

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

# RASTREAMENTO E LOCALIZAÇÃO GEOGRÁFICA DE OBJETOS

**LUCAS ÁQUILA DOS SANTOS**

Orientador: Prof. Dr. Adriano Chaves Lisboa
CEFET MG

Coorientador: Prof. Dr. Alisson Marques da Silva
CEFET MG

BELO HORIZONTE
AGOSTO DE 2021

**LUCAS ÁQUILA DOS SANTOS**

# RASTREAMENTO E LOCALIZAÇÃO GEOGRÁFICA DE OBJETOS

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional.

Linha de pesquisa: Métodos Matemáticos Aplicados.

Orientador: Prof. Dr. Adriano Chaves Lisboa
CEFET MG

Coorientador: Prof. Dr. Alisson Marques da Silva
CEFET MG

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL
BELO HORIZONTE
AGOSTO DE 2021

Santos, Lucas Áquila dos
S237r Rastreamento e localização geográfica de objetos / Lucas Áquila dos Santos. – 2021.
55 f.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientador: Adriano Chaves Lisboa.
Coorientador: Alisson Marques da Silva.
Dissertação (mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Sistemas de informação geográfica – Teses. 2. Rastreamento automático – Teses. 3. Serviços baseados em localização – Teses. I. Lisboa, Adriano Chaves. II. Silva, Alisson Marques da. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. Título.

CDD 003.3

Elaboração da ficha catalográfica pela bibliotecária Jane Marangon Duarte, CRB 6º 1592 / Cefet/MG

# Agradecimentos

Agradeço a Deus por iluminar e abençoar os nossos caminhos. Só Ele sabe o quanto esta conquista foi almejada e quantos obstáculos superei.

À minha mãe Adédica e meu irmão Israel, por estarem sempre ao meu lado, encorajando e proporcionando todo apoio a mim.

À minha companheira 'peta' (Évelin), por iniciar essa caminhada comigo. Além de toda compreensão, suporte e risadas.

A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo financiamento de projetos e que contribuíram pela a minha manutenção no curso bem como crescimento pessoal.

Ao meu orientador Professor Dr. Adriano Chaves Lisboa por incentivar e acreditar no desenvolvimento deste projeto, direcionando esta pesquisa com obséquio e dedicação.

Ao meu coorientador Professor Dr. Alisson Marques da Silva pelas valiosas sugestões e contribuições para o avanço deste trabalho.

Aos professores e colegas que participaram desta árdua jornada dos quais nortearam a minha trajetória.

Um imenso agradecimento a instituição CEFET-MG que proporcionou toda estrutura, materiais de apoio e amparo não apenas para mim, mas para toda a comunidade acadêmica.

*“A ciência nunca resolve um problema sem criar pelo menos outros dez. (George Bernard Shaw)”*

# Resumo

O rastreamento e a localização são métodos que permitem mensurar a posição geográfica de um objeto por meio de imagens capturadas por câmera. A partir da coleta e análise de dados geoespaciais é possível realizar aplicações em inúmeros segmentos do cotidiano, como planejamento urbano e transporte. Neste trabalho, objetiva-se centralizar um objeto de uma imagem e determinar sua localização por meio de uma câmera e uma superfície de relevo. Para tanto, utiliza-se dois modelos, matemático analítico e numérico por interpolação, para o controle da câmera. A partir desta, são extraídas as informações necessárias para a validação de ambos os modelos com o uso das métricas de erro absoluto médio e erro quadrático médio. Sendo assim, obtém-se um erro MSE máximo de aproximadamente duas vezes maior e 1100 vezes mais rápido no modelo analítico em relação a interpolação. Além disso, é constatado por testes estatísticos que o método analítico possui um erro maior do que o método numérico. Por fim, estima-se os parâmetros da câmera por meio de equações não lineares e determina-se a posição geográfica do objeto através do método numérico. A escolha dos métodos dependerá do projeto, que exigirá uma resposta mais rápida ou mais precisa. A localização dependerá da distância entre os dados da superfície local.
**Palavras-chave**: Rastreamento; Localização Geográfica; Georreferenciamento.

# Abstract

Tracking and location are methods that allow you to measure the geographic position of an object using images captured by a camera. From the collection and analysis of geospatial data, it is possible to carry out applications in numerous daily life segments, such as urban planning and transportation. This work aims to centralize an image's object and determine its location using a camera and a relief surface. For this purpose, two models were use, analytical mathematician and numerical by interpolation, to control the camera. From this, the necessary information for the validation of both models is extracted using the metrics of mean absolute error and mean squared error. Thus, a maximum MSE error of approximately twice as large and 1100 times faster is obtained in the analytical model in relation to interpolation. Furthermore, it is verified by statistical tests that the analytical method has a greater error than the numerical method. Finally, the camera's parameters were estimate using non-linear equations and the geographic position of the object is determined using the numerical method. The choice of methods will depend on the project, which will require a faster or more accurate response. The location will depend on the distance between the local surface data.

**Keywords**: Tracking; Geographic Location; Georeferencing.

# Lista de Figuras


Figura 1 – Exemplo de um espaço $\Omega$ de imagem . . . . . 6
Figura 2 – Espaço de cor RGB . . . . . 7
Figura 3 – Exemplo de aplicação do filtro de Gauss . . . . . 7
Figura 4 – Exemplo de *kernel* dos filtros da forma . . . . . 8
Figura 5 – Algoritmo *matchTemplate* . . . . . 9
Figura 6 – Parâmetros da câmera. . . . . 11
Figura 7 – Ilustração dos vetores do modelo analítico. . . . . 12
Figura 8 – Ilustração dos vetores do modelo analítico. . . . . 14
Figura 9 – Divisão da topografia . . . . . 17
Figura 10 – Fluxograma da metodologia abordada no trabalho . . . . . 19
Figura 11 – Exemplo da interpolação baricêntrica . . . . . 24
Figura 12 – Imagem de referência com o *boundbox* no centro da imagem . . . . . 30
Figura 13 – Resultado da localização da região mais semelhante na imagem com o algoritmo *matchTemplate* . . . . . 30
Figura 14 – Malha do resultado do *matchTemplate* com CCOEFF . . . . . 30
Figura 15 – Resultado comparativo visual do centro da imagem de referência com a imagem da instância da máscara. . . . . 31
Figura 16 – *Dataset* bruto . . . . . 32
Figura 17 – *Dataset* com limitações do FoV . . . . . 33
Figura 18 – *Dataset* tratado . . . . . 34
Figura 19 – Erro em cada modelo . . . . . 36
Figura 20 – Erro absoluto em cada modelo . . . . . 37
Figura 21 – Tempo gasto em cada modelo . . . . . 38
Figura 22 – Ambiente Artificial . . . . . 40
Figura 23 – Imagem da câmera gerada artificialmente . . . . . 40
Figura 24 – Resultado da localização(Método Numérico). . . . . 41
Figura 25 – Determinação da área para o estudo. . . . . 42
Figura 26 – Região de estudo para o caso real . . . . . 43
Figura 27 – Foto de referência dos pontos fiduciais . . . . . 44
Figura 28 – Resultado da localização do método numérico para o ponto $e$ . . . . . 45
Figura 29 – Localização do ponto $e$ . . . . . 46
Figura 30 – Resultado da localização do método numérico para o ponto $f$ . . . . . 47
Figura 31 – Localização do ponto $f$ . . . . . 48
Figura 32 – Resultado final . . . . . 48
Figura 33 – Localização do ponto $e$ com os pontos da malha . . . . . 49

# Lista de Tabelas

Tabela 1 – Medidas normalizadas do algoritmo *matchTemplate* . . . . . 10
Tabela 2 – Resultado da acurácia para o modelo analítico . . . . . 37
Tabela 3 – Resultado da acurácia para a Interpolação . . . . . 38
Tabela 4 – Resultado do método numérico . . . . . 42
Tabela 5 – Referência dos Pontos Fiduciais. . . . . 44

# Lista de Abreviaturas e Siglas

| | |
|---|---|
| CCOEFF | Correlação cruzada de desvio médio |
| CCORR | Correlação cruzada |
| CMYK | Sistema de informações geográficas |
| DPI | Pontos por polegadas |
| FoV | Campo de Visão |
| HSV | Sistema de informações geográficas |
| IoT | Internet das coisas |
| IPM | Mapeamento de Perspectiva Inversa |
| MA | Método Analítico |
| MAE | Erro médio absoluto |
| MN | Método Numérico |
| MSE | Erro quadrático médio |
| MT | Casamento de modelo |
| OCR | Reconhecimento óptico de caracteres |
| PTZ | PAN, TILT e ZOOM |
| RAM | Memória de acesso aleatório |
| RGB | Sistema de informações geográficas |
| SIG | Sistema de informações geográficas |
| SQDIFF | Soma das diferenças ao quadrado |

# Sumário


**1 – Introdução . . . . . 1**
1.1 Motivação . . . . . 2
1.2 Objetivos . . . . . 3
1.2.1 Objetivo Geral . . . . . 3
1.2.2 Objetivos Específicos . . . . . 3
1.3 Organização do trabalho . . . . . 3

**2 – Fundamentação Teórica . . . . . 5**
2.1 Imagem . . . . . 5
2.1.1 Algoritmo *matchTemplate* . . . . . 8
2.2 Câmera . . . . . 10
2.2.1 Modelo analítico . . . . . 11
2.3 Interpolação . . . . . 13
2.4 Sistema de equações não lineares . . . . . 13
2.5 Topografia . . . . . 16
2.5.1 Sistemas de Informações Geográficas . . . . . 18

**3 – Rastreamento e Localização Geográfica de Objetos . . . . . 19**
3.1 Rastreamento de Objetos . . . . . 20
3.1.1 Aquisição de imagens . . . . . 20
3.1.2 Interpolação . . . . . 23
3.1.3 Modelo de câmera . . . . . 24
3.1.4 Avaliação dos modelos . . . . . 25
3.2 Localização Geográfica . . . . . 25
3.2.1 Encontrando posição, direção e abertura da câmera . . . . . 26
3.2.2 Localização do objeto . . . . . 26
3.2.2.1 Determinação do mapa de elevação . . . . . 27

**4 – Resultados experimentais . . . . . 29**
4.1 Aquisição de dados para rastreamento de objetos . . . . . 29
4.2 Centralização do objeto na imagem . . . . . 35
4.2.1 Acurácia . . . . . 35
4.2.2 Tempo de Resposta . . . . . 38
4.3 Determinação da posição do objeto no ambiente . . . . . 39
4.3.1 Caso Artificial . . . . . 39
4.3.2 Caso Real . . . . . 42

**5 – Conclusão . . . . . 50**
5.1 Sugestões para trabalhos futuros . . . . . 50

**Referências . . . . . 52**

# 1 Introdução

A visão computacional é uma área da computação que visa utilizar câmeras para analisar e compreender imagens, simulando a visão humana (KLETTE, 2014), tentando repassar para a máquina a capacidade de enxergar (BACKES; JUNIOR, 2016). Essa capacidade envolve a captação de imagens, a remoção de ruídos ou a otimização do contraste, a separação de regiões ou objetos de interesse, a extração de informações, entre outros. Na prática, a visão computacional pode ser empregada em diversas aplicações reais, como reconhecimento óptico de caracteres (OCR, do inglês *Optical Character Recognition*) (HUANG et al., 2020), inspeção de máquina (ZHOU et al., 2019), construção de modelos 3D (3 Dimensões) (KULIKAJEVAS et al., 2019), segurança automotiva (Ragesh; Rajesh, 2019), processamento de imagens médicas (SZELISKI, 2011), sistema de inspeção de controle de qualidade (SHETTY, 2019), sistemas de rede de câmeras para a vigilância de tráfego em tempo real (HO et al., 2019), decodificação automática de expressões faciais (HÖFLING et al., 2020), dentre outras.

Sistemas de Informações Geográficas (SIGs) são sistemas que possuem ferramentas responsáveis por coletar e armazenar informações geográficas e que posteriormente podem ser adquiridos e analisados. Dessa forma, corrobora para a representação, simulação ou análises dos fenômenos e efeitos que ocorrem sobre a superfície adquirida. De acordo com Burrough et al. (2015), os SIGs representam procedimentos computacionais sobre um conjunto de dados pré-existentes no qual permite a elaboração de mapas ou quaisquer elementos adequados para o tratamento e análise dos dados geográficos. Estes sistemas também podem ser utilizados em diversas aplicações, como recurso de gestão, planejamento urbano, marketing e transporte, dentre outros (BAJJALI, 2017; TATEOSIAN, 2015).

Portanto, tanto a visão computacional quanto os SIGs, abordam a visualização e a manipulação dos dados geográficos. Na visão computacional, por exemplo, de uma imagem de um relevo podem ser extraídas informações de distância entre pontos em pixel convertidas em metros. Já para os SIGs, é possível obter as medidas de pontos de uma malha, imagens planares, gráficos planimétricos e entre outros. A conciliação entre esses dois temas é abordada em alguns trabalhos que exploram a extração de características geográficas ou visam mensurar a distância por meio de imagens. Neste contexto, destaca-se o trabalho de Hoekendijk et al. (2015) que utiliza uma câmera para estimar a posição espacial de mamíferos ou aves na superfície marítima. Para isto, faz-se o uso de posições conhecidas presentes na imagem, como a linha costeira, e calcula-se a distância entre os objetos, no caso o mamífero, da câmera por meio de, principalmente, equações trigonométricas e de distância. Na pesquisa de Milosavljević et al. (2017) é utilizada uma câmera *PAN*, *TILT* e *ZOOM* (PTZ) para estimar posições georreferenciadas.

Inicialmente, a câmera PTZ obtém a direção atual por meio do sistema interno. Em seguida, identifica as características do ponto observado de forma manual e obtém as coordenadas geodésicas em 3D. Por fim, estima a posição utilizando um processo de cálculo interativo. Já na pesquisa de Xie et al. (2019), o georreferenciamento é feito por meio de múltiplas câmeras onde cada uma delas, posicionadas em diferentes localizações, identificam o objeto em áreas determinadas. Ao comparar as imagens capturas por cada câmera, faz-se uma estimativa do trajeto do objeto cuja imagem extrapola o campo de visão da câmera. Assim, determina-se a localização e o trajeto do objeto. A pesquisa de Zhu e Fang (2019) propôs um método baseado em aprendizado profundo para prever a distância de um dado objeto em imagens sem a necessidade de utilizar os parâmetros da câmera. E, por fim, o trabalho de Silva et al. (2017) utiliza o modelo clássico de mapeamento de perspectiva inversa (IPM[1]).

Neste trabalho, aborda-se conceitos da visão computacional e do SIG, a fim de extrair informações necessárias para a criação de um modelo analítico de rastreamento por câmera de objeto, bem como a comparação com o modelo numérico por interpolação.

## 1.1 Motivação

Presencia-se uma grande evolução no campo da informática e eletrônica no que tange principalmente a comunicação e processamento de dados. Em meados do século XX, iniciou-se este desenvolvimento com a terceira revolução industrial em que se destacava a eletrônica e, consequentemente, o surgimento da computação e da automação na cadeia produtiva. Atualmente, o presidente executivo do Fórum Econômico Mundial (SCHWAB, 2017) escreveu em seu livro que "A Quarta Revolução Industrial é diferente de tudo o que a humanidade já experimentou". Reitera ainda que as novas tecnologias estão integrando os mundos físicos, digitais e biológicos. Na mesma perspectiva, também denominada indústria 4.0, processos de manufaturas industriais estão se adaptando para este novo modelo de linha de produção. Assim, com o usufruto da internet das coisas (IoT[2]), inteligência artificial e *big data*, os processos tornam-se mais dinâmicos, otimizados e em tempo real. Além disso, nos próximos anos, a implementação das redes 5G potencializarão a comunicação entre diversos aparelhos com baixo consumo de energia e com velocidades cerca de 50 vezes mais rápidas do que as atuais. Deste modo, os sistemas podem se tornar mais autônomos e controlados remotamente em uma velocidade jamais vista. Além dessa perspectiva, os SIGs combinados com a visão computacional, contribuem para a melhoria da sociedade como um todo. No contexto ambiental, por exemplo, cita-se a API AirNow (AIRNOW, 2021) que disponibiliza dados de queimadas, qualidade do ar, imagens em tempo real de forma gratuita. No âmbito epidemiológico, indicadores de dispersão de vírus por meio de API

[1] do inglês *Inverse Perspective Mapping*
[2] do inglês *Internet of Things*

(OpenStreetMap contributors, 2017; HARDER, 2015) mapas de densidade, entre outras aplicações. No tocante a aplicabilidade deste projeto, incluem-se,por exemplo, a estimação da localização de afogamentos ou embarcações em alto-mar, incêndios em áreas longínquas e entre outras que envolvem determinar a posição de objetos. Considerando estes sistemas autônomos e semiautônomos bem como os SIGs, este trabalho aborda o sensoriamento geográfico associado à visão computacional, um fragmento das inúmeras possibilidades desta revolução.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Implementar dois métodos de controle, modelo de interpolação e analítico, para centralizar e determinar a localização de um objeto (i.e., rastrear) por meio de uma imagem de uma câmera e uma superfície de relevo.

### 1.2.2 Objetivos Específicos

- Desenvolver o *dataset* para a interpolação e ajuste do modelo matemático da câmera;
- Extrair informações da direção e posicionamento de pixel com o protótipo;
- Ajustar os parâmetros do modelo matemático da câmera para controlar no cenário real;
- Comparar experimentalmente o uso da interpolação e do modelo matemático para controle da câmera com os dados obtidos;
- Criar um cenário artificial;
- Adquirir dados para a criação da malha do relevo;
- Aplicar o modelo matemático analítico nos ambientes artificial e real;
- Identificar o objeto no cenário real e artificial.

## 1.3 Organização do trabalho

Esta dissertação está compreendida em cinco capítulos. O Capítulo 1 faz uma imersão sobre o tema de rastreamento e localização geográfica de objetos, apontando os objetivos gerais e específicos os quais pretendem-se alcançar. O Capítulo 2 aborda a fundamentação teórica, destacando-se os conceitos básicos da câmera e imagem, técnica e modelo matemático, e aspectos topográficos que agem sobre o posicionamento correto da câmera. O Capítulo 3 aponta a metodologia envolvida no rastreamento dos objetos e na localização geográfica. Os resultados encontram-se no Capítulo 4, que mostra como os algoritmos de rastreamento de objetos e localização geográfica de objetos se comportam

em exemplos artificiais e reais. Finalmente, o Capítulo 5 apresenta o desfecho do trabalho e as propostas de trabalhos futuros.

# 2 Fundamentação Teórica

Este capítulo apresenta os conceitos e as técnicas fundamentais que embasam o desenvolvimento deste trabalho. Inicialmente, na Seção 2.1 foram destacados os conceitos básicos de uma imagem que envolvem a influência das cores e localização em matrizes, operações de suavização da imagem e a apresentação de um algoritmo de buscas de imagens similares. Em seguida, a Seção 2.2 apresenta concepções dos parâmetros da câmera e o direcionamento por meio de vetores. Nas Seções 2.3 e 2.4 definiram-se os métodos utilizados para o rastreamento da câmera, em que o primeiro, faz alusão ao método de interpolação, o segundo, contempla o controle por sistemas de equações não lineares e, seguidamente, aborda-se o método numérico para a localização do objeto. E por fim, a Seção 2.5, na qual aborda-se alguns conceitos topográficos, em especial, sistemas de informações geográficas que realizam a obtenção de informações digitais necessárias para a extração da superfície.

## 2.1 Imagem

Conforme apresentado por Klette (2014), a imagem digital é definida pela integração de várias amostragens de dados analógicos em um espaço de domínio. Este espaço consiste em um conjunto retangular de *pixels* $(x,y,u)$ cuja combinação de $(x,y) \in \mathbb{Z}^2$ determina a localização deste conjunto e o valor de $u \in \mathbb{Z}$ é o endereço do canal de cor. O espaço de *pixel* ($\Omega$) de uma imagem é definido por

$$\Omega = \{(x,y) : 1 \leq x \leq w, 1 \leq y \leq h\} \subset \mathbb{Z}^2, \tag{1}$$

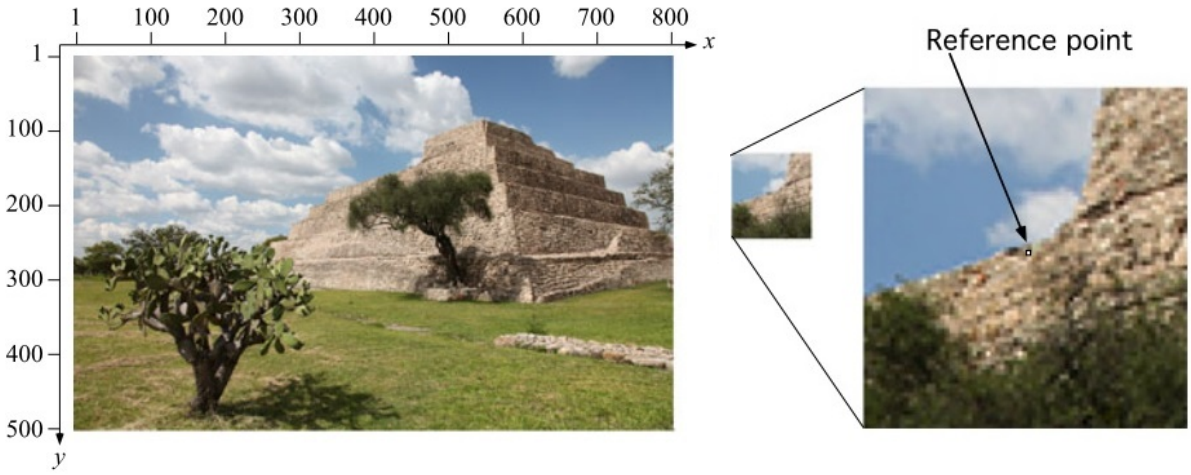
na qual, $w$ e $h$ são os limites do domínio.

Essa característica de mapear o espaço da imagem, possibilita referenciar regiões através da localização dos *pixels*. Como exemplo, a Figura 1a representa a imagem com o padrão de referência das imagens e de dimensões $w = 800$ e $h = 500$, e a Figura 1b é um recorte desta imagem com o tamanho de $w = 77$ e $h = 77$. Por fim, no centro da imagem (Figura 1b), há um ponto de referência cuja localização na imagem principal corresponde ao ponto $(453, 134)$.

Sobre as características da imagem, o espaço de cores define o tamanho da dimensão da matriz de cor e valor do *pixel* o qual é comumente encontrado com o valor de 8 bits (0 a 255 em valores decimais). Existem diversos espaços de cores, tais como: RGB[1], HSV[2]

[1] do inglês *Red, Green, Blue*
[2] do inglês *Hue, Saturation, Value*

Figura 1 – Exemplo de um espaço $\Omega$ de imagem




(a) Figura padrão (b) Ponto de referência

Fonte: Klette (2014, p. 3)

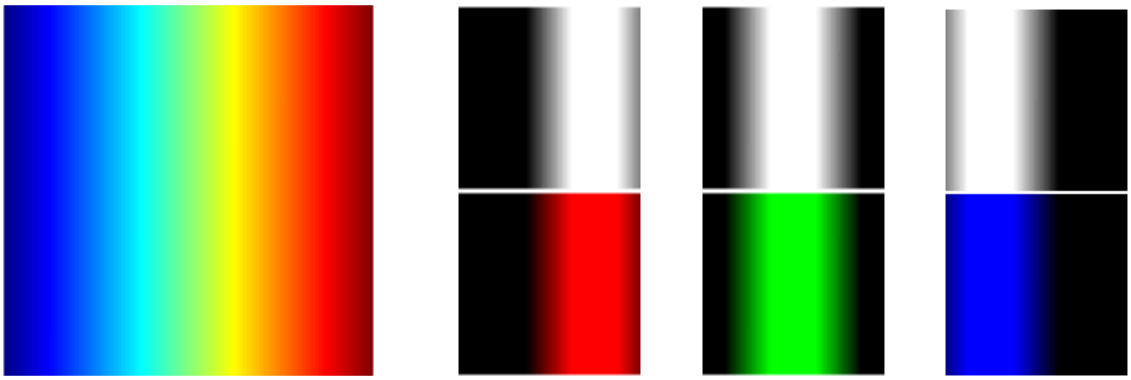
e CMYK[3] (SZELISKI, 2011).

Para efeito exemplificativo, a Figura 2 é uma imagem colorida no espaço de cor RGB que por sua vez é segmentada em três matrizes de cores de mesma dimensão de *pixels* $(w,h)$, ou seja, cada cor possui uma matriz correspondente a cada canal. No caso deste exemplo, a matriz vermelha está disposta na Figura 2b, a matriz verde na Figura 2c e a matriz azul na Figura 2d. Dessa forma, as matrizes na parte superior correspondem a cada canal de cor e, na região inferior, é representada a contribuição de cada canal acima para a formação da Figura 2a. Deste modo, é formado um dos principais espaços de cores RGB. Ressalta-se que para determinar o espaço de tons de cinza, basta encontrar a média pontual do *pixel* das matrizes de cores do espaço de cor RGB.

Em uma imagem podem ser aplicados alguns filtros a fim de arrefecer a quantidade de ruídos de alta frequência. Uma das possibilidades para suavizar uma imagem, consiste em aplicar cálculos de janela deslizante em locais pontuais que visam diminuir valores discrepantes na vizinhança da imagem em questão. Nessa perspectiva, Klette (2014), apresenta as principais técnicas de suavização, tais como: filtro de média, filtro de mediana e, especialmente para este trabalho, filtro de Gauss. Em seguida, determina-se o tamanho da janela $k$, uma matriz de dimensão $(k,k)$ denominada como *kernel* do filtro, que deslizará na imagem de acordo com a influência da vizinhança da imagem. Caso opte-se por uma maior influência dos dados vizinhos, aplica-se uma janela maior; caso contrário, uma janela menor. Nesse contexto, como o próprio nome diz, o filtro de média, realiza a operação de janela deslizante da média dos valores do seu *kernel* para determinar o valor do ponto central.

[3] do inglês *Cyan, Magenta, Yellow, White*

Figura 2 – Espaço de cor RGB



(a) Figura Padrão (b) Canal vermelho (c) Canal verde (d) Canal azul

Fonte: Adaptado de Matlab (2020b)

Figura 3 – Exemplo de aplicação do filtro de Gauss



(a) Figura Padrão (b) Figura com o filtro de Gauss

Fonte: Matlab (2020a)

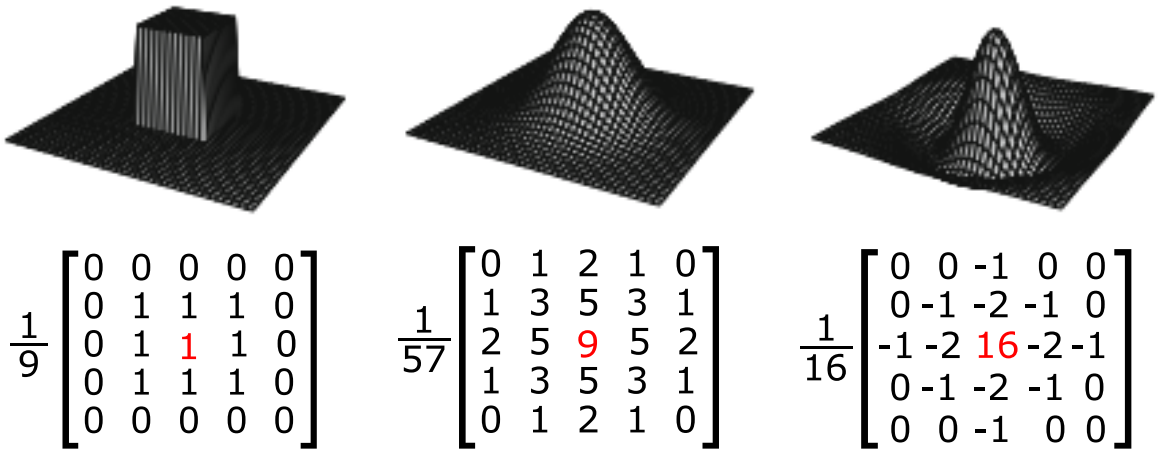
Analogamente, os demais filtros realizam a mesma operação com as suas características.

Ainda de acordo com Klette (2014), filtro de Gauss (Figura 3) é uma convolução local com o filtro de *kernel*. Após determinar o valor para $k$ e $\sigma$, define-se as amostras da função de duas dimensões de Gauss da seguinte forma:

$$G_{\sigma,\mu_x,\mu_y(x,y)} = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{(x-\mu_1)^2+(x-\mu_x)^2}{2\sigma^2}\right)}, \tag{2}$$

em que ($\mu$x, $\mu$y) combina os valores esperados para os componentes x e y, $\sigma$ é o desvio padrão. De forma exemplificativa, a Figura 4 demonstra três tipos de *kernel* com $k = (5{,}5)$:

Figura 4 – Exemplo de *kernel* dos filtros da forma




(a) *Box filter* (b) *Gaussian filter* (c) *Difference filters*

Fonte: Burger e Burge (2016, p. 99)

*Box filter*, *Gaussian filter* e *Difference filters*.

### 2.1.1 Algoritmo *matchTemplate*

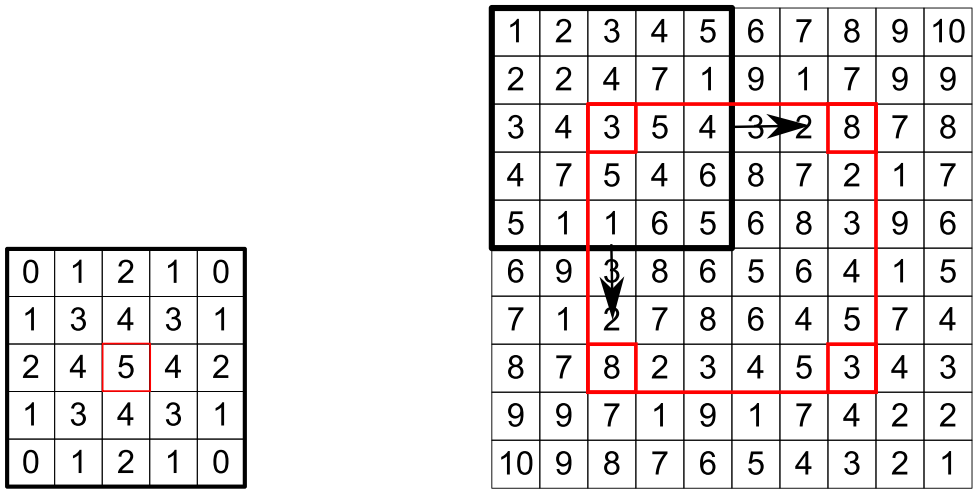
O algoritmo *matchTemplate* é uma técnica no processamento digital de imagens para a determinação da similaridade entre uma instância de uma máscara ($\mathbf{T}$) dentro de uma imagem ($\mathbf{I}$) (ITSEEZ, 2014).

Dessa forma, o algoritmo realiza uma operação de janela deslizante utilizando a instância de máscara sobre uma imagem, comparando estes dois grupos - *template* e o recorte da imagem da imagem - com cada ponto de *pixel* factível. Portanto, dado uma instância de máscara $\Omega_T(w,h)$ e uma imagem $\Omega_I(W,H)$, é feito a operação da medida de desempenho adotada para determinar um valor de similaridade, iniciando em $\Omega_I(W - \frac{w}{2}, H - \frac{h}{2})$ até $\Omega_I(W - \frac{w}{2} + 1, H - \frac{h}{2} + 1)$. A Figura 5 exemplifica esse procedimento no qual a figura do lado esquerdo representa a instância de máscara, o lado direito a operação sendo feita e, em vermelho, o espaço gerado após o algoritmo. Assim, resulta em uma saída de $\Omega_R(W - w + 1, H - h + 1)$ e o valor dessa medida do casamento de máscara $R(x,y)$. Ressalta-se que o maior valor de $R(x,y)$ desta operação, exceto a soma das diferenças ao quadrado (SQDIFF[4]), é considerado o ponto central da imagem em que mais se assemelha da instância de máscara (ITSEEZ, 2014). Ainda de acordo com Itseez (2014), utiliza-se, comumente, as medidas de desempenhos SQDIFF, correlação cruzada (CCORR[5]) e correlação cruzada de desvio médio (CCOEFF[6]).

Para a medida de desempenho da soma das diferenças ao quadrado (SQDIFF), a

[4] do inglês *Square Difference*
[5] do inglês *Cross Correlation*
[6] do inglês *Correlation Coefficient*

Figura 5 – Algoritmo *matchTemplate*




(a) Template (b) Imagem

melhor similaridade é obtida quando o resultado for igual a zero, de acordo com a equação a seguir:

$$R_{SQDIFF}(x,y) = \sum_{x',y'} \left(\mathbf{T}(x',y') - \mathbf{I}\left(x+x',y+y'\right)\right)^2. \tag{3}$$

Já no método da correlação cruzada (CCORR), utiliza a multiplicação para determinar a similaridade de acordo com:

$$R_{CCORR}(x,y) = \sum_{x',y'} \left(\mathbf{T}(x',y') \cdot \mathbf{I}\left(x+x',y+y'\right)\right)^2. \tag{4}$$

Assim, caso o valor seja relativamente alto, significa que existe uma similaridade alta; caso contrário, similaridade baixa.

Por outro lado, a correlação cruzada de desvio médio (CCOEFF), indicada por:

$$R_{CCOEFF}(x,y) = \sum_{x',y'} \left(\mathbf{T}'(x',y') \cdot \mathbf{I}\left(x+x',y+y'\right)\right)^2, \tag{5}$$

é similar ao método CCORR, entretanto, difere-se na correspondência entre o modelo em relação a sua média. Dessa forma, tanto o *template* quanto a imagem, subtraem a média de:

$$\mathbf{T}'(x',y') = \mathbf{T}(x',y') - \frac{1}{w \cdot h} \cdot \sum_{x'',y''} \mathbf{T}(x'',y'') \tag{6}$$

e

$$\mathbf{I}'(x',y') = \mathbf{I}(x+x',y+y') - \frac{1}{w \cdot h} \cdot \sum_{x'',y''} \mathbf{I}(x+x'',y+y''). \tag{7}$$

Essas medidas também possuem suas respectivas formas normalizadas, apresentadas na Tabela 1. Essa forma é útil para contribuir para a redução dos efeitos inerentes à variação da iluminação do meio ambiente e entre outros ruídos.

Tabela 1 – Medidas normalizadas do algoritmo *matchTemplate*

| Método Normalizado | Equação |
|---|---|
| SQDIFF | $R_{SQDIFF-Normed}(x,y) = \frac{R_{SQDIFF}(x,y)}{Z(x,y)}$ |
| CCORR | $R_{CCORR-Normed}(x,y) = \frac{R_{CCORR}(x,y)}{Z(x,y)}$ |
| CCOEFF | $R_{CCOEFF-Normed}(x,y) = \frac{R_{CCOEFF}(x,y)}{Z(x,y)}$ |

A função $Z(x,y)$ indicada na Tabela 1, corresponde a:

$$Z(x,y) = \sqrt{\sum_{x',y'} \left(\mathbf{T}(x',y')\right)^2 \cdot \sum_{x',y'} \left(\mathbf{I}(x+x',y+y')\right)^2} \tag{8}$$

que é igual para todos os métodos de normalização descritos nessa seção.

## 2.2 Câmera

A câmera possui diversas características importantes para a sua escolha. Dentre elas, destacam-se: distância focal, campo de visão (FoV[7]), tamanho da imagem e resolução da imagem. A primeira, distância focal, advém da lei da física na área da óptica de lentes cuja fórmula está disposta em:

$$\frac{1}{z_0} + \frac{1}{z_i} = \frac{1}{f}, \tag{9}$$

na qual,

- $z_0$ distância da lente até o objeto;
- $z_i$ distância da lente até a imagem;
- $f$ distância focal.

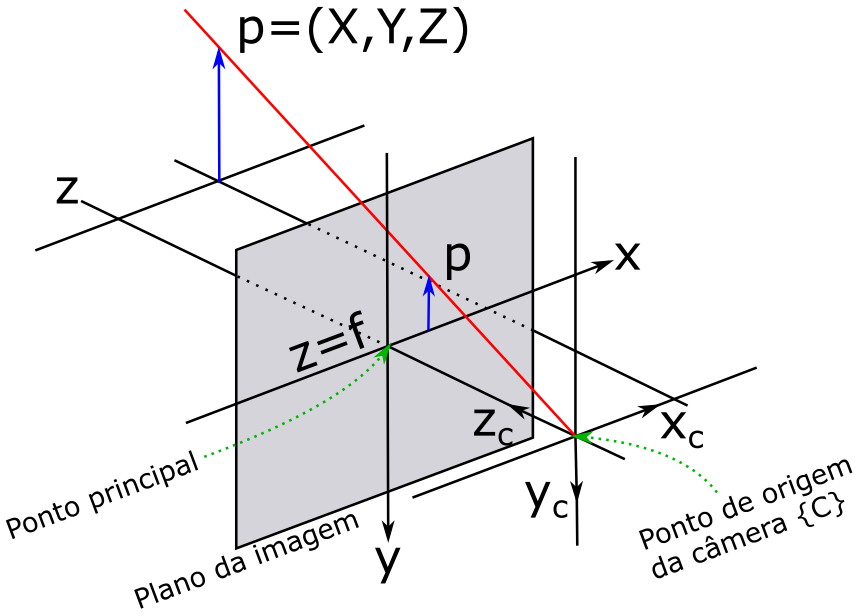
Ressalta-se que a distância focal está diretamente associada a ampliação da imagem sem que haja perda do tamanho da imagem.

Segundo Burger e Burge (2016, p. 8), o tamanho de imagem é determinado pela quantidade de colunas e linhas da matriz da imagem, descritos na Seção 2.1 como $w$ e $h$, respectivamente. Já a resolução é determinada pelas dimensões espaciais uma área quadrada especificada, por exemplo, *pixels* por polegadas (dpi[8]).

De acordo com Corke (2011, p. 252), é comum utilizar o modelo central de geração de imagem em perspectiva, ilustrada na Figura 6. O ponto $C$ é o centro da formação da câmera e o local para onde converge as retas. A região do ponto $z = f$ é o plano onde a imagem foi projetada. O ponto $p = (X,Y,Z)$ é o local de um objeto no ambiente e, por sua vez, é projetado sobre o plano $z = f$ do ponto $p = (x,y)$ na imagem.

[7] do inglês *Field of Vision*
[8] do inglês *Dots per inch*

Figura 6 – Parâmetros da câmera.




Fonte: Adaptado de Corke (2011, p. 252)

Ainda de acordo com Corke (2011, p. 252), para obter uma imagem homogênea, basta aplicar a semelhança de triângulos. Assim, aplica-se:

$$x = f\frac{X}{Z}, y = f\frac{Y}{Z}, \tag{10}$$

em que,

- $x$ e $y$ coordenadas em *pixel* da imagem;
- $X$, $Y$ e $Z$ coordenadas no espaço de um ponto genérico;
- $f$ distância focal.

### 2.2.1 Modelo analítico

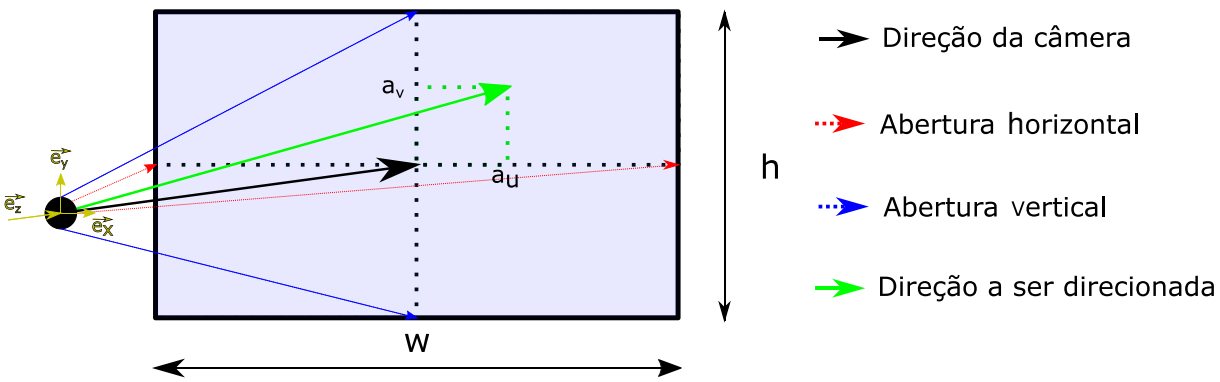
O modelo analítico da câmera descreve a relação da direção do objeto no ambiente com a sua projeção sobre um *pixel* no plano de imagem. Diante disso, o modelo analítico consiste no direcionamento direto dos cossenos diretores dado por:

$$\vec{d}_a \frac{1}{\| \vec{d}_a \|} = (\cos\theta_u, \cos\theta_v, \cos\theta_z)\,. \tag{11}$$

Dessa forma, a Figura 7 demonstra alguns parâmetros utilizados para calcular a direção do posicionamento. O vetor em preto é a direção atual da câmera. O plano em azul é a imagem com os parâmetros de largura $(w)$ e altura $(h)$, ou seja, o espaço de imagem descritos na Seção 2.1. Salienta-se que o ângulo formado entre os vetores em vermelho consiste na abertura da câmera $(\alpha)$. O vetor em verde $(\vec{d}_a)$ representa a direção em que deseja deslocar que origina consequentemente dois parâmetros: $a_u$ destinado a posição

em *pixel* na horizontal e $a_v$ a posição em *pixel* vertical. Os vetores na cor amarela são a base canônica para os cossenos diretores.

Figura 7 – Ilustração dos vetores do modelo analítico.




Diante disso, pode-se encontrar a primeira direção através do rumo do vetor $\vec{d_a}$ em relação a base $\vec{e_x}$, obtendo:

$$\cos\theta_u = \frac{\vec{d_a} \cdot \vec{e_x}}{\| \vec{d_a} \|} = \frac{a_u - \frac{(w-1)}{2}}{\| \vec{d_a} \|}. \tag{12}$$

Analogamente, a segunda direção é obtida em relação a base $\vec{e_y}$, adquirindo:

$$\cos\theta_v = \frac{\vec{d_a} \cdot \vec{e_y}}{\| \vec{d_a} \|} = \frac{a_v - \frac{(h-1)}{2}}{\| \vec{d_a} \|}. \tag{13}$$

Por outro lado, a terceira direção é obtida, diferentemente das outras, com os parâmetros do ângulo de abertura da câmera ($\alpha$), descrita na Seção 2.2, e com a largura da imagem ($w$), em relação a base $\vec{e_z}$,obtêm-se:

$$d = \frac{\frac{w}{2}}{\tan\frac{\alpha}{2}}. \tag{14}$$

Assim, para adquirir a forma do cosseno diretor, faz-se necessário um algebrismo e obtêm-se:

$$\cos\theta_z = \| \vec{d_a} \| = \frac{\frac{w}{2\cdot\tan\frac{\alpha}{2}}}{\| \vec{d_a} \|}. \tag{15}$$

Após determinar todas as direções, determina-se a equação que referencia a direção a ser deslocada. Dessa forma, obtém-se:

$$\vec{\mathbf{d_a}} = \left(a_u - \frac{w-1}{2}\right)\vec{u} + \left(a_v - \frac{h-1}{2}\right)\vec{v} + \left(\frac{w}{2\tan(\alpha/2)}\right)\vec{d}. \tag{16}$$

Por fim, realiza-se o produto vetorial normalizado entre a direção $\vec{z}$ e $\vec{d}$ para obter a direção $\vec{u}$, identificado em:

$$\vec{u} = (\vec{d} \times \vec{z}) \frac{1}{\| \vec{d} \times \vec{z} \|}. \tag{17}$$

E determinar a perpendicularidade dos vetores por meio do produto vetorial entre a direção $\vec{d}$ e $\vec{u}$, apontado em:

$$\vec{v} = \vec{u} \times \vec{d}. \tag{18}$$

Dessa forma, determina-se a perpendicularidade entre cada combinação dos vetores anteriores.

## 2.3 Interpolação

De acordo com Ruggiero e Lopes (1996, p. 212), interpolar uma função $f(x)$ significa aproximá-la por uma outra função $g(x)$, definida entre uma classe de funções e que acate algumas propriedades. Então, utiliza-se a função $g(x)$ para substituir a função requirida. De forma simples, interpolação consiste em utilizar informações do sistema já adquiridas para determinar valores, dentro do intervalo de pontos estimado, de modo a aproximar pontos factíveis não amostrados e otimizados. Diante disso, neste trabalho, utiliza-se a interpolação Baricêntrica que consiste no uso de:

$$r(x) = \frac{\sum_{i=0}^{n} \frac{w_i}{x - x_i} f\left(x_i\right)}{\sum_{i=0}^{n} \frac{w_i}{x - x_i}} \tag{19}$$

(BERRUT; TREFETHEN, 2004, p. 555). Assim, determinando os pesos $w_0, w_1, \cdots, w_n$, é possível encontrar o valor de $r(x)$.

## 2.4 Sistema de equações não lineares

Um sistema de equações diferenciais que não é linear é denominado de não linear (ZILL, 2003; BOYCE; DIPRIMA, 2002). De acordo com Boyce e Diprima (2002, p. 377), alguns sistemas de equações não lineares não são passíveis de solução analítica e podem utilizar como alternativa o uso de métodos numéricos para a resolução desta.

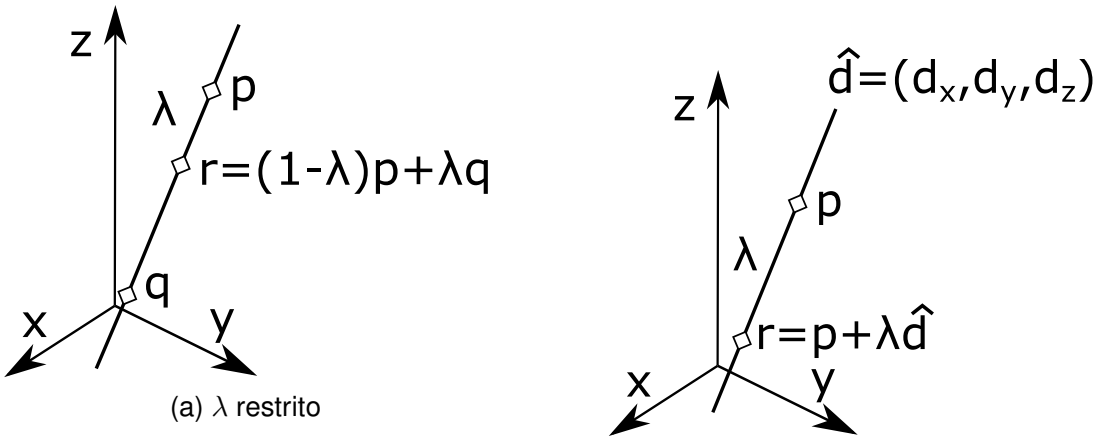
Segundo Szeliski (2011, p. 34), as coordenadas em $3\mathrm{D}$ de uma semirreta podem ser escritas como coordenadas não homogêneas $\boldsymbol{x} = (x, y, z) \in \mathbb{R}^3$, ilustradas pela Figura 8a e representadas por meio:

$$\boldsymbol{r} = (1 - \lambda)\boldsymbol{p} + \lambda \boldsymbol{q}, \tag{20}$$

cujos coeficientes são:

- $\boldsymbol{r}$ posição encontrada no espaço;
- $\boldsymbol{p}$ posição inicial da semirreta no espaço;

Figura 8 – Ilustração dos vetores do modelo analítico.




Fonte: Szeliski (2011)

- $\boldsymbol{q}$ posição final da semirreta no espaço;
- $\lambda$ é um valor restrito no intervalo $[0, 1]$.

Vale ressaltar que existe um caso especial onde há um ponto inicial fixo e um ponto final que pode tender ao infinito, apresentado na Figura 8b. Diante disso, reescreve-se a equação linear não homogênea de $3\mathrm{D}$ em:

$$\boldsymbol{r} = \boldsymbol{p} + \lambda\hat{\boldsymbol{d}}, \tag{21}$$

na qual,

- $\boldsymbol{r}$ posição encontrada no espaço;
- $\boldsymbol{p}$ posição inicial da semirreta no espaço;
- $\lambda$ é o comprimento do segmento de reta e um valor irrestrito;
- $\hat{d}$ é a direção da reta.

Destaca-se que existem problemas que tornam necessário determinar as raízes de uma função, como o ponto ótimo do custo marginal da fabricação de produtos industriais. Diante disso, existem diversas técnicas que visam determinar o ponto mínimo de funções mais complexas do que aquela exemplificada anteriormente. Assim, de acordo com Ruggiero e Lopes (1996), pode-se utilizar o método posição falsa, ponto fixo, Newton-Rapson, secante e, especialmente para este trabalho, bisseção.

Vale salientar que existe a necessidade de segmentar o intervalo para determinar a raiz, pois a curva não será exclusivamente linear. Caso não seja feita essa segmentação, pode não encontrar uma raiz ou uma raiz incorreta. Essa raiz incorreta, neste trabalho, corresponde uma área que não faz parte do campo de visão. Pela simplicidade do algoritmo e incluir a segmentação em subintervalos, utiliza-se o método da bisseção (Algoritmo 1) que consiste em realizar saltos sucessivos com o tamanho da metade do intervalo amostrado anteriormente para o sentido da raiz. Desse modo, continua-se o procedimento até que encontre o valor tolerado como zero ou um valor fixo de interações, caso o algoritmo não

convirja.

**Algoritmo 1:** Método da Bisseção

**Entrada:**
$a_0$ - Ponto inicial do intervalo
$b_0$ - Ponto final do intervalo
$e_s$ - Erro tolerado
$i_{max}$ - Número máximo de interações
$f(x)$ - Função
**Saída:**
$i$ - Interações
$e_a$ - Erro relativo porcentual
$c$ - Ponto médio encontrado
$B_s$ - Resultado final
**Início:**
$i \leftarrow 0$
$e_a \leftarrow 100$
$f_a = f(a_0)$
**while** $(e_a > e_s) \wedge (i < i_{max})$ **do**
  $c_{rold} = c$
  $c \leftarrow \frac{b-a}{2}$
  $f_c = f(c)$
  $i \leftarrow i + 1$
  **if** $c \neq 0$ **then**
    $e_a = ||\frac{c-c_{rold}}{c}||100$
  **end**
  $teste = f_c f_a$
  **if** $teste < 0$ **then**
    $b_0 \leftarrow c$
  **end**
  **else**
    **if** $teste > 0$ **then**
      $a_0 \leftarrow c$
      $f_a \leftarrow f_c$
    **end**
    **else**
      $e_a = 0$
    **end**
  **end**
**end**
$B_s = c$

Fonte: Adaptado de Chapra e Canale (2009, p. 107)

Para estimar o número máximo de interações desse método basta utilizar a seguinte

equação:

$$k > \frac{\log\left(b_0 - a_0\right) - \log(\varepsilon)}{\log(2)}, \tag{22}$$

em que,

- $k$ número de interações;
- $a_0$ início do intervalo fechado;
- $b_0$ final do intervalo fechado;
- $\varepsilon$ precisão adotada.

Por outro lado, para algumas funções não há raiz ou não é possível determinar se, realmente, ela existe. Diante disso, o Teorema 1 afirma que, caso a multiplicação de dois pontos em um intervalo fechado for menor que zero, existe pelo menos uma raiz dentro deste intervalo.

**Teorema 1 (Bolzano-Cauchy)** *Seja $f$ é continua em $[a, b]$. Se $f(a) \cdot f(b) < 0$, então existe $x \in (a, b)$ tal que $f(x) = 0$.*

Assim, utiliza-se o Teorema 1 para determinar a existência de uma raiz no intervalo estimado e diminuir a quantidade de interações. Deste modo, para convergir mais rapidamente, dividiu-se o intervalo em subgrupos e em seguida, utiliza-se o método da bisseção para encontrar a raiz, caso exista.

## 2.5 Topografia

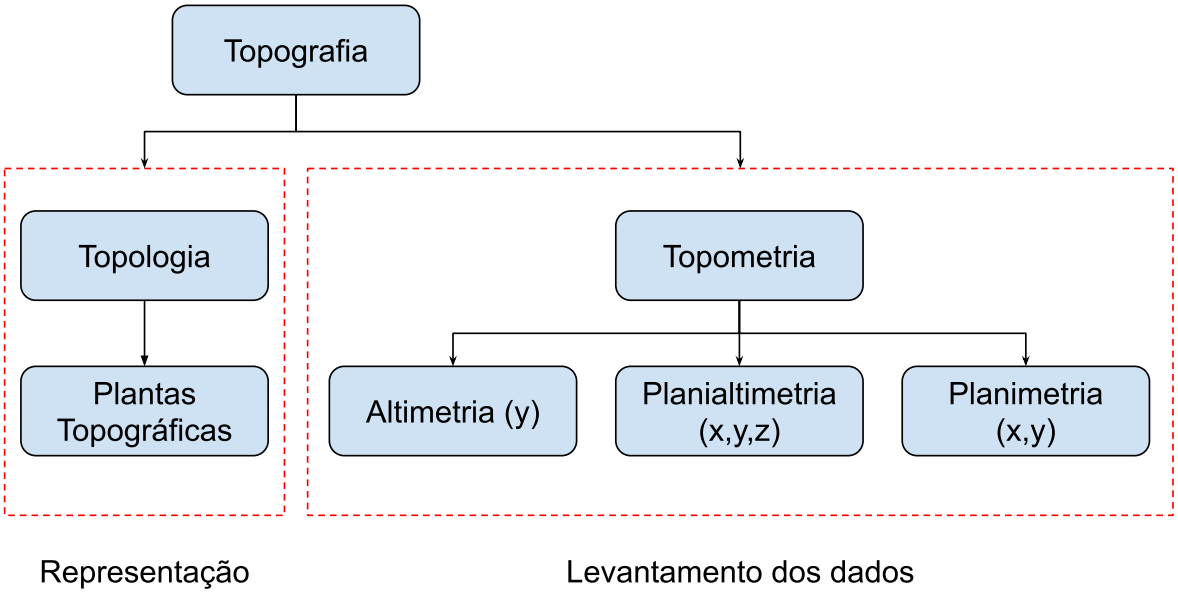
A topografia é uma ciência geográfica que aborda a discriminação e interpretação de dados ligados à superfície terrestre, bem como as suas características de formação e caracterização do relevo. Davis, Sarasua e Mccormac (2016) definem a topografia como:

> "... é uma ciência que trata da determinação das dimensões e contornos (ou características tridimensionais) da superfície física da Terra, através da medição de distâncias, direções e altitudes. A topografia também inclui a locação de linhas e malhas necessárias para a construção de prédios, estradas, barragens e outras estruturas. Além dessas medições de campo, a topografia compreende o cálculo de áreas, volumes e outras quantidades, assim como a preparação dos respectivos mapas e diagramas".

A topografia é dividida, principalmente, em topologia que retrata a representação gráfica e topometria, que se refere ao levantamento de dados, como pode ser observado na Figura 9.

Na engenharia, a topologia é uma ciência destinada a representação gráfica da superfície terrestre, bem como as suas características de formação e caracterização do relevo, por exemplo, a curva de nível. Dessa maneira, o uso dela visa facilitar a representação topográfica a qual permite o controle dos possíveis erros relativos à aquisição dos dados, além de diminuir a quantidade de pontos necessários para a determinação do terreno.

Figura 9 – Divisão da topografia




Na topometria utiliza-se diversos métodos e procedimentos para obter medidas de uma determinada região terrestre, podendo ser dividida em: planimetria, altimetria e planialtimetria, sendo esta última a combinação das outras duas. Dessa forma, a planimetria é a parte da topografia que usufrui de métodos e procedimento que visam estabelecer medidas obtidas de uma determinada região em um plano horizontal $(x,y)$ (COMASTRI, 1992). Segundo este mesmo autor, quando se trata de levantamento de áreas de terrenos relativamente pequenas, não há necessidade de levar em consideração a curvatura terrestre, podendo-se então, considerá-las planas sem erros apreciáveis. Por outro lado, as superfícies são curvilíneas e os mapas que as representam são planos. Assim a minimização do erro devido a desconsideração da curvatura pode ser feita através de equações matemáticas que convertem os dados do sistema geodésico (latitude e longitude) sobre uma elipsoide, ou uma esfera ou esferoide para uma posição representativa sobre uma superfície plana (DAVIS; SARASUA; MCCORMAC, 2016, p. 281). Já a altimetria, trata-se dos métodos e instrumentos empregados no estudo e representação de medidas no plano vertical, ou seja, relevo do solo. Ainda, define-se altimetria como o comprimento perpendicular de um ponto até ao plano horizontal que se denomina de superfície de nível de comparação (COMASTRI; TULER, 1987).

### 2.5.1 Sistemas de Informações Geográficas

Na literatura existem diversas definições para sistemas de informações geográficas (SIG), sendo cada uma delas baseada de acordo com o usuário e ou o uso. De forma geral, Davis, Sarasua e Mccormac (2016, p.262) definem SIG como um conjunto de tarefas voltado aos dados geográficos que inclui desde a coleta dos dados até o uso das informações para a tomada de decisões. De forma mais detalhada, Fitz (2008) define SIG como conjunto de programas computacionais que incluem os dados, equipamentos e pessoas com objetivo de coletar, armazenar, recuperar, manipular, visualizar e analisar dados espacialmente referenciados a um sistema de coordenadas conhecido. Já para Burrough, McDonnell e Lloyd (1986), SIG representam um conjunto de procedimentos computacionais que viabiliza a análise por meio de cálculos e álgebra sobre uma base de dados preexistente.

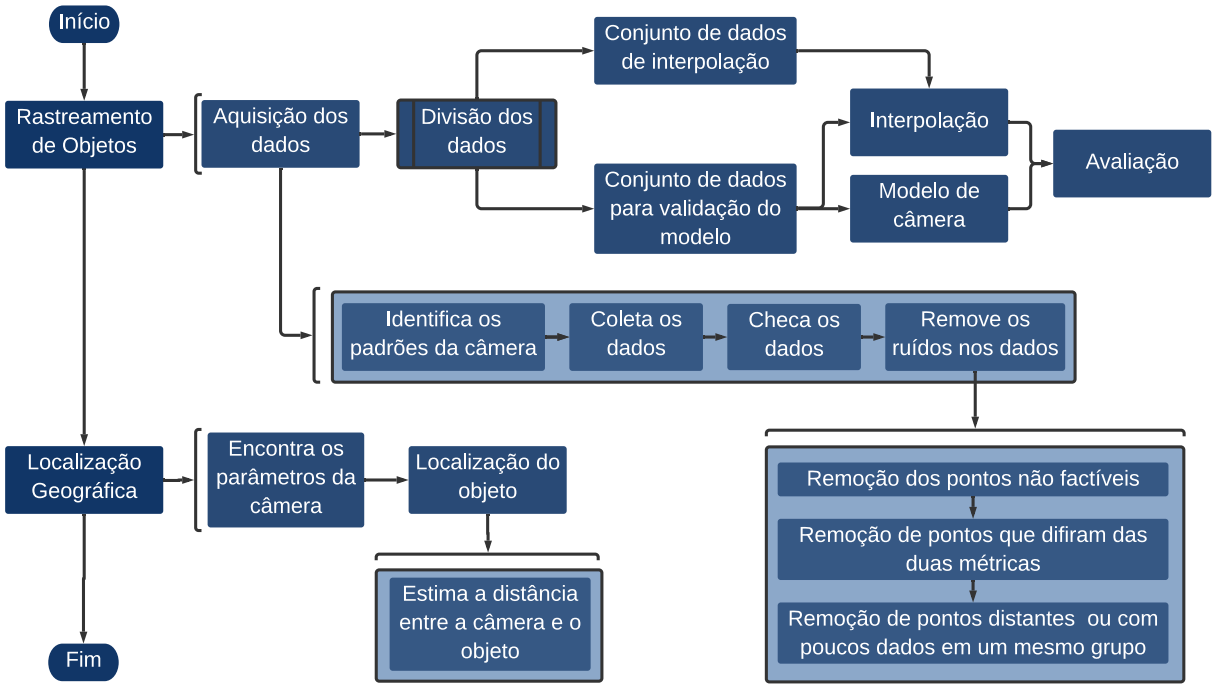
Desse modo, permite a elaboração de mapas ou elementos que sejam mais adequados para análise e tratamentos dos dados geográficos. Neste trabalho, os SIGs são utilizados para extrair as informações da superfícies, ou seja, pontos de distância em metros e a elevação do terreno. Além disso, o uso recursos de imagens para a referenciação do ambiente.

# 3 Rastreamento e Localização Geográfica de Objetos

Este capítulo tem como propósito descrever a metodologia abordada neste trabalho, destacando-se o rastreamento e a localização geográfica de objetos. A etapa de rastreamento contempla a aquisição dos dados da câmera junto a extração de características de correção de posicionamento e remoção de ruídos. Em seguida, incluem-se o controle por interpolação e modelo analítico da câmera, bem como o método utilizado para determinar a acurácia do projeto junto a posição em pixel na imagem, indicado principalmente na Seção 2.4. Na fase de localização são estabelecidos os métodos matemáticos para determinar a localização do objeto por meio da direção da câmera junto a posição em *pixel* na imagem. Nesta fase também ocorre a extração da superfície por meio do SIG, e a conversão da posição geodésica em metros.

Em suma, o capítulo está organizado de acordo com a Figura 10. As seções estão dispostas na coluna à esquerda e as subseções, no mesmo tom de cor das seções, à direita da seção correspondente. As cores mais escuras são as seções abordadas diretamente no trabalho e as claras, os subtemas que estão na forma de conteúdo no texto.

Figura 10 – Fluxograma da metodologia abordada no trabalho



## 3.1 Rastreamento de Objetos

Esta seção apresenta a metodologia utilizada no trabalho para a coleta dos dados a fim de mensurar a acurácia dos modelos propostos, além de ajustar a sincronização do modelo com a câmera.

### 3.1.1 Aquisição de imagens

Inicialmente, são feitas algumas aquisições aleatórias com a finalidade de identificar o padrão de imagem da câmera e os limites de movimentação. Assim, determina-se a região de interesse consistindo em faixas de imagem cujo teor fosse pertinente, ou seja, eliminando regiões como paredes, chão do andar ou andares do prédio da instalação da câmera.

Após delimitar a região de interesse, realiza-se a varredura a fim de criar uma malha de dados que relaciona o quadro de imagem com a direção correspondente, descrita de forma simplificada no Algoritmo 2. Neste contexto, por limitação do computador que coletou os dados, os quadros de imagem são salvos em grupos de tons de cinza, a fim de diminuir o espaço armazenado e possíveis manipulações. Dessa forma, evita-se a perda de velocidade durante o processo de captura dos dados oriundos do envio de informação da memória RAM[1] para o disco rígido quando se exede o limite operacional de memória disponível no sistema. Desse modo, a etapa de aquisição é separada em laços de repetição: o laço interno referente ao ângulo de azimute e o externo à elevação. Dentro do laço interno, é salvo o quadro de imagem e a direção correspondente a ele na sua respectiva matriz de dados e incrementa-se o ângulo para manter o laço de repetição. Assim, ao término do laço interno, salva-se o conjunto de dados criado até o momento e o ângulo do laço externo e incrementa-se o ângulo do laço externo. Então, o algoritmo utiliza a mesma quantidade de memória RAM e mantém a velocidade de aquisição.

Ao final desta captura, cria-se um *script* para verificar de forma semi-manual se há algum erro inesperado durante a captura. Esta verificação semi-manual consiste na criação de um *script* que exibe um trecho dos dados do *dataset* de vídeo, denominado mini *dataset* . Este refere-se ao ciclo completo do laço de repetição interno indicado no Algoritmo 2, com amostragem de 1 segundo por imagem. Cada quadro de imagem possui uma numeração única que difere de qualquer outro dado. Assim, caso haja algum erro, basta indicar o respectivo número do quadro de imagem, que com o auxílio dos dados das direções, é determinado o ponto em que a câmera deve ser posicionada para o reenvio da nova imagem correspondente àquela direção. Corrigido o erro, outro mini *dataset* é examinado até que todo conjunto de dados seja visualizado. Estes erros podem ser, por exemplo, pássaros sobrevoando em frente a câmera, carros ou pessoas movimentando na área durante a coleta da imagem ou quaisquer interferências que possam prejudicar

[1] do Inglês *Random Access Memory* - Memória de Acesso Aleatório

**Algoritmo 2:** Cria o conjunto de dados com os dados da câmera

**Entrada:**
$\theta_{1,i}$ - Limite inicial de Azimute
$\theta_{1,f}$ - Limite final de Elevação
$\Delta\theta_1$ - Incremento no ângulo $\theta_1$
$\theta_{2,i}$ - Limite inicial de Elevação
$\theta_{2,f}$ - Limite final de Elevação
$\Delta\theta_2$ - Incremento no ângulo $\theta_2$
**Saída:**
$V_{(W\times H\times m\times n)}$ - Matriz com imagens capturadas
$D_{(1\times m\times n)}$ - Matriz com as direções de azimute das imagens capturadas
$D_{(2\times m\times n)}$ - Matriz com as direções de elevação das imagens capturadas
**Início:**
$ct_b \leftarrow 1$
**for** $\theta_1 = \theta_{1,i}, \theta_{1,i} + \Delta\theta_1, \cdots, \theta_{1,f}$ **do**
  $ct_a \leftarrow 1$
  **for** $\theta_2 = \theta_{2,i}, \theta_{2,i} + \Delta\theta_2, \cdots, \theta_{2,f}$ **do**
    $D_{((1,2),ct,ct_b)} \leftarrow DirecaoCamera()$
    $V_{(W\times H\times m\times n)} \leftarrow ImagemCamera()$
    $incrementaDirecaoCamera(0,\Delta\theta_2)$
    $ct_a \leftarrow ct_a + 1$
  **end**
  $incrementaDirecaoCamera(\Delta\theta_1, -(\theta_{2,f} - \theta_{2,i}))$
  $ct_b \leftarrow ct_b + 1$
**end**

a extração de dados posteriormente. Após esta etapa, utiliza-se outro computador com memória RAM suficiente para agrupar todos os dados e continuar o processamento.

Após a coleta dos dados é necessário extrair informações para servir como dados de validação e interpolação. Os dados por si só, não possuem características que possam ser usadas para determinar um modelo analítico, pois há apenas imagens e as direções atuais. Por outro lado, existem pontos diferentes do centro da imagem que não possuem a direção com apenas estes dados. Ressalta-se que o ponto central é a própria direção do quadro de imagem. Diante disso, esses pontos diferentes dos pontos centrais, podem ser determinados por outros quadros de imagens salvos. De acordo com esse pressuposto, utiliza-se o algoritmo de casamento de padrões (MT[2]), descrito na Subseção 2.1.1 e representado no Algoritmo 2, para encontrar instâncias de máscara dentro de uma outra imagem, i.e., encontrar a posição de *pixel* dentro de uma imagem de referência. Utilizando a função *filtroGaussiano* aplica-se na imagem de entrada ($I$) um filtro Gaussiano com $k = (5,5)$ que, por sua vez, retorna a imagem ($f$) suavizada e uma quantidade menor de ruídos, conforme descrito na Seção 2.1. Já por meio da função *funcAdaptadaMT*, descrita na Subseção 2.1.1, utiliza-se o algoritmo MT com as entradas da imagem ($f$) e a

[2] do Inglês *matchTemplate*

máscara ($t$), além da saída ($DS$) com a localização em *pixel* ($DS_{((1,2,3,4),ct_c)}$) e o resultado das duas medidas do MT ($DS_{((5,6),ct_c)}$) daquele instante ($ct_c$). Portanto, a localização do *pixel* dessas medidas são: $P_{x-SQDIFF}$, $P_{y-SQDIFF}$, $P_{x-CCORR}$ e $P_{y-CCORR}$; em seguida, os resultados são: $R_{SQDIFF}$ e $R_{CCORR}$. Então, determina-se a posição onde ocorre o casamento de máscara e o valor da maior similaridade nesses valores. Dessa forma, ao localizar essa instância, consequentemente, determina-se a direção em que se deve deslocar para direcionar a câmera para aquela instância. Assim, ao varrer todos os quadros de imagens gravados, determina-se todas as possibilidades de movimentação da câmera. Vale ressaltar que este algoritmo retorna valores dessas medidas e a posição em que ocorre o valor máximo mesmo que não corresponda com a máscara escolhida. Diante disso, é necessário remover estes ruídos dos dados a fim de obter dados concisos e precisos os quais serão descritos a seguir. Após a etapa de coleta dos dados, inicia-se o procedimento para refinar os dados dos ruídos.

**Algoritmo 3:** Extrair dados da câmera

**Entrada:**
$V_{(W\times H\times m\times n)}$ - Matriz com imagens capturadas
$D_{(1:2\times m\times n)}$ - Matriz com as direções das imagens capturadas
$c_{box}$ - caixa determinada para máscara do MT
**Saída:**
$DS_{(1:4\times m\cdot n)}$ - Localização central do pixel ($P_x$ e $P_y$) na imagem em dois métodos diferentes
$DS_{(5:6\times m\cdot n)}$ - Valor de exatidão nos dois métodos
$DS_{(7:8\times m\cdot n)}$ - Direção de entrada da câmera
$DS_{(9:10\times m\cdot n)}$ - Direção de saída da câmera
**Início:** $c_{box} \leftarrow 120$
$ct_c \leftarrow 1$
**for** $ct_{a1} = 1, ct_{a1}+1, \cdots, m$ **do**
  **for** $ct_{a2} = 1, ct_{a2}+1, \cdots, n$ **do**
    $f \leftarrow filtroGaussiano(V_{(:,:,ct_{a1},ct_{a2})})$
    $t \leftarrow f[\frac{W-c_{box}}{2} : \frac{W+c_{box}}{2}, \frac{H-c_{box}}{2} : \frac{H+c_{box}}{2}]$
    **for** $ct_{b1} = 1, ct_{b1}+1, \cdots, m$ **do**
      **for** $ct_{b2} = 1, ct_{b2}+1, \cdots, n$ **do**
        $f \leftarrow filtroGaussiano(V_{(:,:,ct_{b1},ct_{b2})})$
        $DS_{(1:6,ct_c)} \leftarrow funcAdaptadaMT(f,t)$
        $DS_{(7:8,ct_c)} \leftarrow D_{(:,ct_{b1},ct_{b2})}$
        $DS_{(9:10,ct_c)} \leftarrow D_{(:,ct_{a1},ct_{a2})}$
        $ct_c \leftarrow ct_c + 1$
      **end**
    **end**
  **end**
**end**

O primeiro estágio consiste em eliminar imagens que não contêm o padrão de

referência de centro (i.e., está fora do campo de visão da câmera). Essa remoção deve ser feita pois não é possível calcular o deslocamento para centralizar o padrão de referência na imagem da câmera. Assim, mesmo que não seja possível identificar o objeto por se encontrar fora do campo de visão, encontra-se como o objeto da máscara devido à similaridade da imagem.

No segundo estágio, os dados ainda detêm alguns erros devido ao resultado do MT, pois existem diversas imagens semelhantes dentro do FoV[3]. Como exemplo, a vegetação e o parapeito do local em que estava localizada a câmera. Desse modo, utilizam-se dois resultados do MT e estima-se que o erro deveria ser menor que 4 *pixels* tanto para $x$ quanto para $y$. Assim, minimizaria o erro de encontrar duas regiões que, possivelmente, fossem consideradas similares.

Já no terceiro estágio, os erros são provenientes de imagens similares e que não são capazes de serem identificados no MT. Diante disso, divide-se os dados para filtragem de acordo com o número de instância de máscara. Desse modo, para cada grupo são divididos nas direções de $\theta_1$ e $\theta_2$ e, posteriormente, feitos cálculos de distância interquartílica, desvio padrão e, por fim, quantidade de elementos em cada divisão. Assim, caso um desses cálculos extrapole determinado valor, seu ponto correspondente é excluído. Se o conjunto de dados possuísse quantidade inferior a $20\%$ dos elementos, elimina-se todo o grupo por falta de representatividade.

Por fim, após a remoção destes ruídos, utilizam-se estes dados para a interpolação e a validação do modelo de câmera no cenário real.

### 3.1.2 Interpolação

Na etapa de interpolação, o *dataset* é criado e fragmentado em duas partes, cerca de $70\%$ do conjunto de dados para a interpolação e o restante para a validação, os quais são separados aleatoriamente para evitar vieses relativos à seleção das amostras. Assim, com os dados de interpolação, utiliza-se o algoritmo de interpolação linear sobre malhas simpliciais[4] para criar a interpolação dos dados. Este consiste na triangulação dos dados e, em cada triângulo formado, realiza-se uma interpolação baricêntrica linear.

Desta forma simplificada, a interpolação pode ser feita por meio das seguintes etapas:

- calcular a triangulação;
- localizar a qual triângulo $t$ e o ponto $q$ a ser interpolado;
- interpolar linearmente no triângulo utilizando Equação (23).

Assim, de forma genérica e exemplificativa, o resultado do procedimento acima é ilustrado na Figura 11. Ressalta-se que os pontos $P_1$, $P_2$ e $P_3$, são pontos da malha mais

[3] do Inglês *Field of View* - campo de visão

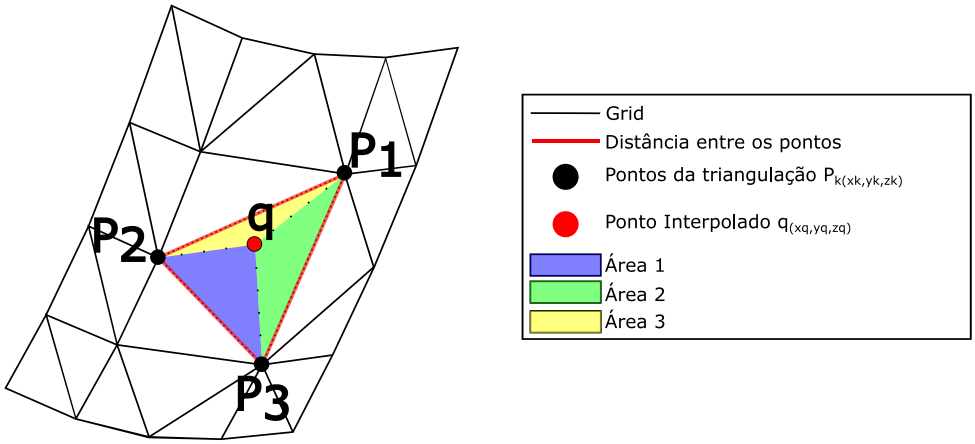
[4] LinearNDInterpolator da biblioteca SciPy (2020) com o uso do Algoritmo do Barber, Dobkin e Huhdanpaa (1996)

próximos do ponto interpolador $q$ o qual realiza-se a triangulação de Delaunay a fim de encontrá-los. Em posse destes pontos, determinam-se os sub triângulos, indicados em três tons diferentes de cores na Figura 11. Salienta-se que a área do sub triângulo é denominada de acordo com o vértice oposto a ele, ou seja, como exemplo: a "área2", localizada na legenda, recebe este nome devido ao ponto $P_2$. Por fim, aplica-se a

$$f(q) = \frac{A_1 f(p_1) + A_2 f(p_2) + A_3 f(p_3)}{A_1 + A_2 + A_3} \tag{23}$$

em que, $A_i$ é a área do sub triângulo de t dividido por q oposto ao vértice $p_i$, para determinar o ponto a ser interpolado.

Figura 11 – Exemplo da interpolação baricêntrica




### 3.1.3 Modelo de câmera

Nesta seção é relatado como se realiza o controle através do modelo de câmera, bem como alguns ajustes desse modelo para ambientá-lo ao comportamento do movimento da câmera. Para o modelo de câmera, definido pela Equação (16) e descrito na Subseção 2.2.1, utiliza-se o mesmo fragmento dos dados da interpolação de validação para estimar a acurácia e compará-los.

No modelo analítico, a direção da câmera é dada na forma de vetores. Entretanto o modelo real da câmera é dado pelos ângulos de azimute e elevação. Houve então a necessidade de encontrar uma relação entre essas duas direções para controlar a câmera. Assim, utilizando a conversão de coordenadas cartesianas em esféricas é possível determinar os ângulos, azimute e elevação, daquela direção do vetor.

Essas transformações são dadas matematicamente por

$$\theta_1 = \arcsin \frac{d_z}{\| \vec{d} \|} \tag{24}$$

e

$$\theta_2 = \arctan \frac{d_x}{d_y}. \tag{25}$$

De maneira similar, para utilizar o modelo analítico é necessário obter a direção atual da câmera que é encontrada através dos dados extraídos da própria câmera. Desse modo, é necessário e reverter a conversão das coordenadas para a forma de vetores.

$$d_z = \| \vec{d} \| \sin\theta_2 \tag{26}$$

$$d_x = \| \vec{d} \| \cos\theta_2 \cos\theta_1 \tag{27}$$

$$d_y = \| \vec{d} \| \cos\theta_2 \sin\theta_1 \tag{28}$$

$$\vec{\mathbf{d}} = \| \vec{d} \| \left(\hat{d_x}, \hat{d_y}, \hat{d_z}\right) \tag{29}$$

Por fim, ressalta-se que a acurácia e o tempo de resposta do modelo são definidos utilizando o conjunto de dados de validação.

### 3.1.4 Avaliação dos modelos

Para este trabalho são utilizados dois tipos de avaliação: acurácia e o tempo de resposta. A acurácia consiste em verificar a qualidade do resultado analisando o quanto os modelos destoam do resultado real, ou seja, o quanto o resultado da interpolação e o modelo analítico diferem do conjunto de dados de validação, citado na Subseção 3.1.2. Deste modo, utiliza-se como métrica de erro o erro quadrático médio (MSE[5]) e o erro absoluto médio (MAE [6]). Assim, quanto menor o erro, mais próximo do sistema real.

Já com relação ao tempo de resposta, verifica-se o quanto o modelo é custoso computacionalmente para realizar o cálculo. Dessa forma, utilizam-se os mesmos dados de validação para mensurar o tempo decorrido nos cálculos, obtendo-se um gráfico de relação entre o tempo gasto pelo número de amostras para cada um dos métodos.

## 3.2 Localização Geográfica

Nesta seção são descritos os conceitos propostos para a localização geográfica do objeto, baseando-se em vetores (ou cossenos diretores). Portanto, são abordados temas relacionados a topografia de uma seção de uma determinada microrregião, como por exemplo, mapa de elevação, além de mostrar como é sincronizada a direção da câmera com a direção do modelo analítico.

[5] do Inglês *Mean Squared Error*
[6] do Inglês *Mean Absolute Error*

### 3.2.1 Encontrando posição, direção e abertura da câmera

Nesta seção é descrito o procedimento para determinar a posição, direção e abertura da câmera, utilizando o modelo referido na Seção 2.4. A determinação da posição, direção e abertura da câmera é dada por um sistema de equações que envolvem posições geográficas e de *pixels* conhecidos. Assim, dada a posição geográfica relativa de um ponto $p_a$, estabelece-se a relação necessária para determinar a posição do objeto ao ambiente ao redor da câmera, dada por:

$$p + \lambda_a \hat{d_a} = p_a \tag{30}$$

em que,

- $p$ é a posição geográfica da câmera;
- $\lambda_a$ é a distância relativa ao ponto $a$ e a câmera;
- $\hat{d_a}$ é a direção entre ponto geográfico $a$ e a câmera.

Desse modo, geram-se as incógnitas $p$, $\lambda_a$, $\hat{d}$ e a abertura da câmera $\alpha$, descritas na Subseção 2.2.1. Vale mencionar que $d_a$ é função da abertura da câmera $\alpha$, direção da câmera $\hat{d_a}$ e posição do *pixel* $(a_v,a_u)$. Sendo assim, cada posição geográfica $p_a = (x_a,y_a,z_a)$ define um sistema de três equações. De forma análoga, outros pontos são definidos por Equação (30) para outras coordenadas geográficas (e.g. $p_b$, $p_c$). Reforça-se que as direções de ambas as posições são normalizadas, ou seja, $\| \hat{d} \| = 1$. Assim, escolhendo três pontos fiduciais, forma-se o sistema não-linear com dez incógnitas e dez equações, de acordo com:

$$\begin{aligned} p + \lambda_a d_a &= p_a \\ p + \lambda_b d_b &= p_b \\ p + \lambda_c d_c &= p_c \\ \| \hat{d} \| &= 1 \end{aligned} \tag{31}$$

### 3.2.2 Localização do objeto

Após a determinação dos parâmetros da câmera, inicia-se o procedimento para determinar a localização do objeto através dos métodos matemáticos abordados na Seção 2.4 e Subseção 2.2.1. A direção da câmera no espaço é definida pelos cossenos diretores $\hat{d_a}$ com a sua posição inicial $p$ e, ainda, pode-se definir que, para uma determinada distância $\lambda_a$, a posição no espaço é dada por:

$$p_a(\lambda_a) = p + \lambda_a \hat{d_a}, \quad \lambda_a \geq 0 \tag{32}$$

na qual,

- $p$ é a posição geográfica da câmera;
- $\lambda_a$ é a distância relativa ao ponto $a$ e a câmera;

- $d_a$ é a direção entre ponto geográfico $a$ e a câmera.

Diante disso, para determinar o ponto de interseção da reta sobre uma superfície, é necessário encontrar o menor valor de $\lambda$ que consiste na menor distância entre a superfície e a câmera, pois pode haver outros pontos que intercedem a superfície com a mesma direção. Dessa forma, escolhe-se a distância mais próxima, que é a imagem vista pela câmera, assim, a função é minimizada de acordo com a diferença entre um ponto genérico $p_g = (x_g,y_g,z_g)$ determinado pela direção da câmera com a sua posição relativa sobre a superfície $z_g(x_g,y_g)$, exposta na Subseção 2.2.1 e definido por:

$$h\left(\lambda_a\right) = p_a\left(\lambda_a\right) \cdot \hat{z} - z_g\left(p_a\left(\lambda_a\right) \cdot \hat{x}, p_a\left(\lambda_a\right) \cdot \hat{y}\right). \tag{33}$$

Vale destacar que alguns pontos da posição relativa sobre a superfície não coincidem com o ponto genérico da direção da câmera, portanto há necessidade de interpolar o *grid* de modo a encontrar as mesmas posições de $(x_g,y_g)$ e, assim, estabelecer a função para determinar a interseção da direção da câmera com a superfície.

Deste modo, utiliza-se o método numérico de busca de raízes, denominado bisseção (BURDEN; FAIRES, 1985), para encontrar a interseção entre a direção da câmera com a superfície. Entretanto, o método carece de complementação para determinar corretamente a raiz global, pois as variações do relevo ambiente podem omitir a primeira incidência da interseção e determinados pontos mais distantes como raiz. Isto não está errado, porém a primeira interseção é aquela que é vista na imagem da câmera. Então, para contornar este empecilho, utiliza-se o Teorema 1, descrito na Seção 2.4, para determinar se naquela direção existe uma raiz e ainda diminuir a amplitude da direção para o método da bisseção. Assim, o método convergirá mais rápido e corretamente para a raiz. Após minimizar a função $h\left(\lambda_a\right)$, determina-se o ponto exato em que o objeto está sobre um relevo. Por fim, neste trabalho utilizam-se dois ambientes, artificial e real, para determinar a posição do objeto por meio da imagem e a superfície local.

#### 3.2.2.1 Determinação do mapa de elevação

Em um ambiente real ou genérico, necessita-se determinar a localização dos pontos ao redor da câmera a fim de encontrar o posicionamento do objeto. Assim, com o direcionamento da câmera com o objeto é capaz de determinar a posição correta do objeto no relevo (superfície). Portanto, nesta seção, destina-se encontrar o relevo com as medidas em metros, descrito na Seção 2.5.

Os valores limitantes da área da superfície são determinados de acordo com o campo de visão da câmera na forma de coordenadas geodésicas. Como todos os cálculos são feitos em metros, converte-se a diferença dos pontos geodésicos em metros. Assim,

essa diferença é calculada através da seguinte equação:

$$d = 2 \cdot r_{earth} \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Phi_1 - \Phi_2}{2}\right) + \cos\left(\Phi_1\right) \cdot \cos\left(\Phi_2\right) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (34)$$

em que,

- $r_{earth}$ é o raio da Terra, aproximadamente 6371 km;
- $\Phi$ é a latitude;
- $\lambda$ é a longitude.

De posse dos limites da superfície, extrai-se o relevo com o *grid* para determinar a localização de um objeto na imagem sobre o ambiente.

# 4 Resultados experimentais

Neste capítulo são apresentados alguns resultados obtidos experimentalmente com o uso do método de rastreamento e localização de objetos, além de ressaltar as principais ferramentas empregadas e como são utilizadas no presente trabalho. Diante dos dados experimentais, é possível determinar o modelo proposto dentro do intervalo e explicitar o controle da câmera através de alguns ajustes do modelo analítico e interpolação dos dados. Assim, na Seção 4.2 são apresentados os resultados referentes aos métodos de controle, como a acurácia e o tempo de resposta. Na Seção 4.3 por sua vez, descreveu-se os resultados relacionados a determinação da posição do objeto no ambiente.

## 4.1 Aquisição de dados para rastreamento de objetos

Nesta seção são descritas a forma da aquisição e a limpeza dos dados para a criação do *dataset*. Portanto, são abordados temas relacionados a imagem e estatísticas. Ao final desta seção, há uma breve descrição das bibliotecas utilizadas bem como do computador para o processamento e a câmera utilizada.

Como dito na Seção 3.1, a captura dos dados inicia-se pela aquisição aleatória de dados a fim de determinar o espaço em que se deve fazer a varredura. Após a determinação deste, inicia-se o processo propriamente dito que consiste na captura da imagem em diversas direções e armazenando-a em *arrays* multidimensionais com a sua respectiva direção. O procedimento de aquisição de um ponto do conjunto dos dados é explicitado nas Figuras 12, 13, 14 e 15. O centro da Figura 12 é uma amostra de *template* que serve de referência para a Figura 13. Em seguida, aplica-se o algoritmo MT sobre estas duas imagens com a cálculo do CCOEFF, resultando na malha disposta na Figura 14 cujo ponto com a maior similaridade está localizado no *pixel* (640, 479) e com o valor de 0,645, indicado na cor em vermelho. Ressalta-se que para outros conjuntos de imagens, são gerados outros valores que podem ou não ser iguais a este. Assim, obtém-se a Figura 15 que demonstra a imagem de referência (Figura 13), com retângulo branco, a imagem ampliada deste retângulo e o *template* da Figura 12. Isto significa que aquela posição em *pixel* (640, 479) é o local onde é encontrado o *template*, e a direção para centralizar este *template* é a mesma da Figura 12.

Analogamente a este procedimento, é repetido esse procedimento para todos as imagens, alternando-as entre imagem de referência e *template*. Este procedimento é descrito na Seção 3.1 com o uso do Algoritmo 3. Assim, obtém-se 1.562.500 dados o que corresponde a 1.250 imagens e/ou direções salvos. Vale salientar que existem ruídos neste *dataset* bruto, como aqueles provocados por regiões similares. E para remover uma

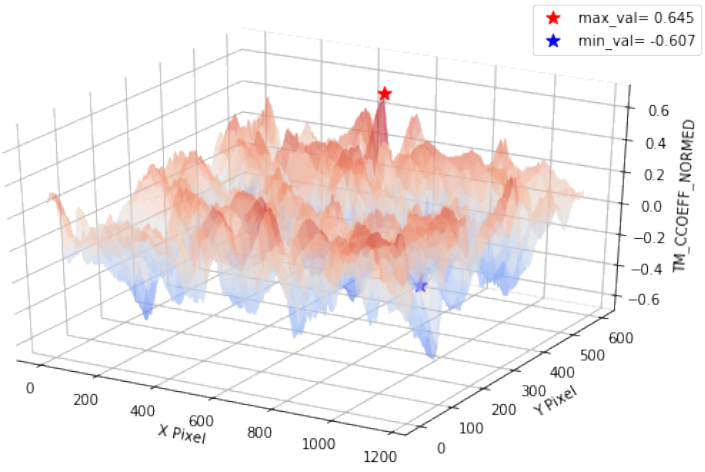
Figura 12 – Imagem de referência com o *boundbox* no centro da imagem



Figura 13 – Resultado da localização da região mais semelhante na imagem com o algoritmo *matchTemplate*



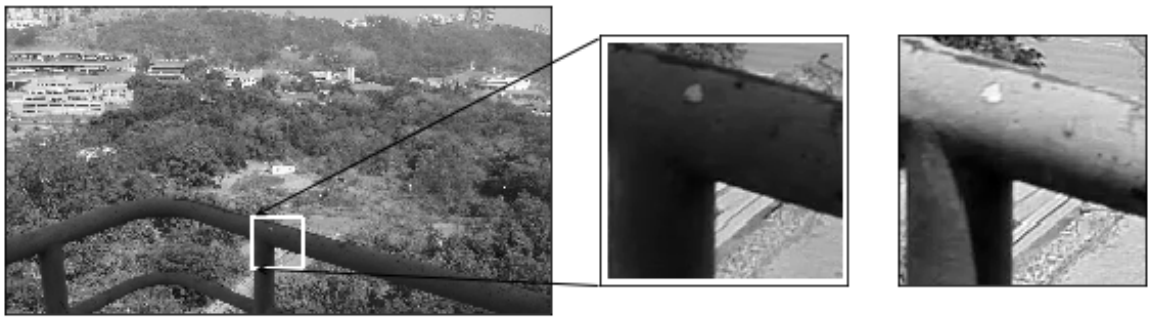
Figura 14 – Malha do resultado do *matchTemplate* com CCOEFF




descontinuidade da direção de azimute, os dados são normalizados[1] os dados de tal forma

[1] Esta etapa será justificada na seção do modelo de interpolação.

Figura 15 – Resultado comparativo visual do centro da imagem de referência com a imagem da instância da máscara.



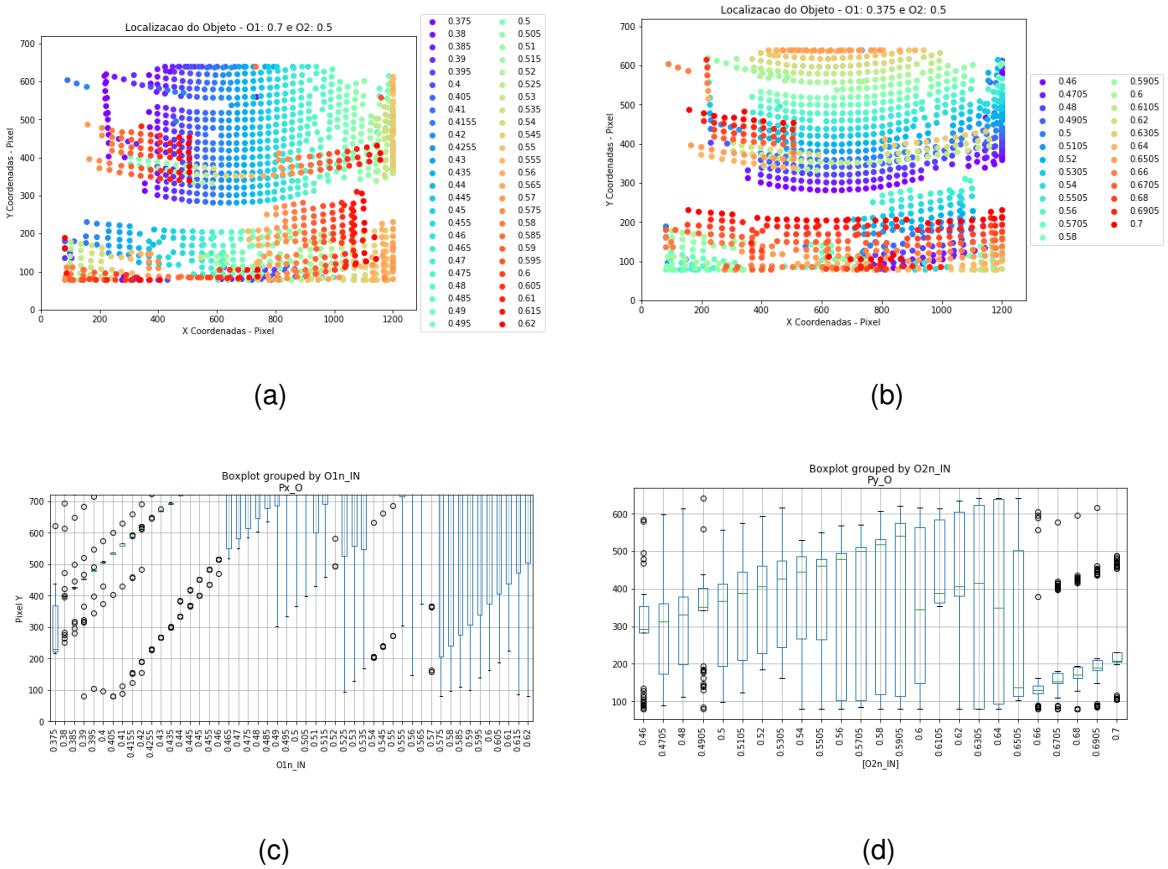
que qualquer dado obtido da câmera esteja dentro do intervalo de $0$ a $1$.

A etapa da identificação dos ruídos é descrita na Subseção 3.1.1. Após escolher o conjunto de dados que correspondem à direção da imagem *template*, $\theta_1 = 0{,}7$ e $\theta_2 = 0{,}5$ como exemplo, obtém-se o resultado da Figura 16. Esta figura demonstra todas as posições de entradas que são encontradas para centralizar o mesmo objeto. Portanto, pontos em *pixel* cujos ângulos de entradas devem ser direcionados para encontrar o mesmo *template*. As cores da Figura 16a ilustram as posições de azimute e a Figura 16b representa as posições de elevação. Deste modo, as cores da legenda presentes em cada ponto correspondem as direções de entrada: $\theta_1$ e $\theta_2$. Observa-se que existem diversos pontos que, por si só, não se relacionam, além de regiões não contínuas que são constituídas por erros provenientes de ruídos. Além disso, pode-se observar na Figura 16 que o gráfico de *boxplot* não apresenta regularidade da quantidade de dados em cada grupo de saída tanto para a saída do ângulo $\theta_1$ (Figura 16a), quanto para o $\theta_2$ (Figura 16b). Este primeiro resultado já era esperado, pois o algoritmo MT retorna valores de similaridade e não se pode afirmar de fato que é a direção correta. Assim, é necessário remover ruídos.

Para contornar este problema, é aplicado primeiramente um limiar que exclui todos os dados que estão fora do campo de visão (FoV) da câmera. Vale salientar que caso seja utilizada a técnica de eliminação dos *outliers* pelo método de Tukey (1977), resultaria na perda de dados corretos. Isto porque os pontos errados deslocam a curva e, ainda assim, mantêm os pontos incorretos, além de não poder garantir que é uma distribuição normal para todos os eventos. Diante disso, por meio do limiar do FoV, origina-se a Figura 17. Nota-se a diminuição da quantidade de itens removidos nas legendas das Figuras 17a e 17b, quando confrontada às Figuras 16a e 16b. Além disso, comparando visualmente a Figura 16 com a Figura 17, observa-se que os dados após a aplicação do limiar apresentam menores dispersão (maior regularidade dos dados) e distância interquartílica.

Ainda assim, existem outros pontos que são provenientes de outros ruídos tais como a similaridade da imagem de vegetação da região e construções. Diante disso, outro limiar é utilizado para remover estes ruídos: diferença dos resultados de duas técnicas do MT que

Figura 16 – *Dataset* bruto




(a) (b)

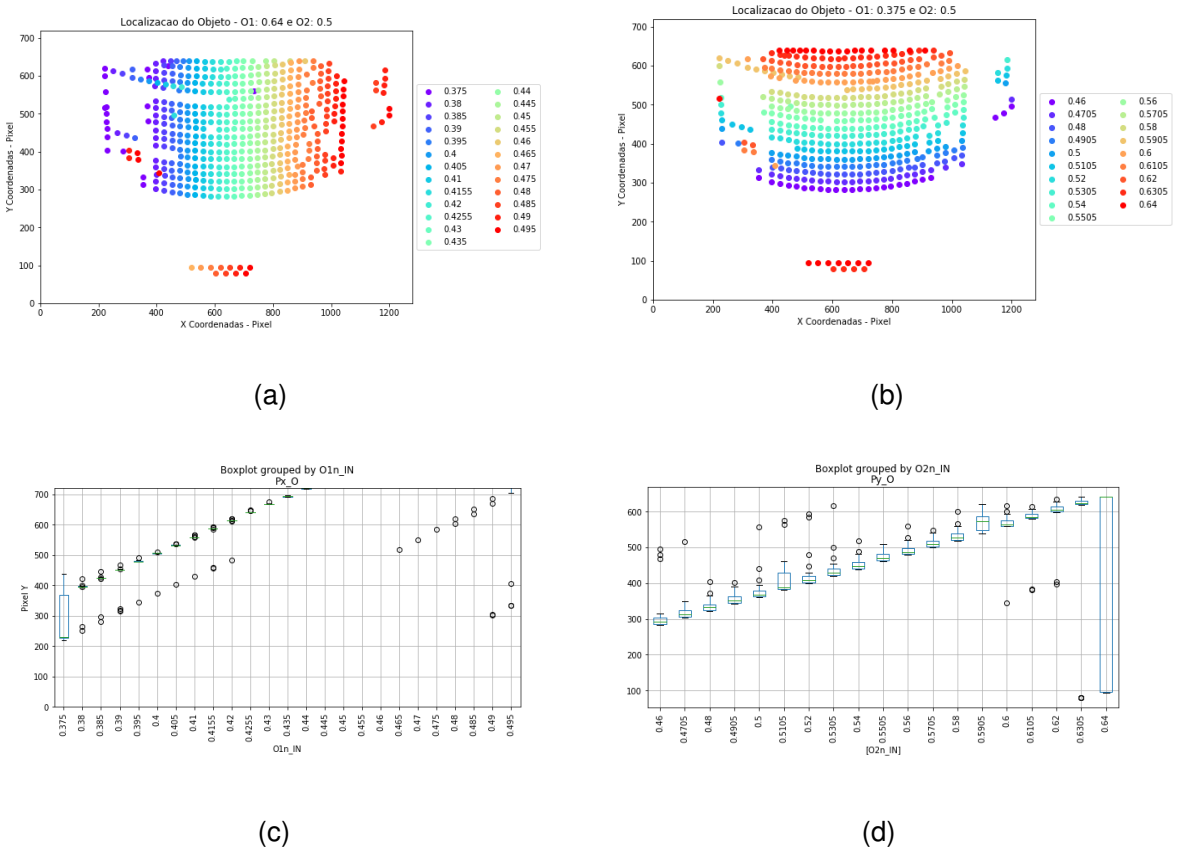
(c) (d)

devem ser aproximadamente iguais. Assim, caso o resultado das medidas do *matchTemplate* sejam maiores que 4 *pixels*, tolerância adotada no projeto, seriam eliminados. Sendo assim, são eliminados os dados dos semigrupos (agrupamento pelo conjunto de cores) com número inferior a 35 e 50 para a elevação e azimute, respectivamente, devido à falta de representatividade, já que existem imagens com alta similaridade sendo incorretas. Ressalta-se que pela quantidade de dados (cerca de 1.562.500 pontos pré-filtragem e 356.752 após a filtragem) é infactível a inspeção manual na determinação de falsos positivos em todo conjunto de dados. Assim, após a verificação de um conjunto e seguindo o pressuposto que é melhor ter um conjunto com dados mais próximos do real do que ter um conjunto grande de falsos positivos, é aumentada a margem de segurança do limiar (diminui-se o limiar) para que mantenha o maior número de pontos próximo do real. Assim, resulta-se nos dados demonstrados na Figura 18, com a dispersão de *pixel* indicados na Figura 18c e Figura 18d.

Por fim, é finalizada a etapa de aquisição dos dados tratados e inicia-se a obtenção dos resultados dos métodos para o controle e localização do objeto no ambiente. A seguir, estão descritas todas as bibliotecas e *softwares* empregados no trabalho, bem como as especificações do computador utilizado para a avaliação.

Neste estudo, os experimentos são realizados na plataforma Jupyter (KLUYVER

Figura 17 – *Dataset* com limitações do FoV




(a) (b)

(c) (d)

et al., 2016) com a linguagem de programação Python com as seguintes bibliotecas: OpenCV (ITSEEZ, 2015), NumPy (WALT; COLBERT; VAROQUAUX, 2011), Pandas (MCKINNEY et al., 2010), Matplotlib (HUNTER, 2007), Scikit-learn (PEDREGOSA et al., 2011) e Scipy(JONES et al., 2001–). Além disso, utiliza-se o *software* Matlab (MATLAB, 2018) e Blender (COMMUNITY, 2020).

O OpenCV[2](ITSEEZ, 2015) é uma biblioteca livre de multiplataforma e multilinguagem que utiliza a visão computacional. Utiliza-se dessa biblioteca a função *matchTemplate* para encontrar características semelhantes que possam ser empregadas como referência para gerar os *dataset*. Além disso, também ocorre a apliação do filtro de gauss na imagem para a suavizar possíveis ruídos.

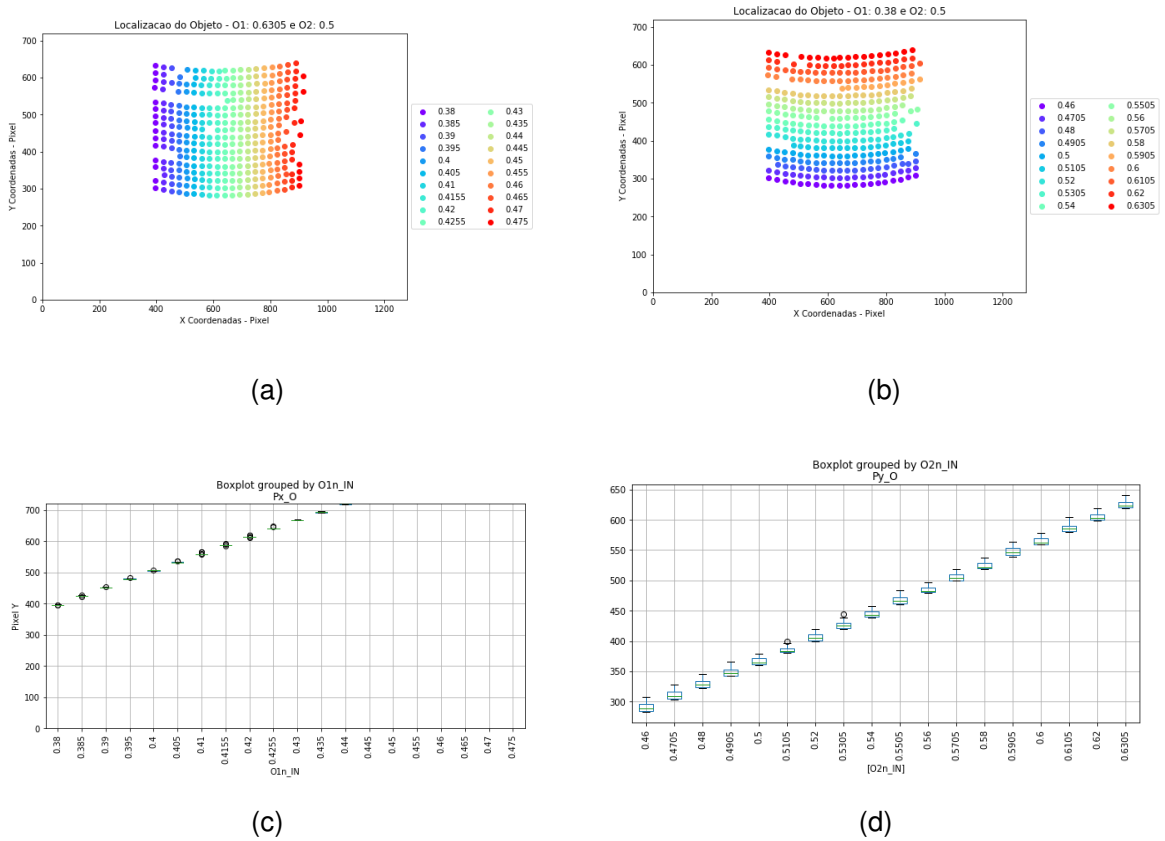
Segundo os autores Walt, Colbert e Varoquaux (2011), o NumPy é uma biblioteca para a linguagem Python que suporta *arrays* unidimensional e multidimensionais, além de possuir funções matemáticas para estas estruturas.

A biblioteca Matplotlib é uma biblioteca livre em Python utilizada para gerar gráficos (HUNTER, 2007). Neste trabalho, além de gerar gráficos, a Matplotlib é utilizada para gerar as superfícies do ambiente.

A biblioteca Pandas é utilizada na linguagem Python para a manipulação e análise

[2] Open-Source Computer Vision

Figura 18 – *Dataset* tratado




(a) (b)

(c) (d)

dos dados. Neste estudo, ela é empregada para estruturar o *dataset*, vinculando os dados da direção de entrada e a localização do objeto com o correspondente correto de direção e de saída (MCKINNEY et al., 2010).

Já a biblioteca Scikit-learn é uma biblioteca livre de aprendizado de máquina para a linguagem Python (PEDREGOSA et al., 2011). Neste estudo, é utilizada para dividir os *dataset* e para avaliar a acurácia e o tempo da interpolação.

A biblioteca Scipy (JONES et al., 2001–), derivada da NumPy, é uma biblioteca livre em Python, destinada à manipulação de dados científicos. Nesta pesquisa, a Scipy é utilizada para a dividir o *dataset* e para a realização de testes estatísticos.

O *software* Matlab (MATLAB, 2018) é utilizado neste estudo, para a visualização das direções dos vetores. A escolha deve-se principalmente pelo fato de não haver implementação para manter os eixos com as mesmas proporções na visualização dos eixos com as bibliotecas em Python utilizadas no projeto. Além de avaliar a acurácia e o tempo do modelo numérico e analítico.

Já o *software* Google Earth combina as imagens de satélite com os dados topográficos pertinentes ao relevo de escolha. Neste contexto, o Google Earth é utilizado para determinar as posições reais da câmera no ambiente, posições fiduciais a fim de definir parâmetros para a sincronização do modelo analítico ou o método por interpolação. Além

disso, verificar a exatidão da Equação (34).

Já o *software* Blender (COMMUNITY, 2020), é uma ferramenta para modelagem, animação, texturização, composição, renderização, e edição de vídeo (BLENDER, 2020). Para este trabalho, este software é utilizado para criar um cenário artificial para servir de comparação ao mundo real.

No que tange a avaliação do tempo de resposta dos métodos, as análises são realizadas em um computador com o processador AMD Ryzen 5 2600, que possui 6 núcleos com uma CPU cuja velocidade é 3,40GHz. A quantidade de memória do computador é de 16 GB de RAM DDR4 com a velocidade de 2666 Mhz. O sistema operacional utilizado é o Linux - Ubuntu 19.04 LTS. E para a captura da imagem e da direção, a câmera Autodome IP 5000 IR[3] com controle PTZ e resolução máxima de 1080p.

## 4.2 Centralização do objeto na imagem

Nesta seção, divide-se em dois âmbitos: acurácia e tempo de execução. Diante disso, são abordados conceitos envolvendo a precisão dos modelos além de determinar o tempo de execução de cada método. Destaca-se a quantidade dos dados brutos, filtrados e utilizados, da etapa descrita na seção anterior.

Os dados obtidos são utilizados para avaliar a acurácia e o tempo de resposta do método de interpolação e analítico. Assim, são selecionados $70\%$ dos dados, com o auxílio da biblioteca Scipy, para constituírem o conjunto de dados para a interpolação e o restante para avaliar ambos os métodos. Esta etapa é descrita na Subseção 3.1.4. Desse modo, a quantidade de pontos total era de 356.752, dos quais são extraídos 249.726 dados para criar o modelo de interpolação. O restante, 107.266 dados, é utilizado para validar a interpolação, entretanto, 393 dados extrapolam o conjunto de dados do modelo interpolador. Assim, são utilizados 106.633 dados para a validação da interpolação e do modelo analítico, ou seja, os dados extrapolados na interpolação são removidos durante esse processo.

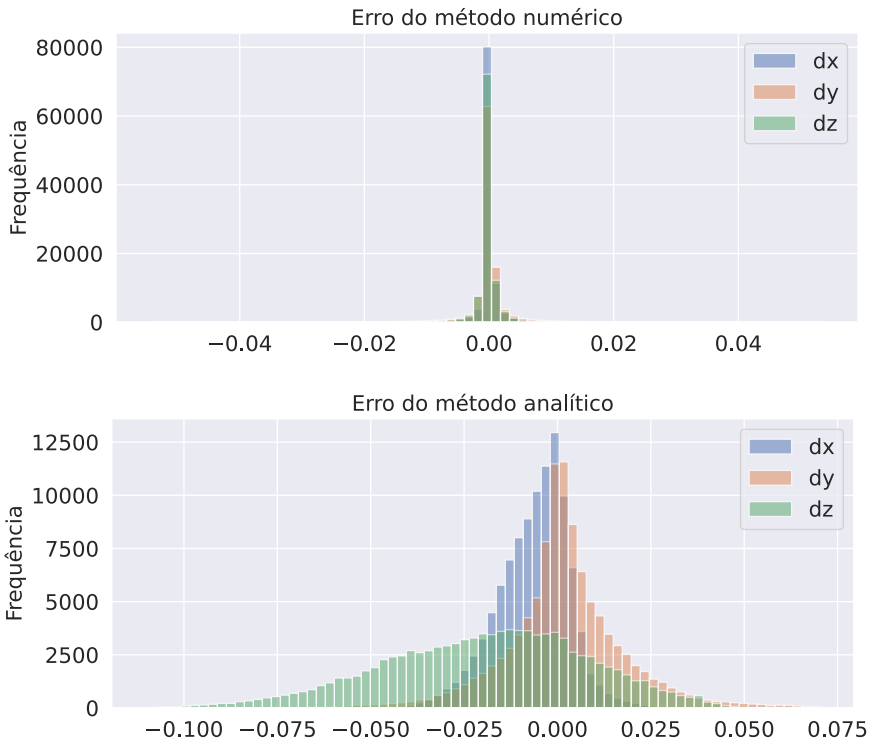
### 4.2.1 Acurácia

Após utilizar esses dados de entrada de validação sobre os métodos, os dados obtidos são aqueles que correspondem à direção a ser posicionada por cada método. Em seguida, subtraem-se as direções obtidas por cada um deles pelos respectivos pontos de validação. Assim, obtêm-se os dados para o histograma de erro dos métodos com a específica direção, ilustrada na Figura 19.

Portanto, os erros de cada direção ($d_x$,$d_y$ e $d_z$) são localizados na Figura 19, bem como, a indicação da acurácia de cada método. Isso reflete, principalmente, na exatidão do

[3] Código do produto: NEZ-5130-IRCW4

Figura 19 – Erro em cada modelo




modelo. Observa-se que o método de interpolação é mais preciso em todas as direções, enquanto o método analítico possui erros maiores e um erro mais expressivo na direção $d_z$.

A fim de determinar qual o teste a ser aplicado, inicia-se o teste de normalidade dos dados (D'AGOSTINO, 1971; D'AGOSTINO; PEARSON, 1973). Vale ressaltar que os dados são submetidos ao valor absoluto de modo que não haja erro negativo, conforme ilustrado na Figura 20. Diante disso, a hipótese nula ($H_0$) e a alternativa ($H_a$) são estruturadas da seguinte forma:

- $H_0$ : Os dados seguem uma distribuição normal;
- $H_a$ : Não se pode afirmar que os dados seguem uma distribuição normal.

Como pode ser observado na Figura 20, nenhuma das duas distribuições são gaussianas. Além disso, com o valor da significância ($\alpha$) igual a $0{,}05$ o p-valor de cada direção é próximo a zero ($[10^-9, 10^-9, 10^-9]$), ou seja, os dados não seguem uma distribuição normal.

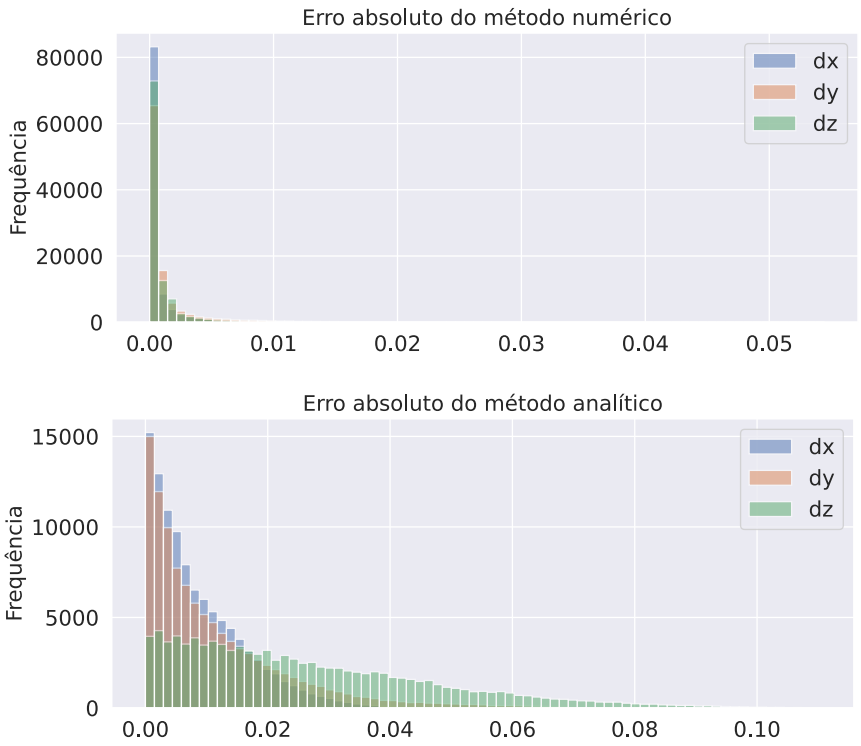
Após a identificação da distribuição não-normal dos dados, utiliza-se o teste U de *Mann-Whitney* (ON. . . , 1947) para determinar qual o menor erro dentre os métodos:

- $H_0 : F(U) \geq G(U) \; \forall \; U$;
- $H_a : F(U) < G(U) \; \forall \; U$.

Em que, F(U) é o erro absoluto do método analítico (MA) e G(U) é o erro absoluto do método numérico (MN).

De acordo com Figura 19, espera-se que o erro de MN seja menor que MA, pois a quantidade de valores em torno do erro próximo a zero é maior em relação ao outro. Diante disso, todas as direções rejeitam a hipótese alternativa com o p-valor igual a

Figura 20 – Erro absoluto em cada modelo




$(0.\overline{9}, 0.\overline{9}, 0.\overline{9}]$), ou seja, o erro do MA é maior do que MN. Portanto, percebe-se graficamente e estatisticamente que o método numérico contém erros menores que o método analítico.

Em seguida, para determinar a qualidade do modelo analítico e da interpolação, são utilizadas três medidas: (1) o erro médio absoluto (MAE[4]), (2) erro médio quadrático (MSE[5]) e (3) raiz do erro médio quadrático (RMSE[6]). Após os experimentos, nas tabelas 2 e 3 são apresentados os resultados de acurácia dos modelos. Portanto, de acordo com estes dados, o método da interpolação apresenta uma maior acurácia com erro MAE e RMSE máximo de $2{,}912 \times 10^{-03\circ}$ e $3{,}142 \times 10^{-04\circ}$ na direção $d_y$. Já para o modelo analítico, o erro máximo está na direção $d_z$ com os valores $2{,}697 \times 10^{-02\circ}$ e $3{,}529 \times 10^{-02\circ}$ de MSE e RMSE, respectivamente.

Tabela 2 – Resultado da acurácia para o modelo analítico

| | Modelo analítico | | |
|---|---|---|---|
| Erro | $MAE(^\circ)$ | $MSE(^\circ)$ | $RMSE(^\circ)$ |
| $d_x$ | $9{,}2107 \times 10^{-03}$ | $2{,}1181 \times 10^{-04}$ | $1{,}4554 \times 10^{-02}$ |
| $d_y$ | $1{,}2514 \times 10^{-02}$ | $5{,}0240 \times 10^{-04}$ | $2{,}2414 \times 10^{-02}$ |
| $d_z$ | $2{,}6966 \times 10^{-02}$ | $1{,}2455 \times 10^{-03}$ | $3{,}5292 \times 10^{-02}$ |
| Total | $3{,}795 \times 10^{-02}$ | $1{,}960 \times 10^{-03}$ | $7{,}225 \times 10^{-02}$ |

[4] do inglês *Mean Absolute Error*
[5] do inglês *Mean Squared Error*
[6] do inglês *Root Mean Square Error*

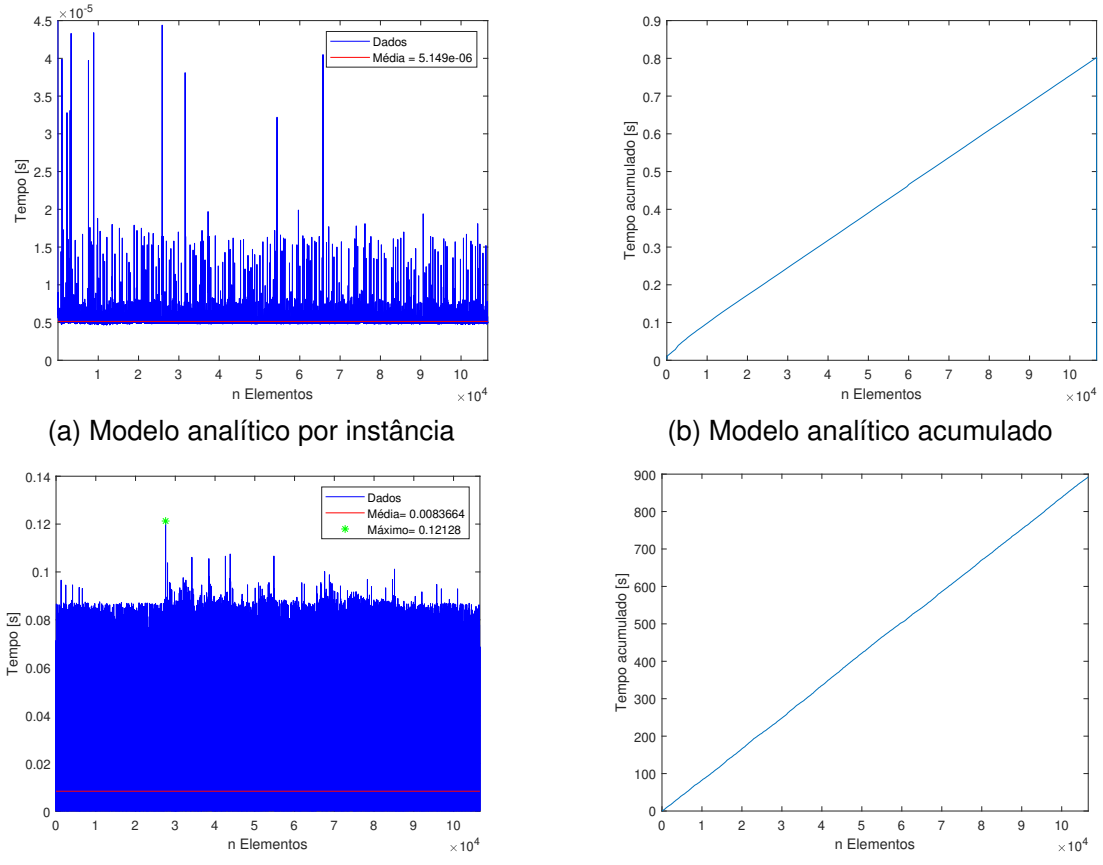
Tabela 3 – Resultado da acurácia para a Interpolação

| | Modelo de interpolação | | |
|---|---|---|---|
| Erro | $MAE(°)$ | $MSE(°)$ | $RMSE(°)$ |
| $d_x$ | $1{,}3851 \times 10^{-03}$ | $9{,}2528 \times 10^{-05}$ | $9{,}6192 \times 10^{-03}$ |
| $d_y$ | $2{,}9117 \times 10^{-03}$ | $3{,}1423 \times 10^{-04}$ | $1{,}7726 \times 10^{-02}$ |
| $d_z$ | $1{,}7067 \times 10^{-03}$ | $9{,}1202 \times 10^{-05}$ | $9{,}5500 \times 10^{-03}$ |
| Total | $6{,}0004 \times 10^{-03}$ | $4{,}979 \times 10^{-04}$ | $3{,}690 \times 10^{-02}$ |

### 4.2.2 Tempo de Resposta

Assim como a acurácia dos modelos é importante, o tempo de resposta é um fator crucial para a determinar a escolha do método, pois interfere diretamente no escalonamento de tarefas. Desta maneira, esta seção aborda o tempo necessário para executar cada tarefa entre os dois métodos propostos cujo resultado está disposto na Figura 21.

Figura 21 – Tempo gasto em cada modelo




(a) Modelo analítico por instância

(b) Modelo analítico acumulado

(c) Interpolação por instância

(d) Interpolação acumulada

Assim, a Figura 21b e Figura 21d, demonstram o tempo acumulado dos cálculos dos $n$ elementos e observa-se que o comportamento é linear. A Figura 21a e Figura 21c representam o tempo de cálculo de cada $n$ elementos, por meio dos quais pode-se constatar que para o modelo analítico o tempo médio é de $7{,}522 \times 10^{-06}$ segundos e com o desvio

padrão de $1{,}598 \times 10^{-05}$ segundos. Já para o modelo de interpolação o tempo médio é maior, com o valor de $8{,}366 \times 10^{-03}$ segundos, porém com desvio padrão de $2{,}018 \times 10^{-02}$ segundos. De acordo com os dados, o tempo de resposta do modelo analítico é cerca de $1.100$ vezes menor do que a interpolação.

## 4.3 Determinação da posição do objeto no ambiente

A seguir, são apresentados os resultados referentes à localização geográfica. Nesta etapa, o ambiente de testes é fragmentado em duas linhas: artificial e real. O ambiente real corresponde a uma região onde a câmera está localizada localizada a câmera cuja direção é usada para determinar a localização do objeto de interesse. O ambiente artificial, representa um cenário artificial que se refere à topografia associada à oscilação de uma membrana (logo do MATLAB). Esta seção está relacionada com a aplicação dos métodos da Seção 3.2.

### 4.3.1 Caso Artificial

No caso artificial, a imagem da câmera consiste na representação do logotipo do Matlab em um ambiente artificial. Ressalta-se que na imagem não ocorre nenhum fenômeno em que possa incorporar erros provenientes da captação de imagens e eixos de rotações diferentes ao modelo analítico.

Inicia-se o caso artificial com a escolha do ambiente que objetiva simular a localização de um objeto dentro de uma imagem, demonstrado na Figura 22, cuja dimensão é $w = 1920$ e $h = 1200$. Salienta-se que é preciso corrigir os ângulos de azimute e elevação para que haja uma sincronização da imagem com modelo analítico, pois os ângulos inseridos no *software Blender* não correspondem diretamente com o já predeterminado. Para isto, determinam-se, anteriormente, os parâmetros de entrada do ambiente, tais como: posição, direção e abertura da câmera, 3 pontos fiduciais que estão dentro da imagem.

Após determinar a região de interesse, são utilizados alguns parâmetros predefinidos da câmera, tais como: $\hat{d} = (0.4364, -0.8729, -0.2182)$, $\alpha = 50°$, $p = (300, 800, 50)$ metros e as posições em *pixels* de $a = (10, 1190)$, $b = (1700, 800)$, $c = (1000, 200)$. Assim, com estes parâmetros fixos, obtém-se a Figura 23 fictícia cuja imagem retrata a visão do ambiente por uma câmera artificial. Além disso, extraem-se as direções desses *pixels* e, consequentemente, os pontos fiduciais em metros, identificados na Figura 23. Os pontos *a*, *b* e *c* nesta figura, são os pontos fiduciais e, para título de exemplo, o ponto $e = (500, 400)$ é escolhido para ser encontrado. Ressalta-se que nesta etapa considera-se apenas as posições fiduciais e de *pixels* dos pontos *a*, *b* e *c*, a posição da câmera $p$, para encontrar o parâmetro de abertura da câmera $\alpha$ e prosseguir na localização do *pixel* de referência *e*.

Assim, com o uso do modelo analítico, descrito na Subseção 2.2.1, determina-se a direção do objeto em relação a câmera e com o auxílio do Teorema 1, obtém-se a Figura 24a.

Figura 22 – Ambiente Artificial

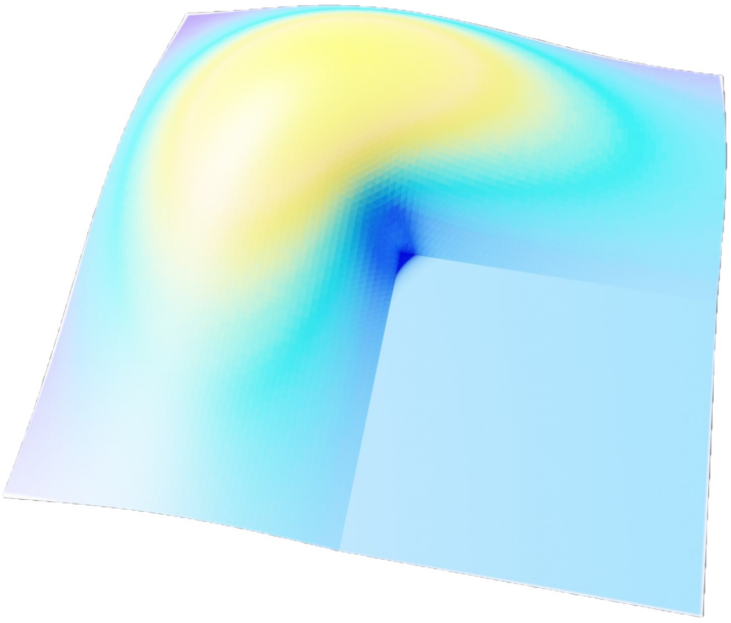
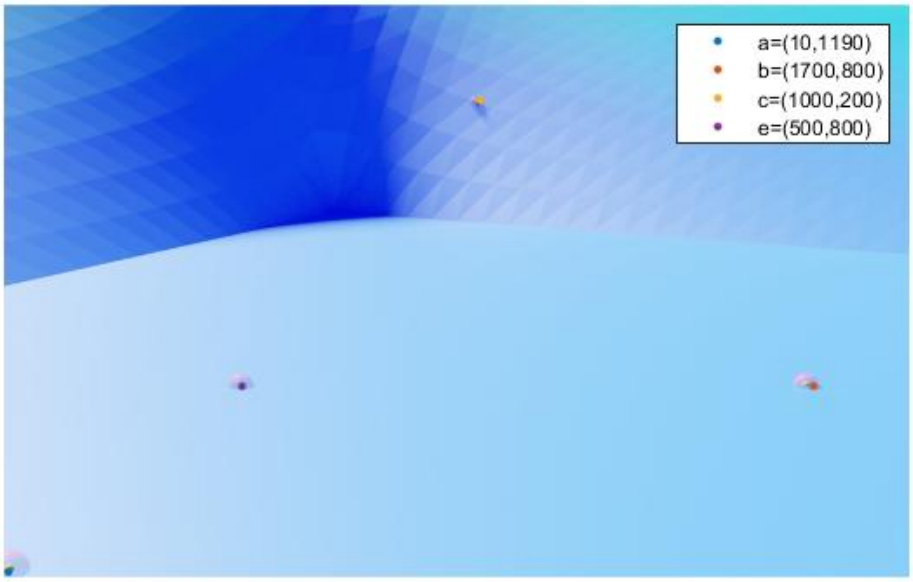


Figura 23 – Imagem da câmera gerada artificialmente




Ainda de acordo com esta figura, a posição da câmera está localizada no asterisco vermelho; a direção do objeto está disposta na reta vermelha; a projeção da direção do objeto sobre o relevo está disposta na reta azul; por fim, os asteriscos verdes representam os $30$ pontos feitos pelo Teorema 1, a fim de determinar se em cada semi-intervalo existe uma raiz. Vale ressaltar que, a título de ilustração, é realizado o cálculo para todos estes os pontos.

Após determinar os intervalos menores, utiliza-se o método numérico com a tolerância de $0{,}0003$ metros, descrito na Subseção 3.2.2, para encontrar a raiz da diferença entre a direção do objeto com a sua projeção no relevo (raiz da Equação (33)). Desta forma, a

Figura 24 – Resultado da localização(Método Numérico).

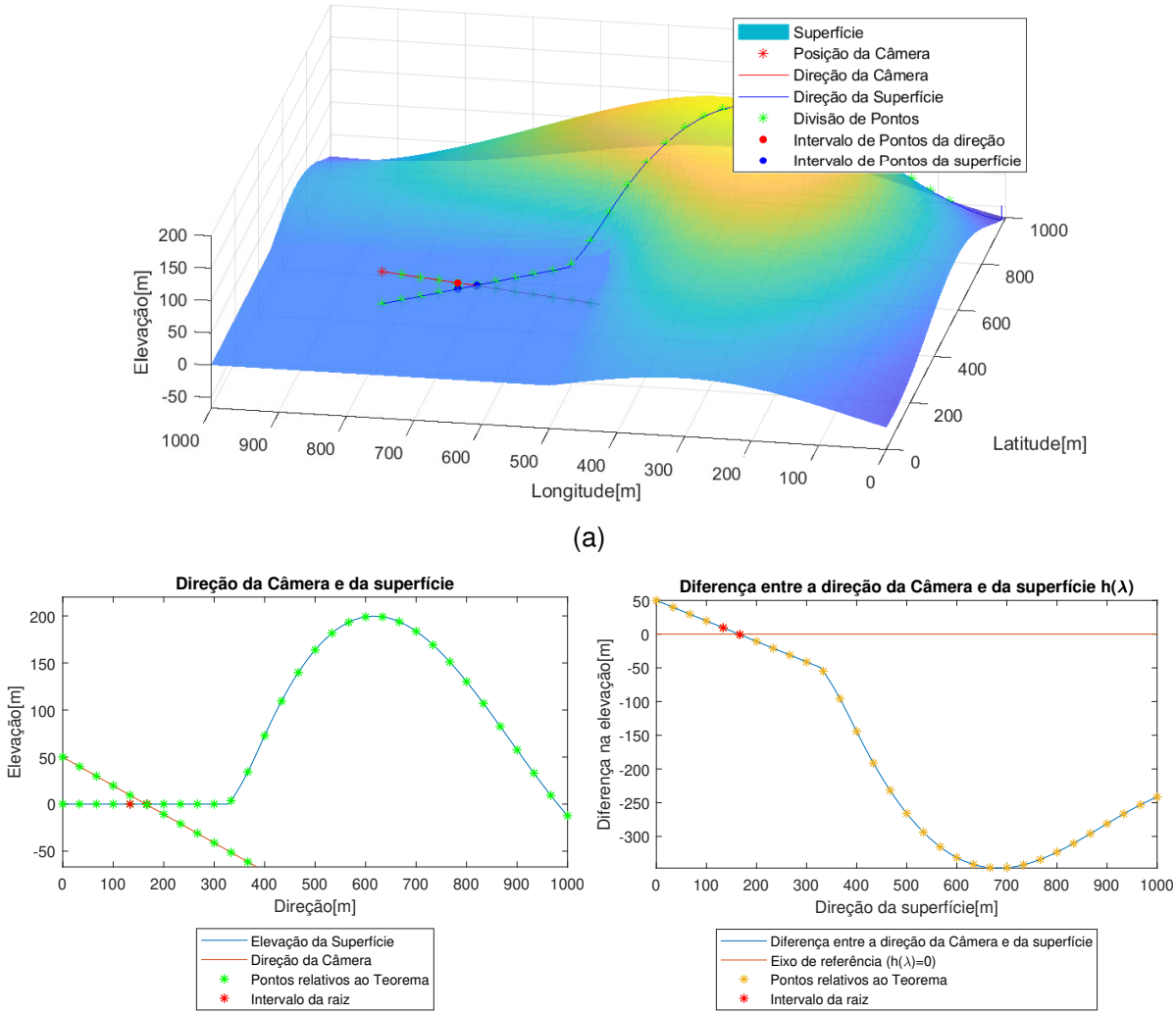



Figura 24 mostra o resultado obtido por estes procedimentos. Cabe mencionar que a reta na Figura 24b retrata a direção do objeto, a curva em azul é a sua projeção na superfície; os pontos em verde são os mesmos da Figura 24a, porém em um plano $\Re^2$; os dois pontos em vermelho, correspondem ao intervalo em que ocorre a primeira raiz. Além disso, a Figura 24c representa a função da diferença entre as direções, onde ocorre o método da bisseção. Ainda, os dois pontos em vermelho presentes nesta figura, são exatamente os mesmos encontrados na Figura 24a, Figura 24b e Figura 24c.

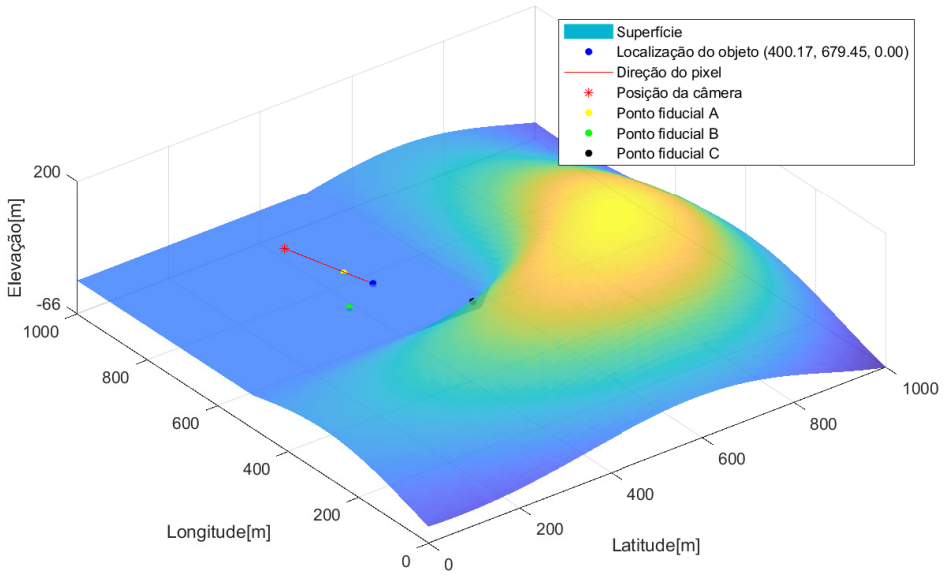
Após determinar os pontos em que ocorre a raiz, intervalo entre os pontos em vermelho, utiliza-se o método bisseção e o resultado de cada interação pode ser observado na Tabela 4. Assim, para esta direção, superfície e a posição de *pixel* escolhidos, obtém-se o resultado em $9$ interações, $\lambda = 164{,}52$ e um erro de $-0{,}00027$ metros, indicados na mesma tabela. Vale lembrar que a malha é dividida em uma área retangular com $100 \times 100$ pontos e realiza-se interpolação para determinar essa direção projetada sobre a superfície. Então,

caso altere o refinamento da malha, pode-se encontrar um erro diferente de acordo com a irregularidade da superfície e as aproximações da interpolação. Por fim, resulta-se na localização do objeto no ponto $p_o = (400{,}01,\ 679{,}65,\ 0{,}00)$ indicado na Figura 25.

Tabela 4 – Resultado do método numérico

| i | $\lambda_i$ | $\lambda$ | $\lambda_f$ | $h(\lambda)$ | $p(\lambda)$ |
|---|---|---|---|---|---|
| 0[7] | 0 | 500,00 | 1000,00 | 50,00000 | 50,00000 |
| 1 | 133,33 | 150,00 | 166,67 | 4,41210 | 4,41210 |
| 2 | 150,00 | 158,33 | 166,67 | 1,87940 | 1,87940 |
| 3 | 158,33 | 162,50 | 166,67 | 0,61311 | 0,61311 |
| 4 | 162,50 | 164,58 | 166,67 | -0,02005 | -0,02006 |
| 5 | 162,50 | 163,54 | 164,58 | 0,29652 | 0,29652 |
| 6 | 163,54 | 164,06 | 164,58 | 0,13823 | 0,13823 |
| 7 | 164,06 | 164,32 | 164,58 | 0,05909 | 0,05909 |
| 8 | 164,32 | 164,45 | 164,58 | 0,01951 | 0,01951 |
| 9 | 164,45 | 164,52 | 164,58 | -0,00027 | -0,00027 |

Figura 25 – Determinação da área para o estudo.




### 4.3.2 Caso Real

No caso real, o ambiente está contido na cidade de Belo Horizonte - Minas Gerais (MG) na regional da Pampulha, explicitado na Figura 26 que contém os pontos geodésicos de referência. Ressalta-se que os eixos dos *pixels* da elevação são invertidos para coincidirem com a referência cartesiana padrão, ou seja, $p_{y-utilizado} = h - p_{y-padrao-das-imagens}$.

Figura 26 – Região de estudo para o caso real




A escolha dos pontos, justificados no Seção 3.1, delimitam a área de estudo que está disposta na Figura 26 entre as linhas em vermelho com o uso do *software* Google Earth. Os pontos representados pela letra $a$ e com o símbolo de uma circunferência são os pontos geodésicos usados para determinar os parâmetros da câmera. A marcação em verde representa o ponto de teste que serve de exemplo para mostrar a exatidão do algoritmo.

Após a determinação da área de estudos, utiliza-se a API Elevation da plataforma do Google para captar os dados referentes à elevação do ambiente. Assim, com os dados de limites e com o tamanho do grid determinado $(480,\ 586)$, obtém-se o relevo com as coordenadas geodésicas. Em seguida, é necessário converter os dados geodésicos em metros para determinar a posição do objeto no ambiente, utilizando para tanto a Equação (34) descrita na Subsubseção 3.2.2.1. Cabe mencionar que, em paralelo a esta equação, utiliza-se uma das ferramentas do Google Earth para medir a distância entre os pontos.

Após determinar o relevo da região de estudo em metros, selecionam-se alguns pontos na imagem, identificados na Figura 27, com as respectivas posições conhecidas no ambiente. Com o auxílio do Google Earth e da Figura 27, por meio de inspeção visual, identificam-se os pontos geodésicos de cada referência, denominado de ***a***, ***b***, ***c*** e ***e***, indicados na Figura 26. De forma análoga a etapa da conversão em metros dos pontos de referência, converte-se em metros esses pontos e os referenciados na imagem. Assim, determina-se a referência de cada posição no ambiente para servir de base para o método descrito na Seção 3.2. Estes dados estão dispostos na Tabela 5.

Após determinar todos os pontos fiduciais, inicia-se a determinação dos parâmetros da câmera como posição inicial, direção e abertura da câmera, descritos na Subseção 3.2.1. Deste modo, encontra-se a posição na posição de $p = (973{,}08,\ 119{,}64,\ 901{,}37)$, direção

Tabela 5 – Referência dos Pontos Fiduciais.

| Referência | Latitude(○) | Longitude($^o$) | x(m) | y(m) | Px | Py | Elevação(m) |
|---|---|---|---|---|---|---|---|
| A1 | -19,874685 | -43,983487 | 0,00 | 1385,57 | - | - | - |
| A2 | -19,874685 | -43,973303 | 1133,68 | 1385,57 | - | - | - |
| A3 | -19,886390 | -43,973303 | 1133,68 | 0,00 | - | - | - |
| A4 | -19,886390 | -43,983487 | 0,00 | 0,00 | - | - | - |
| Câmera | -19,885087 | -43,974836 | 963,03 | 154,25 | - | - | 865,18 |
| a | -19,884362 | -43,975112 | 932,30 | 240,07 | 552 | 238 | 846,43 |
| b | -19,880996 | -43,978154 | 593,67 | 638,52 | 220 | 568 | 842,65 |
| c | -19,875391 | -43,977202 | 699,64 | 1302,00 | 660 | 613 | 835,96 |
| e | -19,877751 | -43,978796 | 522,20 | 1022,64 | 510 | 523 | 846,04 |

Figura 27 – Foto de referência dos pontos fiduciais

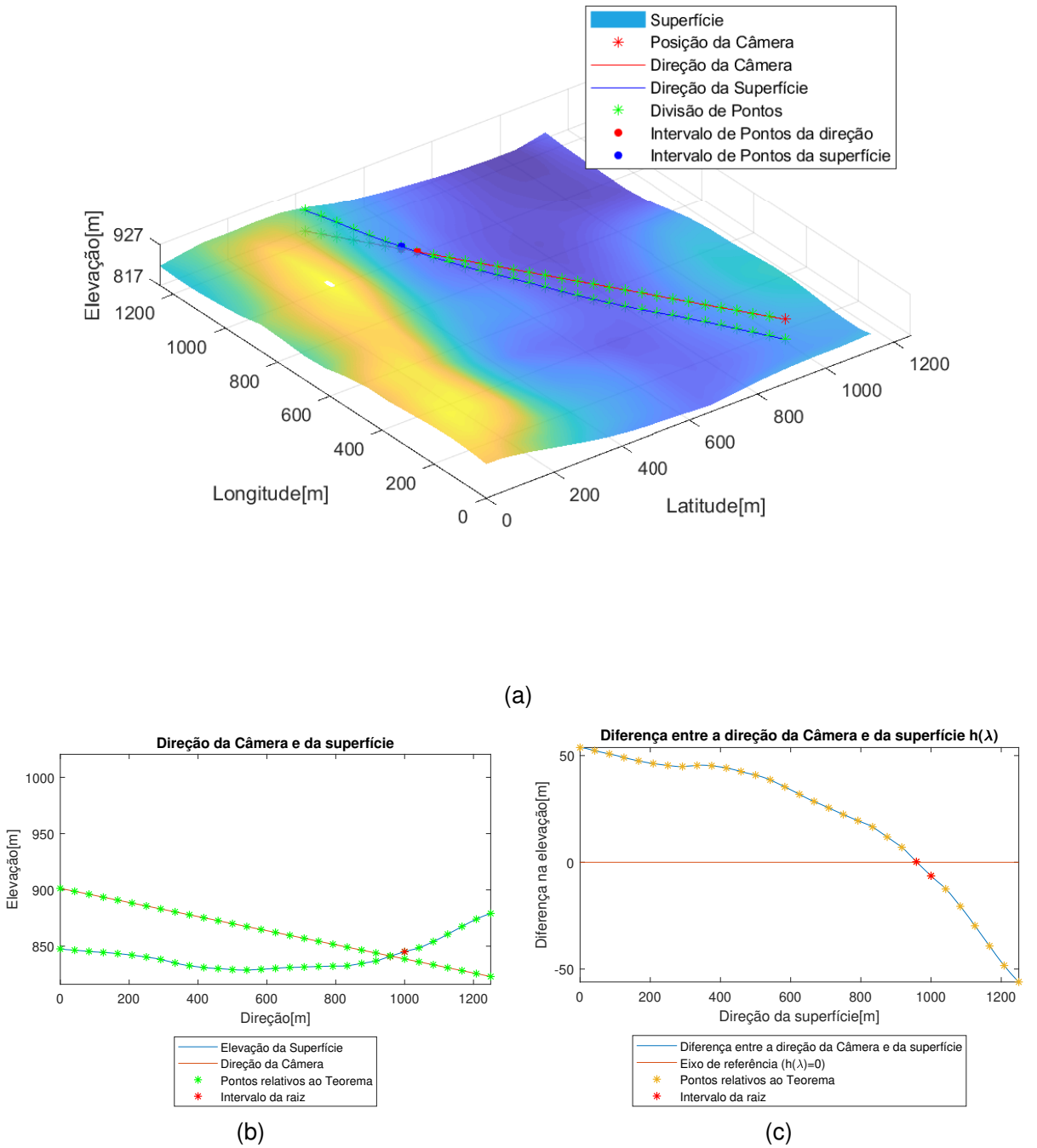



$d = (-0{,}3155,\ 0{,}9040,\ -0{,}2885)$ e abertura da câmera $\alpha = 54.96$.

Em seguida, escolhe-se o *pixel* $e = (510,\ 523)$ da Figura 27 como teste. Deste modo, por meio de inspeção visual da Figura 27, é estabelecida que aquela posição de *pixel e* correspondeu à localização geodésica $p_e = (-19{,}877751, -43{,}978796)$. Assim, demonstrado na Figura 28a, aplica-se o cálculo vetorial da direção do *pixel* e determina-se a direção. A partir disso, utiliza-se interpolação da direção da câmera nos dados da superfície e encontra-se a projeção desta direção sobre o relevo. Esta operação é transformada em um plano, indicada na Figura 28b, onde a reta azul corresponde ao relevo e a reta vermelha à direção da câmera. Os asteriscos nesta figura representam a divisão dos pontos do

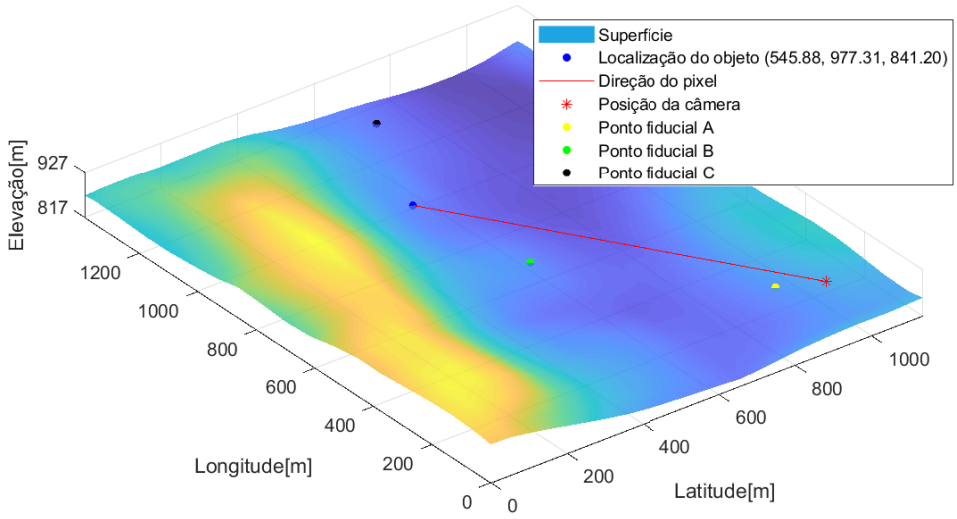
Teorema 1. Em especial, os asteriscos em vermelho correspondem à primeira interseção entre a direção da câmera e o relevo, indicando que naquele intervalo ocorre a troca de sinal na Equação (32). Por meio do ponto de teste, localiza-se a posição do objeto na superfície cujo valor é de $p_e = (545{,}9,\ 977{,}3,\ 841{,}2)$ apontado na Figura 29.

Figura 28 – Resultado da localização do método numérico para o ponto $e$




De forma análoga, tendo como referência a fumaça indicada na Figura 27, o *pixel* $f$ corresponde um possível foco de incêndio. A determinação do *pixel* é encontrada por inspeção visual, obtendo-se o valor de $f = (510,\ 523)$. A partir disso, é considerado um

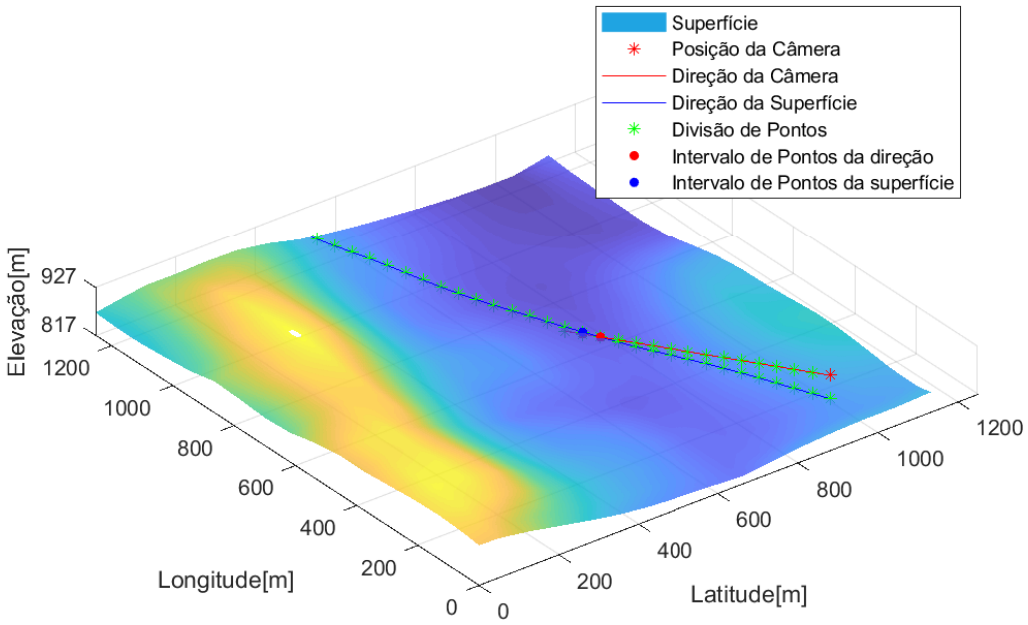
Figura 29 – Localização do ponto $e$



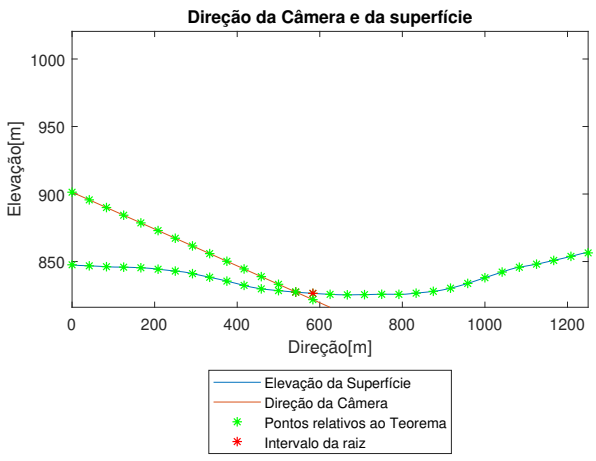

novo caso e determina-se novamente todos os parâmetros da câmera(abertura, direção e a posição da câmera). Obtêm-se os seguintes resultados: $\alpha = 54.97$, $d = (-0{,}3154,\ 0{,}9038,\ -0{,}2894)$ e $p = (973{,}07,\ 119{,}69,\ 901{,}35)$ . A Figura 30 indica a direção daquele *pixel* em vermelho, a projeção desta direção sobre o relevo, pontos do Teorema 1 em verde e o início do intervalo de onde ocorre a interseção entre as funções. Já a Figura 30b e Figura 30c correspondem ao plano das direções presentes Figura 30a. Assim, o ponto $f$ localizado na posição $p_f = (780{,}0,\ 621{,}4,\ 827{,}3)$ e identificado na Figura 31. Finaliza-se este trabalho com a localização dos pontos $e$ e $f$, apresentados na Figura 32.

Tendo em vista a Figura 32, o resultado do ponto *e* destoa do valor obtido pelo método proposto. Em função disso, é feita uma análise do resultado a partir dos pontos oriundos da aquisiçãoda malha em volta do ponto localizado, ilustrado na Figura 33. Os pontos em vermelho representam as aquisições de elevação e o ponto em azul a localização do objeto. Observa-se que a região em torno do ponto em azul, forma-se um plano onde está localizado um edifício cuja região deveria ser a base dele ou a estrutura. Em função disso, constata-se que um dos motivos deste resultado é devido a aquisição dos dados da superfície que não incluía as estruturas ou a precisão adotada pelo SIG.

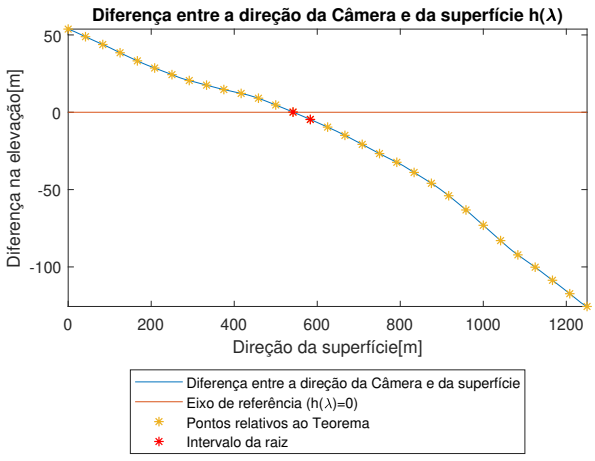
Figura 30 – Resultado da localização do método numérico para o ponto $f$




(a)




(b)




(c)

Figura 31 – Localização do ponto $f$

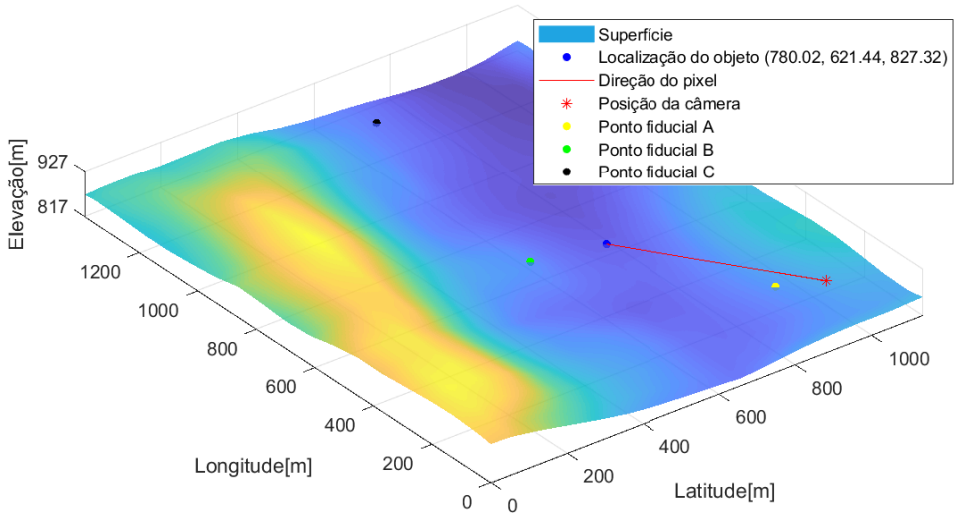
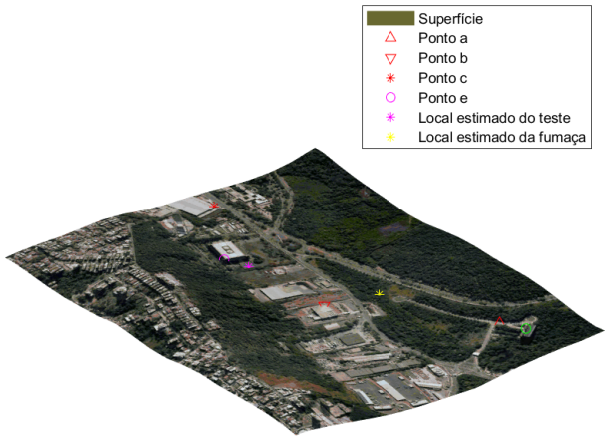



Figura 32 – Resultado final

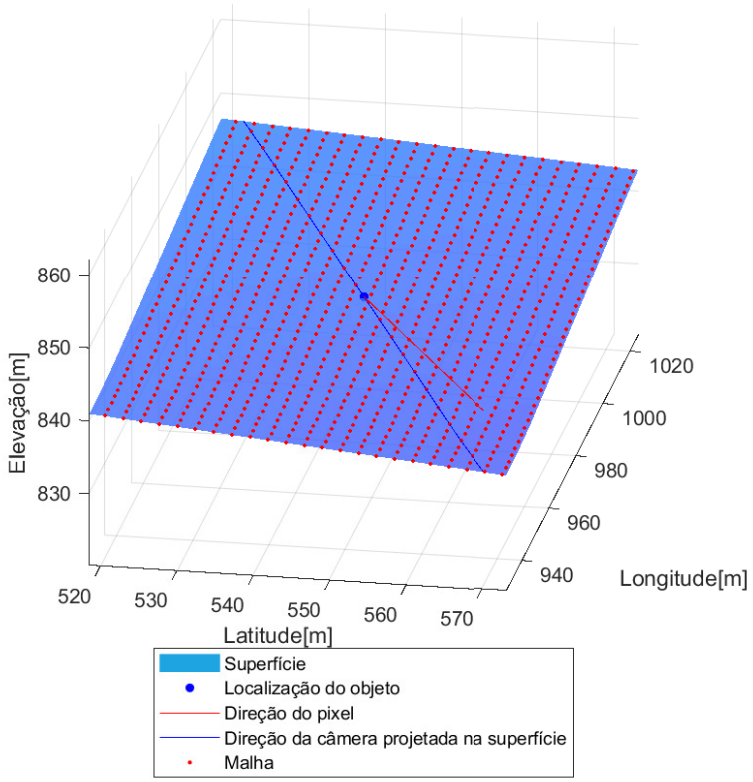


(a) 2D




(b) 3D

Figura 33 – Localização do ponto $e$ com os pontos da malha



# 5 Conclusão

Este trabalho apresentou dois modelos para fazer o controle do posicionamento da câmera: modelo analítico e interpolação. Além disso, utilizou-se sistemas de equações não lineares para determinar os parâmetros da câmera para iniciar a localização. Por fim, empregou-se o método de interpolação de malha e o numérico para determinar a localização do objeto na superfície.

Ambos os modelos, analítico e numérico por interpolação, apresentaram erros aceitáveis, ou seja, próximos à variação da captura de imagem pela câmera (0,1°). A interpolação apresentou um erro(°) MSE $2{,}1181 \times 10^{-04}$, $5{,}0240 \times 10^{-04}$ e $1{,}2455 \times 10^{-03}$, nas direções dx, dy e dz, respectivamente. Enquanto o modelo numérico, apresentou $9{,}2528 \times 10^{-05}$, $3{,}1423 \times 10^{-04}$, $9{,}1202 \times 10^{-05}$, nas direções dx, dy e dz, respectivamente. Destaca-se que o modelo numérico mais fidedigno e é cerca de 3,94 vezes mais preciso na soma dos MSE e para a direção dz, 13,66 vezes mais preciso. Entretanto, o tempo médio para analisar o método analítico foi de $7{,}522 \times 10^{-06}$ segundos e o numérico com $8{,}366 \times 10^{-03}$ segundos os dados obtidos foram 1.100 vezes maior em relação ao outro. Em contrapartida, o modelo analítico pode ser otimizado por meio da alteração dos valores de entrada, incluindo fatores de correções que contribuam para uma precisão mais refinada. Assim, a escolha dos métodos dependerá de acordo com o projeto, que exigirá uma resposta mais rápida ou mais precisa.

A localização do objeto foi obtida por meio da saída dos modelos que apontaram uma direção geográfica. A precisão desta direção está relacionada diretamente ao espaçamento dos pontos da superfície. Cabe ressaltar que para cada direção houve uma curva de relevo diferente a qual demandou nova operação dos cálculos, ou seja, uso de um dos modelos e, em seguida, o algoritmo de interseção do relevo. Assim, caso a superfície local seja muito acidentada o valor da malha deve ser maior do que estimado neste trabalho; caso contrário, igual ou inferior. Por fim, este trabalho alcançou erro próximo à diferença entre captura de imagem e localizou o objeto no ambiente.

## 5.1 Sugestões para trabalhos futuros

- Encontrar uma equação direta com os dados adquiridos. Em posse dos dados de treinamento, é possível determinar a posição em pixel e a direção da câmera em que se deve posicionar no espaço, mas para tanto, deve-se referenciar a posição geográfica daquele pixel (figura *template*). Assim, pode-se determinar o posicionamento da câmera no espaço sem que necessite localizar os parâmetros da câmera novamente;
- Utilizar outros métodos para o controle da câmera. Foi constatado por meio dos dados que é possível controlar a câmera por outros métodos, como redes neurais e

regressões multilineares, com acurácia maior que $97\%$ tanto para o treino quanto para o teste. Entretanto, é necessário também, referenciar a posição geográfica do pixel *template*;

- Aumentar a janela dos resultados dos dados. Para este trabalho foi escolhido aumentar os dados corretos do que a quantidade dos dados, ou seja, decidiu-se ter menos pontos de amostras para garantir que estivessem corretos. Em consequência disso, os erros provenientes de pontos mais distantes do centro foram mitigados na análise;
- Utilizar a aquisição dos dados geográficos com construções. Este trabalho utilizou os dados da API google Elevation que não incluía edifícios ou outras formas que podem alterar a geometria do relevo. Assim, caso a direção estivesse incidindo em uma estrutura, a localização seria encontrada após a estrutura, por exemplo;
- Otimizar a extração de dados. A aquisição foi feita por meio da linguagem python e utilizou-se, principalmente, a CPU para a realização dos cálculos (cerca de 6 dias ininterruptos). Caso utilizasse a GPU ou alterasse a linguagem exclusivamente da extração, esse tempo poderia ser reduzido;
- Incluir o ângulo inicial de elevação da câmera na Equação (17). O produto vetorial é feito pelo eixo $d_z$, do qual se obtém as direções iniciais cujo ângulo de elevação é zero e o ângulo de azimute é obtido por meio de sistemas não-lineares. Isto significa que esse modelo reflete fielmente quando começa na posição inicial de elevação. Os demais ângulos, quando iniciados em outras posições, apresentam um pequeno erro decorrente a está aproximação. Para isto, sugere-se a alteração das configurações da câmera no quesito de *setpoint* das posições iniciais e finais de elevação. Em seguida, compará-las aos limites mecânicos da movimentação de elevação e, por fim, inseri-las na Equação (17).

# Referências


AIRNOW. 2021. Disponível em: <https://www.airnow.gov/>. Acesso em: 26 de fevereiro de 2021. Citado na página 2.

BACKES, A. R.; JUNIOR, J. J. d. M. S. **Introdução à Visão Computacional Usando MATLAB**. [S.l.: s.n.], 2016. 288 p. ISBN 13: 9788550800233. Citado na página 1.

BAJJALI, W. **ArcGIS for Environmental and Water Issues**. [S.l.: s.n.], 2017. ISBN 13: 978-3319611570. Citado na página 1.

BARBER, C. B.; DOBKIN, D. P.; HUHDANPAA, H. The quickhull algorithm for convex hulls. **ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE**, v. 22, n. 4, p. 469–483, 1996. Citado na página 23.

BERRUT, J.-P.; TREFETHEN, L. N. Barycentric lagrange interpolation. **SIAM Review**, v. 46, n. 3, p. 501–517, 2004. Disponível em: <https://doi.org/10.1137/S0036144502417715>. Citado na página 13.

BLENDER. **Blender Foundation**. 2020. Disponível em: <https://www.blender.org/features/>. Acesso em: 04 de janeiro de 2020. Citado na página 35.

BOYCE, W.; DIPRIMA, R. **Equações diferenciais elementares e problemas de valores de contorno**. Livros Técnicos e Científicos, 2002. ISBN 9788521613121. Disponível em: <https://books.google.com.br/books?id=LMXtAAAACAAJ>. Citado na página 13.

BURDEN, R. L.; FAIRES, D. J. **Numerical analysis**. 3. ed. Boston: PWS Publishing Company, 1985. ISBN 0-87150-857-5. Citado na página 27.

BURGER, W.; BURGE, M. J. **Digital Image Processing**: An algorithmic introduction using java. 2. ed. Springer, 2016. ISBN 978-1-4471-6684-9. Disponível em: <https://link.springer.com/book/10.1007/978-3-642-20144-8#toc>. Citado 2 vezes nas páginas 8 e 10.

BURROUGH, P. et al. **Principles of Geographical Information Systems**. [S.l.]: OUP Oxford, 2015. ISBN 9780198742845. Citado na página 1.

BURROUGH, P. A.; MCDONNELL, R. M. . R. A.; LLOYD, C. D. **Principles of Geographical Information Systems**. [S.l.]: Oxford University Press, 1986. ISBN 978-0198233657. Citado na página 18.

CHAPRA, S. C.; CANALE, R. P. **Métodos Numéricos para Engenharia - 5.ed.** McGraw Hill Brasil, 2009. ISBN 9788580550115. Disponível em: <https://books.google.com.br/books?id=ch7MLP20q3MC>. Citado na página 15.

COMASTRI, J. A. **Topografia planimetria**. 2. ed. [S.l.]: Universidade Federal de Viçosa, 1992. Citado na página 17.

COMASTRI, J. A.; TULER, J. C. **Topografia altimetria**. 2. ed. [S.l.]: Universidade Federal de Viçosa, 1987. Citado na página 17.

COMMUNITY, B. O. **Blender - a 3D modelling and rendering package**. Stichting Blender Foundation, Amsterdam, 2020. Disponível em: <http://www.blender.org>. Citado 2 vezes nas páginas 33 e 35.

CORKE, P. **Robotics,Vision and Control**: Fundamental algorithms in matlab. Springer, 2011. ISBN 978-3-642-20143-1. Disponível em: <https://link.springer.com/book/10.1007/978-3-642-20144-8#toc>. Citado 2 vezes nas páginas 10 e 11.

D'AGOSTINO, R.; PEARSON, E. S. Tests for departure from normality. empirical results for the distributions of b2 and b1. **Biometrika**, [Oxford University Press, Biometrika Trust], v. 60, n. 3, p. 613–622, 1973. ISSN 00063444. Disponível em: <http://www.jstor.org/stable/2335012>. Citado na página 36.

D'AGOSTINO, R. B. An omnibus test of normality for moderate and large size samples. **Biometrika**, [Oxford University Press, Biometrika Trust], v. 58, n. 2, p. 341–348, 1971. ISSN 00063444. Disponível em: <http://www.jstor.org/stable/2334522>. Citado na página 36.

DAVIS, W.; SARASUA, W.; MCCORMAC, J. **Topografia**. LTC, 2016. ISBN 9788521627883. Disponível em: <https://books.google.com.br/books?id=MZFavgAACAAJ>. Citado 3 vezes nas páginas 16, 17 e 18.

FITZ, P. R. **Geoprocessamento sem complicação**. Oficina de Textos, 2008. ISBN 978-85-86238-82-6. Disponível em: <https://www.lojaofitexto.com.br/geoprocessamento-sem-complicacao/p>. Citado na página 18.

HARDER, C. **The ArcGIS Book: 10 Big Ideas about Applying Geography to Your World**. 1st. ed. [S.l.]: Esri Press, 2015. ISBN 1589484495. Citado na página 3.

HO, G. T. S. et al. A computer vision-based roadside occupation surveillance system for intelligent transport in smart cities. **Sensors**, MDPI AG, v. 19, n. 8, p. 1796, Apr 2019. ISSN 1424-8220. Disponível em: <http://dx.doi.org/10.3390/s19081796>. Citado na página 1.

HOEKENDIJK, J. P. A. et al. Estimating the spatial position of marine mammals based on digital camera recordings. **Ecology and Evolution**, v. 5, n. 3, p. 578–589, 2015. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.1353>. Citado na página 1.

HÖFLING, T. T. A. et al. Read my face: Automatic facial coding versus psychophysiological indicators of emotional valence and arousal. **Frontiers in psychology**, v. 11, p. 1388, 2020. ISSN 1664-1078. Disponível em: <https://europepmc.org/articles/PMC7316962>. Citado na página 1.

HUANG, Y. et al. Epan: Effective parts attention network for scene text recognition. **Neurocomputing**, v. 376, p. 202 – 213, 2020. ISSN 0925-2312. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231219313839>. Citado na página 1.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Citado na página 33.

ITSEEZ. **The OpenCV Reference Manual**. 2.4.9.0. ed. [S.l.], 2014. Citado na página 8.

ITSEEZ. **Open Source Computer Vision Library**. 2015. <https://github.com/itseez/opencv>. Citado na página 33.

JONES, E. et al. **SciPy: Open source scientific tools for Python**. 2001–. Disponível em: <http://www.scipy.org/>. Citado 2 vezes nas páginas 33 e 34.

KLETTE, R. **Concise Computer Vision: An Introduction into Theory and Algorithms**. [S.l.]: Springer Publishing Company, Incorporated, 2014. ISBN 1447163192. Citado 4 vezes nas páginas 1, 5, 6 e 7.

KLUYVER, T. et al. Jupyter notebooks – a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCHMIDT, B. (Ed.). **Positioning and Power in Academic Publishing: Players, Agents and Agendas**. [S.l.], 2016. p. 87 – 90. Citado na página 33.

KULIKAJEVAS, A. et al. Reconstruction of 3d object shape using hybrid modular neural network architecture trained on 3d models from shapenetcore dataset. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n. 7, p. 1553, 2019. Citado na página 1.

MATLAB. **9.7.0.1190202 (R2019b)**. Natick, Massachusetts: The MathWorks Inc., 2018. Citado 2 vezes nas páginas 33 e 34.

MATLAB. **2-D Gaussian filtering of images**. 2020. Disponível em: <https://www.mathworks.com/help/images/ref/imgaussfilt.html>. Acesso em: 19 de março de 2020. Citado na página 7.

MATLAB. **Display Separated Color Channels of RGB Image**. 2020. Disponível em: <https://www.mathworks.com/help/images/display-separated-color-channels-of-rgb-image.html>. Acesso em: 19 de março de 2020. Citado na página 7.

MCKINNEY, W. et al. Data structures for statistical computing in python. In: AUSTIN, TX. **Proceedings of the 9th Python in Science Conference**. [S.l.], 2010. v. 445, p. 51–56. Citado 2 vezes nas páginas 33 e 34.

MILOSAVLJEVIć, A. et al. A method for estimating surveillance video georeferences. **ISPRS International Journal of Geo-Information**, MDPI AG, v. 6, n. 7, p. 211, Jul 2017. ISSN 2220-9964. Disponível em: <http://dx.doi.org/10.3390/ijgi6070211>. Citado na página 1.

ON a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 18, n. 1, p. 50–60, 1947. ISSN 00034851. Disponível em: <http://www.jstor.org/stable/2236101>. Citado na página 36.

OpenStreetMap contributors. **Planet dump retrieved from https://planet.osm.org** . 2017. <https://www.openstreetmap.org>. Citado na página 3.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python . **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 33 e 34.

Ragesh, N. K.; Rajesh, R. Pedestrian detection in automotive safety: Understanding state-of-the-art. **IEEE Access**, v. 7, p. 47864–47890, 2019. Citado na página 1.

RUGGIERO, M.; LOPES, V. da R. **Cálculo numérico: aspectos teóricos e computacionais**. Pearson Makron Books, 1996. ISBN 9788534602044. Disponível em: <https://books.google.com.br/books?id=kuRDAAAACAAJ>. Citado 2 vezes nas páginas 13 e 14.

SCHWAB, K. **The Fourth Industrial Revolution**. USA: Crown Publishing Group, 2017. ISBN 1524758868. Citado na página 2.

SCIPY. **SciPy v1.4.1 Reference Guide**: Scipy interpolate linearndinterpolator. 2020. Disponível em: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.LinearNDInterpolator.html>. Acesso em: 23 de março de 2020. Citado na página 23.

SHETTY, S. J. Vision-based inspection system employing computer vision & neural networks for detection of fractures in manufactured components. **CoRR**, abs/1901.08864, 2019. Disponível em: <http://arxiv.org/abs/1901.08864>. Citado na página 1.

SILVA, R. et al. On the applicability of inverse perspective mapping for the forward distance estimation based on the hsv colormap. In: . [S.l.: s.n.], 2017. Citado na página 2.

SZELISKI, R. **Computer Vision**. 1. ed. [S.l.]: Springer-Verlag London, 2011. ISBN 978-1-84882-935-0. Citado 4 vezes nas páginas 1, 6, 13 e 14.

TATEOSIAN, L. **Python For ArcGIS**. [S.l.: s.n.], 2015. 538 p. ISBN 978-3-319-18398-5. Citado na página 1.

TUKEY, J. W. **Exploratory Data Analysis**. [S.l.]: Addison-Wesley, 1977. Citado na página 31.

WALT, S. v. d.; COLBERT, S. C.; VAROQUAUX, G. The numpy array: A structure for efficient numerical computation. **Computing in Science & Engineering**, v. 13, n. 2, p. 22–30, 2011. Disponível em: <https://aip.scitation.org/doi/abs/10.1109/MCSE.2011.37>. Citado na página 33.

XIE, Y. et al. Integration of multi-camera video moving objects and gis. **ISPRS International Journal of Geo-Information**, v. 8, n. 12, 2019. ISSN 2220-9964. Disponível em: <https://www.mdpi.com/2220-9964/8/12/561>. Citado na página 2.

ZHOU, Q. et al. An automatic surface defect inspection system for automobiles using machine vision methods. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n. 3, p. 644, 2019. Citado na página 1.

ZHU, J.; FANG, Y. Learning object-specific distance from a monocular image. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 3839–3848. Citado na página 2.

ZILL, D. **Equações Diferenciais com aplicações em Modelagem**. Thomson, 2003. ISBN 9788522103140. Disponível em: <https://books.google.com.br/books?id=qdcM6PrecU4C>. Citado na página 13.