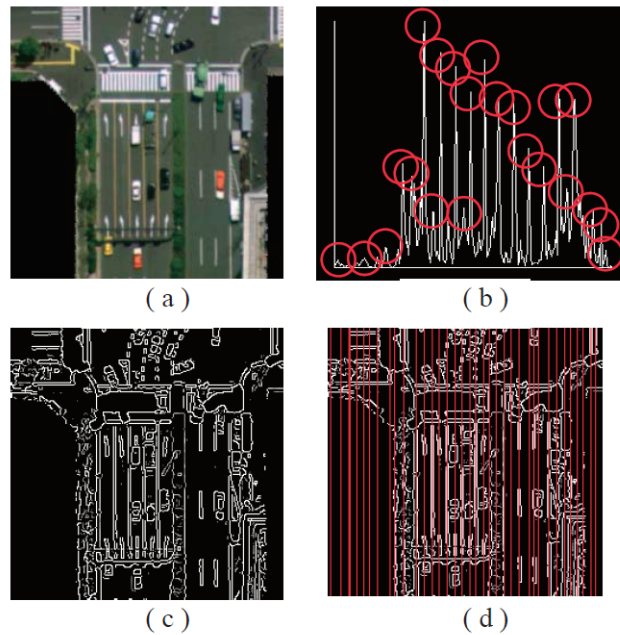


Figura 2.21: Processo de segmentação de marcadores de faixas: (a) imagem a ser processada; (b) histograma da contagem de pixels em cada linha candidata; (c) imagem com todas as arestas segmentadas; (d) sobreposição de linhas candidatas sobre (c) [24].

As arestas com maior valor associado serão os marcadores de faixas. A figura ilustra o exemplo de um resultado.

As imagens aéreas ou de satélite possuem uma visualização da área vista de cima (na maior parte dos casos), no entanto as imagens do veículo possuem o efeito de perspectiva. Por este facto a tarefa de corresponder estes dois tipos de imagens é dificultada. Os seguintes estudos serão explorados com o intuito de demonstrar a complexidade criada pelo efeito de perspectiva.

Bansal, Sawhney, Cheng e Daniilidis [25] ilustram uma abordagem de deteção da correspondência de descritores provenientes de imagens aéreas e imagens de *Street View* (SV). Em cada um dos tipos de imagens identifica-se as fachadas dos edifícios e seguidamente extrai-se destes os descritores “*self-similarity*” e “*HoG*”. Se se identificar os mesmos descritores nas diferentes imagens é possível deduzir a relação entre as posições das duas câmaras. Sendo conhecida a posição da câmara aérea, deduz-se a posição da câmara SV. De modo a validar os resultados escolheu-se manualmente sete correspondências para determinar a matriz de transformação. Com esta abordagem é possível determinar a posição do veículo em áreas urbanas, por existir um grande número de fachadas de edifícios. No entanto, num ambiente rural, os resultados não seriam tão positivos, por não haver edifícios altos.

Muito tipicamente este tipo de localização é realizada recorrendo a CNN’s, pois estes são bastante versáteis em situações muito mutáveis. Kim e Walter [26] exemplificam um sistema que realiza um inquérito com imagem adquirida do veículo de modo a encontrar a imagem aérea associada, empregando CNN’s. Com a imagem aérea identificada é possível localizar o veículo. Para se realizar a correlação entre as imagens



adquiridas do veículo com imagens aéreas usa-se um sistema denominado por *Siamese Neural Network* (SNN) para criar associações entre imagens. Este sistema é a junção de dois CNN's, em que cada um analisa cada fonte (proveniente do veículo ou proveniente do satélite). Os CNN's trabalham em paralelo e cada um realiza, sobre as imagens, transformações não lineares definindo características de aprendizagem. De modo a aumentar a robustez, adicionou-se características de médio nível <sup>2</sup> no treino do sistema, pois estas tendem a invariarem com mudanças na iluminação, tempo meteorológico e mudanças sazonais. Por outro lado características de alto nível fornecem maior robustez face a variações de perspectiva. Por outras palavras, as características de médio nível são mais aptas para discriminação de instâncias em determinados cenários, enquanto que características de alto nível são mais aptas para discriminar cada cenário. A estimação da localização do veículo é realizado com um filtro de partículas. Em cada iteração é usado partículas determinadas anteriormente (*à priori*) e partículas atualizadas com a odometria do carro e do sistema *Siamese Neural Network* (*à posteriori*). O conjunto atual de partículas é atualizada face aos valores e posições dos conjunto de partículas anteriores. Na última fase deste processo as partículas com um "peso" demasiado baixo são descartadas e quando o número de partículas desce abaixo de um certo limite é feito uma re-amostragem. Todo este processo de estimação de localização do veículo é ilustrado na figura 2.22.

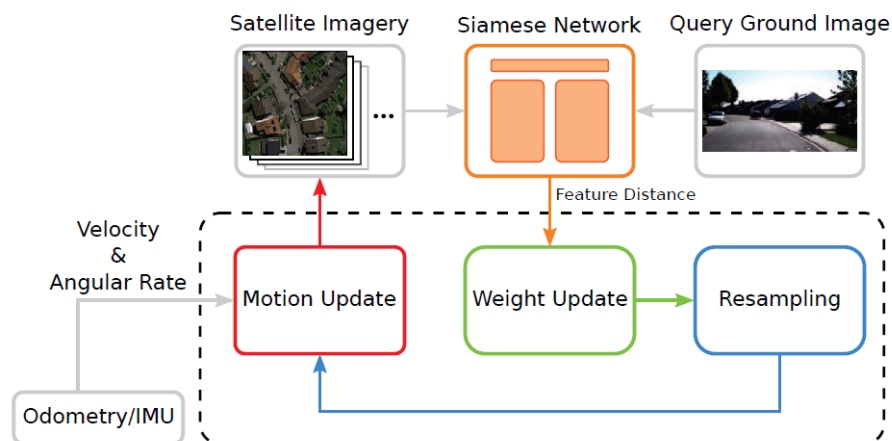


Figura 2.22: Processo de estimação da localização do veículo. Este engloba informação proveniente da odometria e dos inquéritos das imagens. Incluindo, também, a atualização das partículas e re-amostragem das partículas [16].

O sistema em causa é dependente das imagens a que lhe são fornecidas e por isso este falhará quando houver uma mudança radical de uma das localizações a que foi treinado a detetar. Uma dessas situações seria por exemplo quando um novo edifício fosse construído ou quando um que existia anteriormente fosse destruído. Para além desta fragilidade, este sistema não é apto para uma aplicação a uma escala global. Isto porque se este é dependente de imagens a que foi treinado então seria necessário adquirir imagens por todo o globo de todas os troços de estrada. Tornar-se-ia num trabalho

<sup>2</sup>Características de médio nível e características de alto nível são conceitos introduzidos no artigo [26].

exaustivo e insustentável.

Como foi verificado pelos estudos anteriores, quando a imagem adquirida pelo veículo possui o efeito de perspectiva a geolocalização torna-se numa tarefa extremamente complexa para uma aplicação a uma escala global. Portanto, infere-se que a tarefa de geolocalizar o veículo numa imagem de satélite e/ou aérea é simplificada com o uso de imagens IPM, pois a mesma representa a estrada vista de cima.

## 2.6 Apreciação crítica

Este capítulo, “Estado de Arte”, expõe os desenvolvimentos alcançados, até ao momento, relacionados com o método IPM e métodos de geolocalização, e em simultâneo acrescenta uma discussão sobre os mesmos.

O método IPM é um dos focos principais desta dissertação, portanto dedicou-se grande parte deste capítulo na revisão da teoria matemática do mesmo, juntamente com os pré-processamentos associados. Durante a revisão de literatura verificou-se a escassez de estudos que avaliam a precisão das imagens retornadas por este método. Foi também observado uma escassez de estudos que analisam o modo como os tempos de execução são influenciados pela variação de determinados parâmetros inerentes ao método IPM. Decidiu-se, portanto, complementar a escassez de informação referente aos tempos de execução ao longo desta investigação.

Outro tópico fulcral nesta investigação é a geolocalização, mais especificamente, a geolocalização do veículo com recurso a imagens de satélite e/ou aéreas, e imagens adquiridas por câmaras montadas no mesmo. Uma das finalidades desta dissertação é a comparação entre imagens de satélite e imagens IPM (i.e., imagens sem o efeito de perspectiva), no entanto considerou-se importante investigar a geolocalização entre imagens que possuem pontos de vista bastantes distintos entre si. A geolocalização entre estes últimos tipos de imagens força o desenvolvimento de técnicas muito complexas (e.g. SNN’s), sendo que as mesmas não são adequadas para serem aplicadas em qualquer ponto geográfico do planeta. A solução da problemática anterior é o recurso a técnicas de comparação entre imagens de satélite e imagens IPM. De modo a ir ao encontro a esse fim explorou-se a viabilidade de técnicas de comparação (“Template Matching” e “Chamfer Matching”), sendo estas aparentemente boas candidatas para servirem como ferramentas de uma geolocalização precisa e utilizável a uma escala global.

## Capítulo 3

# Metodologia

Este capítulo tem como intuito principal a descrição detalhada dos vários métodos implementados, salientando as limitações, vantagens e desvantagens. De modo a fornecer ao leitor uma melhor compreensão sobre o modo como este capítulo foi estruturado, menciona-se, de seguida, o que se pretende efetuar para cumprir os objetivos propostos.

A finalidade das imagens adquiridas das câmaras instaladas no ATLASCAR2 é a visualização de porções de estrada. Pretende-se aplicar sobre as imagens obtidas o método MM-IPM com o propósito de se obter uma representação da estrada vista de cima e sem obstáculos. O desenvolvimento dos vários métodos IPM focou-se na utilização de imagens provenientes apenas de uma câmara. O seguinte passo consiste em comparar a imagem IPM resultante com imagens de satélite. Pretende-se com esta operação localizar o veículo na imagem satélite e, conseqüentemente, localizar o mesmo em coordenadas geográficas, i.e, geolocalizar o veículo.

Tendo em conta todo o processo de geolocalização do veículo, assume-se como lógico organizar as secções da seguinte forma: IPM e Geolocalização.

### 3.1 Inverse Perspective Mapping

Como referido no Capítulo 2, “Estado de Arte”, o processo de desenvolvimento do método IPM necessita essencialmente de três elementos, sendo estes: 1) a imagem obtida de uma câmara instalada no ATLASCAR2; 2) a relação entre a pose da câmara e a pose arbitrária no mundo; 3) a matriz intrínseca da câmara. A obtenção destes elementos é imprescindível para o desenvolvimento do método IPM e, em função disso, os mesmos serão explorados.

Antes de se iniciar o desenvolvimento do método IPM foi necessário em primeiro lugar obter ficheiros no formato *bagfile*. Estes ficheiros são produzidos pelo sistema ROS e contêm as mensagens provenientes de tópicos durante certos intervalos de tempo. Na fase de prototipagem do método IPM produziu-se um *bagfile* quando o ATLASCAR2 está estacionado na oficina do DEM. O segundo *bagfile* foi produzido do sistema ATLASCAR quando este está a ser conduzido no bairro de Santiago. Os ficheiros mencionados serão designados, respetivamente, por “Target1” e “Target2”. O primeiro ficheiro será alvo de estudo nesta secção e será utilizado como exemplo concreto de extração dos dados necessários para a realização do método IPM. O ficheiro supracitado

é principalmente mencionado no Capítulo 4, “Resultados e Discussão”.

Antes de se utilizar a imagem e a matriz intrínseca provenientes da câmara é fundamental, em primeiro lugar, realizar um processo de calibração sobre a câmara. Isto porque todas as câmaras existentes possuem, de um modo geral, lentes que distorcem a imagem gerada, ou seja, a mesma não representa a realidade com precisão. Este processo foi efetuado com base no tutorial disponível no site oficial do ROS ([6]), cujos passos são, em seguida, enumerados.

De modo a realizar o referido processo é crucial a obtenção de um painel axadrezado de quadrados pretos e brancos, que possuem dimensões idênticas, como ilustrado pela figura 3.1. É imprescindível o conhecimento da dimensão de cada quadrado e, inclusive, o modo como os mesmos estão dispostos nas duas direções (horizontal e vertical).

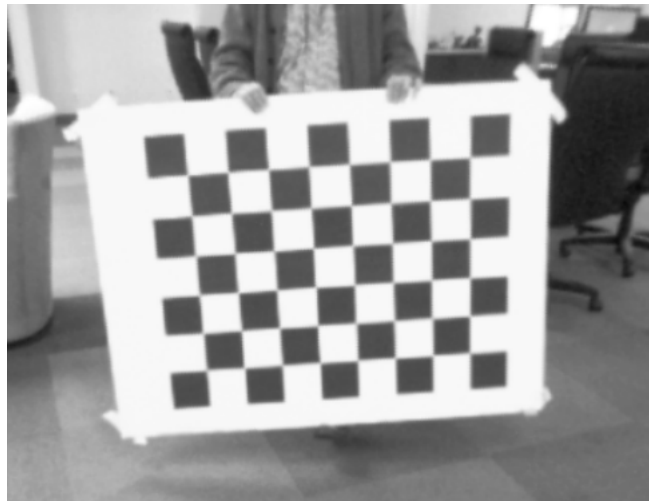


Figura 3.1: Exemplo de um painel axadrezado adequado para a calibração da câmara [6].

Para uma melhor exposição dos passos do tutorial de calibração, existente no site, utiliza-se a situação exemplificada no mesmo. O painel demonstrado pela figura anterior possui uma distribuição de quadrados no formato 8x6 e cada um destes possui uma dimensão de 0.108x0.108 metros. Conhecendo estes valores é possível ativar a interface, que auxilia o utilizador na operação de calibração, com o comando ilustrado pela figura 3.2. É de notar que no exemplo apresentado a imagem é adquirida do tópic “/camera/image\_raw”.

Quando solicitado pela interface, o utilizador deverá mover o painel axadrezado várias posições e ângulos, para uma melhor calibração da câmara. Em cada posição e orientação do painel, a interface apresentará o ilustrado pela figura 3.3. Após se mover o painel suficientes vezes o sistema indicará que a calibração está terminada, podendo-se agora adquirir a imagem e a matriz intrínseca “corrigidas”. Estas serão publicadas

```
roslaunch camera_calibration cameracalibrator.py --size 8x6 --square 0.108 image
:=/camera/image_raw camera:=/camera
```

Figura 3.2: Exemplo de um comando para iniciar a calibração da câmara ([6]).

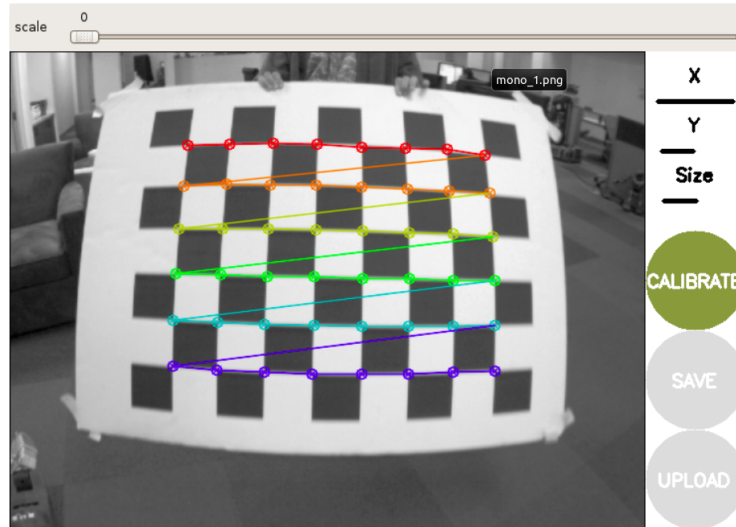


Figura 3.3: Comportamento da interface a novas poses do painel axadreado ([6]).

pelo sistema ROS, respetivamente, em formatos “sensor\_msgs Image” e “sensor\_msgs CameraInfo”, sendo este último possuidor do parâmetro “P” que é a matriz intrínseca calibrada.

O último elemento a ser adquirido é a matriz extrínseca que representa o modo como a câmara se posiciona e orienta em relação ao referencial no mundo, arbitrariamente escolhido. O sistema ATLASCAR2 define vários referenciais, tal como as relações geométricas entre estes, que representam várias zonas importantes no sistema (figura 3.4). O sistema ROS possui ferramentas que permitem a obtenção da matriz extrínseca entre qualquer referencial.

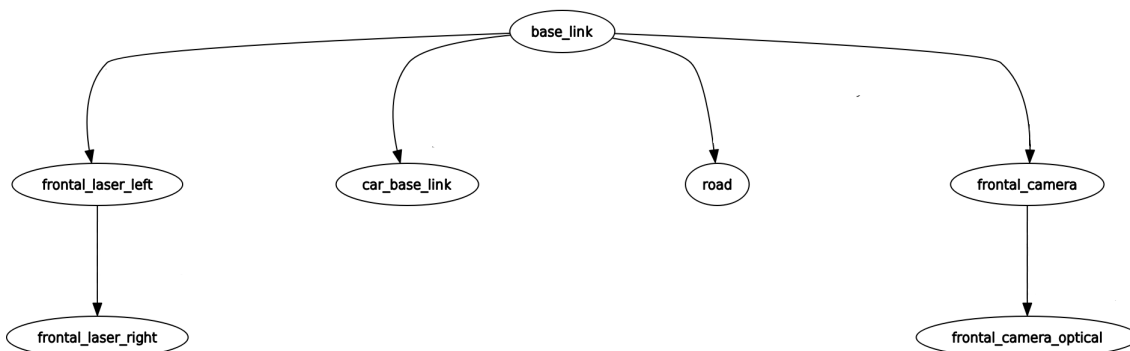


Figura 3.4: Referenciais definidos no sistema ATLASCAR2. O referencial “road” é, neste contexto, equivalente ao referencial “mundo”, que é referido inúmeras vezes ao longo deste documento. Imagem originalmente adquirida através do pacote “rqt” e modificada para uma melhor visualização por parte do leitor.

A construção do método IPM foi fortemente baseada no trabalho de Oliveira e Santos [13] e toda a explicação teórica foi formulada no Capítulo 2, “Estado de Arte”. Segue-se para uma explicação mais específica de como o algoritmo foi construído. Apesar de serem definidas sete abordagens (que serão explicadas mais adiante) para a cons-

trução do algoritmo, todas estas partilham entre si porções de código que podem ser categorizadas em, essencialmente, quatro fases, sendo estas: 1) “Indexing”; 2) “Mapping”; 3) “Masking”; 4) “Interpolation/Smoothing”.

A fase “Indexing” refere-se à porção de algoritmo responsável pela criação das variáveis que contêm as coordenadas cartesianas  $x$  e  $y$  dos píxeis no novo referencial e estas são denominadas, respetivamente, por “Mapx” e “Mapy”. Esta porção de algoritmo aceita como *input* as coordenadas dos píxeis no referencial original (imagem original), a matriz intrínseca e a matriz extrínseca, e retorna como *output* as variáveis “Mapx” e “Mapy”. Foram desenvolvidas três formas distintas de efetuar a operação supramencionada, sendo estas: 1) cálculo pixel-a-pixel por meio de iterações de ciclos; 2) operações matriciais com recurso à biblioteca “numpy”; 3) operações matriciais com recurso à biblioteca “numpy”, juntamente com integração de computação paralela.

A fase “Mapping” corresponde à associação dos píxeis provenientes da imagem original na imagem IPM. Esta fase recorre às variáveis criadas na fase “Indexing” (“Mapx” e “Mapy”) para realizar a referida associação. Esta porção de algoritmo foi construída de duas formas distintas, sendo estas: 1) cálculo pixel-a-pixel por meio de iterações de ciclos; 2) operações matriciais com recurso à biblioteca “numpy”.

As fases “Indexing” e “Mapping” são consideradas as fases essenciais para a implementação do método IPM. No entanto, é necessário ainda a introdução da fase “Masking” para se ter um método MM-IPM, i.e., um método IPM que integra informação proveniente de outros sensores para além das câmaras. Esta fase refere-se à porção de algoritmo responsável pela criação de uma máscara que indica os píxeis representativos de zonas sem obstáculos. Como já foi mencionado no Capítulo 2, “Estado de Arte”, todas as novas coordenadas estarão todas posicionadas num mesmo plano, normalmente definido por  $z = 0$ . Portanto, se existirem objetos com alguma dimensão vertical os mesmos não serão adequadamente representados na imagem IPM final. A definição de uma máscara representativa de zonas sem obstáculos permite a remoção dos referidos obstáculos e, conseqüentemente, representa a realidade com maior precisão. Antes da formação da máscara é necessário recolher a informação proveniente dos sensores lasers instalados no ATLASCAR2, que é representada por uma lista de coordenadas. Estes sensores enviam vários feixes de luz, mas apenas em direções contidas num plano. Em algumas situações, os feixes de luz poderão colidir com alguma superfície que provoca o retorno dos mesmos para o sensor laser que os originou. Nestas situações, o sensor calcula a distância dos obstáculos sabendo o tempo que o feixe de luz demorou a retornar. O sistema ATLASCAR2 possui instalado dois destes sensores em cada lado, como ilustrado pela figura 3.5.

Cada ponto gerado possui informação espacial de três dimensões, no entanto uma destas dimensões é reduntante. A dimensão correspondente à verticalidade do espaço é descartada, uma vez que não será utilizada para a criação da máscara. Com o conjunto de pontos é possível gerar o polígono da área sem obstáculos e sabendo a posição de sensores em relação ao referencial arbitrário do mundo é possível posicionar o polígono no mesmo sistema de coordenadas que a imagem IPM (figura 3.6)(a) e (b). Durante a construção das várias versões desta fase, testou-se quatro modos de utilizar este polígono. No primeiro caso, durante a fase “Indexing”, verificou-se se os píxeis estavam ou não contidos no polígono da área sem obstáculos, recorrendo a funções da biblioteca “shapely” (figura 3.6(c)). Na segunda versão, converteu-se as coordenadas dos pontos gerados pelo laser para coordenadas no referencial da imagem original e criou-se destas



Figura 3.5: Vista frontal do ATLASCAR2.

uma máscara representativa de pixels pertencentes ao polígono da área sem obstáculos (figura 3.7). Esta última versão também utilizou funções da biblioteca “shapely” e funções da biblioteca “cv2” com o objetivo de desenhar o polígono na imagem, sendo estas: 1) “cv2.drawlines”; 2) “cv2.floodfill”.

As duas versões restantes, associadas a esta fase, referem-se ao mesmo conceito de

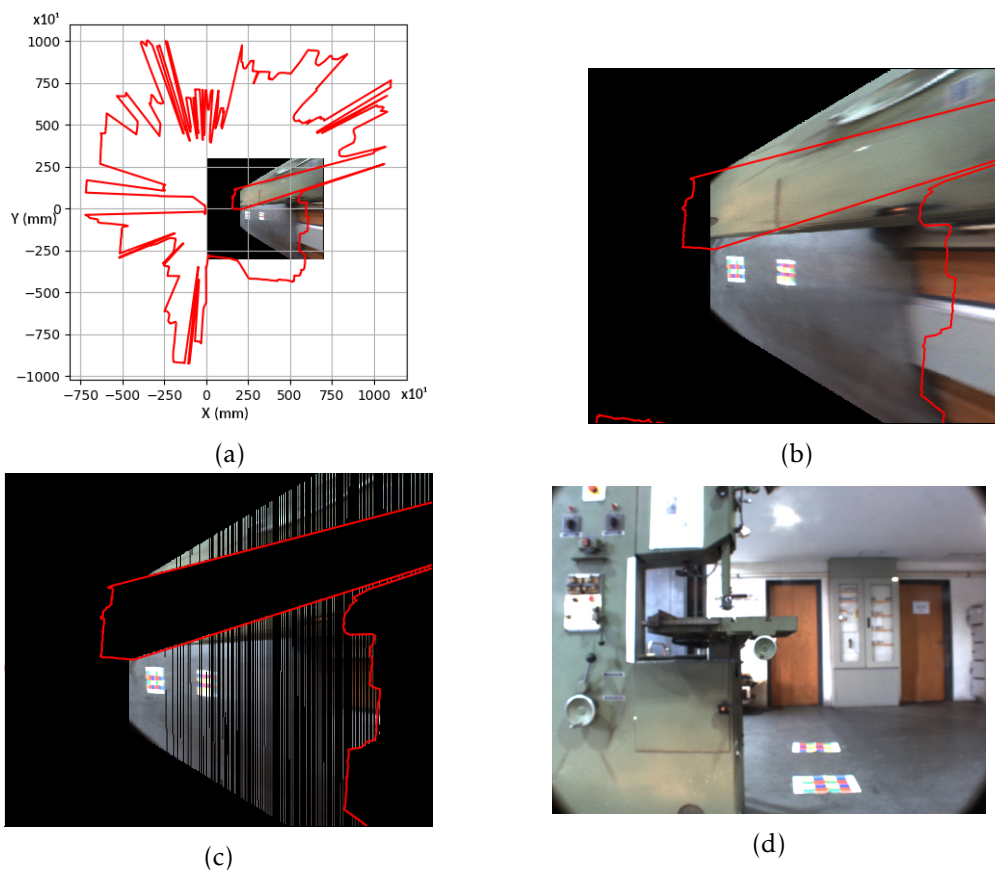


Figura 3.6: Sobreposição do polígono de área sem obstáculos na imagem IPM final. (a) Vista geral; (b) Vista ampliada sobre a área correspondente à imagem IPM; (c) Pixels da imagem IPM contidos dentro do polígono de área sem obstáculos; (d) Imagem original.



Figura 3.7: Conversão do polígono de área sem obstáculos para o referencial da imagem. (a) Sobreposição de polígono de área sem obstáculos na imagem original; (b) Máscara gerada do polígono de área sem obstáculos, no referencial da imagem.

construção de uma máscara recorrendo a coordenadas retornadas pelos sensores lasers, mas em vez de se utilizar as funções da biblioteca “shapely” utilizou-se a função “fillpoly” da biblioteca “cv2”. Estas duas últimas versões diferem-se entre si apenas por criarem máscaras referentes à imagem original ou à imagem IPM. Considerou-se pertinente a criação destas últimas versões da fase “Masking”, uma vez que estas demonstram diferenças em termos de eficiência e precisão da máscara criada (figura 3.8), em comparação com as duas primeiras versões mencionadas.

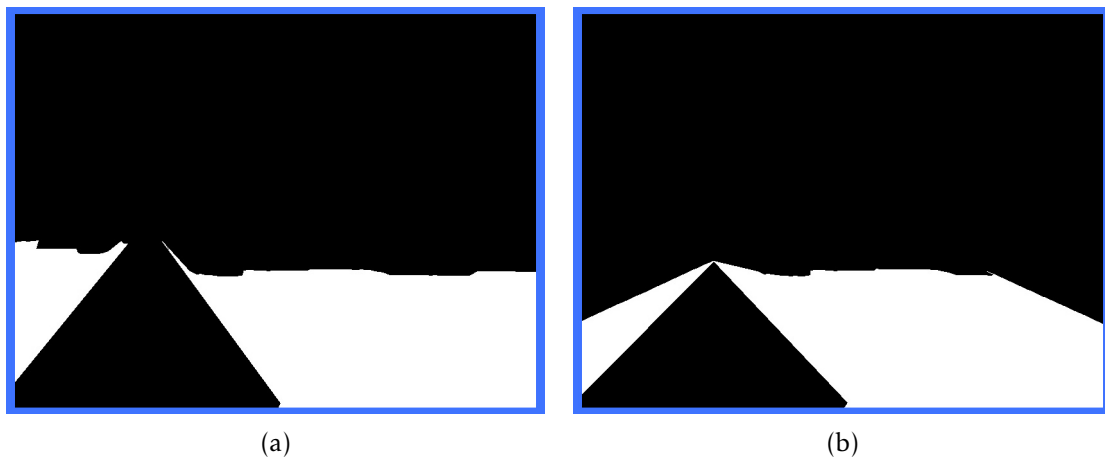


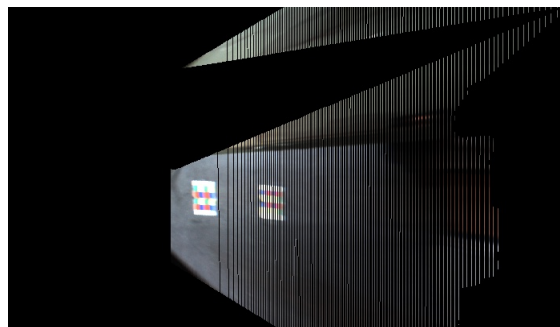
Figura 3.8: Máscaras criadas pela fase “Masking”. (a) Máscara criada pelo método IPM D; (b) Máscara criada pelo método IPM G.

Finalmente, a fase “Interpolation/Smoothing” é, tal como o nome indica, responsável por operações de interpolação ou suavização<sup>1</sup>. Esta fase é, normalmente, a última a ser implementada no método IPM. Antes da sua implementação, a imagem IPM é

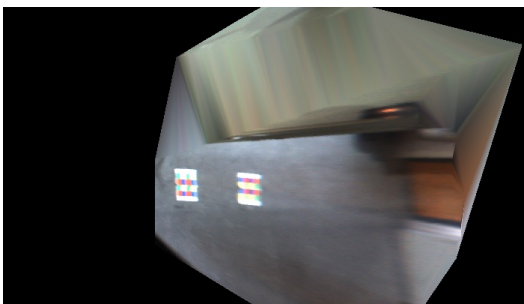
<sup>1</sup>A operação de suavização, que será referida daqui em diante, é uma mediana que é realizada sobre a imagem.



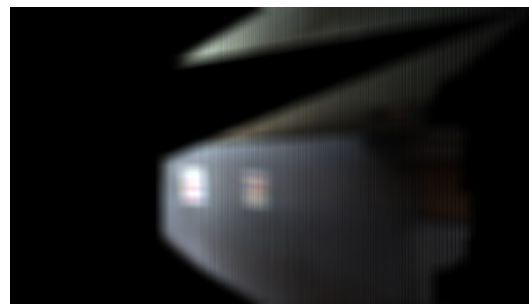
esparsa e de modo a aprontar a mesma para futuras operações é necessário “eliminar” píxeis sem informação por entre píxeis com informação, sendo esta fase crucial para essa “conversão”. Em termos visuais, a operação de interpolação retorna melhores resultados do que a operação de suavização. Apesar disso, esta última poderá também retornar bons resultados com um ajuste dos parâmetros utilizados na realização da mediana. A operação de interpolação poderá originar resultados erróneos se a máscara foi utilizada anteriormente a esta fase, como demonstrado pela figura 3.9 (b). Isto ocorre porque a operação de interpolação é aplicada apenas sobre os píxeis com informação, portanto caso um obstáculo seja detetado no centro da imagem IPM este será “tapado” devido à própria operação de interpolação.



(a)



(b)



(c)

Figura 3.9: Resultados da fase “Interpolation/Smoothing”. (a) Imagem IPM antes da fase “Interpolation/Smoothing”; (b) Imagem IPM após o processo de interpolação; (c) Imagem IPM após o processo de suavização.

Como o método IPM é tipicamente utilizado para aplicações de condução autónoma é crucial um bom desempenho do mesmo, em termos de custo computacional, i.e., necessita de processar os dados em tempo real. Por esta razão, durante esta investigação, a construção deste algoritmo foi abordada de sete modos distintos. As abordagens implementadas serão denominadas por IPM A, IPM B, IPM C, IPM D, IPM E, IPM F e IPM G. As características de cada método estão representadas na tabela 3.1 e estas serão de seguida sucintamente explicadas.

O método IPM A foi o primeiro método a ser desenvolvido e este não possui a fase “Masking”, sendo a sua estrutura bastante linear (figura 3.10). O método IPM B não constrói uma máscara representativa da área sem obstáculos, mas utiliza o polígono gerado pelos sensores lasers na fase “Indexing” de modo a “escolher” os píxeis que serão

Tabela 3.1: Modo como cada método IPM foi construída.

Métodos	“Masking”	“Indexing”	“Mapping”	“Interpolation/Smoothing”
IPM A	N/A	Operações Pixel-a-Pixel	Operações pixel-a-pixel	Interpolação
IPM B	Verificação coordenada-a-coordenada no polígono com recurso à biblioteca “shapely”. Referencial da imagem IPM.	Operações Pixel-a-Pixel	Operações pixel-a-pixel	Interpolação
IPM C	Com recurso à biblioteca “shapely”. Máscara sobre a imagem original.	Operações Pixel-a-Pixel	Operações pixel-a-pixel	Interpolação
IPM D	Com recurso à biblioteca “shapely”. Máscara sobre a imagem original.	Operações matriciais	Operações matriciais	Interpolação
IPM E	Com recurso à biblioteca “shapely”. Máscara sobre a imagem original.	Operações Vetoriais + Computação paralela	Operações matriciais	Interpolação
IPM F	Com recurso à função “fillpoly”. Máscara sobre a imagem IPM.	Operações matriciais	Operações matriciais	Interpolação
IPM G	Com recurso à função “fillpoly”. Máscara sobre a imagem original.	Operações matriciais	Operações matriciais	Suavização

representados na imagem IPM final (figura 3.11). Os métodos IPM C, IPM D, IPM E e IPM G controem uma máscara representativa da área sem obstáculos referente à imagem original. A fase “Indexing” considera apenas píxeis que se encontram presentes na referida máscara e, conseqüentemente, é produzido a imagem IPM com os potenciais obstáculos removidos (figura 3.12). Finalmente, o método IPM F implementa as fases “Indexing”, “Mapping” e “Interpolation/Smoothing” e só depois utiliza uma máscara representativa da área sem obstáculos referente à imagem IPM de modo a remover potenciais obstáculos. Desta forma, a imagem IPM resultante não será distorcida pela operação de interpolação, pois esta operação será aplicada sobre a imagem previamente à remoção dos píxeis da mesma.

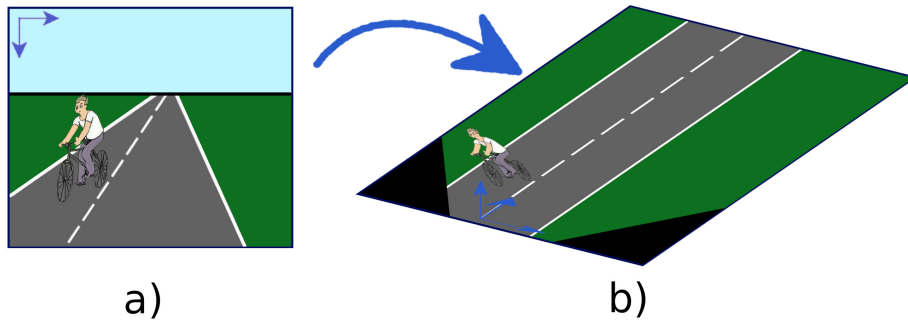


Figura 3.10: Estrutura do método IPM A. (a) Imagem original; (b) Imagem IPM final.

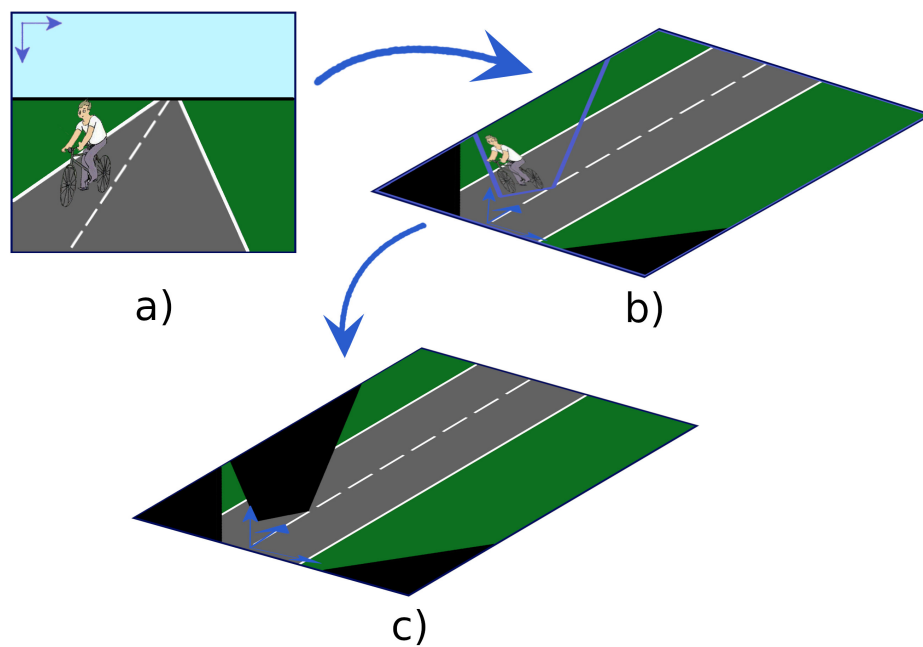


Figura 3.11: Estrutura do método IPM B. (a) Imagem original; (b) Imagem IPM com sobreposição do polígono da área sem obstáculos; (c) Imagem IPM final.

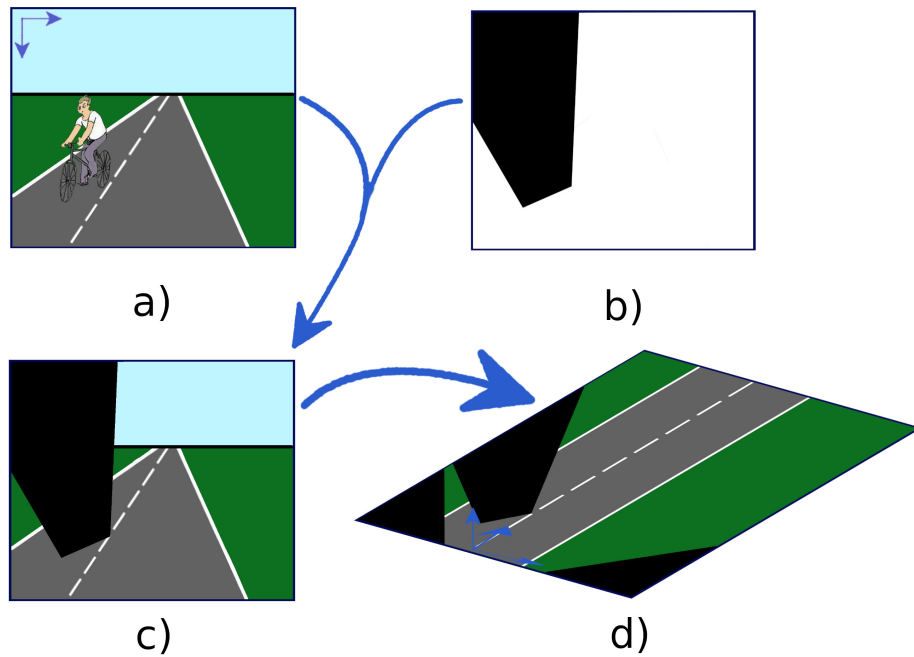


Figura 3.12: Estrutura dos métodos IPM C, IPM D, IPM E e IPM G. (a) Imagem original; (b) Máscara representativa da área sem obstáculos, no referencial da imagem original; (c) Imagem original após a remoção de obstáculos; (d) Imagem IPM final.

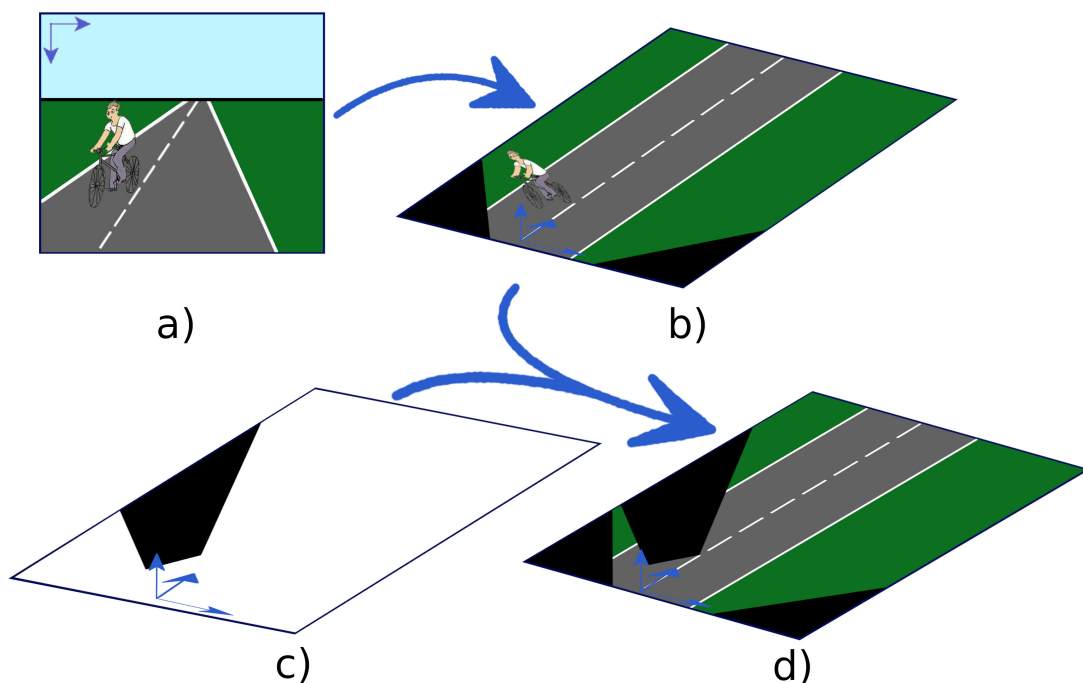


Figura 3.13: Estrutura dos métodos IPM F. (a) Imagem original; (b) Imagem IPM; (c) Máscara representativa da área sem obstáculos, no referencial da imagem IPM; (d) Imagem IPM final.

## 3.2 Geolocalização

Esta secção consiste na localização de uma porção de estrada quando comparada a uma imagem satélite. Será explorado de seguida o modo como as coordenadas geográficas poderão ser associadas a coordenadas de píxeis de uma imagem satélite. Será também alvo de grande foco o estudo de várias abordagens de localização de segmentos de estrada em imagens satélite, sendo estas: *Template Matching* e *Chamfer Matching*. Cada uma destas abordagens possui variantes na implementação, que serão também explicitadas nesta secção. As técnicas de comparação desenvolvidas serão testadas num ambiente simplificado, i.e., os algoritmos de comparação utilizarão uma imagem satélite e um *template* extraído desta.

### 3.2.1 Imagem Satélite

Antes de se prosseguir para a exposição dos algoritmos de comparação entre imagens é pertinente referir de que modo é possível associar as coordenadas geográficas a coordenadas de píxeis. Para se efetuar esta associação entre coordenadas de diferentes referenciais optou-se por criar imagens no formato “tiff”. Este formato de imagem possui a capacidade de atribuir a cada coordenada de píxeis outras informações pertinentes, como por exemplo coordenadas geográficas em latitude e longitude.

De modo a obter imagens no formato “tiff” é necessário, em primeiro lugar, a aquisição de imagens de satélite. Recorreu-se à aplicação “Google Maps”, que possui opções embutidas que permitem a extração de imagens em regiões especificadas. No entanto, antes de se realizar a aquisição da imagem satélite é crucial a remoção de etiquetas<sup>2</sup> existentes na interface, pois estes especificam nomes das ruas, rios, praças e outras informações. Caso não se removam as referidas etiquetas, as mesmas iriam aparecer visíveis na imagem de satélite e, conseqüentemente, estas iriam interferir nos algoritmos de comparação.

Recorreu-se ao *software* “QGIS Desktop” que possui ferramentas próprias para converter a imagem adquirida no formato “tiff”. De um modo sucinto, utiliza-se a ferramenta “Georeferencer”, importa-se a imagem satélite e adiciona-se, nos locais adequados, pontos de referência com as respetivas coordenadas geográficas. O último passo de todo este processo é indicar os parâmetros de georeferenciamento, ou seja, deve-se escolher o tipo de aproximação (aproximação linear, polinomial, etc...) e o modelo do globo que deve ser utilizado. Foi escolhido o tipo de aproximação “polynomial 1” e o modelo “WGS 84/Pseudo-Mercator”, que é considerado o modelo *standard*. Todo o processo de utilização do *software* é ilustrado pela figura 3.14.

---

<sup>2</sup>Elementos inerentes à aplicação “Google Maps”.

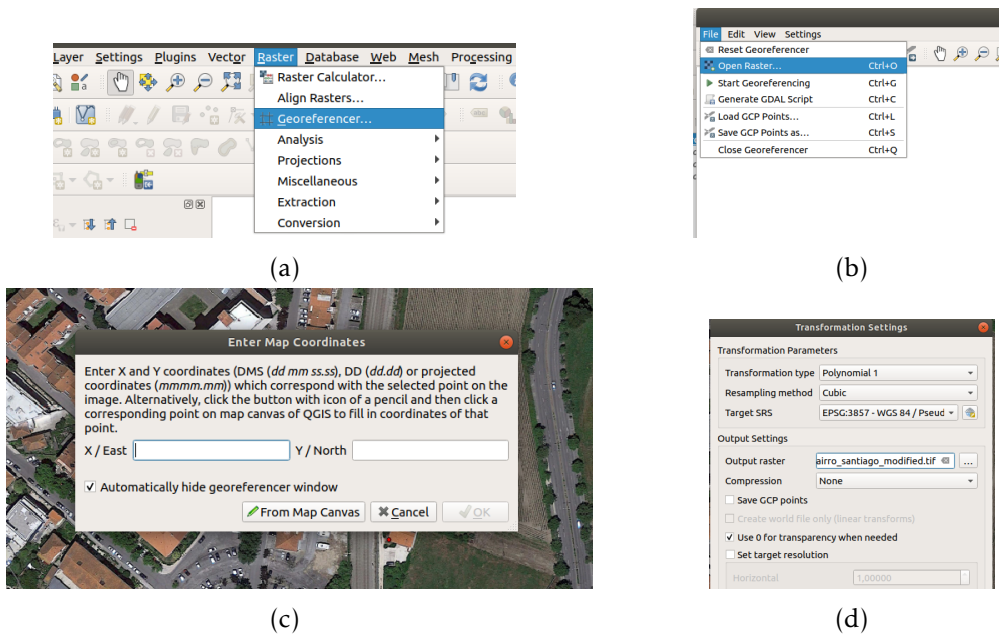


Figura 3.14: Processo de criação de uma imagem no formato “tiff”. (a) Iniciação da ferramenta “Georeferencer”, responsável pela criação do ficheiro pretendido; (b) Após se iniciar a ferramenta “Georeferencer” adiciona-se a imagem que se pretende converter para o formato “tiff”; (c) Após adição da imagem, ao se clicar em algum ponto (sobre a imagem) uma janela apresenta-se e indica ao utilizador para atribuir coordenadas geográficas no ponto onde este clicou; (d) Configurações finais antes de se gerar o ficheiro pretendido.

### 3.2.2 Template Matching

Como referido no Capítulo 2, “Estado de Arte”, existem vários modos de realizar geolocalização e uma das técnicas que se optou por desenvolver e testar foi o *Template Matching*. De um modo sucinto, este método “desliza” o *template* sobre a imagem alvo e em cada posição são executadas determinadas operações matemáticas, previamente mencionadas no Capítulo 2, “Estado de Arte”. Depois de se posicionar o *template* por toda a imagem alvo é retornado, pelo algoritmo, uma matriz cujos elementos possuem valores de correlação, i.e., cada elemento representa o grau de similaridade com o *template*. Cada coordenada de cada elemento da matriz de correlação representa o centro da região na imagem alvo da qual foi comparada com o *template*. Em condições simuladas, sabe-se *à priori* o ângulo a que o *template* necessita de estar rodado para localizar o mesmo na imagem alvo. No entanto, em condições reais, o *template* é uma porção de estrada adquirida localmente e pretende-se localizar o mesmo numa imagem de satélite. Supõem-se à partida que não se sabe o ângulo a que o *template* está rodado, portanto a sua localização na imagem alvo implica a execução do método *Template Matching* para cada rotação diferente do *template*.

Para além de haver necessidade em executar o *Template Matching* várias vezes, tornando-o muito ineficiente, é necessário ainda modificar o algoritmo de comparação de maneira a suportar a rotação do *template*. Isto porque a rotação do *template* gera píxeis pretos (píxeis sem informação), como ilustrado pela figura 3.15. Com o intuito de resolver

esta problemática é criada uma máscara referente à informação pertinente e rodou-se a mesma em simultâneo com o *template*. Esta versão de *Template Matching* é denominada por *Classic Template Matching* (CTM).

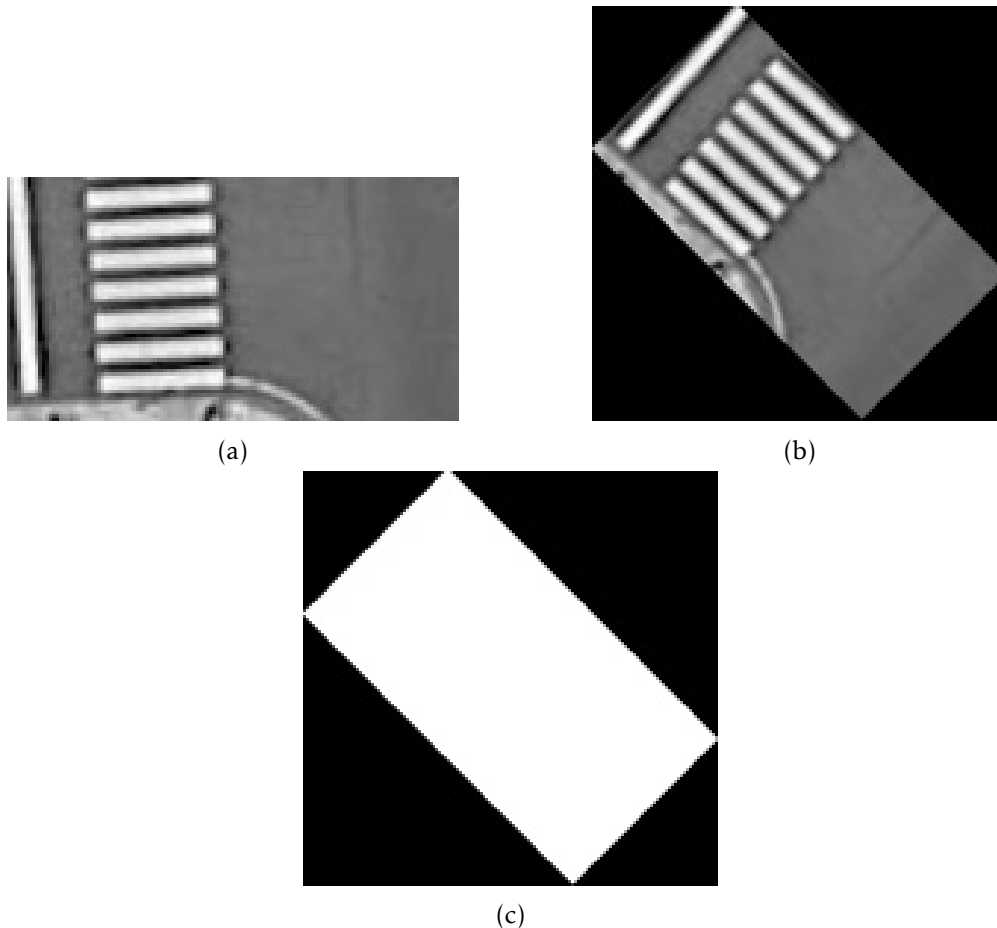


Figura 3.15: Rotação do *template*. a) *Template* com nenhuma rotação aplicada; b) *Template* com uma rotação de 45 graus. c) Máscara correspondente do *template* com uma rotação de 45 graus.

Algumas imagens de satélite utilizadas para testar este algoritmo ilustram porções de estrada compridas, sem traços distinguíveis ao longo das mesmas. Supõem-se por isto que se o ângulo de rotação do *template* diferir demasiado do ângulo *ground truth* a matriz de correlação poderá indicar erroneamente a localização do *template* na imagem alvo.

De modo a mitigar este possível problema introduziu-se uma matriz cujos valores dos elementos são organizados com uma distribuição gaussiana, i.e., quanto mais afastado o elemento da matriz se encontra do centro da distribuição menor será o seu valor. O centro desta distribuição gaussiana será colocado no local mais provável de ser a localização do *template*. Tanto nas condições simuladas como nas condições reais a distribuição gaussiana se irá centrar nas posições *ground truth* do *template*. A posição *ground truth* em condições reais foi determinada manualmente, mas numa hipotética implementação em tempo real a posição onde se iria centrar a distribuição gaussiana

seria determinada por meio de outros sistemas de localização no veículo. Esta matriz é multiplicada com a matriz de correlação obtida do CTM originando uma nova matriz de correlação. Este algoritmo será denominado por *Gaussian Template Matching* (GTM). A figura 3.16 ilustra de que forma a matriz de correlação se comporta quando esta é multiplicada por uma matriz com uma distribuição gaussiana.

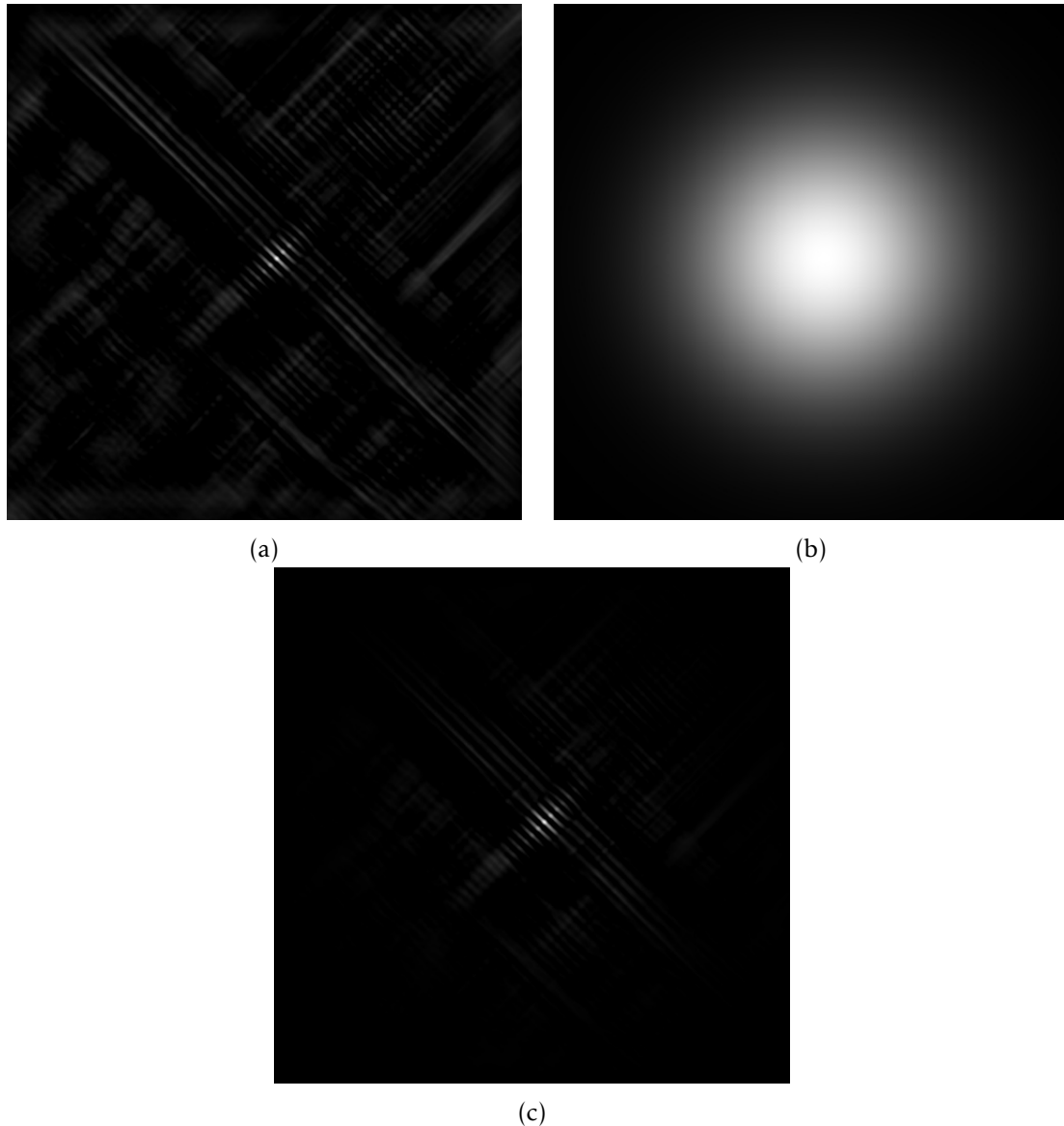


Figura 3.16: Exemplo de matrizes intrínsecas ao método *Template Matching*. a) Matriz de correlação do algoritmo CTM; b) Distribuição Gaussiana; c) Matriz de correlação do algoritmo GTM.

### 3.2.3 Chamfer Matching

Durante o desenvolvimento dos algoritmos CTM e GTM, foi verificado que estes são demasiado ineficientes para aplicações em tempo real e que um pequeno desfazamento



do ângulo de rotação do *template* com o ângulo *ground truth* gera um grande desfazamento da posição em relação à posição *ground truth*. Por esta razão investigaram-se outras possíveis abordagens a serem aplicadas e decidiu-se testar a performance do chamado *Chamfer Matching*. Tal como o método *Template Matching* este foi também mencionado no Capítulo 2, “Estado de Arte”. Pelas mesmas razões justificadas na secção 3.2.2, deve-se também executar o método *Chamfer Matching* para vários ângulos de rotação do *template*. Neste método é realizado sobre o *template* um algoritmo de deteção de arestas, mas o mesmo continuará, daqui em diante, a ser denominado como *template*.

Durante o desenvolvimento deste método analisaram-se os algoritmos de deteção de arestas mais adequados a serem aplicados tanto na imagem de satélite como no *template*. Após se aplicar o algoritmo de deteção de arestas sobre a imagem de satélite é aplicado de seguida uma função de distância. A rotação do *template* varia a dimensão do mesmo e, conseqüentemente, varia a dimensão das matrizes de *padding*<sup>3</sup> a serem adicionadas à imagem alvo. Uma solução para esta problemática é a adição de matrizes de *padding* de dimensões suficientemente grandes para considerar as maiores variações do tamanho do *template* e só depois aplicar a função de distância. De seguida, remove-se as matrizes de *padding*, nas quais também foram aplicadas a função de distância. Aquando a verificação da dimensão do *template*, após a sua rotação, adiciona-se à imagem alvo as matrizes de *padding*, previamente removidas, mas agora com as dimensões corretas. Todo este procedimento é ilustrado pela figura 3.17.

Similarmente aos métodos CTM e GTM, existe a versão original do *Chamfer Matching* denominada por *Classic Chamfer Matching* (CCM) e uma versão que introduz uma distribuição gaussiana denominada por *Gaussian Chamfer Matching* (GCM). Ao contrário do algoritmo GTM, a distribuição gaussiana introduzida em GCM é invertida, ou seja, quanto mais afastados os elementos se encontram do centro da distribuição maior será os seus valores. A figura 3.18 ilustra o modo como a matriz de correlação se comporta quando este é aplicado sobre uma matriz com uma distribuição gaussiana.

---

<sup>3</sup>As matrizes de *padding*, neste contexto, referem-se a matrizes que se adicionam à imagem alvo de modo a que os *templates* utilizados em algoritmos de comparação percorram todos os píxeis da imagem original e atribuem valores aos mesmos.

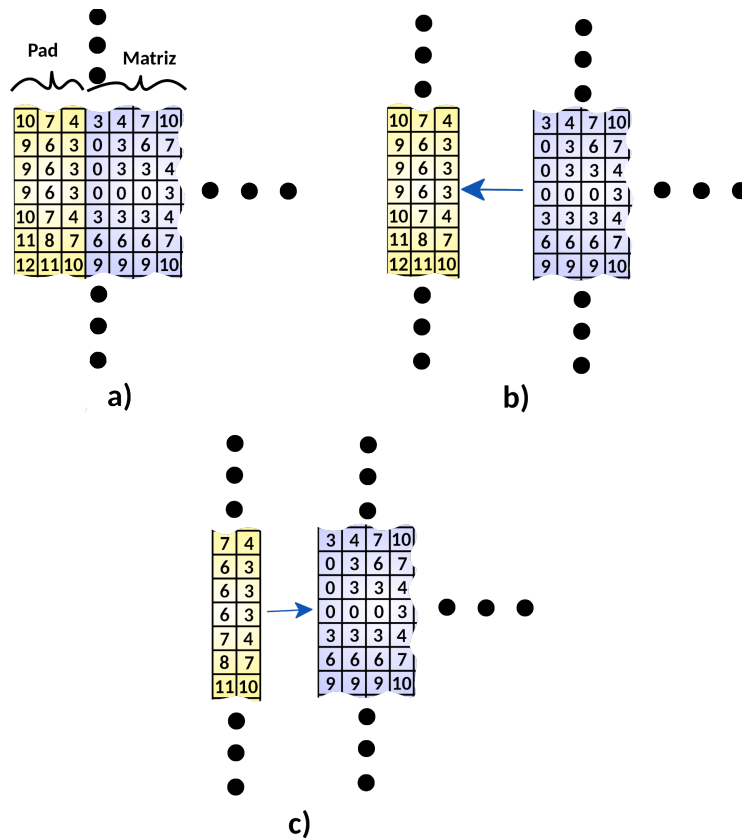


Figura 3.17: Procedimento de adição da matriz de *padding* de dimensões adequadas. O esquema exemplifica uma situação em que no pior caso a matriz de *padding* possui três colunas, mas que na situação atual apenas é necessário uma matriz de *padding* com duas colunas. A azul é representado elementos da matriz que não serão removidos, independentemente da dimensão do *template*. A amarelo é representado elementos das matrizes de *padding*. Os pontos a preto representam a continuidade das matrizes ilustradas, i.e., é apenas ilustrado pequenas porções das matrizes, para fins exemplificativos. (a) Matriz inicial resultante da função de distância; (b) Remoção das matrizes de *padding*; (c) Adicionamento das matrizes de *padding*, previamente removidas, agora com dimensões reduzidas.

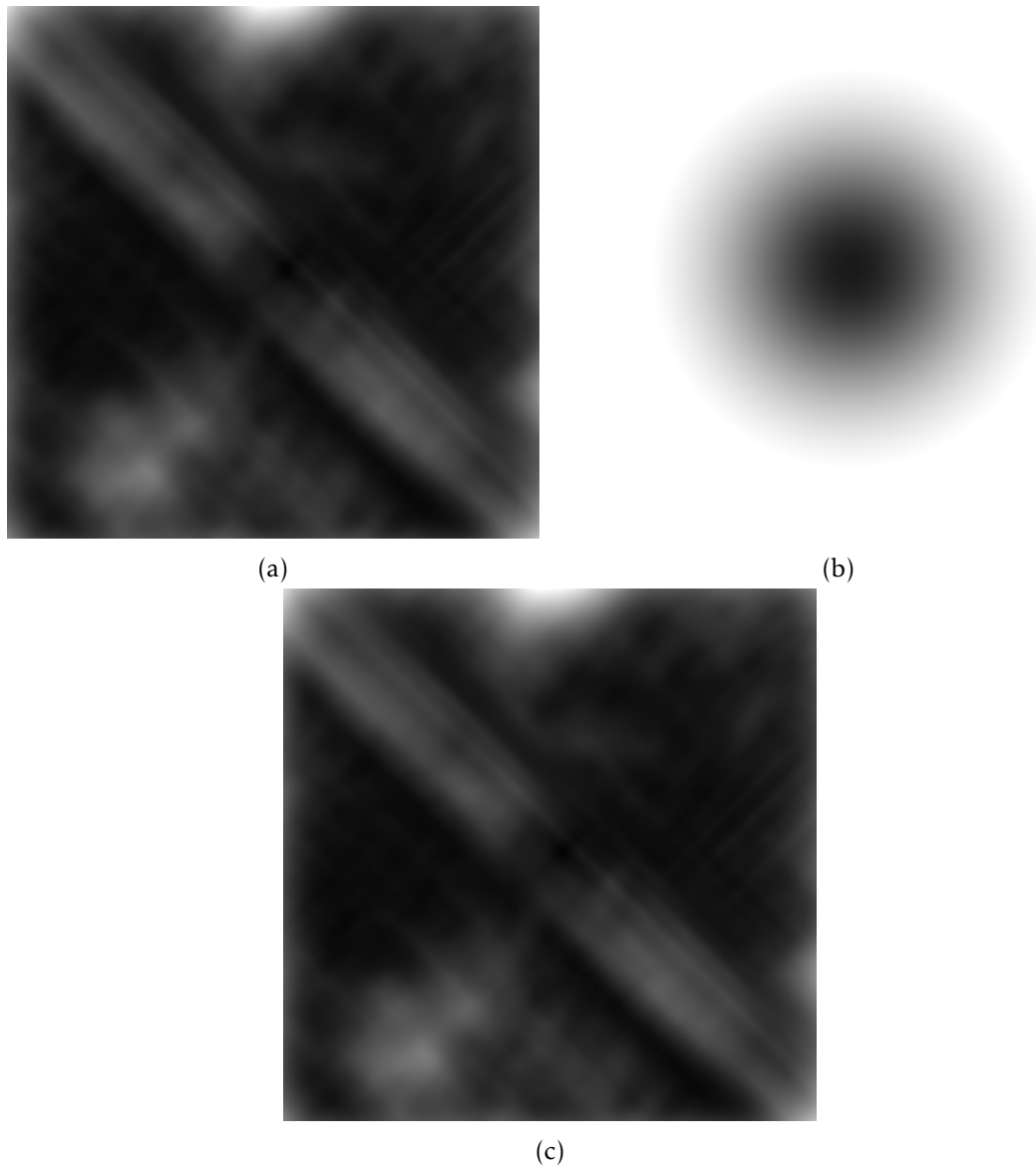


Figura 3.18: Exemplo de matrizes intrínsecas ao método *Chamfer Matching*. a) Matriz de correlação do algoritmo CCM; b) Distribuição Gaussiana invertida; c) Matriz de correlação do algoritmo GCM.



## Capítulo 4

# Resultados e Discussão

Este capítulo possui como objetivo principal a exposição e discussão dos resultados obtidos, tanto do desenvolvimento das várias abordagens do método IPM como das várias abordagens dos métodos *Template Matching* e *Chamfer Matching*, sendo estas últimas relacionadas com a geolocalização. Estes métodos são independentes entre si, o que permitiu um desenvolvimento e validação de resultados independentes entre si. No caso do método IPM, a validação será focada na performance em termos de tempos de execução. Isto porque desde o início desta investigação sempre se teve em conta a aplicação do algoritmo completo em veículos, necessitando por isso que o algoritmo final seja executado em tempo real. No caso dos algoritmos de geolocalização, considera-se pertinente tanto o custo computacional como a precisão dos resultados, uma vez que é pretendido, neste caso, localizar um veículo. É, portanto, importante chegar ao resultado o mais rápido possível com a maior precisão possível.

### 4.1 Performance do método Inverse Perspective Mapping

Como já foi referido no Capítulo 3, “Metodologia”, desenvolveram-se várias versões do método IPM e cada uma destas possui determinadas características que influenciam o custo computacional de alguma forma. Desenvolveram-se os métodos designados por IPM A, IPM B, IPM C, IPM D, IPM E, IPM F e IPM G, cujas diferenças entre si foram explicitadas no Capítulo 3, “Metodologia”. No decorrer desta secção estas diferenças serão sucintamente explicadas de modo a contextualizar os resultados obtidos.

Os dados que serão analisados nesta secção foram adquiridos do bagfile “Target1”, ou seja, adquiridos do sistema ATLASCAR2 quando este está estacionado na oficina do DEM. A imagem a ser utilizada para a recolha de dados é ilustrada pela figura 4.1 e a mesma possui uma resolução de 1224x1624 píxeis. Como é possível observar, a imagem não ilustra um segmento de estrada, mas sim uma porção da oficina do DEM. Este facto não é pertinente, isto porque apenas se pretende expor a eficiência dos métodos desenvolvidos em termos de custo computacional.

Antes de se prosseguir para a discussão dos métodos desenvolvidos é necessário expor de que modo os valores dos tempos de execução foram adquiridos. Para a aquisição dos tempos de execução de cada algoritmo desenvolvido executou-se o mesmo cinco vezes, mantendo constante os parâmetros com que se pretende analisar o método, e realizou-se a média dos resultados obtidos. Ao contrário do que se pensava inicial-



Figura 4.1: Imagem adquirida do sistema ATLASCAR2 quando este está estacionado na oficina do DEM.

mente a média realizada sobre os tempos de execução não é suficiente para mitigar a variação dos mesmos, em algumas situações. No entanto, a flutuação dos valores dos tempos de execução não é suficiente para afetar a discussão de resultados. Para além dos métodos serem bastante diferenciáveis entre si, em termos de custo computacional, a variação dos tempos de execução é, na maior parte dos casos, proporcional à variação de alguns parâmetros, nos casos em que estes realmente afetam a performance do método. Ilustra-se na figura 4.2 os tempos de execução da fase “Masking”. A média e o desvio padrão representados foram calculados recorrendo aos valores dos tempos de execução dos métodos IPM C, IPM D e IPM E. Como é observável pelo gráfico, juntamente com a tabela no Apêndice, o valor do tempo de execução, na criação da máscara, é inicialmente relativamente elevado aquando a utilização de uma imagem de pouca resolução. Com o aumento da resolução o valor do tempo de execução diminui e o mesmo volta a aumentar para os valores mais elevados de resolução. Esperava-se que o aumento do valor do tempo de execução fosse diretamente proporcional ao aumento da resolução, mas tal não é verificável devido à variabilidade dos valores adquiridos.

No resto desta secção será exposto o comportamento (tempo de execução) de cada um dos métodos desenvolvidos face à variação de determinados parâmetros. Para além de se estudar o modo como o tempo de execução entre cada método varia, pretende-se também observar o comportamento do tempo de execução de cada fase de cada método. Cada um destes métodos pode ser dividido em, essencialmente, quatro fases, sendo estas: 1) “Masking”; 2) “Mapping”; 3) “Indexing”; 4) “Interpolation/Smoothing”. Analisar-se-á ao longo desta secção o comportamento de cada método face à variação de determinados parâmetros inerentes ao método IPM, finalizando com uma última análise sobre o método mais adequado a ser aplicado como algoritmo final. Os

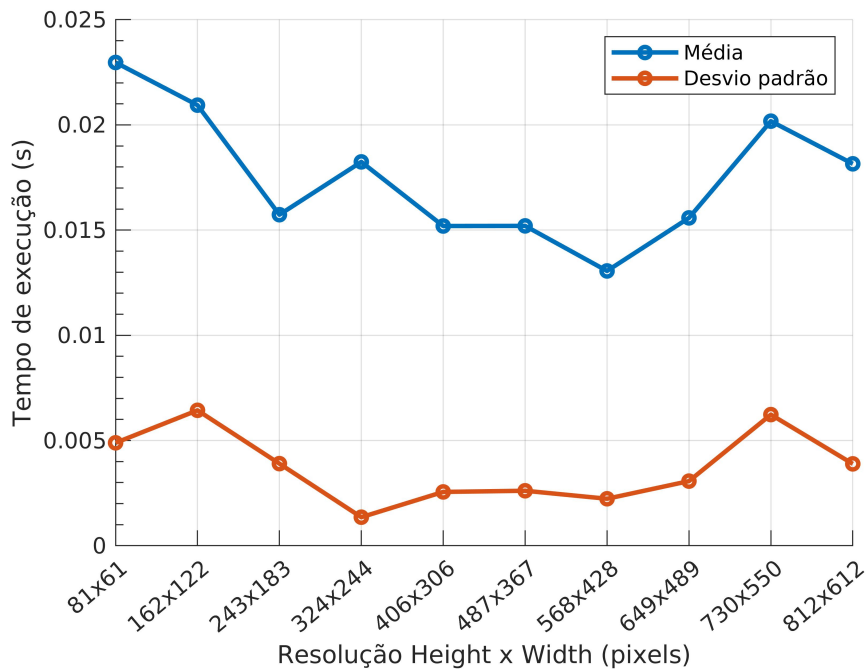


Figura 4.2: Evolução da média e desvio padrão dos tempos de execução, na construção da máscara representativa da área sem obstáculos, consoante a variação da resolução da imagem original.

parâmetros a serem utilizados, à exceção de quando indicado o contrário, são 10% de diminuição de resolução da imagem original, um RSF de 10% e uma área de polígono de 5 por 5 metros.

#### 4.1.1 Variação da resolução

A resolução da imagem a ser tratada é um fator importante no método IPM, uma vez que esta define a “quantidade” de informação que a imagem IPM irá possuir. Por outras palavras, se a imagem original possui maior resolução, por consequência, a imagem IPM final possuirá também um maior número de píxeis com informação. A resolução original (neste caso) é excessivamente elevada e desnecessária, no entanto a diminuição excessiva da imagem original implicará a perda de alguma informação que poderá ser pertinente. Para além de se querer balancear a resolução da imagem com a sua perda de informação é também crucial considerar que uma resolução pequena implica pouco custo computacional e vice-versa. Este custo computacional advém do facto da resolução ser diretamente proporcional ao número de píxeis dos quais devem ser calculadas as novas coordenadas no novo referencial.

Como já foi referido no Capítulo 3, “Metodologia”, a fase “Masking” possui três versões e verifica-se, por observação do gráfico 4.3, que os tempos de execução dos mesmos permanecem constantes consoante a variação da resolução da imagem. Os tempos de execução permanecem constantes porque a fase “Masking” não depende do número de píxeis com informação, mas sim do tamanho da imagem que é gerada.

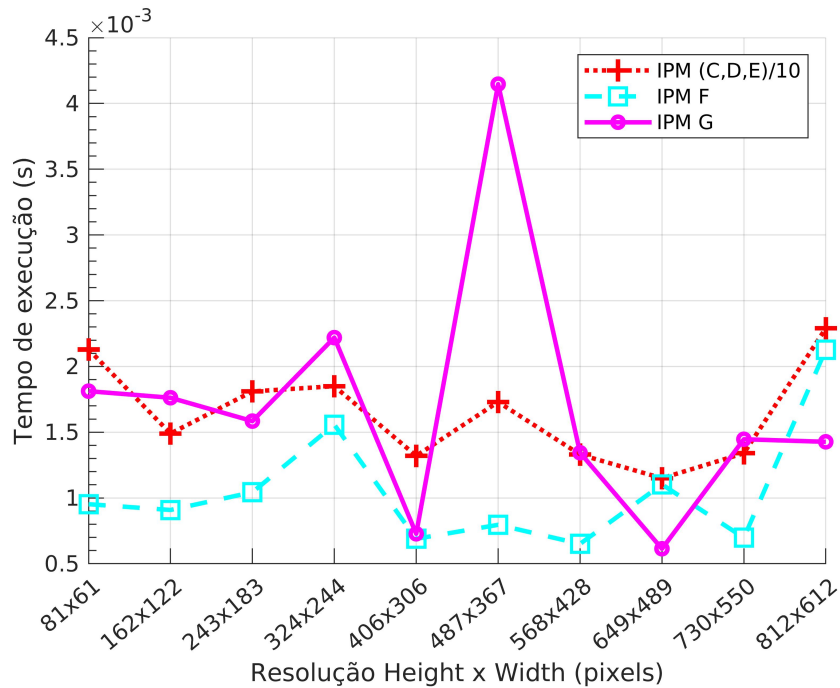


Figura 4.3: Evolução dos tempos de execução da fase “Masking” entre todos os métodos, face à variação da resolução da imagem original.

A fase “Interpolation/Smoothing” possui duas versões que são, tal como o nome indica, operações de interpolação e suavização. Por observação do gráfico 4.4 verifica-se que os tempos de execução dos métodos IPM A, IPM B, IPM C, IPM D, IPM E e IPM F (interpolação) aumentam proporcionalmente com o aumento da resolução da imagem original. Isto ocorre porque uma maior resolução gera um maior número de píxeis na imagem resultante e, conseqüentemente, gera um maior número de píxeis que o algoritmo de interpolação deverá processar. A diferença essencial entre os métodos IPM A e IPM D é o facto de o primeiro não incluir a fase “Masking”, ou seja, não é criada a máscara que representa zonas da imagem original que não possuem obstáculos. No caso do método IPM F a máscara só é aplicada após a operação de interpolação, portanto esta operação irá processar todos os píxeis, inclusive píxeis que representam obstáculos, tal como no método IPM A. Esta máscara, em situações em que o ATLASCAR2 se depara com obstáculos, cria uma imagem IPM com menos píxeis e, em consequência, menos píxeis são processados na operação de interpolação. O gráfico 4.4 demonstra que o método IPM C, nesta fase, é mais eficiente do que o método IPM A, o que corrobora a afirmação anterior. O método IPM G, que recorre a uma operação de suavização, para além de possuir tempos de execução consideravelmente mais reduzidos do que a operação de interpolação os mesmos mantêm-se constantes com o aumento da resolução. A suavização mantém-se invariável com a variação de resolução uma vez que não depende da quantidade de píxeis com informação. A operação de interpolação é o que gera os maiores tempos de execução, excedendo os valores adequados para aplicações em tempo real.

É de seguida analisado a performance de cada método em relação à implementação da fase “Indexing”. Com a observação do gráfico 4.5 verifica-se de imediato que o



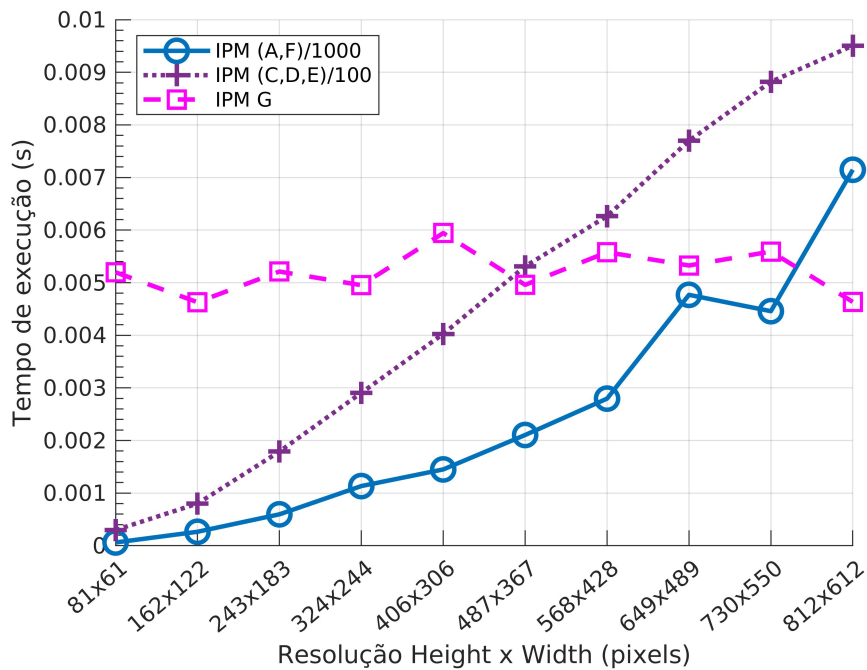


Figura 4.4: Evolução dos tempos de execução da fase “Interpolation/Smoothing” entre todos os métodos, face à variação da resolução da imagem original.

método IPM B é consideravelmente menos eficiente do que todos os outros, sendo os respetivos tempos de execução superiores em várias ordens de grandeza, em relação a todos os outros métodos. Este método em específico verifica se cada nova coordenada existe dentro de um polígono representativo da área sem obstáculos, no novo sistema de coordenadas. Este processo recorre à biblioteca “shapely” e, pela análise dos dados, verifica-se que o mesmo é computacionalmente custoso.

O método IPM A possui tempos de execução maiores que os do método IPM C, sendo isto explicado pelo facto de o primeiro calcular as novas coordenadas dos píxeis, independentemente se os mesmos representam obstáculos. É importante notar que os tempos de execução do método IPM E estão ao nível dos tempos de execução dos métodos IPM A e IPM C. O método IPM E calcula as novas coordenadas recorrendo a operações matriciais, tal como os métodos IPM D, IPM F e IPM G, pois este tipo de operações é mais eficiente do que realizar o cálculo das novas coordenadas pixel-a-pixel. No entanto, este método introduz multi-processamento por entre as operações matriciais, tornando-o mais lento, ao contrário do que se assumiu inicialmente.

Nesta fase, os tempos de execução dos métodos IPM D, IPM F e IPM G são, entre os restantes métodos, significativamente mais pequenos. O método IPM C é o que possui tempos de execução que se aproximam mais dos do método IPM D (por exemplo), mas a diferença entre estes possui várias ordens de grandeza. Para além da grande diferença entre os tempos de execução, o incremento do tempo de execução consoante o aumento da resolução da imagem é menor nos métodos IPM D, IPM F e IPM G, comparativamente ao método IPM C. Este facto foi verificado ao se realizar uma regressão linear nas duas versões mencionadas, obtendo desta forma o declive de cada grupo

de dados. Foi utilizado como valores no eixo das abcissas a percentagem referente ao quanto se diminuiu a imagem original e obteve-se as fórmulas 4.1 e 4.2, que correspondem às equações de regressão linear dos métodos IPM C e IPM D, respetivamente. Por observação do declive das fórmulas é evidente que o “crescimento” do tempo de execução é muito mais lento no método IPM D.

$$y = 0,0646 * x - 0,6426 \quad (4.1)$$

$$y = 0,0005 * x - 0,0048 \quad (4.2)$$

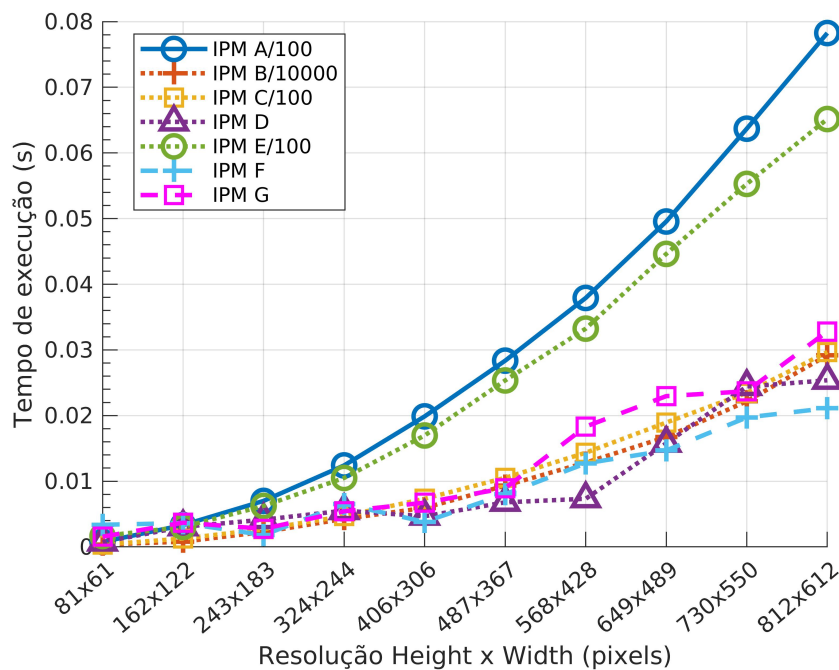


Figura 4.5: Evolução dos tempos de execução da fase “Indexing” entre todos os métodos, face à variação da resolução da imagem original.

Finalmente, a fase “Mapping”, que corresponde à associação de píxeis da imagem original na imagem IPM, é implementada com uma abordagem pixel-a-pixel nos métodos IPM A, IPM B e IPM C e implementado com base em operações vetoriais nos métodos IPM D, IPM E, IPM F e IPM G. Tal como nas fases anteriores, esta irá ter tempos de execução diferentes entre o método IPM A e IPM C, pois o primeiro não lida com píxeis representativos de obstáculos, ao invés do segundo método. O gráfico 4.6 confirma que os tempos de execução do método IPM A são superiores aos do método IPM C. Ainda referente ao mesmo gráfico, observa-se novamente que os métodos IPM D, IPM F e IPM G são consideravelmente mais eficientes, comparativamente aos restantes métodos nesta fase, devendo-se isto à utilização de operações matriciais.

Os dados utilizados para a criação do gráficos apresentados nesta subsecção estão representados em tabelas no Apêndice.

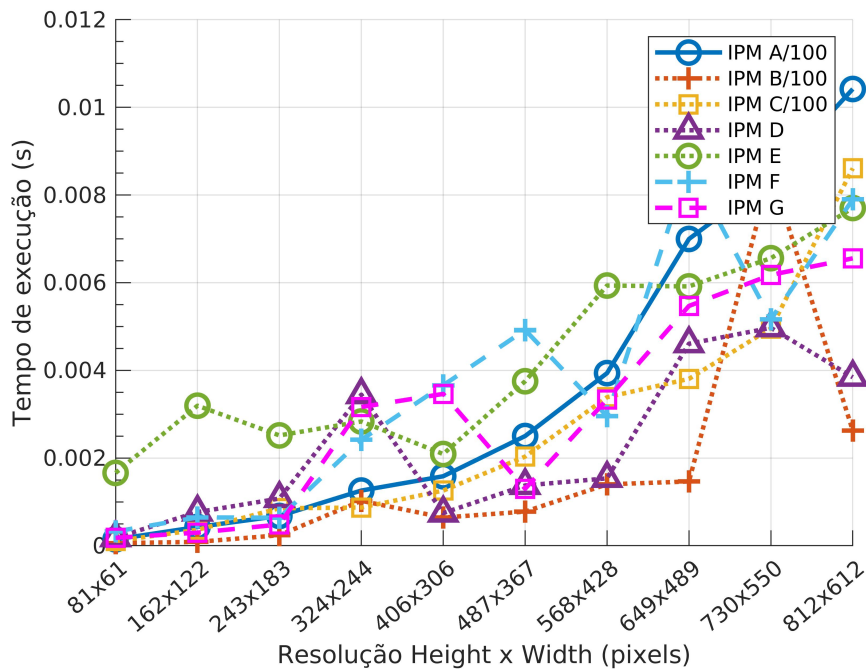


Figura 4.6: Evolução dos tempos de execução da fase “Mapping” entre todos os métodos, face à variação da resolução da imagem original.

#### 4.1.2 Variação da polígono de percepção

O polígono de percepção é essencialmente a porção de área no novo sistema de coordenadas da qual se pretende ilustrar na imagem IPM final. O aumento desta área varia proporcionalmente o número de píxeis que serão representados na imagem IPM. No entanto, o número de píxeis da imagem original, dos quais serão calculadas as novas coordenadas, não será influenciado pela variação da área do polígono de percepção.

A fase “Interpolation/Smoothing” é a única fase em que a variação da área do polígono de percepção influencia significativamente os tempos de execução da mesma, como demonstrado pelo gráfico 4.7. A diferença do custo computacional entre os métodos foi explicada na secção 4.1.1.

No caso da fase “Masking”, em teoria, o tempo de execução do método IPM F deveria aumentar consoante o aumento da área do polígono uma vez que a máscara criada refere-se à imagem IPM. No entanto, os dados apresentados pelo gráfico 4.8, possuem demasiada variabilidade (dentro de valores na mesma ordem de grandeza) para se tirar conclusões sobre o comportamento dos tempos de execução dos métodos. Apesar disso, se a variação dos tempos de execução fosse substancial consoante o aumento da área do polígono a representação da mesma, no referido gráfico, seria mais aparente.

As fases “Indexing” e “Mapping” geram tempos de execução que permanecem constantes independentemente do aumento da área do polígono de percepção, como verificado pelos gráficos 4.9 e 4.10. Isto ocorre porque o número de píxeis da imagem original, sobre os quais serão realizadas operações, não varia.

Os dados utilizados para a criação dos gráficos aqui apresentados estão representados em tabelas no Apêndice.

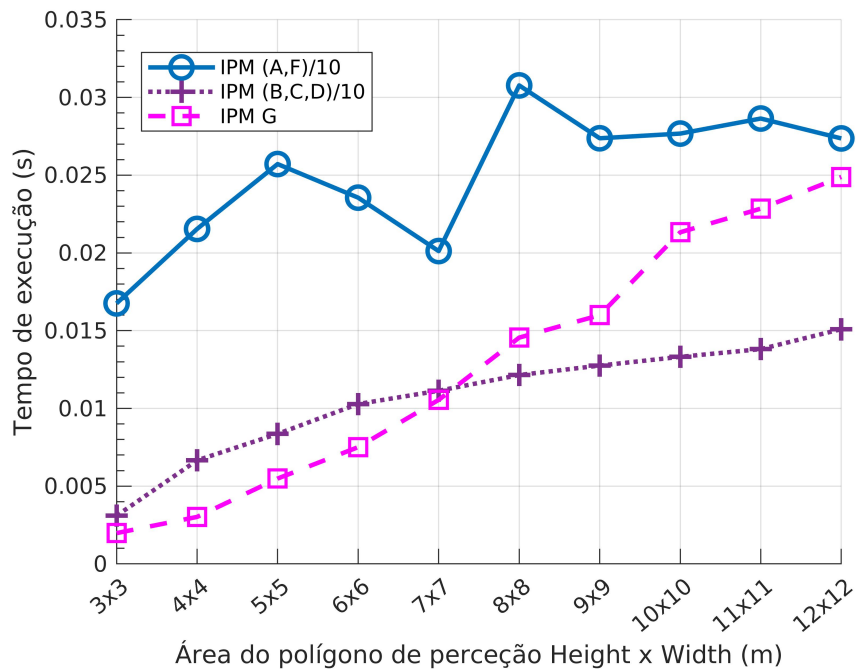


Figura 4.7: Evolução dos tempos de execução da fase “Interpolation/Smoothing” de todos os métodos, face à variação da área do polígono de percepção.

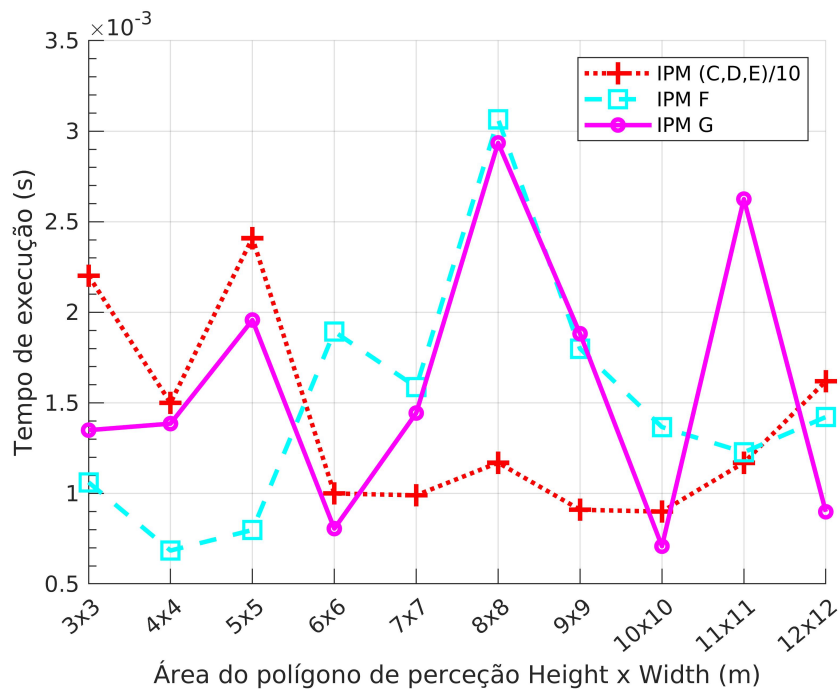


Figura 4.8: Evolução dos tempos de execução da fase “Masking” de todos os métodos, face à variação da área do polígono de percepção.

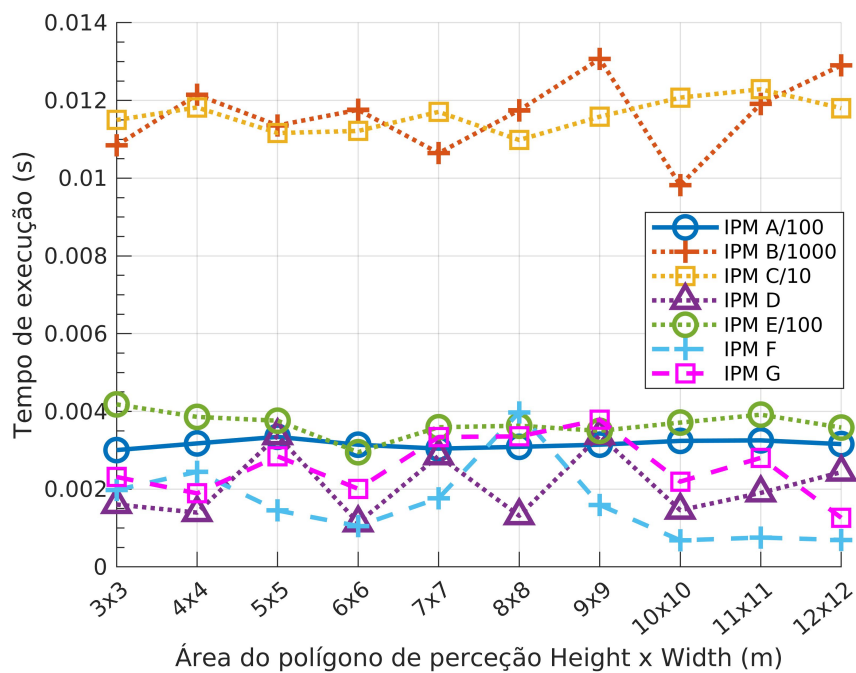


Figura 4.9: Evolução dos tempos de execução da fase “Indexing” de todos os métodos, face à variação da área do polígono de percepção.

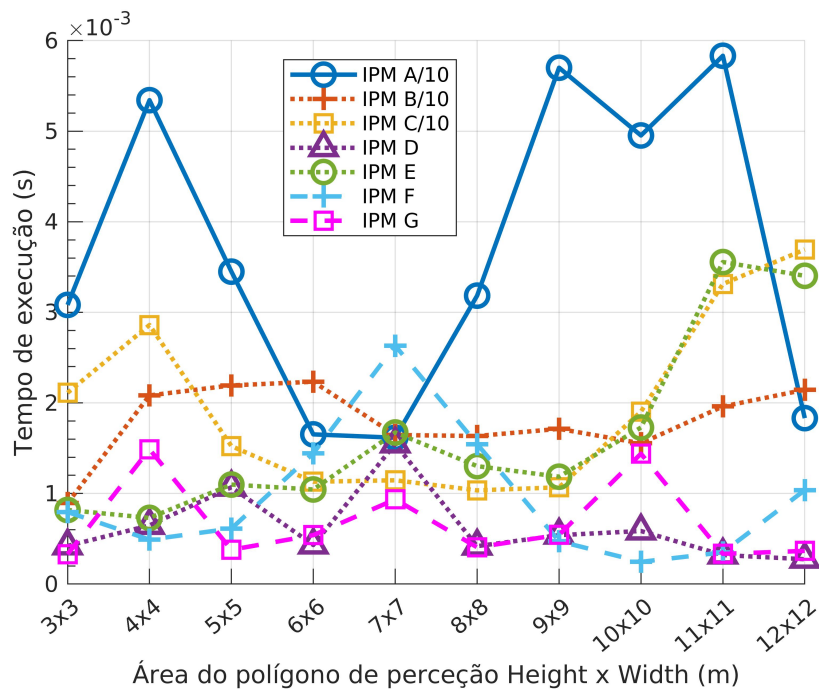


Figura 4.10: Evolução dos tempos de execução da fase “Mapping” de todos os métodos, face à variação da área do polígono de percepção

### 4.1.3 Variação de RSF

A variável RSF não interfere no número de píxeis que serão utilizados da imagem original nem no número de píxeis que serão colocados na imagem IPM. Se a variável RSF possuir um valor significativamente baixo (e.g., RSF=2%) os píxeis que serão colocados na imagem IPM poderão ser sobrepostos. No entanto, as operações responsáveis pelo posicionamento dos píxeis na nova imagem (“Indexing” e “Mapping”) apresentam o mesmo custo computacional, independentemente desta sobreposição de informação. Em virtude do mencionado, prevê-se que os tempos de execução nas fases “Indexing” e “Mapping” se mantenham invariáveis consoante o aumento do valor de RSF, e tal é verificado por observação dos gráficos 4.12, 4.13, respetivamente. No caso da fase “Masking”, os tempos de execução do método IPM F aumentam proporcionalmente com o aumento da escala RSF, ao contrário dos restantes métodos (figura 4.11). Isto ocorre porque no caso do método IPM F a máscara irá ser referente à imagem IPM, que é afetada pela escala RSF. As diferenças dos tempos de execução entre os métodos foram explicadas nas subsecções anteriores e, por isso, não se considera pertinente justificar novamente o que torna um método mais eficiente do que outro.

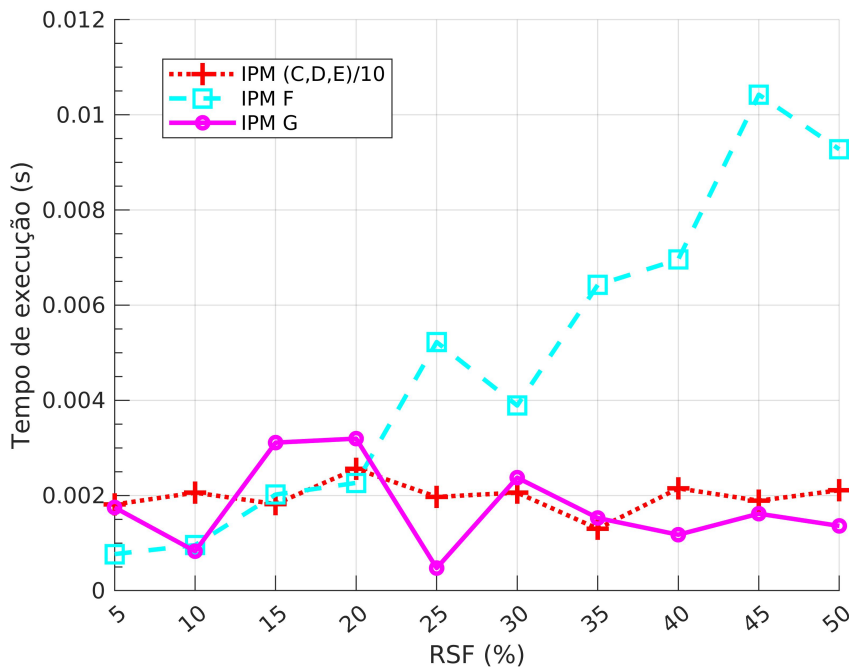


Figura 4.11: Evolução dos tempos de execução da fase “Masking” de todos os métodos, face à variação do RSF.

Os tempos de execução da fase “Interpolation” aumentam proporcionalmente com o aumento da escala RSF, como demonstrado pelo gráfico 4.14. Este comportamento ocorre porque um maior valor de RSF implica a criação de uma imagem IPM de maior resolução e, em consequência, um maior custo computacional por parte da operação de interpolação. Tal como nas secções 4.1.1 e 4.1.2, a operação de suavização terá sempre o custo computacional significativamente inferior em comparação à operação de interpolação.

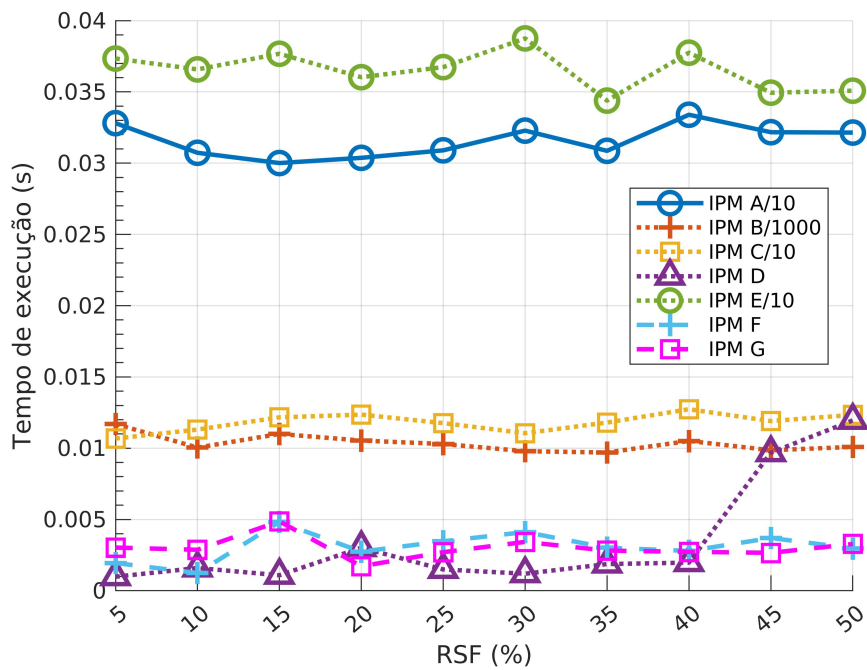


Figura 4.12: Evolução dos tempos de execução da fase “Indexing” de todos os métodos, face à variação do RSF.

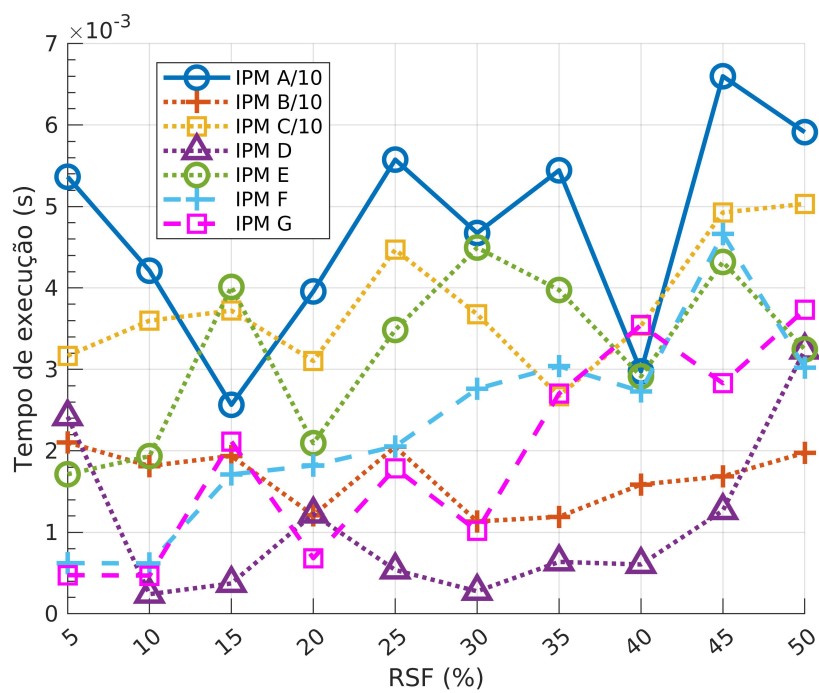


Figura 4.13: Evolução do tempo de execução da fase “Mapping” de todos os métodos, face à variação do RSF.

Os dados utilizados para a criação dos gráficos apresentados nesta subsecção estão representados em tabelas no Apêndice.

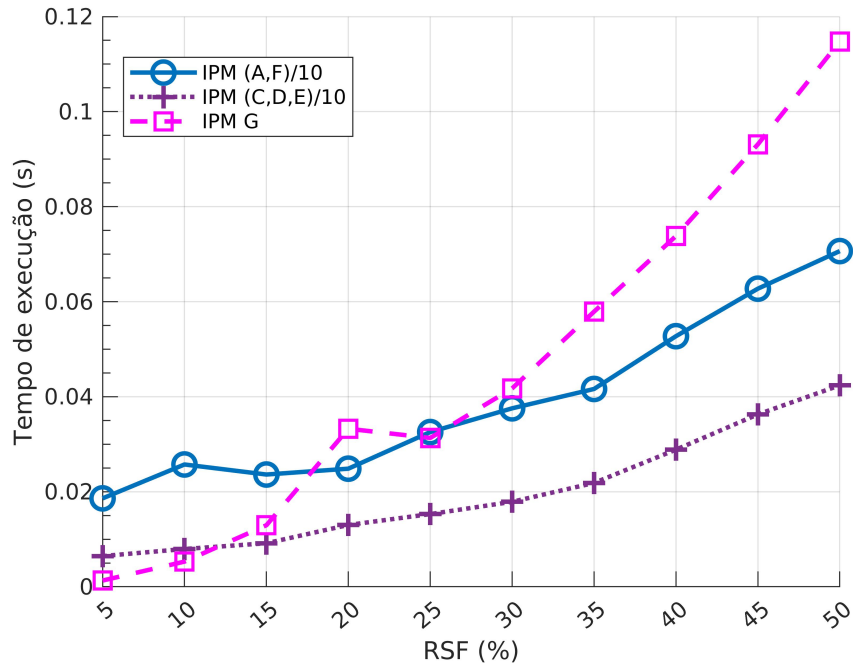


Figura 4.14: Evolução dos tempos de execução da fase “Interpolation” de todos os métodos, face à variação do RSF.

#### 4.1.4 Escolha do melhor método IPM

Com base nos dados ilustrados nas secções anteriores, o método IPM G é o que produz consistentemente os menores tempos de execução. Este método utiliza na fase “Interpolation/Smoothing” uma operação de suavização, sendo o custo computacional associado a esta fase muito pequeno em comparação com a operação de interpolação. O método IPM F utiliza uma operação de interpolação tornando-o menos eficiente em comparação com o método IPM G. A operação de interpolação é o maior responsável pelos grandes tempos de execução em quase todos os métodos desenvolvidos que a utilizam. Para além da fase “Interpolation/Smoothing”, a fase “Masking” nos métodos IPM C, IPM D e IPM E gera tempos de execução superiores em várias ordens de grandeza em comparação com as fases “Indexing” e “Mapping”. As variações da fase “Masking” que apresentam os menores tempos de execução foram aplicadas nos métodos IPM F e IPM G. O gráfico 4.15 confirma as afirmações anteriores, no entanto devido à diminuição das escalas em grande parte dos métodos é difícil comparar os tempos de execução de algumas fases entre os métodos. De modo a resolver esta problemática disponibilizou-se no Apêndice a tabela 5.23 que apresenta os valores utilizados no gráfico 4.15.



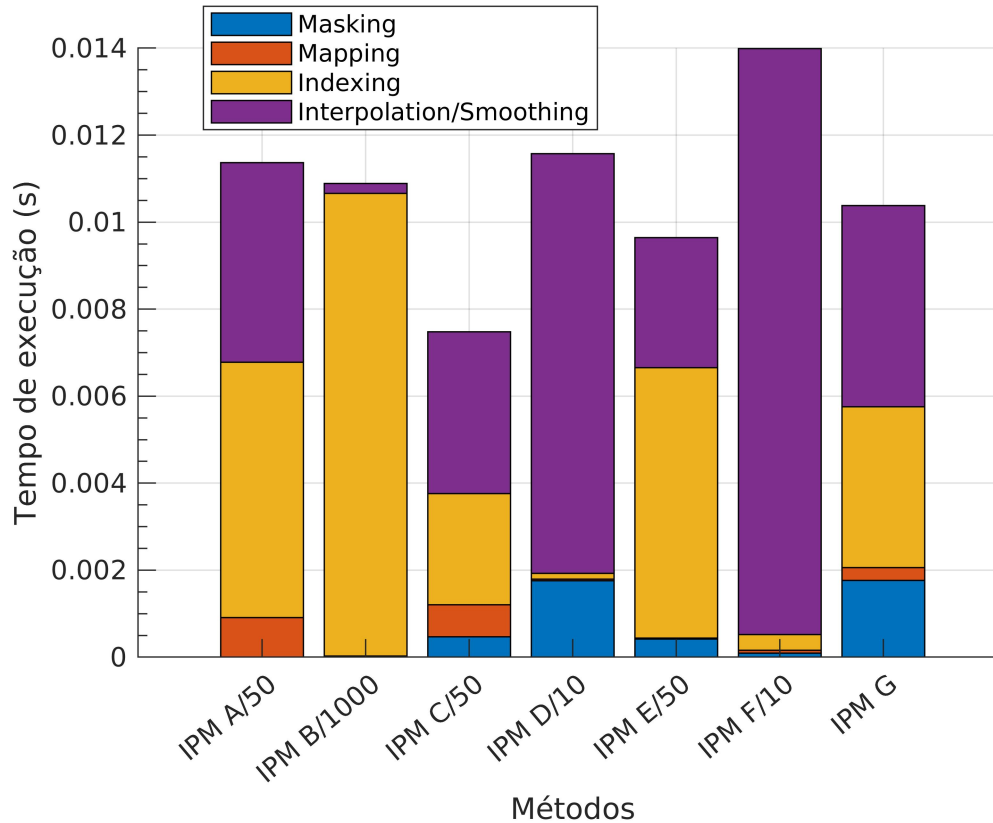


Figura 4.15: Tempos de execução de cada fase de cada método IPM desenvolvido.

## 4.2 Geolocalização

Esta secção foca-se na análise de algoritmos de localização desenvolvidos ao longo desta investigação. Nas duas primeiras subsecções a análise será focada principalmente num ambiente simulado. Pretende-se testar os algoritmos desenvolvidos e validar os resultados dos mesmos face às várias combinações de parâmetros.

Tipicamente, o *template* “desliza” pela imagem alvo pixel-a-pixel, atribuindo a cada um destes o valor proveniente das operações inerentes ao método. Alternativamente, o *template* poderá deslizar pela imagem alvo com passos superiores a um pixel, sendo este parâmetro denominado, daqui em diante, por “passo de píxeis” (PP). De forma similar, cada algoritmo terá que ser executado para cada variação do ângulo do *template*, sendo este parâmetro denominado, daqui em diante, por “passo de ângulo” (PA). O aumento do PP diminui o tempo de execução do algoritmo para cada ângulo e o aumento do PA diminui o tempo de execução do algoritmo quando este executa um conjunto de ângulos.

No seguimento desta secção a validação dos algoritmos será, como já foi referido, realizada num ambiente simulado, no entanto pretende-se, ainda assim, testar estes algoritmos numa grande variedade de situações. Para esse efeito utilizou-se vinte imagens de satélite de diferentes regiões e os *templates*, extraídos destes, ilustram ou não, marcas salientes (e.g. linha tracejada da via, passadeira). A figura 4.16 exemplifica os

tipos de imagens que os *templates* podem representar. Alguns algoritmos, nomeadamente o CTM, processam o *template* a cada ângulo a elevados tempos de execução. Por esta razão, de maneira a se obter resultados pertinentes e rápidos, todos os algoritmos analisam apenas um conjunto de cem ângulos, centrado no ângulo *ground truth*.

Nas seguintes subsecções, os valores referentes à distância euclidiana e correlação, representados nos gráficos, provêm da média destes valores em todo o conjunto de imagens em cada ângulo de rotação do *template*, à exceção de quando indicado o contrário.

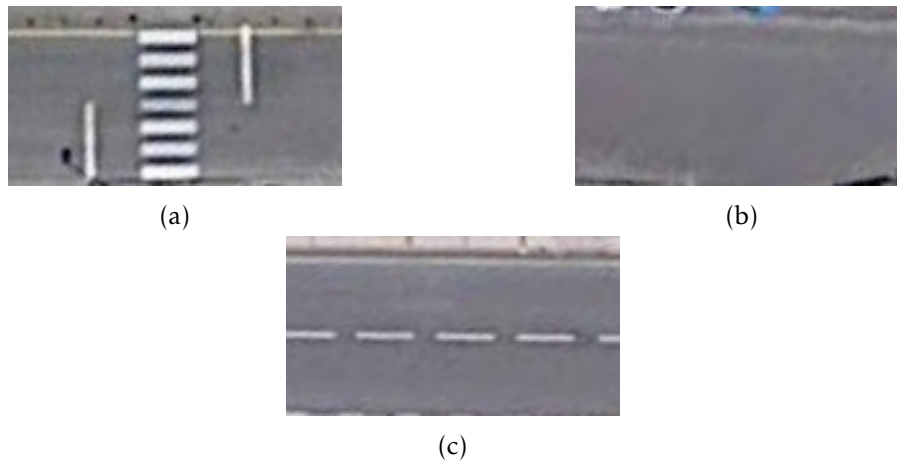


Figura 4.16: Tipos de *templates* utilizados na validação dos algoritmos de localização. a) *Template* com a marcação de uma passadeira; b) *Template* com nenhuma marcação rodoviária; c) *Template* com uma linha tracejada da via.

No caso da secção 4.2.1, o objetivo principal é a validação e a comparação dos resultados obtidos pelos algoritmos CTM e GTM, face à variação dos parâmetros PP e PA. A secção 4.2.2 foca-se principalmente na validação dos algoritmos CCM e GCM. Na secção 4.2.3 realiza-se uma análise sobre os algoritmos desenvolvidos em situações reais, i.e., o *template* é gerado pelo ATLASCAR e a imagem alvo será uma imagem de satélite sobre a porção de estrada, de onde o *template* foi adquirido. Os dados utilizados, à exceção da imagem de satélite, são provenientes do *bagfile* “Target2” quando o ATLASCAR é conduzido pelo bairro de Santiago, em Aveiro. Foram utilizados quatro imagens IPM, das quais uma destas não possui qualquer marcação rodoviária. Pretende-se principalmente verificar se os algoritmos são igualmente válidos em condições simuladas e reais.

#### 4.2.1 Classic Template Matching e Gaussian Template Matching

Esta subsecção visa a identificar, entre os métodos CTM e GTM, qual destes produzem resultados mais precisos. O gráfico 4.17 apresenta as distâncias euclidianas obtidas destes métodos com os parâmetros PP e PA com o maior potencial, em teoria, de retornarem posições mais precisas (PP=1 e PA=1). Por observação deste gráfico é evidente que GTM é o algoritmo que retorna, de um modo geral, as distâncias euclidianas mais reduzidas. O algoritmo CTM apenas retorna posições com distâncias euclidianas inferiores a dez píxeis quando o ângulo a que o *template* é rodado praticamente coincide com o ângulo *ground truth*. Por outro lado GTM obtém distâncias euclidianas inferiores

a dez píxeis quando a rotação do *template* difere até dez graus (aproximadamente) do ângulo *ground truth*. Dito isto, é evidente que o GTM é o que resulta em posições mais precisas, com os parâmetros indicados.

Para além do algoritmo GTM retornar as distâncias euclidianas mais reduzidas, em geral, comparativamente ao algoritmo CTM este apresenta também um menor desvio padrão das distâncias euclidianas, como ilustrado pela figura 4.18. Por outras palavras, os valores das distâncias euclidianas, retornados pelo algoritmo GTM variam menos do que os retornados pelo algoritmo CTM.

Em termos de valores de correlação obtidos para cada desfazamento com o ângulo *ground truth*, ambos os algoritmos possuem uma distribuição de valores considerada adequada. Isto porque em ambos os algoritmos o valor de correlação é mais elevado quando o desfazamento com o ângulo *ground truth* é zero e esse valor diminui proporcionalmente com o aumento deste desfazamento (figura 4.19). Este comportamento implica que independentemente do valor do parâmetro PA o ângulo do *template* que é escolhido como o mais adequado entre todos os outros será sempre o mais próximo do ângulo *ground truth*.

Ao se analisar os resultados provenientes dos algoritmos CTM e GTM com os melhores parâmetros de PP e PA, considera-se pertinente verificar como estes algoritmos se comportam face a maiores valores destes. O gráfico 4.20 ilustra as distâncias euclidianas de cada algoritmo face a parâmetros de PP e PA com valores, respetivamente, de cinco e quatro. Igualmente como foi ilustrado anteriormente, o algoritmo GTM continua a retornar as menores distâncias euclidianas. Nestas condições o algoritmo CTM não possui, independentemente do ângulo, nenhuma distância euclidiana inferior a dez píxeis, sendo este muito impreciso e inadequado para ser utilizado como algoritmo de localização, com estes parâmetros.

Como se esperava, por observação da figura 4.21, o algoritmo GTM demonstra novamente menores valores de desvio padrão das distâncias euclidianas, comparativamente ao algoritmo CTM, com PP=5 e PA=4. Este comportamento do desvio padrão implica que o algoritmo GTM retorna, como anteriormente, resultados mais consistentes.

No caso dos valores de correlação, com estes parâmetros de PP e PA, os algoritmos em questão continuam-se a comportar de um modo similar, i.e., o maior valor de correlação estará, nos dois algoritmos, mais próximo do ângulo *ground truth* (figura 4.22).

Por observação dos gráficos mencionados, conclui-se que entre os algoritmos GTM e CTM o primeiro é, independentemente dos parâmetros PP e PA, o que retorna as posições do *template* mais próximas da posição *ground truth*. Tal facto é ainda mais evidenciado pela observação das figuras 4.23, 4.24, 4.25 e 4.26. Nestas figuras é ilustrado o comportamento de ambos os algoritmos em relação às distâncias euclidianas e o desvio padrão das mesmas, consoante a variação dos parâmetros PP e PA.

Os gráficos mencionados ilustram, em ambos os algoritmos, que não existe variação significativa nas distâncias euclidianas e respetivos desvios padrões para grandes desfazamentos com o ângulo *ground truth*. No entanto, para pequenos desfazamentos com o ângulo *ground truth* é mais perceptível a variação destas distâncias e respetivos desvios padrões, como demonstrado pelos gráficos 4.27, 4.28 e 4.29.

Em toda a análise de dados realizada até ao momento foi considerado um desfazamento de apenas 50 graus do ângulo *ground truth*. Esta restrição é necessária devido ao grande custo computacional que os algoritmos podem acarretar. No entanto, esta restrição impede a análise do comportamento de cada algoritmo em todos os 360 graus de

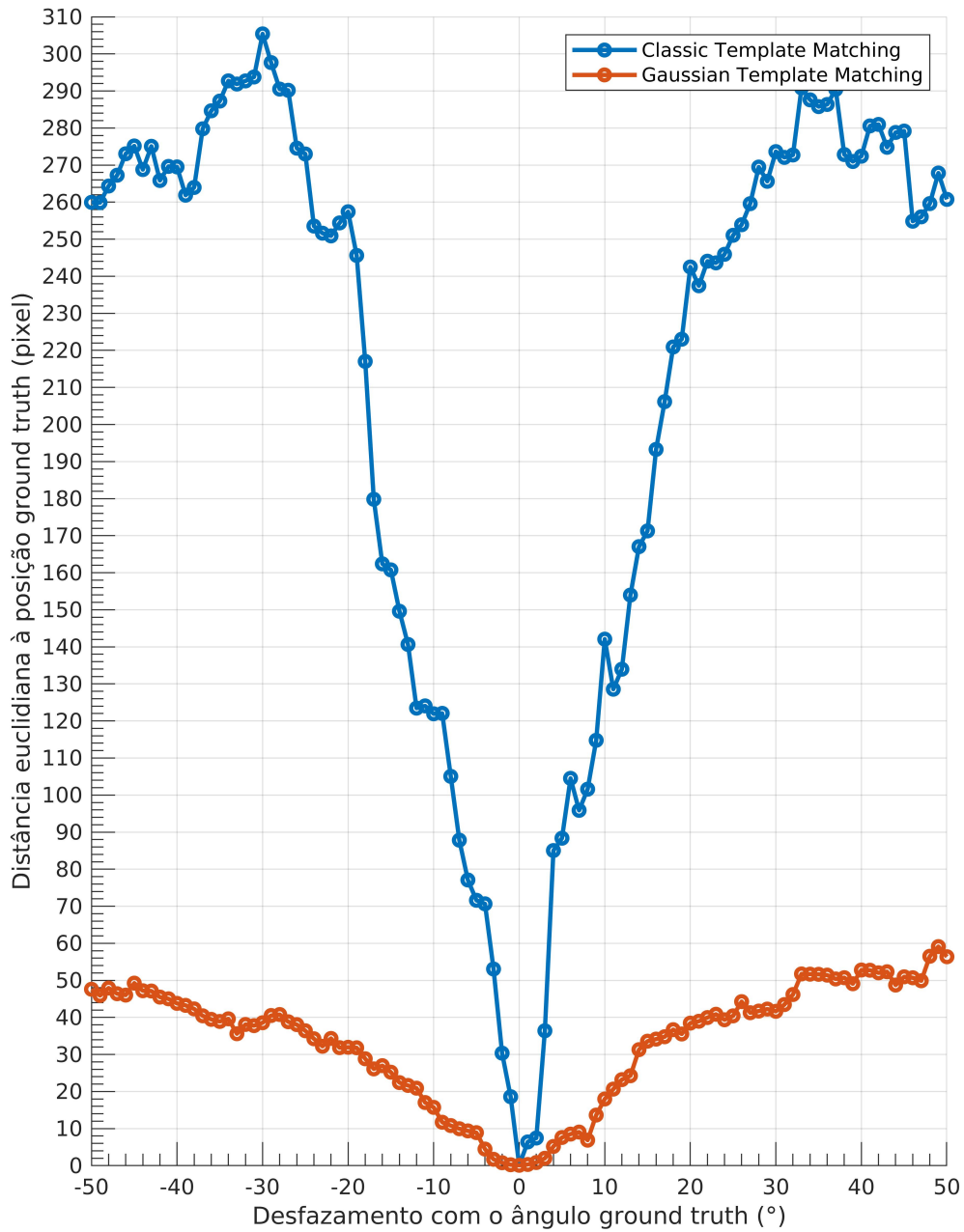


Figura 4.17: Distâncias euclidianas entre a posição calculada pelos algoritmos CTM e GTM e a posição *ground truth*, face à variação do ângulo a que o *template* é rodado. Com os parâmetros PP=1 e PA=1.

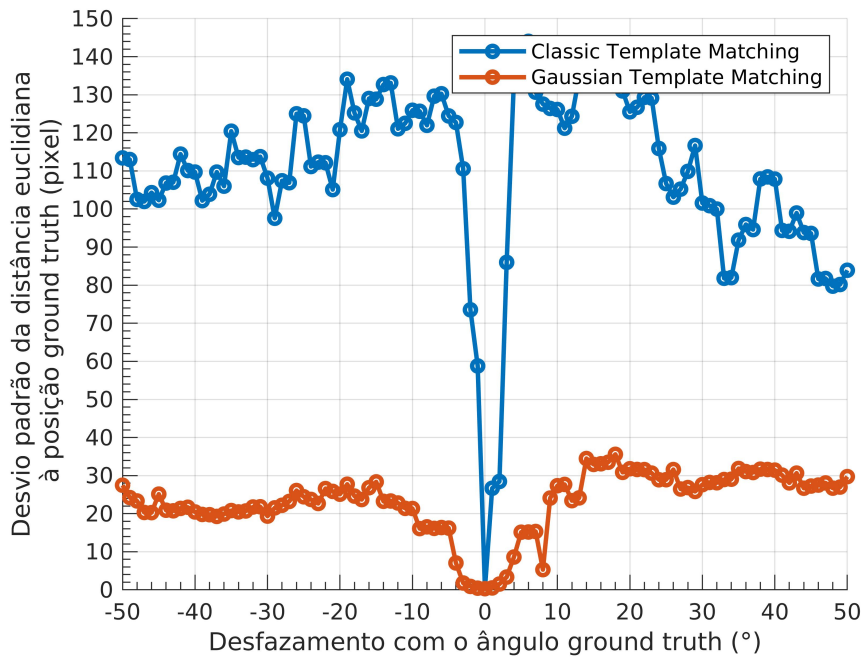


Figura 4.18: Desvio padrão das distâncias euclidianas entre a posição calculada pelos algoritmos CTM e GTM e a posição *ground truth*, face à variação do ângulo a que o *template* é rodado. Com os parâmetros PP=1 e PA=1.

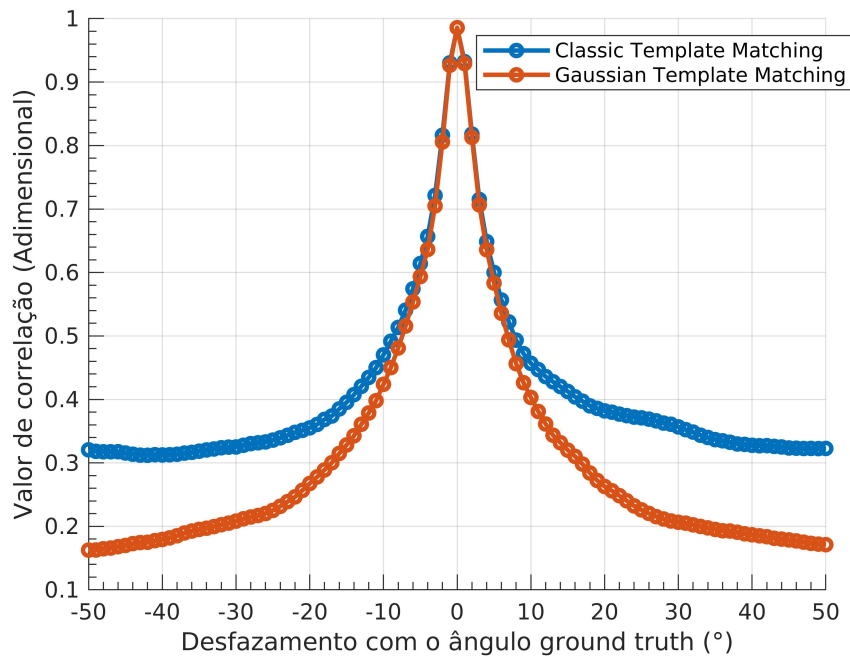


Figura 4.19: Valores de correlação de CTM e GTM, face à variação do ângulo a que o *template* é rodado. Com os parâmetros PP=1 e PA=1.

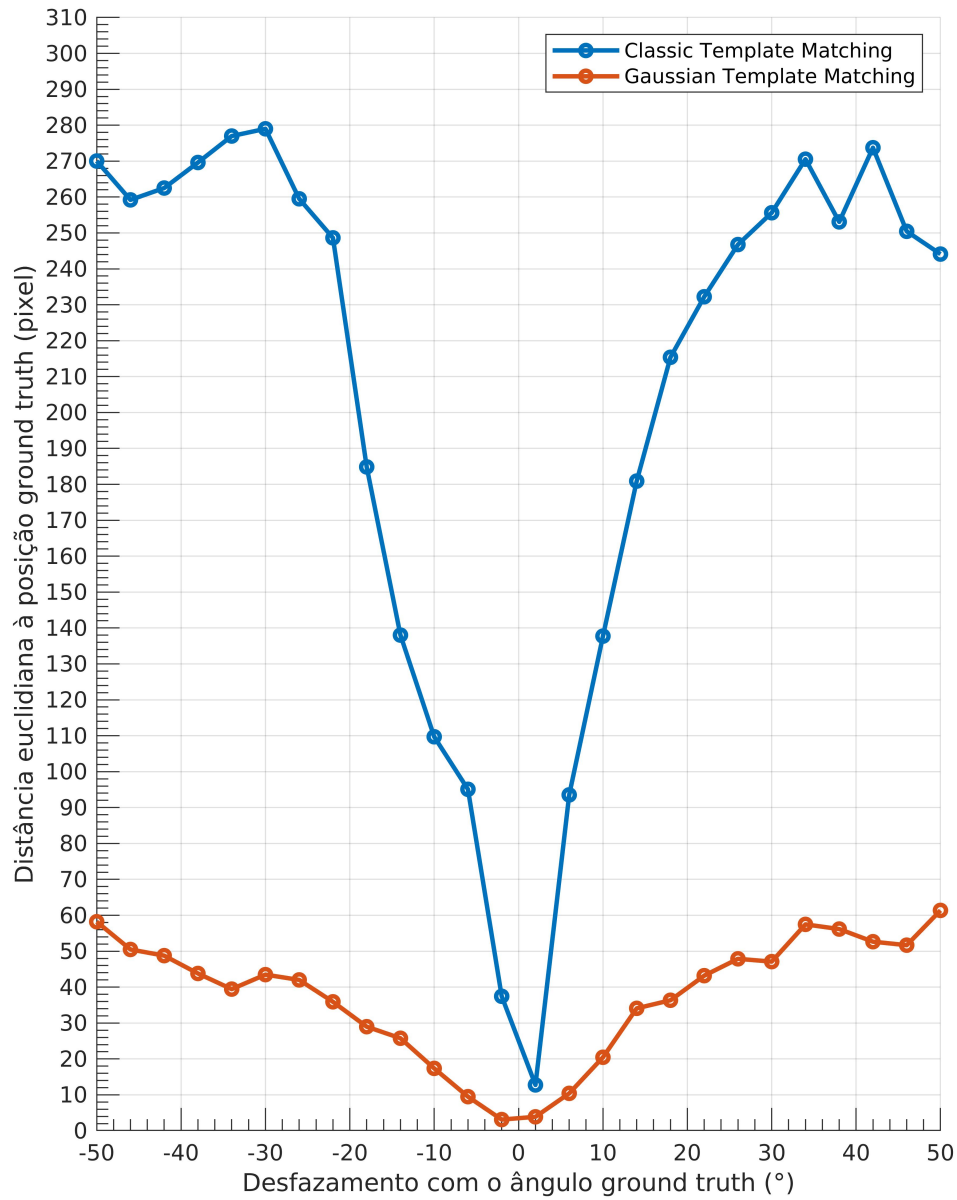


Figura 4.20: Distância euclidiana entre a posição calculada pelos algoritmos CTM e GTM e a posição *ground truth*, face à variação do ângulo a que o *template* é rodado. Com os parâmetros PP=5 e PA=4.

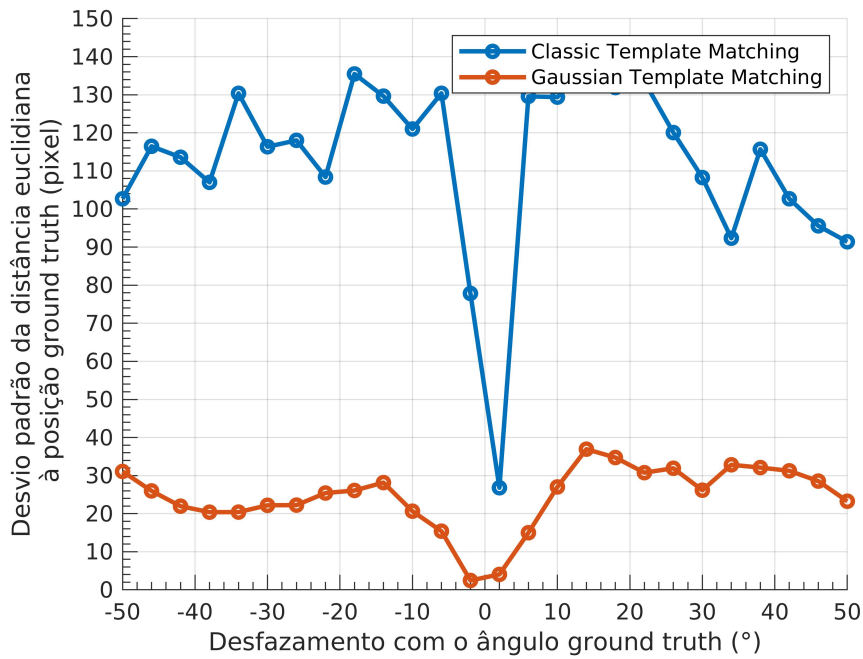


Figura 4.21: Desvio padrão das distâncias euclidianas entre a posição calculada pelos algoritmos CTM e GTM e a posição *ground truth*, face à variação do ângulo a que o *template* é rodado. Com os parâmetros PP=5 e PA=4.

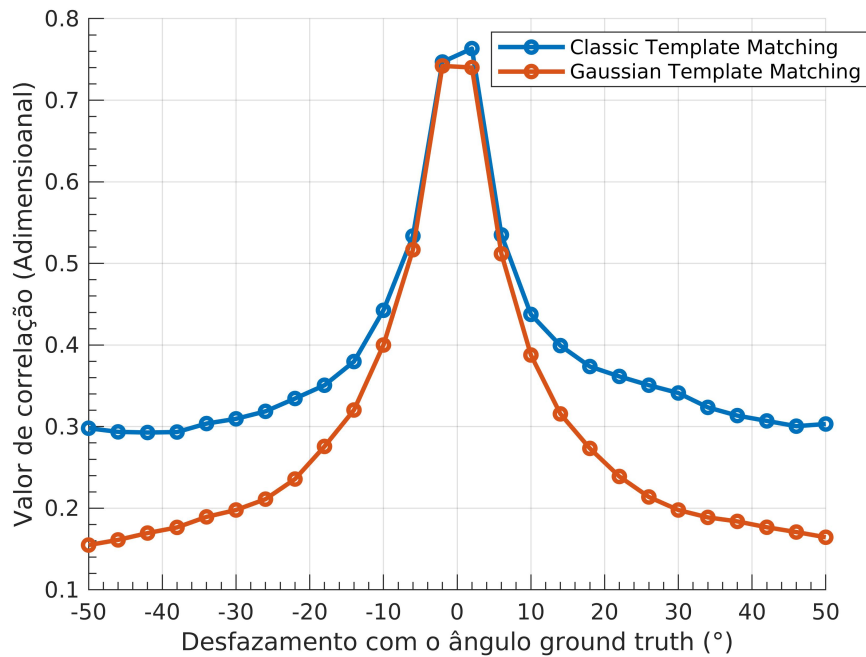


Figura 4.22: Valores de correlação de CTM e GTM, face à variação do ângulo a que o *template* é rodado. Com os parâmetros PP=5 e PA=4.

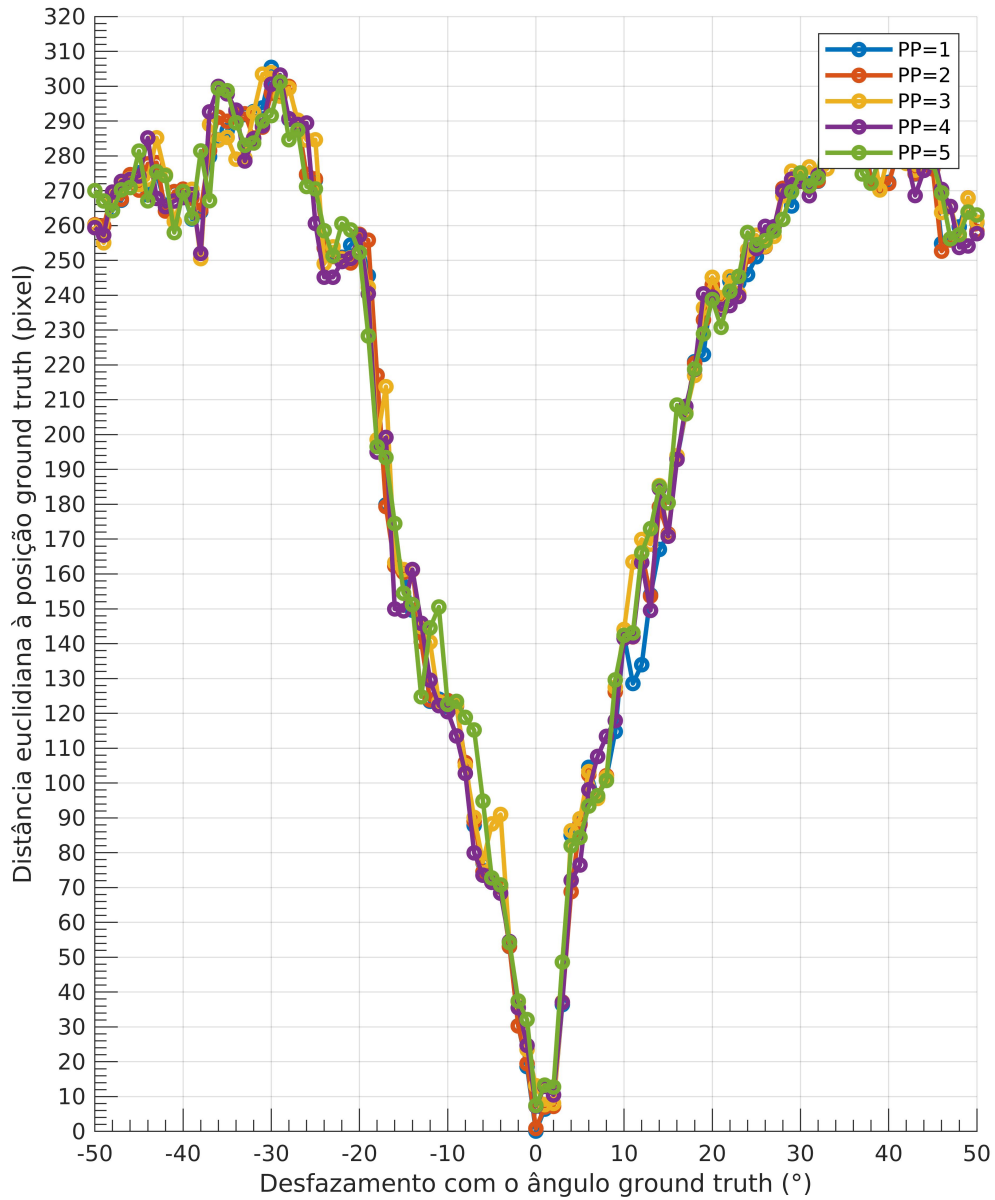


Figura 4.23: Distâncias euclidianas do algoritmo CTM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado.



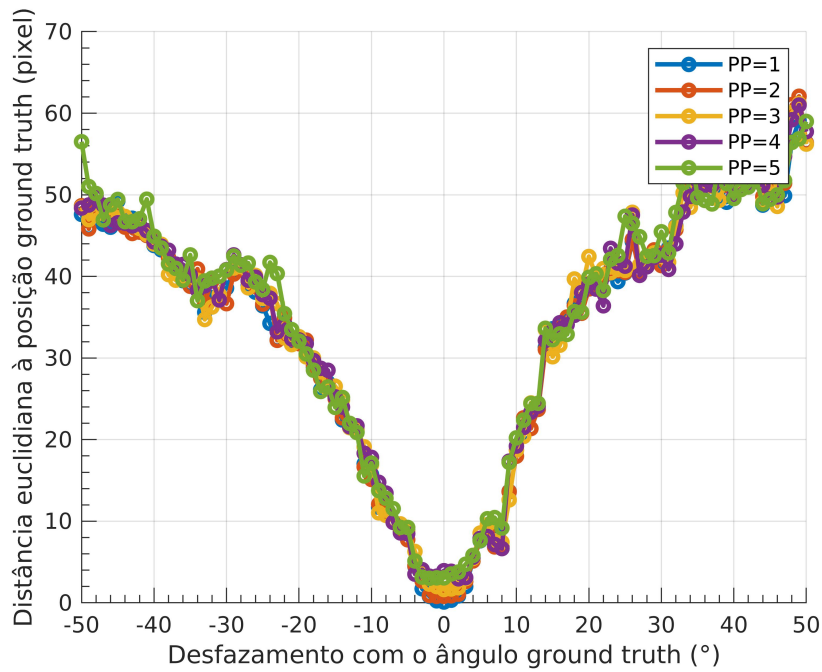


Figura 4.24: Distâncias euclidianas do algoritmo GTM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado.

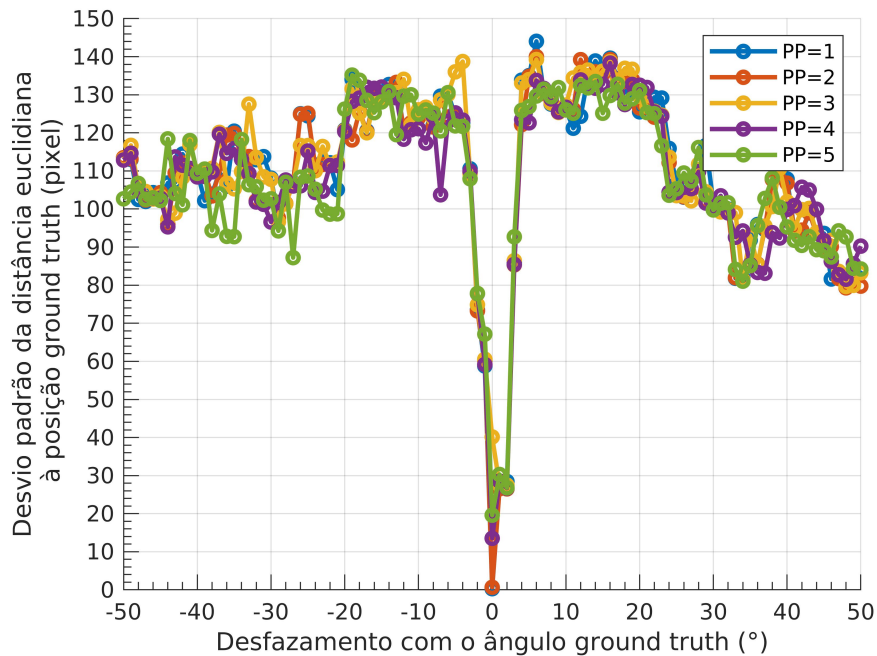


Figura 4.25: Desvio padrão das distâncias euclidianas do algoritmo CTM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado.

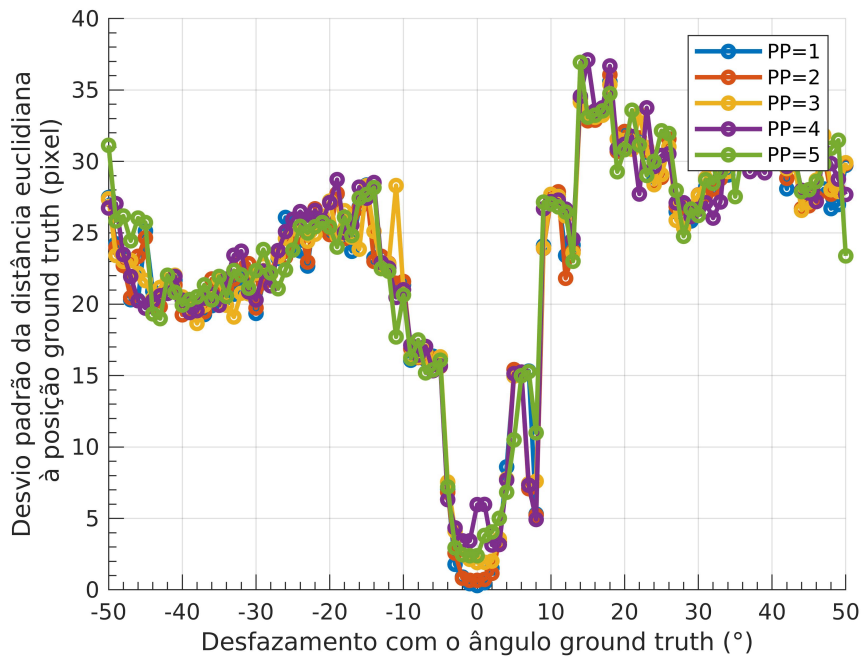


Figura 4.26: Desvio padrão das distâncias euclidianas do algoritmo GTM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado.

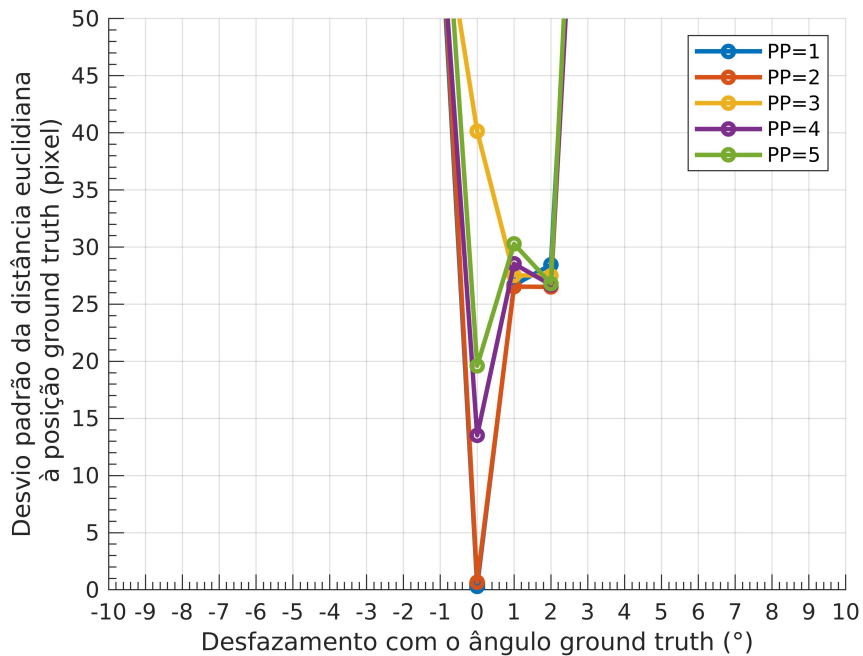


Figura 4.27: Desvio padrão do algoritmo CTM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado (Zoom sobre a figura 4.24).

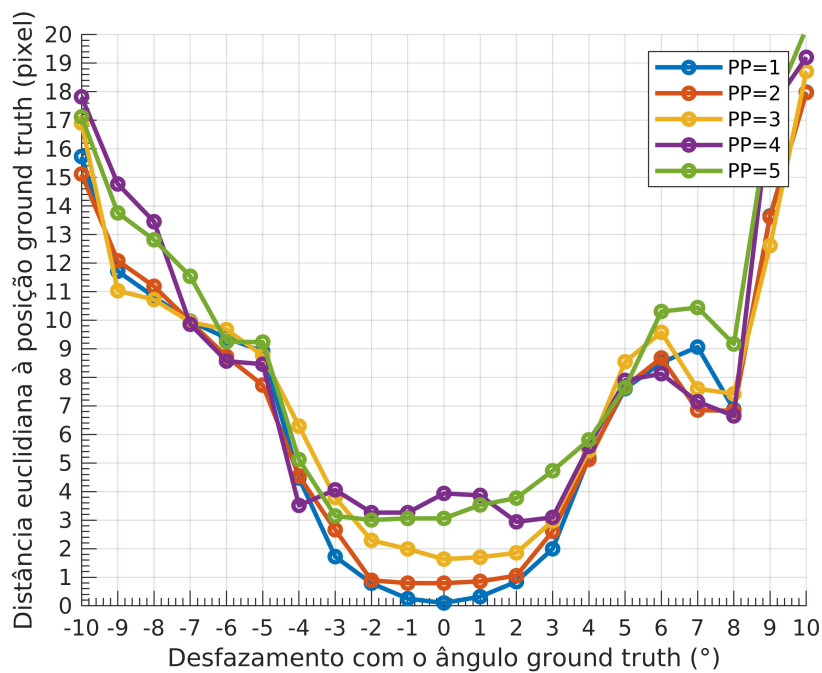


Figura 4.28: Desvio padrão do algoritmo GTM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado (Zoom sobre a figura 4.25).

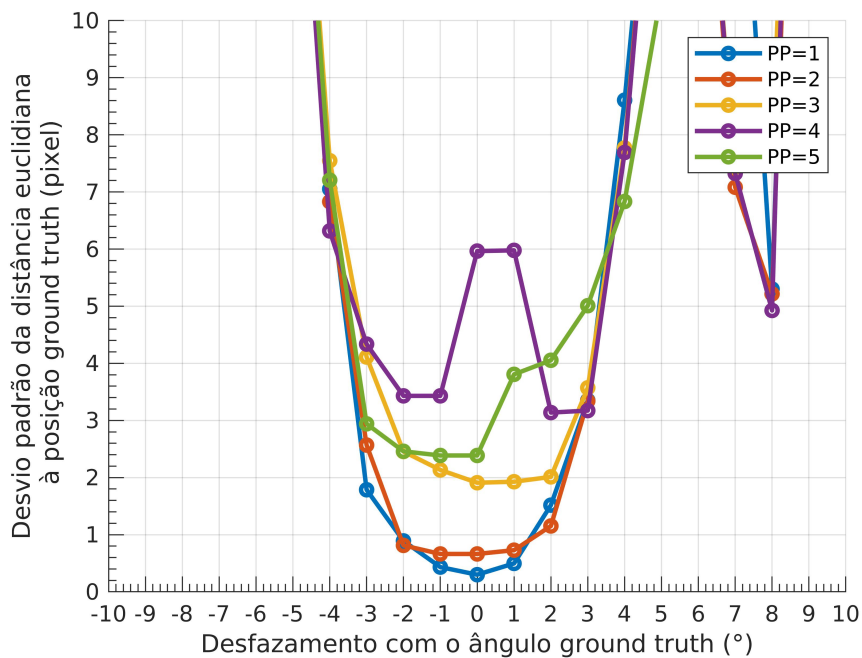


Figura 4.29: Distâncias euclidianas do algoritmo GTM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado (Zoom sobre a figura 4.26).

rotação do *template*, podendo existir comportamentos imprevistos. A título de exemplo, se o *template* apresentar bastante simetria existe a possibilidade da ocorrência de um falso positivo quando o *template* é rodado 180 graus do ângulo *ground truth*. O aumento do parâmetro PP poderá aumentar a probabilidade de ocorrerem estes falsos positivos. A fim de provar que a simetria do *template* pode criar falsos positivos executou-se os algoritmos considerando um conjunto de ângulos centrados em 180 graus de desfazamento do ângulo *ground truth*. Considerou-se os parâmetros PP=1 e PP=5 para os algoritmos CTM e GTM, e obteve-se os gráficos 4.30, 4.31, 4.32 e 4.33. Ao contrário do que se assumiu inicialmente, a ocorrência de falsos positivos aparenta ser um evento improvável, uma vez que em todos os gráficos os ângulos mais próximos do ângulo *ground truth* ilustram consistentemente maiores valores de correlação.

De modo a finalizar a análise relativa à comparação da performance entre os algoritmos CTM e GTM, considerou-se pertinente verificar quais as distâncias euclidianas e desvios padrões finais retornados destes algoritmos após se analisar um conjunto de ângulos. Calcula-se a média destes valores para o conjunto de vinte imagens variando os parâmetros PP e PA. Construiu-se, com os dados mencionados, os gráficos 4.34 e 4.35. Confirma-se novamente que o algoritmo GTM apresenta as menores distâncias euclidianas e menores variações das mesmas.

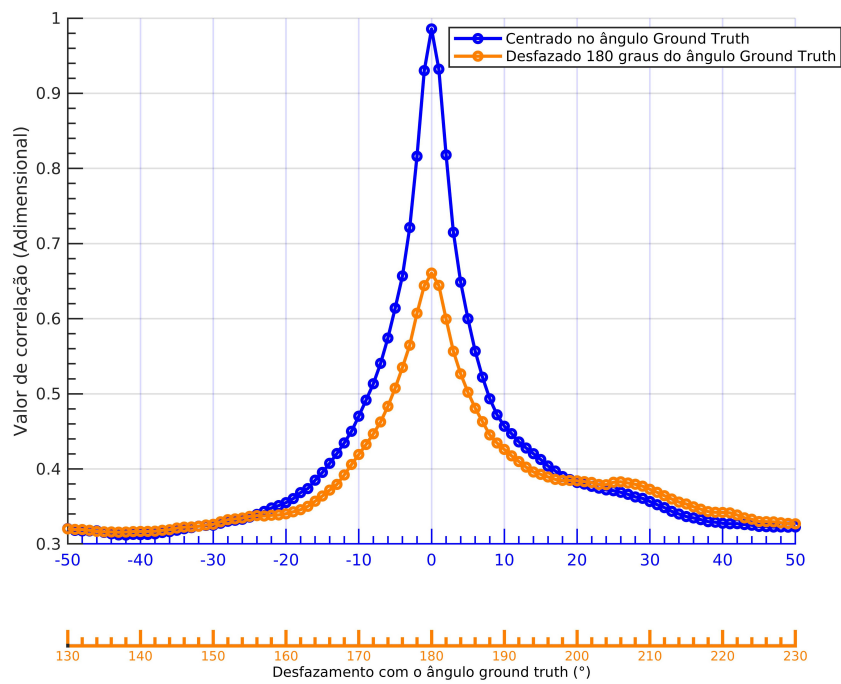


Figura 4.30: Comparação dos valores de correlação para valores de ângulos centrado ou desfasado 180 graus do ângulo *ground truth* (Algoritmo CTM e PP=1).

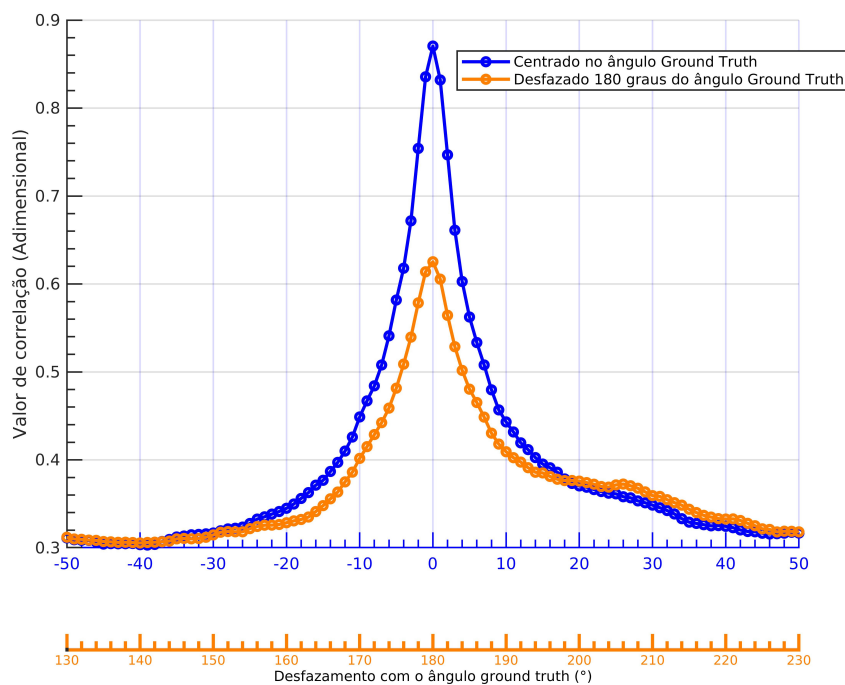


Figura 4.31: Comparação dos valores de correlação para valores de ângulos centrado ou desfasado 180 graus do ângulo *ground truth* (Algoritmo CTM e PP=5).

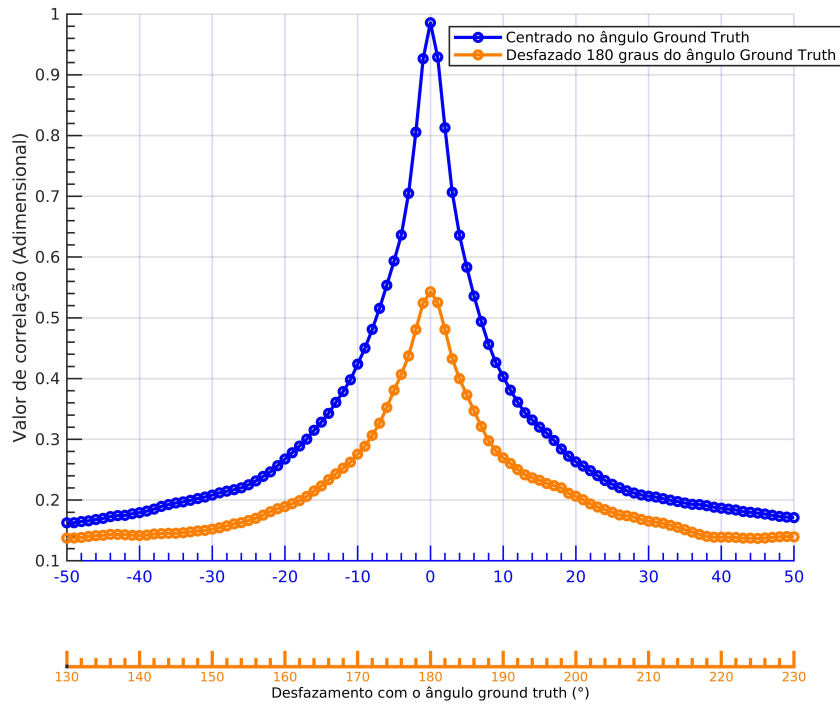


Figura 4.32: Comparação dos valores de correlação para valores de ângulos centrado ou desfasado 180 graus do ângulo *ground truth* (Algoritmo GTM e PP=1).

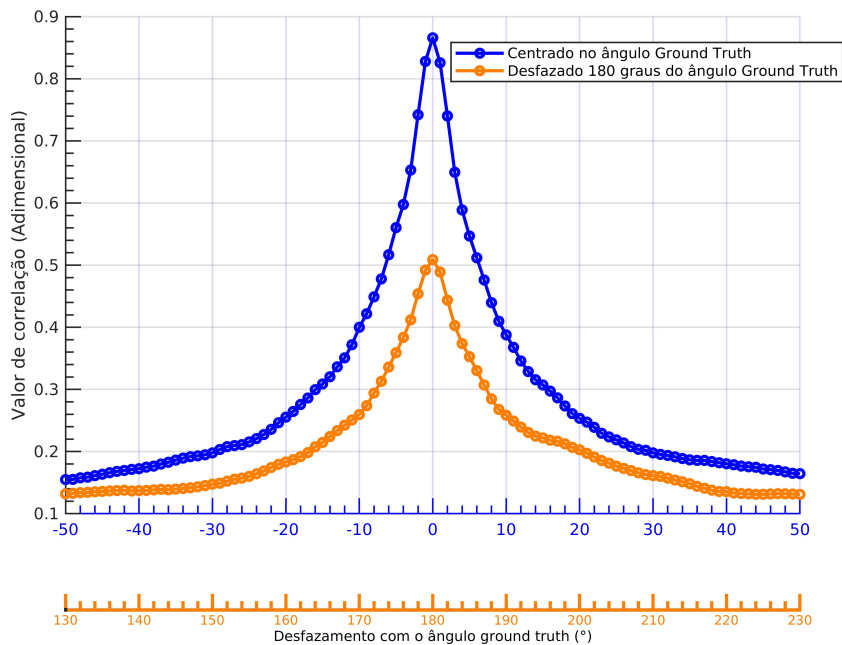


Figura 4.33: Comparação dos valores de correlação para valores de ângulos centrado ou desfasado 180 graus do ângulo *ground truth* (Algoritmo GTM e PP=5).

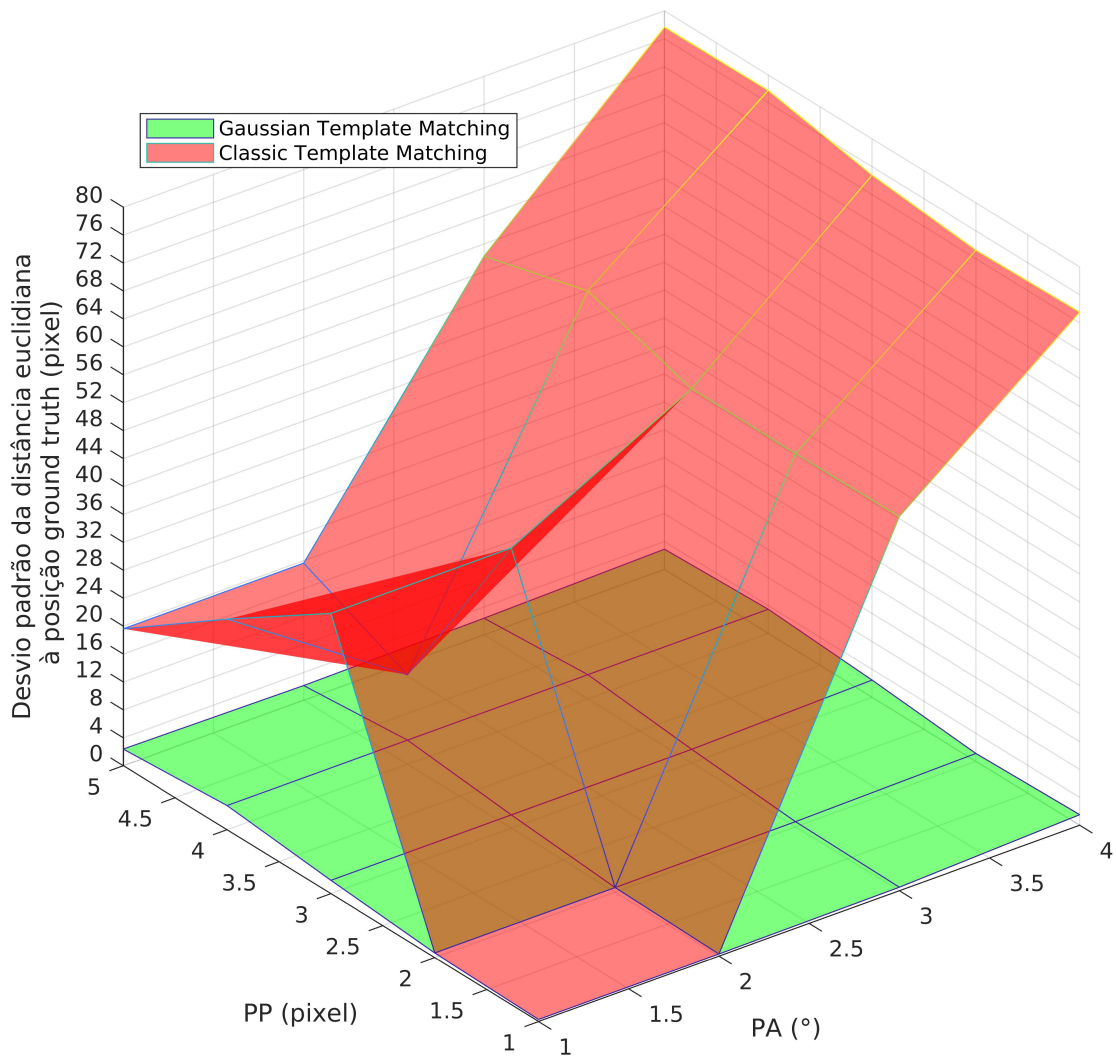


Figura 4.34: Distâncias euclidianas retornadas dos algoritmos GTM e CTM, face à variação dos parâmetros PP e PA

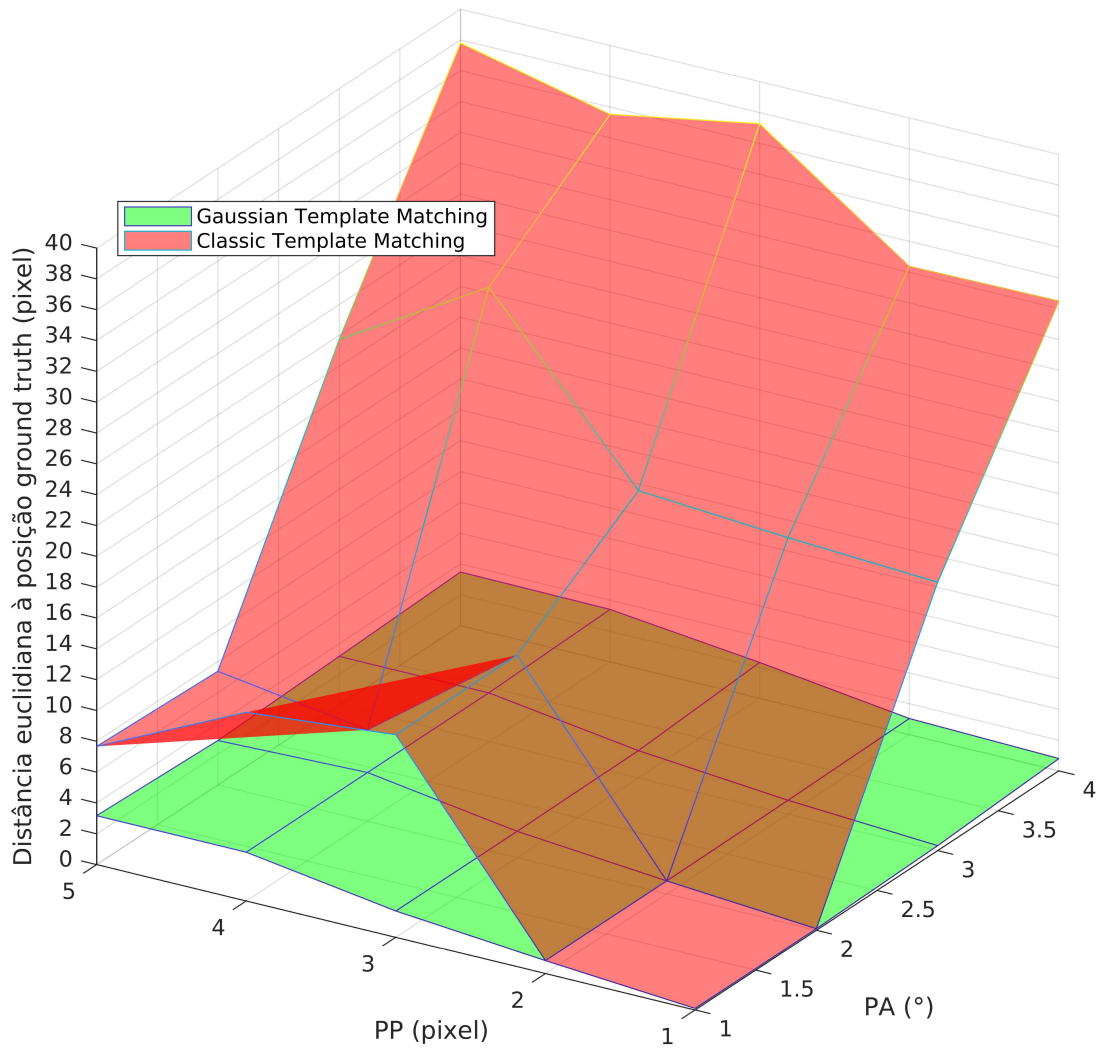


Figura 4.35: Desvio padrão das distâncias euclidianas retornadas dos algoritmos GTM e CTM, face à variação dos parâmetros PP e PA



### 4.2.2 Classic Chamfer Matching e Gaussian Chamfer Matching

O algoritmo GTM aparenta ser adequado para ser utilizado como algoritmo de geolocalização, uma vez que este apresenta consistentemente posições próximas do *ground truth*. No entanto, acarreta grandes custos computacionais, mesmo com os maiores valores dos parâmetros PP e PA. Tendo isto em conta houve necessidade de desenvolver os algoritmos CCM e GCM, que, em teoria, são mais robustos e com menor custo computacional associado do que o *Template Matching*.

Por observação dos gráficos 4.36 e 4.37, tanto o algoritmo CCM como o GCM, apesar de possuírem um custo computacional reduzido (comparativamente aos algoritmos da subsecção anterior) não aparentam ser adequados para aplicações relacionadas com localização. Isto porque as distâncias euclidianas são demasiado grandes, mesmo quando o ângulo do *template* coincide com o ângulo *ground truth*. A implementação de uma distribuição gaussiana (GCM) reduziu a distância euclidiana, mas não o suficiente para se considerar aplicável para localização de porções de estrada.

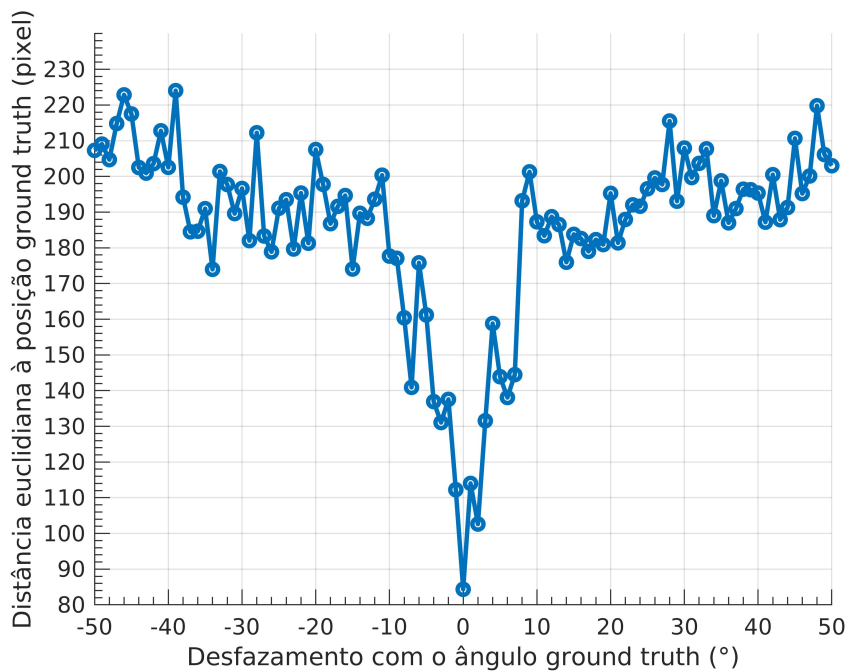


Figura 4.36: Distâncias euclidianas do algoritmo CCM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado

Existem vários tipos de circunstâncias responsáveis por estes resultados erróneos. A título de exemplo, em alguns casos, os algoritmos de deteção de arestas quando recebem o *template* como *input* retornam uma imagem a preto, isto porque, nestes casos, o *template* poderá não possuir características salientes o suficiente para serem detetadas pelo algoritmo (figura 4.38 (b)). Outro caso típico de ocorrer advém quando o algoritmo de deteção de arestas retorna representações de arestas, das quais os algoritmos CCM e GCM realizam correspondências noutras posições para além da posição *ground truth* (figura 4.38 (c)). Isto ocorre porque tipicamente os *templates* a serem analisados ilustram porções de estrada com marcações simples (e.g. marcas delimitadoras da via,

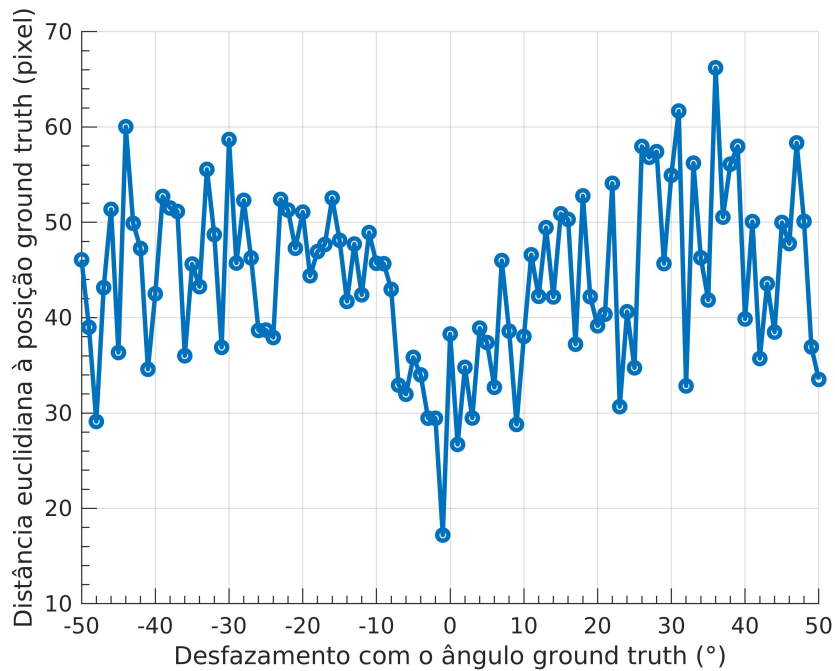


Figura 4.37: Distâncias euclidianas do algoritmo GCM, face à variação do parâmetro PP e do ângulo a que o *template* é rodado.

linhas tracejadas da via) e consequentemente as representações das arestas serão também simples. Como as representações das arestas são simples existe possibilidade de ocorrer correspondências noutras posições.

Por visualização dos dados, escolheu-se um conjunto de imagens, cujos *templates* extraídos destes são equivalentes ao da figura 4.38 (a), sendo estas as imagens 1, 2, 5, 6, 8, 14, 15, 16, 17, 18 e 19. Os *templates* que não serão utilizados daqui a diante são equivalentes aos das figuras 4.38 (b) e 4.38 (c). Ao se executar os mesmos algoritmos apenas com os *templates* que serão utilizados obtém-se os gráficos 4.39 e 4.40. Ao não se considerar imagens que não possuem representações características de arestas observou-se uma grande melhoria na precisão das posições retornadas. No entanto, a obtenção dessa boa performance implicou uma remoção manual de imagens cujos *templates* eram propensos a gerar, pelos algoritmos, posições erróneas. Significando isto que o método *Chamfer Matching* não é tão robusto como se assumiu inicialmente e o mesmo não é adequado para a localização de imagens de estrada em imagens de satélite. Em termos de custo computacional, o algoritmo GCM produz menores tempos de execução do que o algoritmo GTM, como observado pela tabela 4.1.

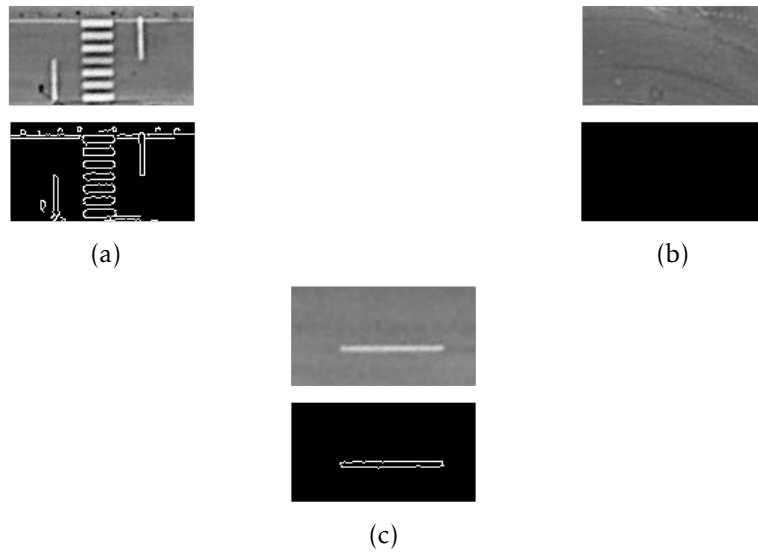


Figura 4.38: *Templates* que geram situações diferentes nos algoritmos *Chamfer Matching*. Cada subfigura possui em cima o *template* a ser analisado e em baixo a respectiva representação das arestas. (a) *Template* do qual é identificado a posição correta do mesmo na imagem-alvo; (b) *Template* cuja representação de arestas não possui arestas; (c) *Template* cuja representação de arestas é demasiado simples, podendo a sua localização na imagem-alvo ser errônea.

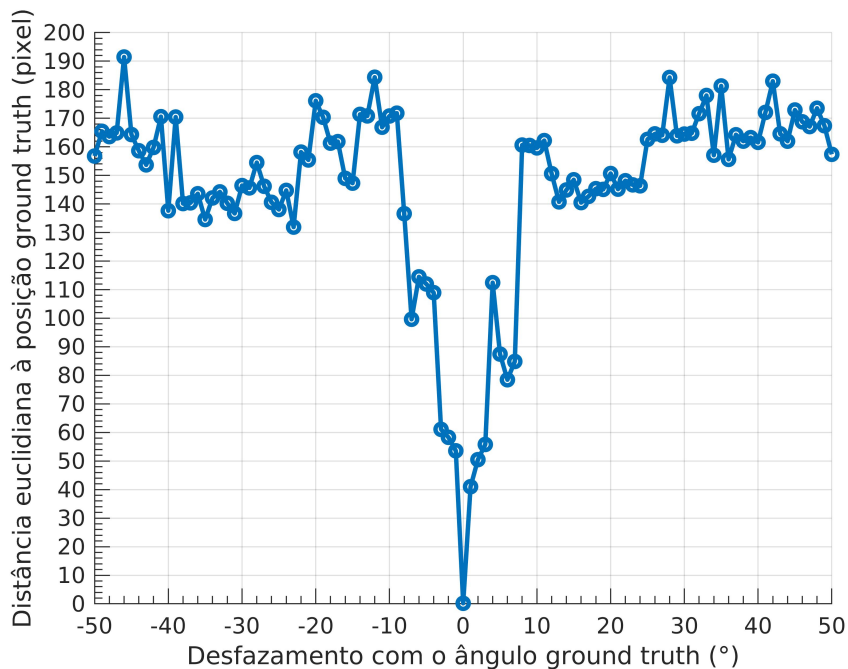


Figura 4.39: Distâncias euclidianas do algoritmo CCM, considerando apenas *templates* adequados, face à variação do ângulo a que este é rodado.

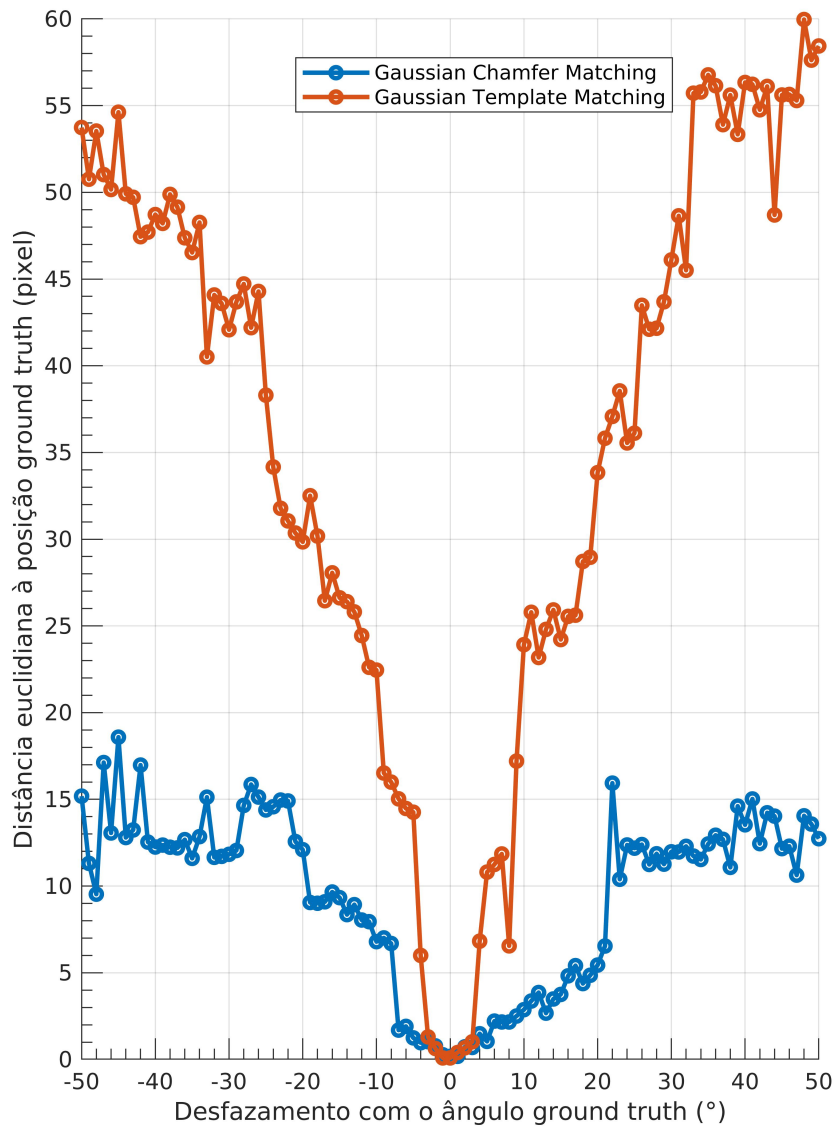


Figura 4.40: Distâncias euclidianas do algoritmo GCM, considerando apenas *templates* adequados, face à variação do ângulo a que este é rodado.

Tabela 4.1: Tempos de execução dos algoritmos GTM e GCM, com a variação do parâmetro PP.

Métodos	Tempos de execução (s)				
	(PP=1)	(PP=2)	(PP=3)	(PP=4)	(PP=5)
GTM	88,9552	24,8045	10,7798	5,7240	4,6881
GCM	7,7656	2,4280	0,9589	0,6013	0,3747

### 4.2.3 Análise dos algoritmos em condições reais

Anteriormente, recorreu-se a um ambiente simulado para testar os vários algoritmos de localização. Os resultados obtidos neste tipo de ambientes demonstraram-se adequados para efetuar a análise do custo computacional, assim como a precisão das posições retornadas, face à variação dos parâmetros PP e PA. No entanto, em ambientes de condições reais, existem inúmeros fatores que influenciam os resultados, tais como diferenças de luminosidade entre o *template* e imagem satélite, distorções derivadas do próprio método IPM, ou a existência de objetos na imagem satélite não existentes no *template*. Como tal, considera-se que a análise dos resultados de cada algoritmo, em condições reais, é fulcral.

Os *templates* a serem utilizados em condições reais são imagens IPM geradas pelo sistema ATLASCAR em vários locais na sua rota. De modo a proceder à referida análise dos algoritmos, foram geradas quatro imagens IPM com os dados obtidos do *bag-file* “Target2”, denominadas por “*template1* (T1)”, “*template2* (T2)”, “*template3* (T3)” e “*template4* (T4)”, ilustradas na figura 4.41. Através destas imagens IPM, foram obtidas as imagens de satélite correspondentes, de dimensão 425x500 píxeis.

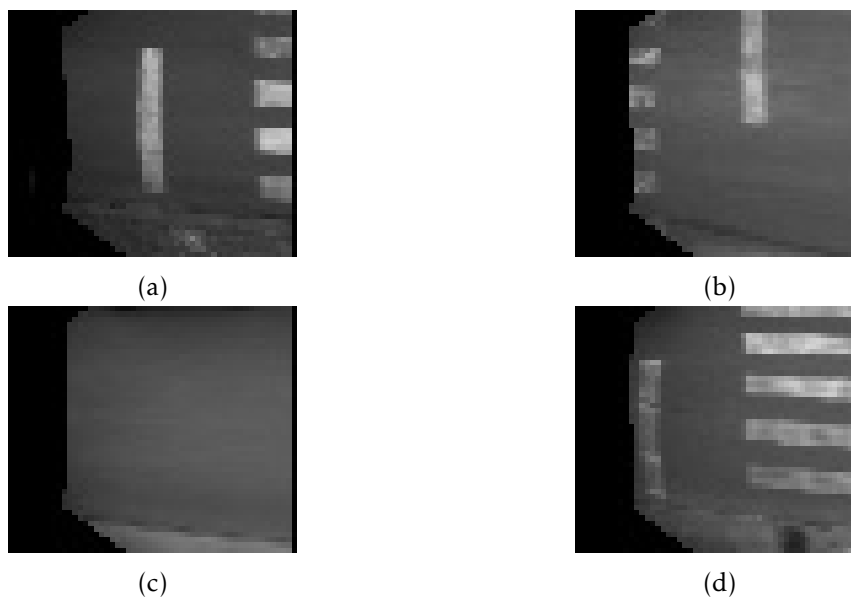


Figura 4.41: *Templates* utilizados na validação dos algoritmos em condições reais. a) “*template1*”; b) “*template2*”; c) “*template3*”; d) “*template4*”

Executaram-se os algoritmos CTM e GTM em condições reais e produziram-se os gráficos ilustrados pelas figuras 4.42, 4.43, 4.44 e 4.45. Em ambos os algoritmos, as distâncias euclidianas e os valores de correlação demonstraram-se incoerentes. Comparativamente ao CTM, o algoritmo GTM produziu, no geral, menores distâncias euclidianas, apesar destas apresentarem uma larga flutuação face à variação do ângulo de rotação do *template*. Para cada um dos *templates*, observou-se que as distâncias euclidianas se aproximam de zero para alguns ângulos. No entanto, apesar de determinados ângulos de rotação do *template* produzirem distâncias euclidianas próximas de zero,

estes, assim como as suas respectivas posições geradas, não se iriam apresentar como os valores finais do algoritmo na maioria dos casos, visto que o valor de correlação correspondente a esse ângulo não é o mais elevado.

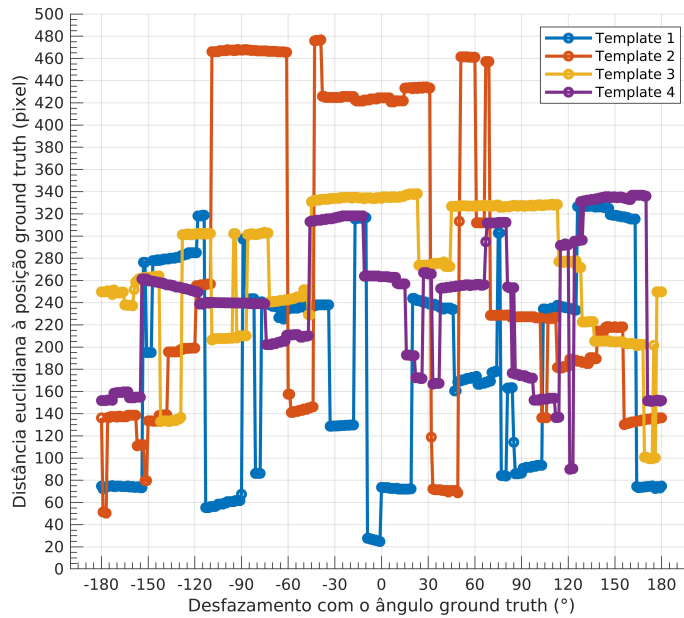


Figura 4.42: Distâncias euclidianas obtidas do algoritmo CTM sobre cada imagem de teste, face à variação do ângulo a que o *template* é rodado.

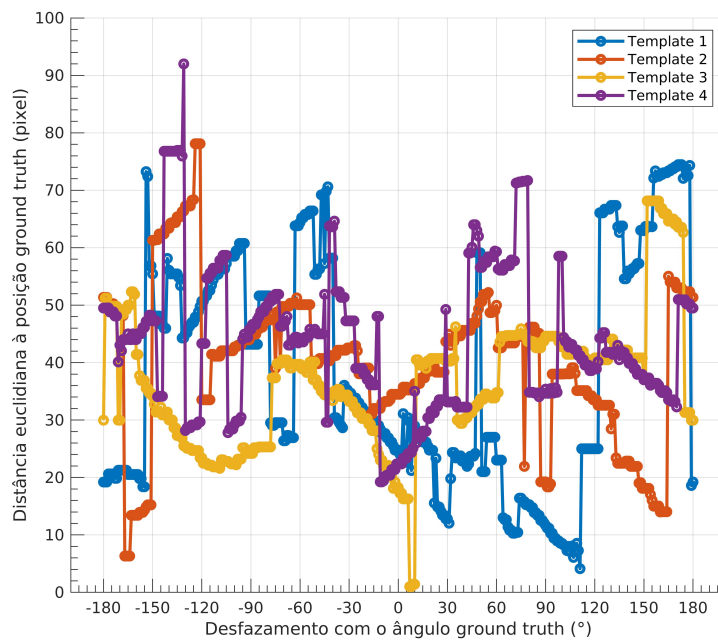


Figura 4.43: Distâncias euclidianas obtidas do algoritmo GTM sobre cada imagem de teste, face à variação do ângulo a que o *template* é rodado.

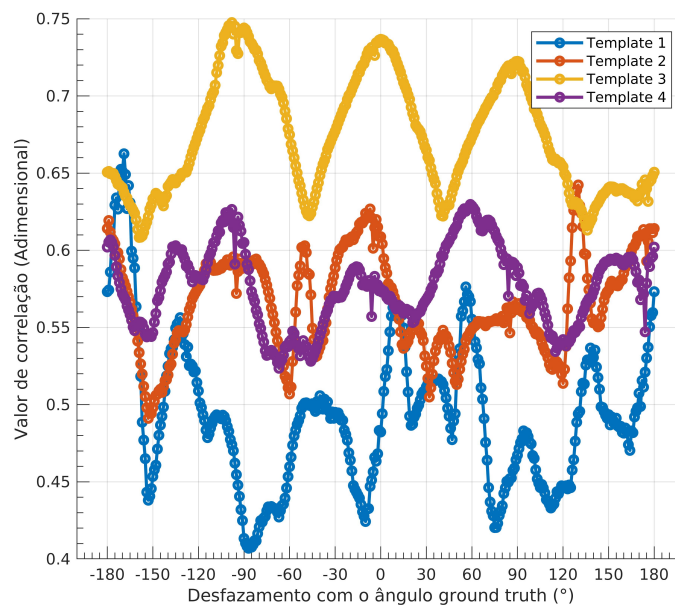


Figura 4.44: Valores de correlação obtidas do algoritmo CTM sobre cada imagem de teste, face à variação do ângulo a que o *template* é rodado.

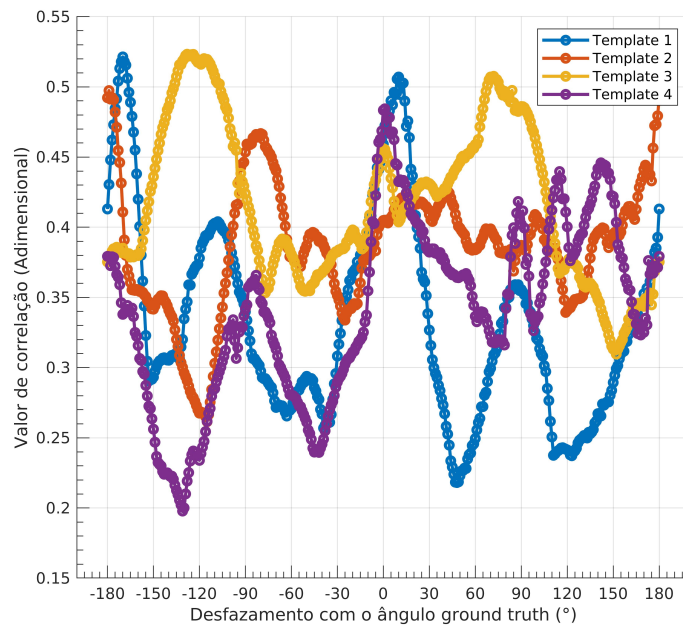


Figura 4.45: Valores de correlação obtidas do algoritmo GTM sobre cada imagem de teste, face à variação do ângulo a que o *template* é rodado.

Foram, de seguida, aplicados os algoritmos CCM e GCM, obteve-se as figuras 4.46, 4.47, 4.48 e 4.49. O CCM apresentou muita aleatoriedade, similarmente aos algoritmos anteriores. Por outro lado, o GCM apresentou valores de correlação e distâncias euclidianas mais “estáveis”, comparativamente aos outros algoritmos. Adicionalmente, considerou-se a localização do T4 corretamente determinada pelo algoritmo GCM. O menor valor de correlação obtido pelo GCM com T4 foi identificado próximo do ângulo *ground truth*, com a distância euclidiana, associada a esse ângulo, inferior a 5 píxeis. No entanto, uma vez que se identificou a localização de apenas um *template*, estes algoritmos, similarmente aos anteriores, demonstraram-se inadequados quanto à sua aplicabilidade em localizações relativas a porções de estrada.



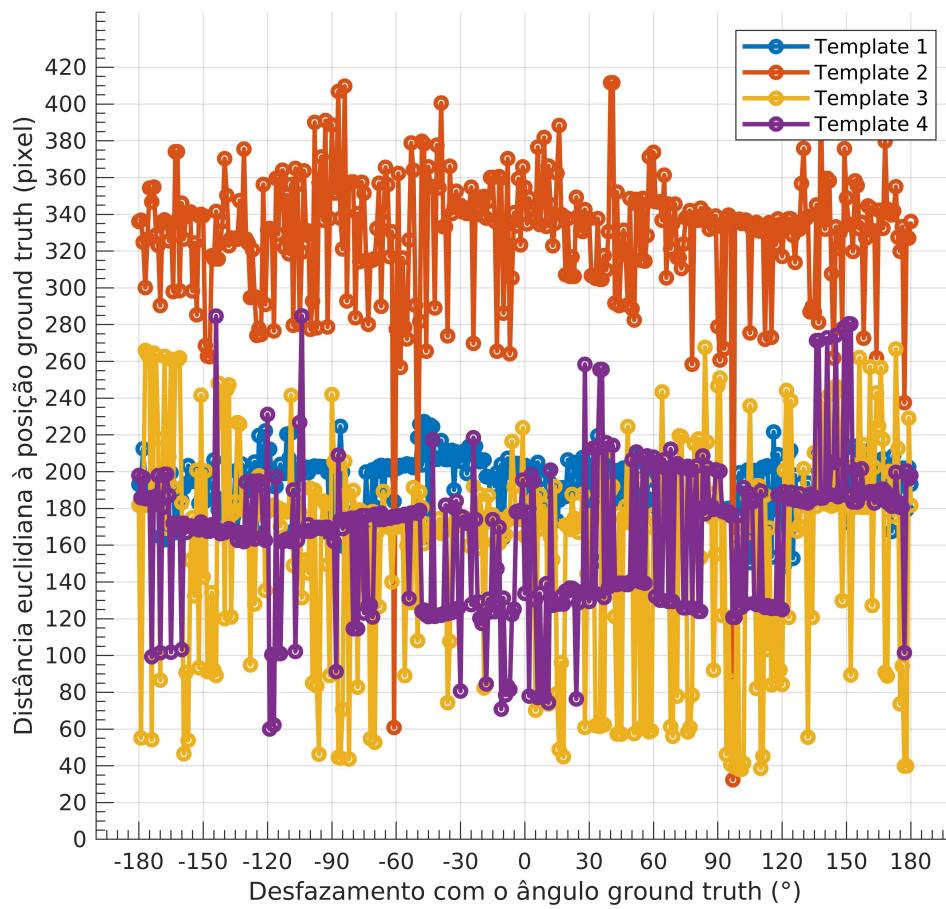


Figura 4.46: Distâncias euclidianas obtidas do algoritmo CCM sobre cada imagem de teste, face à variação do ângulo a que o *template* é rodado.

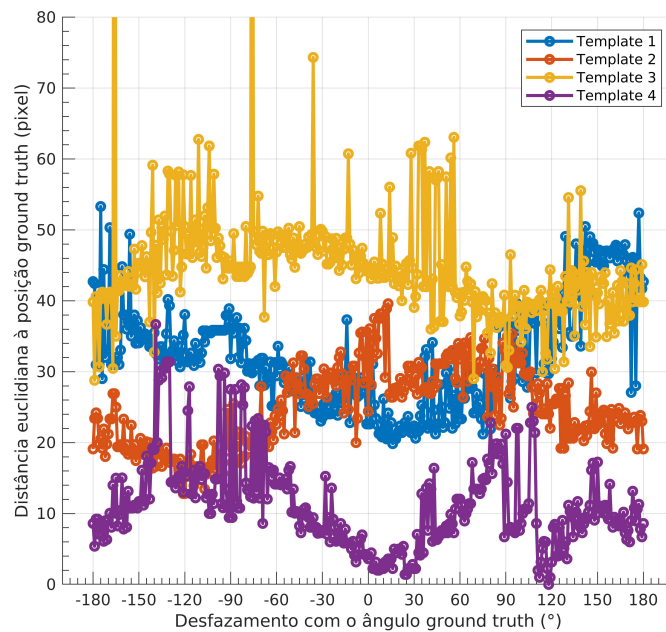


Figura 4.47: Distâncias euclidianas obtidas do algoritmo GCM sobre cada imagem de teste, face à variação do ângulo a que o *template* é rodado.

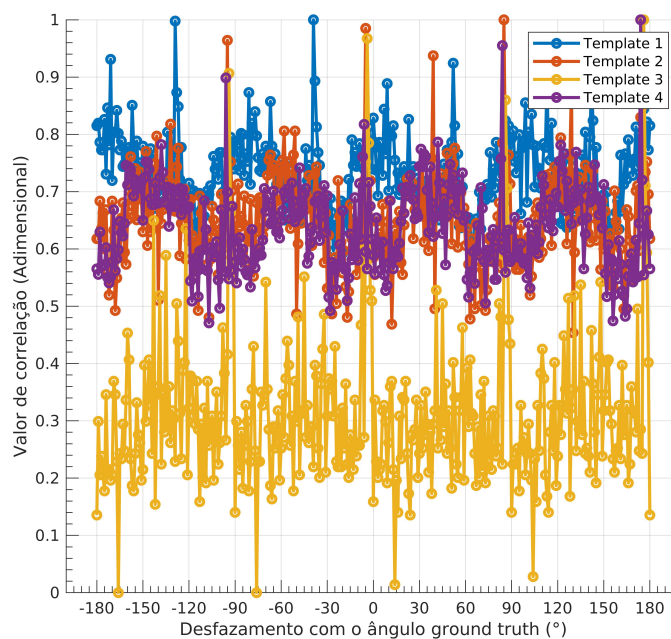


Figura 4.48: Valores de correlação obtidas do algoritmo CCM sobre cada imagem de teste, face à variação do ângulo a que o *template* é rodado.

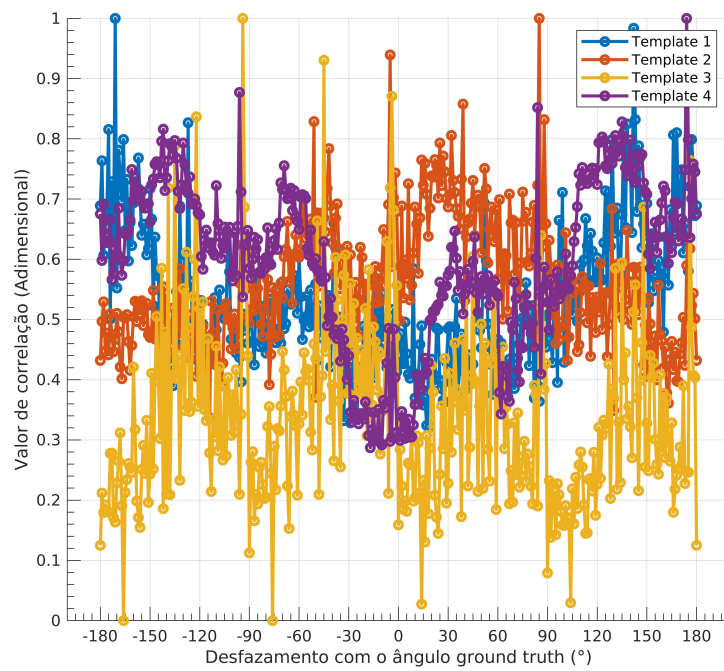


Figura 4.49: Valores de correlação obtidas do algoritmo GCM sobre cada imagem de teste, face à variação do ângulo a que o *template* é rodado.



## Capítulo 5

# Conclusão

A presente investigação procurou centrar a sua análise em dois objetivos principais, sendo estes o desenvolvimento do método IPM e a investigação de potenciais métodos de geolocalização. Posto isto e após alguma reflexão, em questões de âmbito teórico e perspetivadas no estudo, é chegado o momento de se efetuar uma análise que permita conjugar a síntese dos objetivos que me propus alcançar.

De uma forma genérica, no que respeita ao primeiro caso, pretendeu-se desenvolver um método IPM com a capacidade de remover obstáculos (e.g., carros, peões) e com tempos de execução adequados para uma posterior aplicação em tempo real. No segundo caso, o objetivo principal consistiu na investigação dos vários métodos de geolocalização existentes e o desenvolvimento de alguns destes, recorrendo a imagens IPM. Como estes dois métodos são independentes entre si, de certa forma, haverá também duas conclusões, que serão de seguida fundamentadas.

Entre as múltiplas variações do método IPM desenvolvidas, constatou-se o IPM G como a abordagem com tempos de execução mais reduzidos. Este método utiliza uma operação de suavização ao invés de uma operação de interpolação, uma vez que este último, apesar de representar a imagem IPM com maior aproximação da realidade, é mais computacionalmente custoso.

Como referido no Capítulo 2, “Estado de Arte”, procurou-se investigar os diferentes métodos de geolocalização. Neste sentido, e com o propósito de se efetuar a geolocalização com base nas teorias já existentes, desenvolveram-se os algoritmos CTM, GTM, CCM e GCM, em condições simplificadas. Os algoritmos CTM e GTM apresentaram uma precisão satisfatória, mas possuem um custo computacional demasiado elevado para serem aplicáveis em condução autónoma. No caso dos algoritmos CCM e GCM, apesar de estes possuírem tempos de execução mais reduzidos em comparação com os algoritmos anteriores não são ainda aplicáveis em sistemas que exigem processamentos em tempo real. Para além disso, ainda referente aos algoritmos CCM e GCM, verificou-se em condições simuladas que estes algoritmos retornavam, em alguns casos, posições erróneas. A causa destes valores advém do próprio método *Chamfer Matching*, i.e., o próprio método não é robusto o suficiente para evitar ambiguidade dos resultados, em alguns casos. Quando se removeu da análise de resultados as imagens que geravam falsos positivos a precisão das posições eram igualáveis aos algoritmos CTM e GTM. Estes resultados podem ser analisados e confirmados no Capítulo 4, “Resultados e discussão”.

Em resumo, todos estes algoritmos de localização possuem tempos de execução su-

periores ao que é considerado processamento em tempo real. Em condições reais, nenhum dos algoritmos desenvolvidos demonstrou uma boa precisão na determinação da localização dos *templates*, nas respectivas imagens de satélite. Das imagens IPM utilizadas, apenas a posição de uma destas foi considerada corretamente localizada na imagem satélite, por parte do algoritmo GCM. A imprecisão dos algoritmos poderá advir de três possíveis fatores, que podem ser independentes entre si ou não, sendo estes: 1) as imagens IPM possuem alguma distorção; 2) existe diferenças de luminosidade entre as imagens IPM e as imagens de satélite; 3) os próprios algoritmos utilizados não são robustos o suficiente para suportar possíveis ambiguidades existentes em condições reais.

Até este ponto da conclusão foi sucintamente revisto os principais resultados dos métodos IPM e algoritmos de geolocalização desenvolvidos, juntamente com uma avaliação crítica da performance dos mesmos. Prossegue-se para uma discussão de perspectivas futuras.

Referente ao método IPM, considera-se necessário otimizar a operação de interpolação para que esta seja apta para sistemas de tempo real. É também necessário a adição de pré-processamentos que atribuem mais robustez ao método IPM (e.g., cálculo do ponto de convergência). Por último, existe uma escassez de técnicas que avaliam a precisão das imagens resultantes do método IPM. Essas técnicas permitiriam analisar com fundamentos matemáticos a precisão de métodos IPM com a junção de determinados pré-processamentos.

No caso do tópico de geolocalização, os métodos *Template Matching* e *Chamfer Matching* possuem o potencial de serem aplicáveis como métodos de geolocalização. No entanto, é necessário investigar técnicas que reduzem os tempos de execução dos mesmos várias ordens de grandeza, pois estes métodos são computacionalmente custosos. Verificou-se que a aplicação dos algoritmos desenvolvidos em imagens reais retorna resultados insatisfatórios. Requer-se, por isso, mais investigação sobre métodos similares ao *Template Matching* ou *Chamfer Matching*, que sejam mais robustos, ou métodos totalmente inovativos, mas que continuem a incluir uma análise entre imagens que possuem o mesmo ponto de vista, i.e., vistas de cima.

---

# Bibliografia

- [1] M. Oliveira, V. Santos, and A. D. Sappa, "Multimodal inverse perspective mapping," *Information Fusion*, vol. 24, pp. 108–121, 2015.
- [2] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE transactions on image processing*, vol. 7, no. 1, pp. 62–81, 1998.
- [3] R. F. Berriel, E. de Aguiar, V. V. de Souza Filho, and T. Oliveira-Santos, "A particle filter-based lane marker tracking approach using a cubic spline model," in *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*. IEEE, 2015, pp. 149–156.
- [4] W. Yang, B. Fang, and Y. Y. Tang, "Fast and accurate vanishing point detection and its application in inverse perspective mapping of structured road," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 5, pp. 755–766, 2018.
- [5] A. Borkar, M. Hayes, M. T. Smith, and S. Pankanti, "A layered approach to robust lane detection at night," in *Computational Intelligence in Vehicles and Vehicular Systems, 2009. CIVVS'09. IEEE Workshop on*. IEEE, 2009, pp. 51–57.
- [6] [Online]. Available: [http://wiki.ros.org/camera\\_calibration/Tutorials/MonocularCalibration](http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration)
- [7] J. J. Rolison, S. Regev, S. Moutari, and A. Feeney, "What are the factors that contribute to road accidents? an assessment of law enforcement views, ordinary drivers opinions, and road accident records," *Accident Analysis & Prevention*, vol. 115, pp. 11–24, 2018.
- [8] A. Sathyanarayana, P. Boyraz, and J. H. Hansen, "Information fusion for robust 'context and driver aware' active vehicle safety systems," *Information Fusion*, vol. 12, no. 4, pp. 293–303, 2011.
- [9] J. Lewis, "Fast normalized cross-correlation/fast template matching, vision interface," 1995.
- [10] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 1. IEEE, 2005, pp. 654–661.
- [11] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision," pp. 153–177, 2003.

- [12] C. Wang and Z.-k. Shi, "A novel traffic stream detection method based on inverse perspective mapping," *Procedia engineering*, vol. 29, pp. 1938–1943, 2012.
- [13] M. Oliveira and V. Santos, "Multi-camera active perception system with variable image perspective for mobile robot navigation," in *8th Conference on Mobile Robots and Competitions, Festival Nacional de Robótica 2008, Aveiro, April 2008*, 2008.
- [14] M. Aly, "Real time detection of lane markers in urban streets," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 7–12.
- [15] J. Jeong and A. Kim, "Adaptive inverse perspective mapping for lane map generation with slam," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on*. IEEE, 2016, pp. 38–41.
- [16] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, "Fast vanishing-point detection in unstructured environments," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 425–430, 2012.
- [17] P.-M. Damon, H. Hadj-Abdelkader, H. Arioui, and K. Youcef-Toumi, "Inverse perspective mapping roll angle estimation for motorcycles," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 349–354.
- [18] S. Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, "Efficient lane detection and tracking in urban environments," in *European Conference on Mobile Robots*. ECMR, 2007.
- [19] C. Wang and Z.-k. Shi, "A novel traffic stream detection method based on inverse perspective mapping," *Procedia engineering*, vol. 29, pp. 1938–1943, 2012.
- [20] A. Kaso, "Computation of the normalized cross-correlation by fast fourier transform," *PloS one*, vol. 13, no. 9, p. e0203434, 2018.
- [21] B. Liu, X. Shu, and X. Wu, "Fast screening algorithm for rotation and scale invariant template matching," *arXiv preprint arXiv:1707.05647*, 2017.
- [22] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," Tech. Rep., 1977.
- [23] G. Borgefors, "Distance transformations in digital images," *Computer vision, graphics, and image processing*, vol. 34, no. 3, pp. 344–371, 1986.
- [24] Y. Ishino and H. Saji, "Extraction of road markings from aerial images," in *SICE Annual Conference, 2008*. IEEE, 2008, pp. 2180–2183.
- [25] M. Bansal, H. S. Sawhney, H. Cheng, and K. Daniilidis, "Geo-localization of street views with aerial image databases," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 1125–1128.
- [26] D.-K. Kim and M. R. Walter, "Satellite image-based localization via learned embeddings," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2073–2080.



# Apêndice



Tabela 5.1: Evolução da média e desvio padrão dos tempos de execução, na construção da máscara representativa de área sem obstáculos, consoante a variação da resolução da imagem original. Dados utilizados dos métodos IPM C, IPM D e IPM E.

Height x Width (Píxeis)	Média (s)	Desvio Padrão (s)
81x61	0,0230	0,0049
162x122	0,0209	0,0064
243x183	0,0157	0,0039
324x244	0,0182	0,0014
406x306	0,0152	0,0026
487x367	0,0152	0,0026
568x428	0,0131	0,0022
649x489	0,0156	0,0031
730x550	0,0202	0,0062
812x612	0,0181	0,0039

Tabela 5.2: Variação dos tempos de execução de cada fase do método IPM A, consoante a variação da resolução da imagem a ser utilizada.

Height x Width (Píxeis)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
81x61	0,0000	0,0145	0,0755	0,0614
162x122	0,0000	0,0428	0,3325	0,2651
243x183	0,0000	0,0679	0,7007	0,5965
324x244	0,0000	0,1258	1,2409	1,1336
406x306	0,0000	0,1585	1,9879	1,4502
487x367	0,0000	0,2502	2,8364	2,1046
568x428	0,0000	0,3937	3,7913	2,7962
649x489	0,0000	0,6997	4,9555	4,7692
730x550	0,0000	0,8385	6,3690	4,4582
812x612	0,0000	1,0419	7,8257	7,1469

Tabela 5.3: Variação dos tempos de execução de cada fase do método IPM B, consoante a variação da resolução da imagem a ser utilizada.

Height x Width (Píxeis)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/Smoothing (s)
81x61	0,0000	0,0059	3,5765	0,0370
162x122	0,0000	0,0088	7,9436	0,1136
243x183	0,0000	0,0240	22,6833	0,3660
324x244	0,0000	0,1024	42,6358	0,5949
406x306	0,0000	0,0643	59,3452	0,9188
487x367	0,0000	0,0778	93,2869	1,0265
568x428	0,0000	0,1400	128,3335	2,4287
649x489	0,0000	0,1465	167,3016	2,6522
730x550	0,0000	0,8541	221,7753	2,4076
812x612	0,0000	0,2624	291,8487	3,6526

Tabela 5.4: Variação dos tempos de execução de cada fase do método IPM C, consoante a variação da resolução da imagem a ser utilizada.

Height x Width (Píxeis)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/Smoothing (s)
81x61	0,0213	0,0105	0,0356	0,0439
162x122	0,0149	0,0363	0,1267	0,1655
243x183	0,0181	0,0854	0,2701	0,3919
324x244	0,0185	0,0871	0,4689	0,6544
406x306	0,0132	0,1257	0,7330	0,8746
487x367	0,0173	0,2033	1,0523	1,1895
568x428	0,0133	0,3388	1,4343	1,8687
649x489	0,0115	0,3806	1,8940	2,2523
730x550	0,0134	0,4959	2,3462	2,5657
812x612	0,0229	0,8609	2,9682	3,7956

Tabela 5.5: Variação dos tempos de execução de cada fase do método IPM D, consoante a variação da resolução da imagem a ser utilizada.

Height x Width (Píxeis)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
81x61	0,0180	0,0002	0,0007	0,0298
162x122	0,0180	0,0008	0,0031	0,0800
243x183	0,0189	0,0011	0,0041	0,1790
324x244	0,0198	0,0034	0,0055	0,2906
406x306	0,0188	0,0007	0,0047	0,4026
487x367	0,0115	0,0014	0,0068	0,5310
568x428	0,0102	0,0015	0,0073	0,6265
649x489	0,0190	0,0046	0,0158	0,7698
730x550	0,0284	0,0050	0,0244	0,8819
812x612	0,0182	0,0038	0,0254	0,9505

Tabela 5.6: Variação dos tempos de execução de cada fase do método IPM E, consoante a variação da resolução da imagem a ser utilizada.

Height x Width (Píxeis)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
81x61	0,0296	0,0017	0,1652	0,0492
162x122	0,0299	0,0032	0,3042	0,1125
243x183	0,0102	0,0025	0,6237	0,2175
324x244	0,0165	0,0028	1,0500	0,3187
406x306	0,0136	0,0021	1,6969	0,4251
487x367	0,0168	0,0038	2,5349	0,5862
568x428	0,0157	0,0059	3,3239	0,6967
649x489	0,0162	0,0059	4,4622	0,8644
730x550	0,0187	0,0066	5,5281	0,9466
812x612	0,0134	0,0077	6,5161	0,9994

Tabela 5.7: Variação dos tempos de execução de cada fase do método IPM F, consoante a variação da resolução da imagem a ser utilizada.

Height x Width (Píxeis)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
81x61	0,0010	0,0003	0,0034	0,0409
162x122	0,0009	0,0006	0,0036	0,135
243x183	0,0010	0,0006	0,0018	0,2561
324x244	0,0016	0,0024	0,0065	0,4295
406x306	0,0007	0,0037	0,0038	0,6209
487x367	0,0008	0,0049	0,0080	0,7789
568x428	0,0007	0,0030	0,0127	0,9548
649x489	0,0011	0,0087	0,0147	1,2151
730x550	0,0007	0,0052	0,0197	1,2491
812x612	0,0021	0,0079	0,0211	1,4251

Tabela 5.8: Variação dos tempos de execução de cada fase do método IPM G, consoante a variação da resolução da imagem a ser utilizada.

Height x Width (Píxeis)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
81x61	0,0018	0,0002	0,0016	0,0052
162x122	0,0018	0,0003	0,0037	0,0046
243x183	0,0016	0,0005	0,0028	0,0052
324x244	0,0022	0,0032	0,0054	0,0050
406x306	0,0007	0,0035	0,0067	0,0059
487x367	0,0041	0,0013	0,0089	0,0050
568x428	0,0013	0,0033	0,0183	0,0056
649x489	0,0006	0,0055	0,0230	0,0053
730x550	0,0014	0,0062	0,0237	0,0056
812x612	0,0014	0,0066	0,0328	0,0046

Tabela 5.9: Variação dos tempos de execução de cada fase do método IPM A, consoante a variação da área do polígono da imagem a ser utilizada.

Height x Width (m)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
3x3	0,0000	0,0308	0,3005	0,1675
4x4	0,0000	0,0534	0,3177	0,2154
5x5	0,0000	0,0345	0,3346	0,2571
6x6	0,0000	0,0165	0,3143	0,2355
7x7	0,0000	0,0162	0,3043	0,2012
8x8	0,0000	0,0318	0,3089	0,3077
9x9	0,0000	0,0570	0,3144	0,2737
10x10	0,0000	0,0495	0,3242	0,2767
11x11	0,0000	0,0583	0,3255	0,2864
12x12	0,0000	0,0183	0,3160	0,2736
13x13	0,0000	0,0247	0,3287	0,3268

Tabela 5.10: Variação dos tempos de execução de cada fase do método IPM B, consoante a variação da área do polígono da imagem a ser utilizada.

Height x Width (m)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
3x3	0,0000	0,0089	10,8439	0,1074
4x4	0,0000	0,0208	12,1445	0,1462
5x5	0,0000	0,0219	11,3563	0,1384
6x6	0,0000	0,0223	11,7586	0,1827
7x7	0,0000	0,0165	10,6450	0,1822
8x8	0,0000	0,0163	11,7489	0,1833
9x9	0,0000	0,0171	13,0675	0,2083
10x10	0,0000	0,0155	9,8179	0,2012
11x11	0,0000	0,0196	11,9114	0,2532
12x12	0,0000	0,0214	12,9020	0,2693
13x13	0,0000	0,0209	12,2989	0,2442

Tabela 5.11: Variação dos tempos de execução de cada fase do método IPM C, consoante a variação da área do polígono da imagem a ser utilizada.

Height x Width (m)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
3x3	0,0150	0,0211	0,1150	0,1222
4x4	0,0241	0,0286	0,1182	0,1115
5x5	0,0100	0,0152	0,1116	0,1429
6x6	0,0099	0,0113	0,1121	0,1430
7x7	0,0117	0,0115	0,1171	0,1549
8x8	0,0091	0,0103	0,1098	0,1591
9x9	0,0090	0,0107	0,1158	0,1818
10x10	0,0117	0,0190	0,1207	0,1864
11x11	0,0162	0,0331	0,1229	0,2190
12x12	0,0249	0,0369	0,1180	0,2531
13x13	0,0200	0,0359	0,1225	0,3001

Tabela 5.12: Variação dos tempos de execução de cada fase do método IPM D, consoante a variação da área do polígono da imagem a ser utilizada.

Height x Width (m)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
3x3	0,0163	0,0004	0,0016	0,0310
4x4	0,0275	0,0006	0,0014	0,0666
5x5	0,0251	0,0011	0,0034	0,0836
6x6	0,0185	0,0004	0,0011	0,1028
7x7	0,0227	0,0015	0,0029	0,1113
8x8	0,0211	0,0004	0,0013	0,1215
9x9	0,0193	0,0005	0,0034	0,1275
10x10	0,0199	0,0006	0,0015	0,1332
11x11	0,0187	0,0003	0,0019	0,1381
12x12	0,0209	0,0003	0,0024	0,1510
13x13	0,0187	0,0003	0,0014	0,1601



Tabela 5.13: Variação dos tempos de execução de cada fase do método IPM E, consoante a variação da área do polígono da imagem a ser utilizada.

Height x Width (m)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
3x3	0,0348	0,0008	0,4181	0,0564
4x4	0,0351	0,0007	0,3859	0,1130
5x5	0,0228	0,0011	0,3760	0,1197
6x6	0,0125	0,0010	0,2950	0,1335
7x7	0,0096	0,0017	0,3593	0,1774
8x8	0,0097	0,0013	0,3631	0,1746
9x9	0,0223	0,0012	0,3500	0,1825
10x10	0,0199	0,0017	0,3713	0,1924
11x11	0,0216	0,0036	0,3916	0,2076
12x12	0,0194	0,0034	0,3585	0,2091
13x13	0,0173	0,0035	0,4101	0,2510

Tabela 5.14: Variação dos tempos de execução de cada fase do método IPM F, consoante a variação da área do polígono da imagem a ser utilizada.

Height x Width (m)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
3x3	0,0011	0,0008	0,0020	0,0541
4x4	0,0007	0,0005	0,0024	0,0938
5x5	0,0008	0,0006	0,0015	0,1077
6x6	0,0019	0,0014	0,0011	0,1348
7x7	0,0016	0,0026	0,0018	0,1538
8x8	0,0031	0,0015	0,0040	0,1693
9x9	0,0018	0,0005	0,0016	0,1632
10x10	0,0014	0,0002	0,0007	0,1684
11x11	0,0012	0,0003	0,0008	0,1917
12x12	0,0014	0,0010	0,0007	0,1986
13x13	0,0014	0,0005	0,0008	0,2112

Tabela 5.15: Variação dos tempos de execução de cada fase do método IPM G, consoante a variação da área do polígono da imagem a ser utilizada.

Height x Width (m)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
3x3	0,0013	0,0003	0,0023	0,0020
4x4	0,0014	0,0015	0,0019	0,0030
5x5	0,0020	0,0004	0,0028	0,0055
6x6	0,0008	0,0005	0,0020	0,0075
7x7	0,0014	0,0009	0,0033	0,0105
8x8	0,0029	0,0004	0,0034	0,0146
9x9	0,0019	0,0005	0,0038	0,0160
10x10	0,0007	0,0014	0,0022	0,0213
11x11	0,0026	0,0003	0,0028	0,0228
12x12	0,0009	0,0004	0,0013	0,0249
13x13	0,0025	0,00016	0,0040	0,0298

Tabela 5.16: Variação dos tempos de execução de cada fase do método IPM A, consoante a variação da escala RSF.

RSF (%)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
5	0,0000	0,0537	0,3280	0,1860
10	0,0000	0,0421	0,3073	0,2574
15	0,0000	0,0257	0,3001	0,2362
20	0,0000	0,0396	0,3037	0,2485
25	0,0000	0,0558	0,3090	0,3256
30	0,0000	0,0467	0,3228	0,3756
35	0,0000	0,0544	0,3086	0,4163
40	0,0000	0,0298	0,3340	0,5274
45	0,0000	0,0660	0,3217	0,6274
50	0,0000	0,0591	0,3214	0,7065

Tabela 5.17: Variação dos tempos de execução de cada fase do método IPM B, consoante a variação da escala RSF.

RSF (%)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
5	0,0000	0,0210	11,6988 0	1024
10	0,0000	0,0181	10,0427	0,1241
15	0,0000	0,0194	11,0014	0,1515
20	0,0000	0,0121	10,5376	0,2033
25	0,0000	0,0205	10,2962	0,2177
30	0,0000	0,0113	9,7907	0,2991
35	0,0000	0,0119	9,6996	0,3037
40	0,0000	0,0159	10,5095	0,3785
45	0,0000	0,0169	9,8585	0,4698
50	0,0000	0,0198	10,0912	0,5080

Tabela 5.18: Variação dos tempos de execução de cada fase do método IPM C, consoante a variação da escala RSF.

RSF (%)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
5	0,0181	0,0317	0,1068	0,1034
10	0,0206	0,0360	0,1131	0,1619
15	0,0182	0,0372	0,1216	0,1489
20	0,0256	0,0310	0,1236	0,1895
25	0,0197	0,0447	0,1176	0,2279
30	0,0206	0,0368	0,1105	0,3546
35	0,0130	0,0267	0,1180	0,3548
40	0,0215	0,0354	0,1274	0,4321
45	0,0189	0,0493	0,1190	0,4969
50	0,0211	0,0503	0,1234	0,5554

Tabela 5.19: Variação dos tempos de execução de cada fase do método IPM D, consoante a variação da escala RSF.

RSF (%)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
5	0,0181	0,0024	0,0010	0,0646
10	0,0170	0,0002	0,0016	0,0795
15	0,0180	0,0004	0,0011	0,0916
20	0,0198	0,0012	0,0030	0,1301
25	0,0172	0,0005	0,0015	0,1529
30	0,0159	0,0003	0,0012	0,1788
35	0,0214	0,0006	0,0019	0,2185
40	0,0254	0,0006	0,0020	0,2886
45	0,0218	0,0013	0,0097	0,3628
50	0,0207	0,0032	0,0120	0,4239

Tabela 5.20: Variação dos tempos de execução de cada fase do método IPM E, consoante a variação da escala RSF.

RSF (%)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
5	0,0290	0,0017	0,3734	0,1143
10	0,0321	0,0019	0,3658	0,1158
15	0,0238	0,0040	0,3767	0,1576
20	0,0261	0,0021	0,3603	0,1712
25	0,0187	0,0035	0,3674	0,2118
30	0,0278	0,0045	0,3875	0,2374
35	0,0213	0,0040	0,3439	0,2856
40	0,0185	0,0029	0,3775	0,3542
45	0,0165	0,0043	0,3494	0,3954
50	0,0135	0,0032	0,3508	0,4406

Tabela 5.21: Variação dos tempos de execução de cada fase do método IPM F, consoante a variação da escala RSF.

RSF (%)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
5	0,0008	0,0006	0,0019	0,1263
10	0,0010	0,0006	0,0012	0,1029
15	0,0020	0,0017	0,0048	0,1333
20	0,0023	0,0018	0,0027	0,1640
25	0,0052	0,0021	0,0034	0,1783
30	0,0039	0,0028	0,0041	0,2487
35	0,0064	0,0030	0,0030	0,2641
40	0,0070	0,0027	0,0028	0,3059
45	0,0104	0,0047	0,0037	0,4027
50	0,0093	0,0030	0,0030	0,4423

Tabela 5.22: Variação dos tempos de execução de cada fase do método IPM G, consoante a variação da escala RSF.

RSF (%)	Masking (s)	Mapping (s)	Indexing (s)	Interpolation/ Smoothing (s)
5	0,0017	0,0005	0,0030	0,0013
10	0,0008	0,0005	0,0029	0,0053
15	0,0031	0,0021	0,0049	0,0129
20	0,0032	0,0007	0,0017	0,0333
25	0,0005	0,0018	0,0027	0,0313
30	0,0024	0,0010	0,0034	0,0417
35	0,0015	0,0027	0,0028	0,0579
40	0,0012	0,0035	0,0027	0,0738
45	0,0016	0,0028	0,0027	0,0931
50	0,0014	0,0037	0,0032	0,01147

Tabela 5.23: Tempos de execução de cada fase de cada método.

Métodos	Masking	Mapping	Indexing	Interpolation/ Smoothing
IPM A	0,0000	0,0455	0,2935	0,2294
IPM B	0,0000	0,0224	10,6348	0,2343
IPM C	0,0233	0,0369	0,1278	0,1860
IPM D	0,0176	0,0003	0,0013	0,0965
IPM E	0,0208	0,0010	0,3108	0,1496
IPM F	0,0010	0,0006	0,0012	0,1029
IPM G	0,0008	0,0005	0,0029	0,0053

