

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
FACULDADE DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Maria Júlia Rosa Aguiar

Montagem de Panorâmicas 360 através de Multi-band Blending com ajuste  
Ótimo via Meta-heurísticas

Juiz de Fora  
2021

Maria Júlia Rosa Aguiar

Montagem de Panorâmicas 360 através de Multi-band Blending com ajuste  
Ótimo via Meta-heurísticas

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas de Energia Elétrica

Orientador: Prof. Dr. Leonardo de Mello Honório

Coorientador: Prof. Dr. Ivo Chaves da Silva Junior

Juiz de Fora

2021

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Aguiar, Maria Júlia Rosa .

Montagem de Panorâmicas 360 através de Multi-band Blending com  
ajuste Ótimo via Meta-heurísticas / Maria Júlia Rosa Aguiar. – 2021.

106 f. : il.

Orientador: Leonardo de Mello Honório

Coorientador: Ivo Chaves da Silva Junior

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade  
de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2021.

1. Panoramas. 2. Bundle Adjustment. 3. Metaheurísticas. I. Honório,  
Leonardo de Mello, orient. II. Junior, Ivo Chaves da Silva , co-orient.  
Montagem de Panorâmicas 360 através de Multi-band Blending com ajuste  
Ótimo via Metaheurísticas.

Maria Júlia Rosa Aguiar

Montagem de Panorâmicas 360 através de Multi-band Blending com ajuste Ótimo via Meta-heurísticas

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas de Energia Elétrica

Aprovada em 23 de março de 2021

BANCA EXAMINADORA



Digitally signed by Leonardo de Mello Honório  
DN: C=BR, OU=Faculdade de Engenharia,  
O=Universidade Federal de Juiz de Fora,  
CN=Leonardo de Mello Honório,  
E=leonardo.honorio@ufjf.edu.br  
Reason: I am approving this document  
Location: Juiz de Fora  
Date: 2021.06.16 09:19:24-03'00'  
Foxit Reader Version: 10.1.23

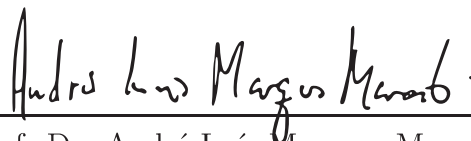
---

Prof. Dr. Leonardo de Mello Honório - Orientador  
Universidade Federal de Juiz de Fora



---

Professor Dr. Ivo Chaves da Silva Junior - Coorientador  
Universidade Federal de Juiz de Fora



---

Prof. Dr. André Luís Marques Marcato  
Universidade Federal de Juiz de Fora



---

Prof. Dr. Carlos Henrique Valério de Moraes  
Universidade Federal de Itajubá

Dedico aos meus pais, por sempre acreditarem em mim e por todo amor e carinho que recebi durante a elaboração desse trabalho.

## AGRADECIMENTOS

Agradeço, primeiramente, a Deus pelo dom da vida, por me dar energia e forças para concluir esse trabalho em meio a tantos momentos difíceis desta caminhada.

Aos meus pais, Sandra e Sebastião, por sempre acreditarem em mim e não medirem esforços para me dar todo o apoio necessário.

Ao meu irmão, Felipe, por estar sempre presente nos momentos que precisei.

Aos meus companheiros de laboratório que tornaram essa jornada mais fácil e divertida e por todo conhecimento compartilhado. Um agradecimento especial ao meu colega, Tiago Alves, pela grande contribuição e apoio durante essa dissertação disponibilizando seu tempo para troca de ideias.

Aos meus orientadores, professores Leonardo e Ivo, pela paciência, atenção e por estarem sempre a disposição para me ajudar e fornecerem todos os recursos necessários para a confecção deste trabalho.

E, aos meus familiares e amigos que direta ou indiretamente fizeram parte desse processo.

À todos, meus sinceros agradecimentos.

## RESUMO

O *Stitching* de imagens é o alinhamento de múltiplas imagens em composições maiores que representam partes de uma cena 3D. A construção automática de panoramas a partir de múltiplas imagens digitais é uma área de grande importância, encontrando aplicações em diferentes setores como sensoriamento remoto, inspeção e manutenção em ambientes de trabalho e medicina. Diversos algoritmos de mosaico de imagens foram propostos nos últimos anos. Ao mesmo tempo, o advento contínuo de novos métodos de mosaico torna muito difícil escolher um algoritmo apropriado para uma finalidade específica. Este trabalho apresenta técnicas para a montagem de panorâmicas 360° a partir de imagens tiradas por um sistema robótico desenvolvido. Foram utilizados os algoritmos de otimização bioinspirados *Grey Wolf Optimizer* e *Bat Algorithm* com intuito de se obter um ajuste ótimo no posicionamento das imagens sendo responsáveis por um *Bundle adjustment*. Após, o ajustamento das imagens para se corrigir possíveis diferenças de coloração e discrepâncias nas imagens utiliza-se a metodologia *Multi-band Blending* para se obter, ao final, uma imagem uniforme. A comparação entre os algoritmos envolverá análise da variabilidade das soluções e características de convergência.

Palavras-chave: Panoramas. *Bundle Adjustment*. Metaheurísticas. *Multi-band Blending*.

## ABSTRACT

Image stitching is the alignment compositions of multiple images that represent parts of a 3D scene. The automatic construction of panoramas from multiple digital images is a technic of great importance, finding applications in different areas such as remote sensing, inspection and maintenance in many work environments and medicine field. In recent years, a myriad of image mosaic algorithms has been proposed. Therefore, it is extremely difficult to choose an appropriate algorithm for a specific purpose. This work presents a set of techniques for assembling 360° panoramas from images taken by a robotic system. The bio-inspired optimization algorithms Gray Wolf Optimizer and Bat Algorithm were used to obtain the optimal fitness value in the positioning of the images responsible for a Bundle adjustment. Afterwards, to adjust the possible color differences and discrepancies in the images, were used the Multi-band Blending methodology, resulting at the end, a uniform image. The comparison between the algorithms will involve analysis of the variability of the solutions and convergence characteristics.

Keywords: Panoramas. Bundle Adjustment. Metaheuristics. Multi-band Blending.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Visão Humana x Visão Computacional . . . . .	14
Figura 2 – Imagem Panorâmica 360° . . . . .	14
Figura 3 – Representação de <i>Stitching</i> de Imagens . . . . .	16
Figura 4 – Matriz de imagem(direita), como armazenada em um computador, de uma pequena porção da imagem (esquerda representada pelo quadrado vermelho) . . . . .	21
Figura 5 – Imagem RGB Representada por Três Matrizes . . . . .	21
Figura 6 – Modelo da Câmera Pinhole . . . . .	22
Figura 7 – Geometria Epipolar . . . . .	25
Figura 8 – Detector de <i>Harris Corner</i> . . . . .	27
Figura 9 – Detecção de Pontos de Interesse - Adaptado de Lowe (59) . . . . .	28
Figura 10 – Descrição do ponto - SIFT: À esquerda tem-se a representação dos gradientes dos pixels e à direita tem-se sub-região de 4x4 pixels . . . . .	30
Figura 11 – Exemplo de Correspondência de Características entre um Par de Imagens . . . . .	31
Figura 12 – <i>Matches</i> Incorretos entre Duas Imagens . . . . .	31
Figura 13 – Estimacão de reta a um conjunto de dados bidimensionais . . . . .	34
Figura 14 – Exemplo de Erro de Re-projecção . . . . .	35
Figura 15 – Panorâmica sem <i>Bundle Adjustment</i> Aplicado . . . . .	36
Figura 16 – Projecção Cilíndrica . . . . .	37
Figura 17 – Projecção Esférica . . . . .	38
Figura 18 – Etapas de desenvolvimento . . . . .	39
Figura 19 – Visão Geral da Estrutura . . . . .	40
Figura 20 – Fluxograma para Extração e Detecção de <i>Features</i> e <i>Matches</i> . . . . .	41
Figura 21 – Correspondência entre Imagens Vizinhas com Destaque nos <i>Outliers</i> . . . . .	42
Figura 22 – Correspondências entre Imagens após Filtragem . . . . .	43
Figura 23 – Hierarquia dos Lobos Cinzentos . . . . .	46
Figura 24 – Pseudocódigo do <i>Grey Wolf Optimizer (GWO)</i> . . . . .	47
Figura 25 – Sistema de Ecolocalização . . . . .	49
Figura 26 – Pseudocódigo do <i>Bat Algorithm (BA)</i> . . . . .	50
Figura 27 – Exemplificação dos Indivíduos . . . . .	52
Figura 28 – Cantos do <i>Frustum</i> da Imagem . . . . .	53
Figura 29 – Projecção Esférica de Imagens . . . . .	56
Figura 30 – Combinação e Mesclagem de Imagens com <i>Multi-band Blending</i> . . . . .	57
Figura 31 – Pirâmide Gaussiana . . . . .	57
Figura 32 – Pirâmide Laplaciana . . . . .	58
Figura 33 – Protótipo Scanner 3D . . . . .	61
Figura 34 – Faculdade de Artes e Design (UFJF) - Vista 1 . . . . .	63
Figura 35 – Faculdade de Artes e Design (UFJF) - Vista 2 . . . . .	64

Figura 36 – Faculdade de Artes e Design (UFJF) - Vista 3 . . . . .	64
Figura 37 – Subestação de Santos Dumont . . . . .	65
Figura 38 – Gráficos de Convergência em 10 simulações - Caso 1 . . . . .	66
Figura 39 – <i>Boxplot</i> dos Valores Ótimos da Função Objetivo nas 10 Simulações para o Caso 1 . . . . .	67
Figura 40 – Comparação das Melhores soluções em Relação à Original - Caso 1 . .	68
Figura 41 – Faculdade de Artes e Design (UFJF) Vista 1 sem <i>Multi-band Blending</i> - <i>Bat</i> <i>Algorithm</i> . . . . .	69
Figura 42 – Faculdade de Artes e Design (UFJF) Vista 1 sem <i>Multi-band Blending</i> - <i>Grey</i> <i>Wolf Optimizer</i> . . . . .	69
Figura 43 – Gráficos de Convergência em 10 simulações - Caso 2 . . . . .	70
Figura 44 – <i>Boxplot</i> dos Valores Ótimos da Função Objetivo nas 10 Simulações para o Caso 2 . . . . .	71
Figura 45 – Comparação das Melhores soluções em Relação à Original - Caso 2 . .	72
Figura 46 – Faculdade de Artes e Design (UFJF) Vista 2 sem <i>Multi-band Blending</i> - <i>Bat</i> <i>Algorithm</i> . . . . .	73
Figura 47 – Faculdade de Artes e Design (UFJF) Vista 2 sem <i>Multi-band Blending</i> - <i>Grey</i> <i>Wolf Optimizer</i> . . . . .	73
Figura 48 – Gráficos de Convergência em 10 simulações - Caso 3 . . . . .	74
Figura 49 – <i>Boxplot</i> dos Valores Ótimos da Função Objetivo nas 10 Simulações para o Caso 3 . . . . .	75
Figura 50 – Comparação das Melhores soluções em Relação à Original - Caso 3 . .	76
Figura 51 – Faculdade de Artes e Design (UFJF) Vista 3 sem <i>Multi-band Blending</i> - <i>Bat</i> <i>Algorithm</i> . . . . .	77
Figura 52 – Faculdade de Artes e Design (UFJF) Vista 3 sem <i>Multi-band Blending</i> - <i>Grey</i> <i>Wolf Optimizer</i> . . . . .	78
Figura 53 – Gráficos de Convergência em 10 simulações - Caso 4 . . . . .	78
Figura 54 – <i>Boxplot</i> dos Valores Ótimos da Função Objetivo nas 10 Simulações para o Caso 4 . . . . .	79
Figura 55 – Comparação das Melhores soluções em Relação à Original - Caso 4 . .	80
Figura 56 – Subestação de Energia sem <i>Multi-band Blending</i> - <i>Bat Algorithm</i> . . .	81
Figura 57 – Subestação de Energia sem <i>Multi-band Blending</i> - <i>Grey Wolf Optimizer</i>	81
Figura 58 – Faculdade de Artes e Design (UFJF) Vista 1 com <i>Multi-band Blending</i> - <i>Bat</i> <i>Algorithm</i> . . . . .	82
Figura 59 – Faculdade de Artes e Design (UFJF) Vista 1 com <i>Multi-band Blending</i> - <i>Grey</i> <i>Wolf Optimizer</i> . . . . .	82
Figura 60 – Faculdade de Artes e Design (UFJF) Vista 2 com <i>Multi-band Blending</i> - <i>Bat</i> <i>Algorithm</i> . . . . .	83

Figura 61 – Faculdade de Artes e Design (UFJF) Vista 2 com <i>Multi-band Blending - Grey Wolf Optimizer</i> . . . . .	83
Figura 62 – Faculdade de Artes e Design (UFJF) Vista 3 com <i>Multi-band Blending - Bat Algorithm</i> . . . . .	84
Figura 63 – Faculdade de Artes e Design (UFJF) Vista 3 com <i>Multi-band Blending - Grey Wolf Optimizer</i> . . . . .	84
Figura 64 – Subestação de Santos Dumont com <i>Multi-band Blending - Bat Algorithm</i>	85
Figura 65 – Subestação de Santos Dumont com <i>Multi-band Blending - Grey Wolf Optimizer</i> . . . . .	85
Figura 66 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista 1 sem <i>Multi-band Blending - Bat Algorithm</i> . . . . .	97
Figura 67 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista 1 sem <i>Multi-band Blending - Grey Wolf Optimizer</i> . . . . .	98
Figura 68 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista 2 sem <i>Multi-band Blending - Bat Algorithm</i> . . . . .	99
Figura 69 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista 2 sem <i>Multi-band Blending - Grey Wolf Optimizer</i> . . . . .	101
Figura 70 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista2 sem <i>Multi-band Blending - Bat Algorithm</i> . . . . .	102
Figura 71 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista 3 sem <i>Multi-band Blending - Grey Wolf Optimizer</i> . . . . .	103
Figura 72 – Resultados das 10 Simulações da Subestação de Energia sem <i>Multi-band Blending - Bat Algorithm</i> . . . . .	104
Figura 73 – Resultados das 10 Simulações da Subestação de Energia sem <i>Multi-band Blending - Grey Wolf Optimizer</i> . . . . .	105

## LISTA DE TABELAS

Tabela 1 – Configuração de Simulação . . . . .	60
Tabela 2 – Valores de Parâmetros para os Algoritmos Comparativos . . . . .	61
Tabela 3 – Comparação da Função Objetivo - Caso 1 . . . . .	67
Tabela 4 – Comparação de Valores de Função Objetivo obtidos pelas Meta-heurísticas Extraídos do <i>Boxplot</i> para o Caso 1 . . . . .	68
Tabela 5 – Tempo Médio Computacional para o Caso 1 . . . . .	69
Tabela 6 – Comparação da Função Objetivo - Caso 2 . . . . .	71
Tabela 7 – Comparação de Valores de Função Objetivo obtidos pelas Meta-heurísticas Extraídos do <i>Boxplot</i> para o Caso 2 . . . . .	72
Tabela 8 – Tempo Médio Computacional para o Caso 2 . . . . .	73
Tabela 9 – Comparação da Função Objetivo - Caso 3 . . . . .	75
Tabela 10 – Comparação de Valores de Função Objetivo obtidos pelas Meta-heurísticas Extraídos do <i>Boxplot</i> para o Caso 3 . . . . .	76
Tabela 11 – Tempo Médio Computacional para o Caso 3 . . . . .	77
Tabela 12 – Comparação da Função Objetivo - Caso 4 . . . . .	79
Tabela 13 – Comparação de Valores de Função Objetivo obtidos pelas Meta-heurísticas Extraídos do <i>Boxplot</i> para o Caso 4 . . . . .	80
Tabela 14 – Tempo Médio Computacional para o Caso 4 . . . . .	81

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>13</b>
1.1	IDENTIFICAÇÃO DO PROBLEMA E MOTIVAÇÃO . . . . .	16
1.2	OBJETIVOS . . . . .	18
<b>1.2.1</b>	<b>Objetivos Gerais . . . . .</b>	<b>18</b>
<b>1.2.2</b>	<b>Objetivos Específicos . . . . .</b>	<b>18</b>
1.3	ESTRUTURA DO TRABALHO . . . . .	19
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>20</b>
2.1	PROCESSAMENTO DE IMAGENS . . . . .	20
2.2	MODELAGEM DE CÂMERA . . . . .	21
2.3	PARÂMETROS DA CÂMERA . . . . .	23
<b>2.3.1</b>	<b>Parâmetros Intrínsecos . . . . .</b>	<b>23</b>
<b>2.3.2</b>	<b>Parâmetros Extrínsecos . . . . .</b>	<b>24</b>
2.4	GEOMETRIA EPIPOLAR . . . . .	24
<b>2.4.1</b>	<b>Matriz Fundamental . . . . .</b>	<b>25</b>
2.5	PONTOS DE INTERESSE . . . . .	26
2.6	DESCRITORES . . . . .	29
<b>2.6.1</b>	<b>Scale-Invariant Feature Transform (SIFT) . . . . .</b>	<b>29</b>
2.7	CORRESPONDÊNCIAS DE PONTOS . . . . .	30
2.8	HOMOGRAFIA . . . . .	32
2.9	RANDOM SAMPLE CONSENSUS (RANSAC) . . . . .	33
2.10	<i>BUNDLE ADJUSTMENT</i> . . . . .	34
2.11	PROJEÇÃO DE IMAGENS . . . . .	36
<b>2.11.1</b>	<b>Projeção Cilíndrica . . . . .</b>	<b>36</b>
<b>2.11.2</b>	<b>Projeção Esférica . . . . .</b>	<b>37</b>
<b>3</b>	<b>METODOLOGIA PROPOSTA E ALGORITMOS UTILIZADOS</b>	<b>39</b>
3.1	EXTRAÇÃO E DETECÇÃO DE <i>FEATURES</i> E <i>MATCHES</i> . . . . .	40
3.2	ETAPA DE <i>BUNDLE ADJUSTMENT</i> . . . . .	43
<b>3.2.1</b>	<b>Modelagem matemática do problema . . . . .</b>	<b>43</b>
<b>3.2.2</b>	<b>Métodos de Otimização . . . . .</b>	<b>45</b>
<i>3.2.2.1</i>	<i>Grey Wolf Optimizer (GWO) . . . . .</i>	<i>46</i>
<i>3.2.2.2</i>	<i>Bat Algorithm (BA) . . . . .</i>	<i>49</i>
<b>3.2.3</b>	<b>Formulação do problema . . . . .</b>	<b>51</b>
3.3	<i>STITCHING</i> . . . . .	53
<b>3.3.1</b>	<b>Multi-band Blending . . . . .</b>	<b>56</b>
<b>4</b>	<b>RESULTADOS . . . . .</b>	<b>60</b>
4.1	PARÂMETROS UTILIZADOS . . . . .	60
4.2	AQUISIÇÃO DE IMAGENS . . . . .	61

4.3	RESULTADOS INICIAIS . . . . .	63
4.4	RESULTADOS DAS TÉCNICAS DE OTIMIZAÇÃO . . . . .	65
4.4.1	Estudo de Caso 1 - Faculdade de Artes e Design (UFJF) . . . .	66
4.4.2	Estudo de Caso 2 - Faculdade de Artes e Design (UFJF) . . . .	70
4.4.3	Estudo de Caso 3 - Faculdade de Artes e Design (UFJF) . . . .	74
4.4.4	Estudo de Caso 4 - Subestação Santos Dumont . . . . .	78
4.5	RESULTADOS OBTIDOS PELO <i>MULTI-BAND BLENDING</i> . . . . .	82
4.6	ANÁLISE DOS RESULTADOS . . . . .	86
5	<b>CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>	<b>87</b>
5.1	CONCLUSÃO . . . . .	87
5.2	TRABALHOS FUTUROS . . . . .	88
	<b>REFERÊNCIAS . . . . .</b>	<b>89</b>
	<b>APÊNDICE A – Resultados Encontrados pelos Métodos de Otimização em cada Simulação . . . . .</b>	<b>97</b>

## 1 INTRODUÇÃO

A visão permite que os seres humanos, observem e tenham consciência do mundo que os cerca. A visão humana consiste em várias etapas. Primeiro, a luz é refletida na imagem e entra nos olhos pela córnea. Em seguida, a córnea direciona a luz para as pupilas e a íris, que trabalham juntas para controlar a quantidade de luz que entra no olho. Assim que a luz passa pela córnea, ela entra na retina a qual possui sensores especiais chamados cones e bastonetes, que estão envolvidos na visão de cores (1). Portanto, a visão é, uma tarefa de processamento de informação, pois para entender o que é uma imagem, o cérebro deve ser capaz de representar essa informação como: cor, forma, movimento e profundidade, por exemplo.

Com o intuito de automatizar tarefas que dependem da percepção humana, surgiu a visão computacional. Um campo de Inteligência Artificial e Ciência da Computação que visa dar aos computadores uma compreensão visual do mundo, manipulando e interpretando eletronicamente as imagens através de três componentes principais de processamento: aquisição de imagem, processamento de imagem, análise e compreensão de imagens. Trata-se de automatização das informações das imagens, podendo significar modelos 3D, posição de câmeras, detecção de objetos e reconhecimento de imagens (2). É um ramo que visa modelar matematicamente os processos de percepção visual em seres vivos, e gerar modelos, algoritmos e programas que permitem a simulação dessas habilidades visuais utilizando a capacidade de computadores.

A Figura 1 apresenta uma analogia do comportamento no processamento de informações entre a visão humana e a visão computacional. Percebe-se, então, que a visão computacional, pode ser definida como a simulação tecnológica da visão humana. É um campo que se utiliza de software e hardware para tornar possível que computadores ou máquinas vejam, identifiquem e processem imagens da mesma forma que os humanos. Entretanto, o computador, visualiza as imagens como matrizes (como será explicado no Capítulo 2).

Ao longo dos anos, diversas técnicas e métodos de simulação da visão computacional têm sido desenvolvidos permitindo que os mesmos possam extrair informações de modo a entender o ambiente que os cercam através do uso de sensores e câmeras sendo tratadas como dispositivos de detecção do ambiente. Essa tecnologia tornou-se importante e, tem sido utilizada como parte central de um vasto número de aplicações atualmente, tendo evoluído como um assunto multidisciplinar com foco na extração, representação e uso de informações visuais em inteligência artificial (3), robótica (4), análise de imagens médicas (5, 6), sistemas de vigilância (7) e outras aplicações (8, 9, 10).

À medida que computadores e câmeras se tornaram mais acessíveis, o uso de imagens digitais cresceram. Tal fato acabou propiciando a utilização de imagens fotográficas para

Figura 1 – Visão Humana x Visão Computacional



Fonte: Elaborada pela autora (2021)

a resolução de muitos problemas do cotidiano. A imagem panorâmica, por exemplo, é um parte importante deste grande uso de equipamentos digitais. Com o crescente desenvolvimento tecnológico se tornaram bastante úteis em diversas áreas como *tours* virtuais para navegação e exploração de ambientes, realidade virtual e aumentada e direção não tripulada, por terem seus recursos de visualização completa dada por  $360^\circ \times 180^\circ$  para acessar diferentes pontos de vista (11). A Figura 2 é uma representação de um panorama  $360^\circ$  no sentido horizontal e  $180^\circ$  no sentido vertical ( $360^\circ \times 180^\circ$ ).

Figura 2 – Imagem Panorâmica  $360^\circ$ 

Fonte: Elaborada pela autora (2021)

Mas a necessidade de se obter panoramas existia desde o início da fotografia, pois o campo de visão da câmera é menor do que o campo de visão humano. Além disso, muitas vezes, objetos grandes não podem ser capturados em uma única imagem (12). A utilização de imagens panorâmicas vem se tornando cada vez mais comum, podendo obter estas por



meio de câmeras ou lentes especiais. Um tipo de câmera muito empregada para geração de panoramas é a câmera olho de peixe (*fisheye*) pois conseguem obter um grande campo de visão (13). Porém, tais equipamentos possuem lentes com distorções geométricas, incluindo distorções tangenciais e em barril (14). Dessa forma, uma outra maneira de se construir imagens panorâmicas é a partir de imagens capturadas em um local onde estas devem ser alinhadas e mescladas para formar a imagem final (15). A geração dessas imagens a partir de sequências de imagens é um dos mais antigos campos da visão computacional (16).

Para a criação de imagens panorâmicas a partir de múltiplas imagens utiliza-se, geralmente, o método de *stitching* que é a junção das imagens com base nas suas áreas de sobreposição (17). Tal metodologia parte do princípio de unir imagens com partes sobrepostas que foram fotografadas em diferentes tempos, ângulos visuais ou sensores, a fim de obter um imagem de alta resolução contínua.

Segundo Ye (18), o mosaico de imagens como também são chamados os panoramas é o processo de efetivamente aumentar o campo de visão de uma câmera através da combinação de várias visões parciais de uma cena em uma única visão mais abrangente.

Panoramas 360° são muito utilizados para retratar um ambiente, empresas como a *Google* e a *Microsoft* retratam ruas e rodovias por meio deste recurso, assim o usuário pode “caminhar” pelas vias de uma cidade que desejar sem sair de casa. Outras empresas fazem o uso de panoramas 360° para apresentar melhor seus produtos e serviços, como por exemplo, sites de hotéis e pousadas que possibilitam aos clientes fazer um *tour* virtual nos ambientes do estabelecimento (19).

Também são muito utilizadas em ambiente industriais em locais onde necessita-se de inspeção e manutenção de ambientes. O trabalho (20) aborda o monitoramento de infraestrutura civil que dependem de abordagens visuais para a avaliação das condições de tais sistemas, não necessitando que um inspetor viaje até o local, sendo capaz de avaliar remotamente as estruturas. Em (21) tem-se a criação de imagens panorâmicas para inspeção em túneis. Outra abordagem foi a utilização de panorâmicas para manter a segurança dos seres humanos em manutenção e coleta de informações em linhas de transmissão (22).

O mosaico de imagens também forma a base tecnológica de sistemas de realidade virtual, tais como Quicktime VR (23) e vídeos e imagens em 360 graus que se tornaram muito populares com o advento de visualizadores de 360 graus fáceis de usar, como *Cardboard*<sup>1</sup> e *GearVR*<sup>2</sup>. Também pode ser utilizado no jornalismo quando, por exemplo, o *New York Times* publicou mais de 20 filmes em 360 graus (24).

Esta dissertação propõe a reconstrução de ambientes do mundo real através da captura e processamento de dados para que os resultados possam ser usados em aplicações

<sup>1</sup> *Google Cardboard*: <https://vr.google.com/cardboard>

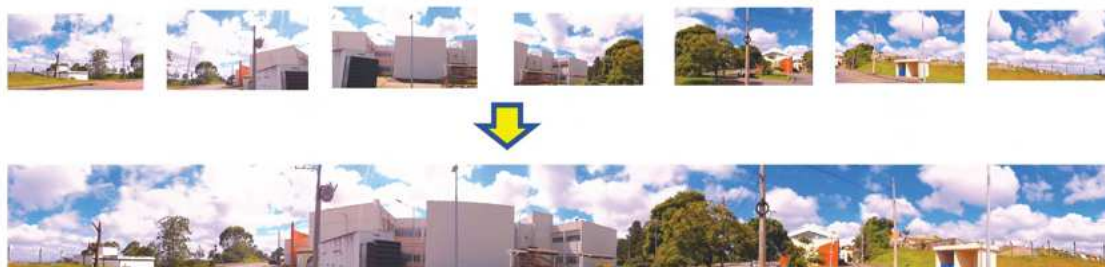
<sup>2</sup> *Samsung GearVR*: [www.samsung.com/us/gearvr](http://www.samsung.com/us/gearvr)

de realidade virtual como auxílio nos procedimentos de treinamento e manutenção. Foi desenvolvido um hardware para auxiliar a virtualização para uso em tempo real através de uma técnica mista de LIDAR com visão monocular para realizar a aquisição de dados. Neste estudo em particular, a metodologia de *Stitching* que será usada para resolver um problema de “costura” de panorâmica 360° concentra-se em realizar uma nova abordagem para o estágio de alinhamento de imagem do sistema e realizar o *Bundle Adjustment*. Para isso, alguns algoritmos de otimização bioinspirados serão implementados afim de verificar a eficiência dos algoritmos ao problema em questão.

### 1.1 IDENTIFICAÇÃO DO PROBLEMA E MOTIVAÇÃO

A criação de imagens panorâmicas, ou *stitching*, é um processo de sobreposição de um conjunto de imagens em um sistema de coordenadas obtido em diferentes pontos de vista e tempos diferentes para gerar uma imagem panorâmica de visualização mais ampla. A Figura 3 é uma representação de múltiplas imagens tiradas em tempos e posições distintas, e depois alinhadas para formar uma imagem final com o campo de visão total. O *stitching* de imagens tem sido uma área de pesquisa ativa nas últimas tempos devido à sua importância e implicações em muitas aplicações, como visão computacional (25), imagens médicas modernas (26), e sensoriamento remoto (27);

Figura 3 – Representação de *Stitching* de Imagens



Fonte: Elaborada pela autora (2021)

A construção de imagens panorâmicas envolve diversas etapas de processamento de imagem: registro, re-projeção e mesclagem. Muitos algoritmos, hoje, são capazes de obter imagens sobrepostas da mesma cena e juntá-las para criar um panorama no qual as transições entre componentes de imagens diferentes não são perceptíveis. Esses algoritmos, no entanto, geralmente têm uma série de requisitos para produzir resultados satisfatórios: variação de iluminação limitada, configurações de exposição semelhantes entre imagens e movimento limitado de objetos na cena. Infelizmente, a maioria dos algoritmos falham em algum grau quando a cena exibe movimento substancial e o resultado geralmente contém objetos cortados ou borrões pesados (28). Embora há avanços nessa área de pesquisa nos

últimos anos, o mosaico de imagens continua a ser um desafio devido a fatores como o registro e a mesclagem (29).

O registro não é apenas uma etapa importante na criação do mosaico, mas também o fundamento do mesmo. É o processo de sobrepor duas ou mais imagens da mesma cena tiradas em momentos diferentes referindo-se ao estabelecimento de correspondência geométrica entre um par de imagens que representam o mesmo ambiente (27). Os algoritmos que permitem que as imagens sejam alinhadas e perfeitamente “costurada” estão entre os mais antigos e mais usados em visão computacional. Em ordem, para registrar um conjunto de imagens, é necessário estimar as transformações que alinham as imagens em relação a uma referência de imagem dentro desse conjunto. Após esse processo tem-se o alinhamento das fotografias em uma coordenada comum usando as transformações geométricas computadas. O objetivo dessa etapa é sobrepor as imagens alinhadas em uma imagem maior mesclando valores de pixel das partes sobrepostas e mantendo pixels onde nenhuma sobreposição ocorre. Uma vez que as imagens são combinadas com sucesso, elas são mescladas para formar uma imagem panorâmica de visualização mais ampla, de forma que a borda seja contínua (30). Essa metodologia abrange duas categorias de abordagens que é a suavização de transição e localização ideal da “costurada” (31).

No processo de *stitching* de imagens diversos erros surgem quando as imagens contêm uma grande quantidade de dados não ideais. Por exemplo, quando imagens são agrupadas usando este sistema, desalinhamentos geométricos e fotométricos muitas vezes resultam em indesejáveis descontinuidades de objetos e visibilidade nas proximidades do limite entre duas imagens e, até correspondências incorretas podem surgir (32).

Dessa forma, a maioria dos algoritmos compartilham desse pipeline o qual estima-se as transformações que alinham as imagem sobrepostas. Porém, esse alinhamento perfeito raramente é alcançado, assim, a maioria dos esforços nessa linha de pesquisa são focados em conceber melhores técnicas de alinhamento ou composição para reduzir ou ocultar os desalinhamentos (33). Por exemplo, *seam cutting* (34, 35) que otimizam a seleção de pixels entre as imagens sobrepostas para minimizar costuras visíveis, o método de mesclagem baseado em pirâmide laplaciana (36) que é uma técnica avançada de combinação de pixels e a mesclagem de imagem de Poisson (37) que minimiza o desfoque devido a desalinhamentos ou diferenças de exposição. Tais algoritmos são utilizados a fim de minimizar as descontinuidade de coloração na aparência da imagem final gerada. No entanto, mesmo com a utilização desses algoritmos de combinação, o efeito pode permanecer insatisfatório.

Embora seja importante para produzir visualmente resultados aceitáveis, tais rotinas de pós-processamento são, no entanto, imperfeitos e pode não funcionar o tempo todo. Portanto, é estratégico tentar reduzir os erros durante a etapa de alinhamento. A pesquisa sobre o alinhamento de imagens para *stitching* culminou, de certa forma, no uso

de *bundle adjustment* (38) para otimizar simultaneamente as posições relativas das imagens de entrada que são, então, usadas para alinhar todas as imagens a um quadro de referência comum. O *Bundle Adjustment* é uma técnica comumente utilizada em reconstruções 3D (39), porém também se utiliza com intuito de *stitching* (40).

Na técnica de *Bundle Adjustment* para a geração e montagem de imagens panorâmicas, geralmente, utiliza-se do método clássico Levenberg-Marquardt para otimização dos parâmetros intrínsecos e extrínsecos da câmera (41, 42), porém tal técnica tem como característica encontrar soluções mínimas locais (43). Sendo assim, outras técnicas podem ser desenvolvidas e implementadas para solucionar o problema dos mínimos locais.

O problema acima levanta uma forte motivação para estudar novos métodos de alinhamento para *stitching* de imagens para formação de panorâmicas, por exemplo. Para este fim, propõe-se uma nova técnica baseada no *bundle adjustment* utilizando abordagens meta-heurísticas de otimização comprovadas como uma boa solução para problemas de otimização multimodal, que tendem a ser afetados por mínimos locais (44) com objetivo de ajustar a matriz de rotação das imagens, assim como focos e centros óticos, afim de se encontrar o melhor alinhamento entre as imagens. E, ainda, para tornar a imagem final com colorações uniformes e suavizar as diferenças devido a exposições, utiliza-se a técnica de mesclagem de imagens *Multi-band Blending*.

## 1.2 OBJETIVOS

Nesta seção serão apresentados os objetivos gerais e específicos dessa dissertação.

### 1.2.1 Objetivos Gerais

A presente proposta tem como objetivo o desenvolvimento de uma metodologia para montagem de imagens panorâmicas 360 digitais de alta resolução, por meio de imagens obtidas por um sistema robótico desenvolvido, baseada em otimização para aplicação em tarefas de inspeção ambientes externos e internos como subestações, que requerem inspeções periódicas por meio de virtualização de ambientes. O intuito desse trabalho será a utilização de algoritmos de otimização para integrar múltiplas imagens utilizando meta-heurísticas, de forma a aumentar a qualidade das imagens panorâmicas finais dada a grande dificuldade unir diversas imagens em posições diferentes.

### 1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Propor uma nova metodologia de otimização ótima de posicionamento de câmeras para a produção de imagens panorâmicas utilizando meta-heurísticas;

- Implementar os algoritmos *Grey Wolf Optimizer* e *Bat Algorithm*
- Desenvolver imagens panorâmicas com o mínimo de artefatos e/ou distorções visuais;

### 1.3 ESTRUTURA DO TRABALHO

Além deste capítulo introdutório, esta dissertação é dividida em mais quatro capítulos.

O Capítulo 2 aborda os fundamentos teóricos necessários para a compreensão da pesquisa, reforçando os conceitos de processamento de imagens, uma breve descrição da modelagem de uma câmera digital, além de conceitos geométricos epipolares, *bundle adjustment* e projeções de imagens.

O Capítulo 3 apresenta a metodologia utilizada para solucionar o problema de formação de imagens panorâmicas 360°, contendo o detalhamento da modelagem do problema e, também, são descritos os algoritmos de otimização *Grey Wolf Optimizer* (GWO) e *Bat Algorithm* (BA) utilizados.

Os resultados serão abordados no Capítulo 4, onde serão discutidos e analisados os diferentes cenários com relação aos parâmetros dos algoritmos e curva de convergência.

Por fim, o Capítulo 5 expõe a conclusão dessa dissertação, descrevendo o que foi obtido com este trabalho e apresentando as propostas para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta os fundamentos teóricos relevantes acerca dos temas abordados por este trabalho. A Seção 2.1 conceitua o processamento digital de imagens e formação de imagens. A seção 2.2 apresenta uma modelagem de câmeras pinhole e, em seguida, na Seção 2.3 os seus parâmetros intrínsecos e extrínsecos. A seção 2.4 apresenta o conceito de geometria epipolar assim como o de matriz fundamental. A Seção 2.5 apresenta a definição de pontos de interesse nas imagens e sua extração e detecção na Seção 2.6, apresentando o algoritmo SIFT utilizado neste trabalho. A seção 2.7 mostra a obtenção de correspondências entre imagens. A seção 2.8 apresenta as diversas formas de se obter a matriz de homografia. Na Seção 2.9 o algoritmo RANSAC é apresentado como metodologia para eliminação de pontos de falsas correspondências. Em 2.10 apresenta-se o conceito de *Bundle Adjustment* para se tratar o erro de re-projeção de *features*. Por fim na Seção 2.11 apresentou-se o conceito de projeção de imagens e os dois tipos de projeções muito utilizadas em confecção de imagens panorâmicas.

### 2.1 PROCESSAMENTO DE IMAGENS

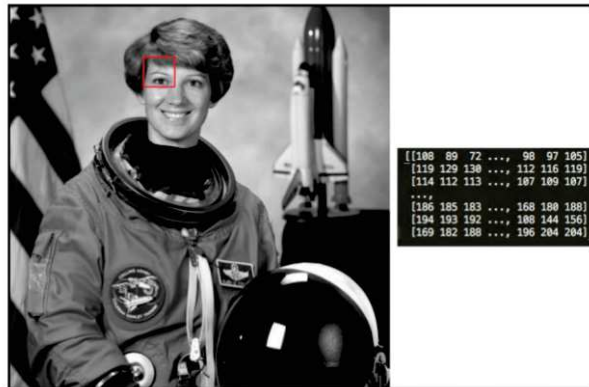
A Visão computacional pode ser considerado um tipo de algoritmo em que se tem um modelo matemático para o processamento de imagens. A tecnologia de processamento de imagens refere-se principalmente ao uso de computadores para o processamento de informações de imagens.

Devido às características da própria imagem, de ter um grande número de pontos, linhas e cores os elementos são manipulados pelo computador e razoavelmente remontados, divididos e ajustados para obter um processamento de imagem eficaz (45). Os seres humanos veem as imagens como uma grade bidimensional com cada célula preenchida com um valor de cor, também denominado de pixel (46). Uma imagem em um computador é representada por uma matriz bidimensional com cada célula da matriz armazenando os valores de pixel correspondentes (47). A Figura 4 é um exemplo de uma matriz de imagem. A matriz representa a parte da imagem marcada pelo quadrado vermelho realçado na figura à direita.

Imagens são dados indexados por duas coordenadas espaciais, sendo as variáveis  $x$  e  $y$  para se referir aos eixos horizontal e vertical da imagem, respectivamente. O valor do pixel representa a cor ou intensidade de cada pixel e a colocação dos pixels com a matriz corresponde ao seu posicionamento dentro da imagem.

Caso seja necessário mais de um valor para representar as informações, a imagem é representada por uma matriz multidimensional. Considerando as cores, um padrão muito utilizado é o RGB (do inglês *Red-Green-Blue*), onde são consideradas as cores primárias: vermelho, verde e azul, respectivamente. Neste caso, cada pixel carregará informações

Figura 4 – Matriz de imagem(direita), como armazenada em um computador, de uma pequena porção da imagem (esquerda representada pelo quadrado vermelho)

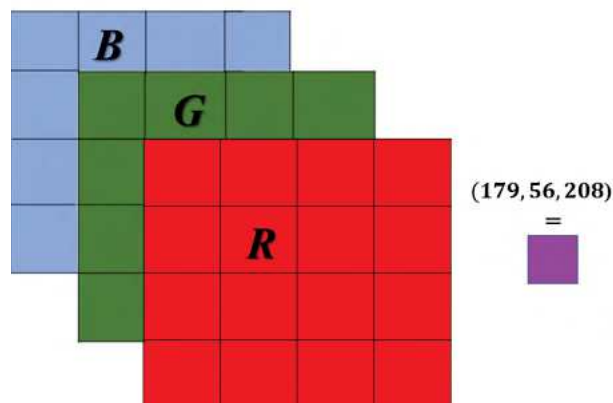


Fonte: Kapur (2017)

referentes a cada uma dessas cores , e a variação de intensidade atribuída a cada uma delas, formando as demais cores. Por exemplo, a representação de uma imagem *RGB*, onde cada pixel possui três valores, contém 3 matrizes, uma para cada coloração do pixel. Neste caso, se cada um dos componentes tem um intervalo de 0 a 255, resulta em  $255^3$  diferentes cores possíveis em uma imagem (48).

A Figura 5 é uma representação de três matrizes e como o valor de cada pixel gera um valor de cor novo. O valor RGB (179, 56, 208), por exemplo, produz um pixel roxo.

Figura 5 – Imagem RGB Representada por Três Matrizes



Fonte: Elaborado pela autora (2021)

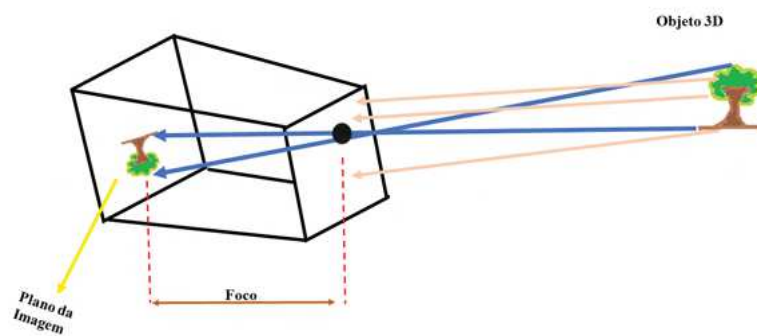
## 2.2 MODELAGEM DE CÂMERA

A câmera é uma das ferramentas mais essenciais na visão computacional. É o mecanismo pelo qual pode-se registrar o mundo ao redor e usar sua saída (fotografias )

para várias aplicações. O modelo geométrico de câmera mais comumente usado em visão computacional é a câmera pinhole (2).

Ao usar um modelo de câmera pinhole, o mapeamento geométrico de 3D para o 2D é chamado de projeção em perspectiva. Este sistema de câmera pode ser projetado colocando uma barreira com uma pequena abertura entre o objeto 3D e um filme fotográfico ou sensor. Como mostra a Figura 6, cada ponto no objeto 3D emite raios de luz para o exterior. Sem uma barreira no lugar, todos os pontos do filme serão influenciados pelos raios de luz emitidos de todos os pontos do objeto 3D. Devido à barreira, apenas um (ou alguns) desses raios de luz passa pela abertura e atinge o filme. Portanto, pode-se estabelecer um mapeamento um a um entre pontos no objeto 3D e no filme. O resultado é que o filme é exposto por uma “imagem” do objeto 3D por meio desse mapeamento. Este modelo simples é conhecido como modelo de câmera pinhole.

Figura 6 – Modelo da Câmera Pinhole



Fonte: Elaborado pela autora (2021)

De acordo com Hartley e Zisserman (49), considerando  $(u, v)$  como coordenadas 2D da imagem e  $(x, y, z)$  coordenadas 3D do objeto 3D. As principais equações de projeção em perspectiva o plano de imagem são:

$$u = f \frac{x}{z} \quad (2.1)$$

$$v = f \frac{y}{z} \quad (2.2)$$

Onde,  $f$  representa a distância focal da câmera, uma propriedade fundamental para relacionamento da imagem com o mundo tridimensional (2).

Se os pontos no espaço 3D e os pontos no plano da imagem são representados por coordenadas homogêneas, a equação 2.3 pode ser escrita em forma de notação matricial



como :

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.3)$$

Em que  $\lambda = Z$  é o fator de escala homogêneo

## 2.3 PARÂMETROS DA CÂMERA

### 2.3.1 Parâmetros Intrínsecos

O plano focal no modelo da câmera pinhole está embutido em  $R^3$ , portanto, precisa-se ter um mapeamento que traduza os pontos no plano da imagem em pixels. Esta transformação é representada por uma matriz triangular superior 3x3. Esta matriz contém os parâmetros internos da câmera ou parâmetros intrínsecos (49).

A matriz pode ser escrita de várias formas (50). A mais comum está expressada na equação 2.4.

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Os parâmetros  $f_x$  e  $f_y$  são comprimentos focais independentes para as dimensões  $x$  e  $y$  do sensor. Esses parâmetros são responsáveis por redimensionar as coordenadas da imagem em pixels. O componente  $s$  é chamado de inclinação e é usado para corrigir pixels inclinados causados pelo sensor não ser exatamente perpendicular ao eixo óptico. Os parâmetros  $c_x$  e  $c_y$  representam o centro óptico da câmera, em pixels. Em muitos casos, especialmente quando se trata de câmeras de última geração, é possível ter uma matriz intrínseca mais simples. Assumindo  $s = 0$  e o mesmo comprimento focal nas dimensões  $x$  e  $y$ , a matriz  $K$  pode ser reescrita como:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Tais equações descrevem o mapeamento de projeção central do mundo para as coordenadas da imagem. Se essas coordenadas forem representadas por vetores homogêneos, pode-se escrever as relações expressas em 3.26a e 2.2 em forma de matriz :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.6)$$

### 2.3.2 Parâmetros Extrínsecos

Parâmetros extrínsecos são aqueles que descrevem a postura (ou seja, posição e orientação) de um sensor em relação a um *frame* de referência externo (49).

Os parâmetros extrínsecos definem a localização e a orientação do quadro de referência da câmera em relação a um quadro de referência conhecido. Uma escolha típica para descrever a transformação entre a câmera e o quadro mundial é usar um vetor translacional 3D  $T$  que descreve as posições relativas dos dois quadros e uma matriz rotacional  $R$  que traz os eixos correspondentes dos dois quadros um ao outro. Dessa forma, tem-se:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.7a)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = K[R|t] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.7b)$$

A equação 2.7b mostra a representação da matriz de pose da câmera, dada por  $P = K[R|t]$ . Essa matriz, permite realizar a projeção de um ponto 3D para qualquer plano de imagem.

A matriz extrínseca representada por  $\zeta$  pode ser obtida conforme equação 2.8. Tal matriz representa a orientação da câmera dado um *frame* de referência.

$$\zeta = \begin{bmatrix} R & t \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (2.8)$$

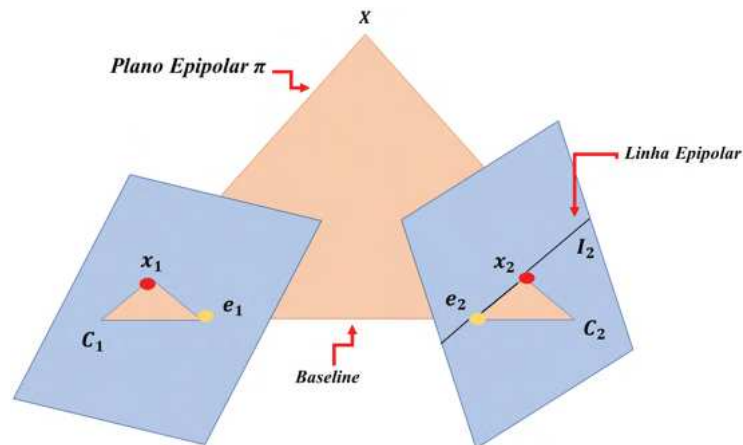
O centro ótico da câmera é dado pela equação 2.9.

$$C = -R^t t \quad (2.9)$$

## 2.4 GEOMETRIA EPIPOLAR

Considerando o caso de duas câmeras olhando para a mesma cena. A geometria epipolar é a restrição básica que surge da existência de dois pontos de vista. Ela descreve as relações que existem entre duas imagens. Cada ponto em um plano que passa por dois centros de projeção serão projetados em cada imagem na interseção deste plano com o plano da imagem correspondente. Portanto, essas duas linhas de interseção contêm uma correspondência epipolar (51).

Figura 7 – Geometria Epipolar



Fonte: Elaborada pela autora (2021)

A Figura 7 apresenta a configuração da geometria epipolar envolvendo duas câmeras observando o mesmo ponto 3D  $X$ , cuja projeção em cada um dos planos da imagem estão localizados em  $x_1$  e  $x_2$ . Os centros da câmera são localizados em  $C_1$  e  $C_2$  e, a reta que liga o centro  $C_1$  da primeira imagem com o centro  $C_2$  da segunda, é chamada de *baseline*. O plano definido pelos centros das duas câmeras e o ponto  $X$  é denominado plano epipolar. Os pontos que estão contidos no plano da imagem e no eixo gerado entre as duas câmeras são denominados de epipolo ( $e_1, e_2$ ). E, as linhas definidas pela interseção do plano epipolar e os planos das imagens são conhecidas como linhas epipolares, na figura 7 a linha epipolar da imagem 2 está representada por  $I_2$  (49).

#### 2.4.1 Matriz Fundamental

A matriz fundamental é a representação algébrica da geometria epipolar (49). Uma alternativa para se encontrar os pontos de projeções  $x_1$  e  $x_2$ , é realizando o mapeamento entre o ponto e a linha epipolar. Conforme a figura 7, dado um par de imagens onde cada ponto  $x_1$  em uma imagem, possui uma linha epipolar correspondente  $I_2$  na outra imagem. Qualquer ponto  $x_2$  na segunda imagem que corresponda ao ponto  $x_1$  deve estar na linha epipolar  $I_2$ . Dessa forma, a matriz fundamental mapeia os pontos de uma imagem para as suas correspondentes linhas epipolares na outra imagem (50).

Sendo assim, a matriz fundamental  $F$ , satisfaz a condição de que para um par de pontos correspondentes  $(x_1, y_1, 1)$  e  $(x_2, y_2, 1)$ , tem-se :

$$[x_2, y_2, 1]F[x_1, y_1, 1]^T = 0 \quad (2.10)$$

Com base em (49) considera-se que  $F$  é uma matriz  $3 \times 3$ , com 7 graus de liberdade e posto 2. Cada par de correspondências gera-se uma equação, sendo que com pelo

menos sete correspondências ( $n \geq 7$ ) é possível realizar o cálculo da matriz fundamental. Considerando  $q_i \leftrightarrow q'_i$ , um sistema linear de equações cuja solução determina F pode ser obtido pela equação 2.11.

$$\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2x_2 & x'_2y_2 & x'_2 & y'_2x_2 & y'_2y_2 & y'_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \quad (2.11)$$

Segundo Hartley e Zisserman (49), uma restrição de singularidade deve ser imposta substituindo F pela matriz  $F'$ , sendo que  $F'$  minimiza  $\|F - F'\|$  e possui a restrição de que o determinante de  $F'$  é igual a zero. Essa equação pode ser resolvida utilizando SVD, onde a matriz F pode ser decomposta em  $F = UDV^T$ , em que  $D = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$  que satisfaz a  $\sigma_1 \geq \sigma_2 \geq \sigma_3$  e  $F'$  é calculado por  $F' = U \text{diag}(\sigma_1, \sigma_2, \sigma_3, 0) V^T$  para minimizar  $\|F - F'\|$ .

## 2.5 PONTOS DE INTERESSE

Pontos de interesse, *features*, *keypoints* ou características são um ponto em uma imagem que possui uma posição bem definida e que pode ser detectado de forma robusta (52). Geralmente, associado com uma mudança significativa em uma ou mais propriedades de imagens como intensidade, cor e textura. Uma *feature* é usada para denotar uma informação que é relevante de uma imagem. Pode se referir a estruturas específicas na imagem em si, variando de estruturas simples, como pontos ou arestas para estruturas mais complexas, como objetos (53).

Os chamados pontos de interesse, são, na maioria das vezes, cantos ou *blobs* (regiões homogêneas) na imagem. Enquanto descritores globais representam melhor conceitos da visão humana, como cor, forma ou textura, descritores locais são mais robustos a transformações de imagens (54).

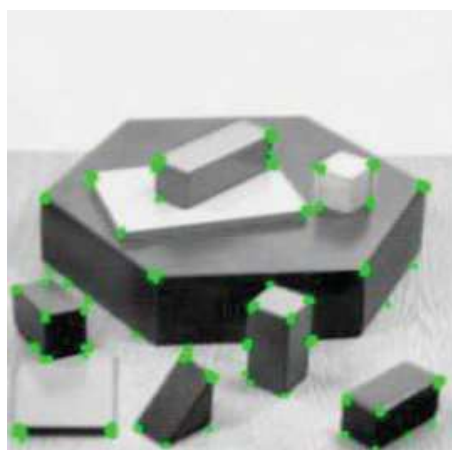
Tuytelaars e Mikolajczyk (55) descrevem que bons pontos de interesses possuem propriedades como :

- *Repetibilidade*: Considerando duas imagens de um mesmo ambiente ou objeto, tiradas de pontos de vistas distintos, uma alta porcentagem de características detectadas no ambiente devem ser detectadas em ambas imagens.

- *Distinção*: Os padrões de intensidade das características detectadas devem apresentar inúmeras variações para que sejam distintas.
- *Localidade*: A característica deve ser local e permitir um modelo simples de aproximação quando há deformação geométrica e fotométrica entre duas imagens, adquiridas de vistas e condições diferentes.
- *Quantidade*: O número de características detectadas devem ser suficientemente grande para que haja representação até de objetos pequenos. No entanto, este número depende da aplicação.
- *Acurácia*: A característica detectada deve ter acurácia na sua localização dentro da imagem com respeito a escala e formato.
- *Eficiência*: Preferivelmente, a detecção de características na imagem deve ocorrer no menor tempo possível.

As características nas imagens, geralmente, são encontrados por detectores de pontos e possuem normalmente as seguintes informações: uma coordenada 2D na imagem, uma orientação e uma escala (56). Um dos detectores de pontos de interesses mais conhecidos é o algoritmo *Harris Corner* (57) que é um detector de canto. O algoritmo *Harris Corner* calcula uma pontuação de canto a partir dos gradientes da imagem e rotula valores acima de um limite definido como cantos. O canto de Harris é invariante à rotação, a translação e as alterações aditivas à intensidade. No entanto, não é invariante para a escala de intensidade e escala, logo, caso ocorra mudança significativa na escala entre imagens, esses pontos de interesse não serão repetidos. A Figura 8 apresenta o comportamento do algoritmo de *Harris Corner*.

Figura 8 – Detector de *Harris Corner*

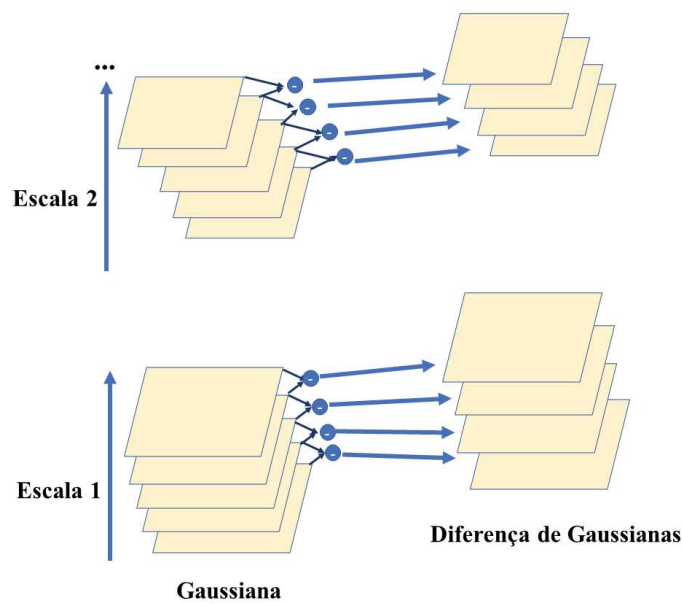


Fonte: (58)

O Detector *Harris Corner* é apenas um dos muitos algoritmos que ajudam a encontrar os pontos de interesse em imagens. Um outro algoritmo para detecção de *feature* foi desenvolvido por Lowe (59) e é um método que detecta pontos de interesse extremamente distintos, possuem alta repetibilidade e são invariantes a rotação e escala. Diferente do método de Harris citado anteriormente, associa a escala e a orientação em cada ponto de interesse por meio de dois estágios, detecção e descrição.

O método de Lowe, denominado SIFT (*Scale Invariant Feature Transform*), é baseado na detecção de extremos no espaço de escala, na localização do ponto de interesse e na sua orientação. Utiliza-se de diferença gaussiana, do inglês *Difference of Gaussians* (*DoG*), para identificar esses pontos de interesse e, também, um fator de escala para variar a distribuição do filtro gaussiano entre as imagens. A representação dessa metodologia pode ser verificada na Figura 9.

Figura 9 – Detecção de Pontos de Interesse - Adaptado de Lowe (59)



Fonte: Elaborada pela autora

Inicialmente, um espaço de escala é construído a partir da convolução repetida de imagem utilizando um filtro gaussiano, com mudanças de escala e agrupamento das saídas em oitavas, conforme a Equação 2.12.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.12)$$

onde  $*$  é o operador de convolução,  $G(x, y, \sigma)$  é um filtro Gaussiano com escala variável  $\sigma$  e  $I(x, y)$  é a imagem de entrada. Após a construção do espaço de escala, as imagens de

Diferença de Gaussianas são calculadas a partir das imagens adjacentes de Gaussianas em cada oitava, conforme Equação 2.13.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.13)$$

## 2.6 DESCRITORES

Na seção anterior foi apresentado como os pontos de interesse podem ser detectados porém, para que os mesmos sejam comparados, devem possuir alguma forma de descrição.

O descritor de imagem é representado na forma de um conjunto de números, os quais, quando comparados, podem ser empregados no reconhecimento de objetos. Os descritores são basicamente representações vetoriais que caracterizam matematicamente uma região na imagem. Esse processo é oriundo da combinação dos descritores do objeto detectado em uma imagem, juntamente com o descritor de características de uma imagem previamente conhecida (60).

Como as imagens analisadas no processo de *Stitching* podem apresentar mudança de rotação e translação, deve-se optar por um detector de *features* do tipo *blobs* que é mais robustos a mudanças de formatos que um detector de cantos (55). Diversos tipos de descritores já foram propostos na literatura, por exemplo, SIFT (*Scale-Invariant feature transform*)(61), SURF (*Speed up robust features*) (62), ORB (63) e AKAZE (64). Dentre esses métodos o algoritmo SIFT foi escolhido para este trabalho. Logo, a seção a seguir apresenta uma visão geral do algoritmo utilizado tendo em vista sua aplicação na descrição de imagens.

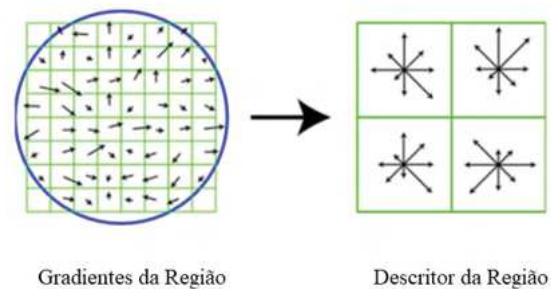
### 2.6.1 Scale-Invariant Feature Transform (SIFT)

SIFT encontra pontos de interesse em várias regiões e escalas da imagem e atribui uma orientação a este. Um vetor que represente cada ponto de interesse é construído para futura correspondência entre características de diferentes imagens, chamado de descritor de pontos de interesse.

Kabbai et al. (65) , apresenta o processo de descrição de pontos de interesse utilizando o algoritmo SIFT a partir de duas etapas principais: (i) atribuição de orientação; e (ii) descrição do ponto de interesse. O SIFT encontra, primeiramente, a magnitude do gradiente e a orientação dos pixels vizinhos ao local do ponto. Em uma vizinhança, considera-se uma janela de 16 x 16 pixels ao redor dos pontos de interesse. Em seguida, particiona as medições em sub-regiões de 4 x 4 pixels. Orientações de gradiente e magnitudes são calculados em cada pixel na janela, ponderado por uma função gaussiana. Um histograma de orientação de gradiente ponderado é calculado e apresenta 8 variações de orientação, resultando em um vetor com 128 (16 x 8) elementos ( existem 16 células 4

x 4 de 8 bins cada). A figura 10 exemplifica como os *keypoints* são criados. Na imagem da parte esquerda o gradiente no qual mostra a magnitude e orientação em cada ponto de amostra de imagem em uma região em torno da localização do ponto de interesse, à direita tem-se a representação dos histogramas em sub-regiões 4 x 4. Esta figura mostra uma matriz de descritor 2 x 2 calculada a partir de um conjunto de amostras 8 x 8.

Figura 10 – Descrição do ponto - SIFT: À esquerda tem-se a representação dos gradientes dos pixels e à direita tem-se sub-região de 4x4 pixels



Fonte: (59)

O SIFT é altamente robusto para um descritor. É invariável em escala e rotação, pode lidar com mudanças no ponto de vista (até 60 graus fora da rotação do plano) e pode lidar com mudanças significativas na iluminação. Os descritores obtidos a partir da utilização do SIFT são considerados robustos para uma ampla família de transformações de imagem, como pequenas mudanças de ponto de vista, ruído, desfoque, mudanças de contraste, deformação de cena, enquanto permanecem discriminativos o suficiente para fins de correspondência (66).

## 2.7 CORRESPONDÊNCIAS DE PONTOS

A correspondência ou *matching* de pontos entre duas imagens trata-se do processo de extrair *keypoints* e, procurar os pontos correspondentes em cada uma, assim como exemplificado na Figura 11. A comparação de pontos é baseada nas características similares dos descritores correspondentes.

O processo para encontrar correspondência é utilizando a distância euclidiana entre as *features*. Os pontos que podem apresentar a melhor correspondência são aqueles em que descritores possuam a menor distância euclidiana em relação aos demais.

Entretanto, o resultado produzido pelos algoritmos de correspondências de características contém sempre um número de correspondências incorretas, alguns pontos instáveis (*outliers*) são detectados ao longo do processo, levando a falsas correspondências. A Figura



Figura 11 – Exemplo de Correspondência de Características entre um Par de Imagens



Fonte: Elaborada pela autora.

12 apresenta o *matching* entre as duas imagens de uma estátua onde tem-se ressaltados os matches incorretos entre os pontos de interesse.

Figura 12 – *Matches* Incorretos entre Duas Imagens



Fonte: Elaborada pela autora.

O Trabalho de Mikolajczyk e Schmid (67) apresenta uma avaliação de desempenho e comparações de descritores, e o SIFT, geralmente, apresenta melhor desempenho. O algoritmo utilizado para identificar vizinhos mais próximos e com uma probabilidade maior é chamado de *Best-Bin-First* (BBF) desenvolvido por Lowe e Beis (68), o algoritmo encontra suas correspondências nas outras imagens, comparando-as com todos os recursos

no conjunto de recursos SIFT da outra imagem. Isso é implementado por um algoritmo de árvore k-d modificado.

Utiliza-se um método de comparação entre a menor distância com a segunda menor, ou seja, para formar uma correspondência entre os descritores de duas imagens utiliza-se a abordagem de relação da distância da melhor correspondência com a distância da segunda melhor, selecionando somente os pontos correspondentes próximo de um limiar (59). Lowe rejeitou todas os matches em que a relação de distância é superior a 0.8 entre os dois descritores, o que elimina 90% das falsas correlações, porém apenas descartando menos de 5% das correspondências corretas. Em outras palavras, rejeita-se correspondências que contenham alta incerteza.

Em seguida, utilizando as correspondências obtidas até o momento, existem algoritmos e metodologias capazes de realizar a filtragem dessas características. O algoritmo *Random Sampling Consensus* (RANSAC) tenta eliminar as falsas correspondências (*outliers*) entre as imagens (69). Outra abordagem, foi apresentada em (70) onde testa-se se as correspondências são compatíveis.

## 2.8 HOMOGRAFIA

A homografia, ou transformação projetiva, é uma relação linear entre pontos nos planos e entre duas vistas desempenhando um papel importante na geometria de múltiplas vistas (71). Uma homografia é um mapeamento invertível de pontos e linhas no plano projetivo  $P^2$ . Outros termos para essa transformação incluem colineação, projetividade e transformação projetiva planar (72).

Calcular a homografia por um plano entre duas vistas é um passo importante para a calibração (73), reconstrução 3D (74) e outras aplicações que usam geometria de perspectiva, como registro e mosaico de imagens (75, 76).

Imagens de pontos em um plano em uma vista são relacionadas aos pontos de imagem correspondentes em outra vista por uma homografia plana usando uma representação homogênea. A homografia transfere pontos de uma vista para a outra como se fossem imagens de pontos no plano, essa matriz obtém linhas retas das imagens, entretanto, não preserva a orientação, ângulos, comprimento e paralelismos (72).

Segundo Hartley e Zisserman (49) dado um conjunto de pontos  $x_i$  em  $P^2$  e um conjunto correspondente de pontos  $x'_i$  também em  $P^2$ , calcula-se a transformação projetiva que leva cada  $x_i$  a  $x'_i$ . Considerando  $x_i$  e  $x'_i$  pontos de duas imagens, onde cada imagem é um plano projetivo  $P^2$ . Os pontos  $x_i$  e  $x'_i$  em duas imagens, são relacionados por 2.14.

$$x'_i = Hx_i \quad (2.14a)$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_4 & h_7 \\ h_2 & h_5 & h_8 \\ h_3 & h_6 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.14b)$$

Onde,  $H$  é a matriz de homografia, sendo invertível e  $3 \times 3$ . A transformação projetiva pode reduzir a correspondência de imagens da pesquisa no espaço 2D para transformação de imagem de um para um. A matriz  $H$  possui nove elementos cujo fator de escala não pode ser determinado e o número total de graus de liberdade é 8 (49).

A matriz de homografia também pode ser obtida utilizando os parâmetros intrínsecos e extrínsecos das câmeras. Quando um objeto plano é fotografado de vários pontos de vista ou quando uma cena é fotografada por câmeras com o mesmo centro óptico, as imagens são relacionadas por uma homografia única. Um plano definido por  $\pi^T X = 0$  com  $\pi = [v, 1]^T$  (49). Dessa forma a homografia induzida por um plano pode ser representada pela equação 2.15.

$$H = A - av^T \quad (2.15)$$

Onde,  $A \in R^{3 \times 3}$  e  $a \in R^3$ .

Uma outra relação de homografia pode ser formulada quando se tem a transformação entre as câmeras puramente rotacional (77). Nesse caso, a homografia apresenta a forma apresentada na equação 2.16.

$$H = k' R k^{-1} \quad (2.16)$$

Sendo:  $K'$  e  $K$  são matrizes  $3 \times 3$  de calibração das respectivas câmeras e,  $R$  é a rotação que conecta as imagens.

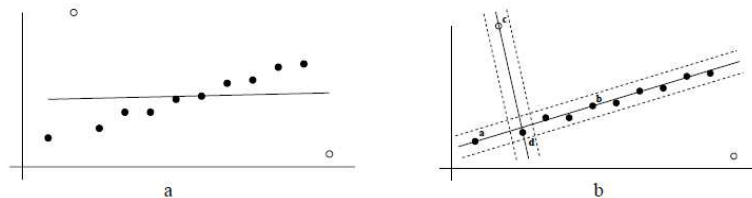
## 2.9 RANDOM SAMPLE CONSENSUS (RANSAC)

O algoritmo *Random Sample Consensus* comumente chamado de RANSAC foi proposto por Fischler e Bolles em 1981 (78). É um método de estimação robusto projetado para extração de bons pontos dentro de uma tolerância de erro. É robusto no sentido de que pode estimar parâmetros com um grau de acerto relevante mesmo quando o número de *outliers*, ou seja, pontos fora de tolerância de erro seja significativo. Possibilita encontrar as correspondências geometricamente consistentes, usadas para resolver a matriz de homografia entre pares de imagens ;

O RANSAC é muito utilizado em problemas de visão computacional, como, para resolver problemas de *matches* entre imagens e estimar a matriz fundamental, explicada na seção 2.4.1. É utilizado para construção de panoramas de imagens utilizando descritores extraídos das imagens com o algoritmo SIFT (36), por exemplo.

A Figura 13, mostra um exemplo apresentado por Hartley e Zirsserman (49). Nele é estimada uma reta ajustada a um conjunto de pontos bidimensionais. Podendo ser feito estimando uma transformação unidimensional  $x' = ax + b$  entre os pontos correspondentes entre as duas linhas.

Figura 13 – Estimação de reta a um conjunto de dados bidimensionais



Fonte: (49).

A Figura13a ilustra o problema de encontrar a reta que minimiza a soma das distâncias a partir de um conjunto de pontos 2D. Deseja-se a reta que melhor se ajusta aos dados e a classificação em pontos válidos.

Já com o uso do algoritmo RANSAC Figura13b, dois pontos são selecionados aleatoriamente, e esses pontos definem uma reta. Essa seleção aleatória é repetida um número de vezes e a reta com maior número de pontos que ficam dentro de um limiar de distância é considerada como ajuste robusto. Na figura, os pontos sem preenchimento estão representando os *outliers*, enquanto os preenchidos os *inliers*. Na Figura13b as linhas pontilhadas indicam a distância do limiar, a linha passando pelos pontos a e b constitui a linha ajustada, o ponto c é um outlier.

O número de iterações é escolhido grande o suficiente para assegurar uma probabilidade ( $\rho$ ) que normalmente é estabelecida para 0.99. Define-se um valor de probabilidade ( $\epsilon$ ) de um dado selecionado ser um *outlier*, sendo atualizado de acordo com as iterações. Para um conjunto de  $s$  elementos, o número de iterações  $n$  necessárias pode ser calculada pela Equação 2.17.

$$n = \frac{\log(1 - \rho)}{\log(1 - (1 - \epsilon)^s)} \quad (2.17)$$

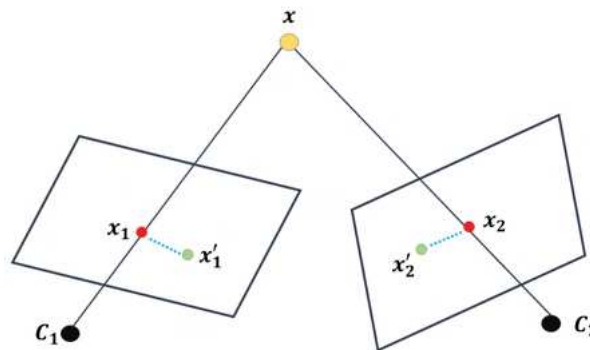
## 2.10 BUNDLE ADJUSTMENT

O *Bundle adjustment* é baseado na projeção de todos os pontos da imagem enquanto extrai as coordenadas 3D que descrevem a estrutura da cena, os parâmetros de movimento relativos e os parâmetros ópticos da câmera (38). O *bundle adjustment* descreve a soma dos erros entre a coordenada de pixel medida e as coordenadas de pixel reprojeta. Dado um conjunto de correspondências geometricamente consistentes entre as imagens, utiliza-se

tal metodologia para resolver todos os parâmetros da câmera em conjunto. Trata-se de um método unificado para estimar os parâmetros intrínsecos (distância focal, ponto principal, etc) e extrínsecos (rotação e translação) das câmeras, além da coordenada 3D do ponto de uma cena (79).

Problemas de ajustes como *Bundle Adjustment* são formulados como problemas de mínimos quadrados não lineares, nos quais a função de custo que quantifica o erro é o erro de re-projeção. O erro de re-projeção é a distância euclidiana entre a posição em que o ponto 3D foi projetado na imagem e a posição da feature que originou esse ponto. A figura 14 exemplifica o erro de re-projeção das *features*.

Figura 14 – Exemplo de Erro de Re-projeção



Fonte: Elaborada pela autora.

Onde,  $x$  representa o ponto no espaço 3D,  $C_1$  e  $C_2$  são origens das câmeras,  $x_1$  e  $x_2$  são as *features* que deram origem ao ponto  $x$  e,  $x'_1$  e  $x'_2$  são as projeções de  $x$  no plano da imagem das câmeras e as linhas pontilhadas representam o erro de re-projeção.

Este algoritmo busca minimizar a soma dos quadrados dos erros de projeção. Para isso, cada *feature* é projetada em todas as imagens às quais corresponde, e a soma dos quadrados é minimizada em relação aos parâmetros da câmera. Se a  $k$ -ésima característica em uma imagem corresponde à  $l$ -ésima característica em outra, obtém-se o resíduo de projeção (36) conforme equação 3.7:

$$r_{ij}^k = u_i^k - p_{ij}^k \quad (2.18)$$

Onde,  $u_i^k$  representa a  $k$ -ésima característica na  $i$ -ésima imagem,  $r_{ij}^k$  é o resíduo da projeção da  $k$ -ésima *feature* da  $j$ -ésima imagem na  $i$ -ésima imagem, e  $p_{ij}^k$  é a projeção da imagem  $j$  para imagem  $i$  do ponto correspondente  $u_i^k$ .

Geralmente, a otimização é realizada com o método *Levenberg-Marquardt* que é capaz de resolver problemas não lineares de mínimos quadrados (80). Porém, existem outras abordagens para a realização de *bundle adjustment* como nos trabalhos (81, 82).

Na construção de imagens panorâmica, geralmente, é necessário utilizar *bundle adjustment* pois quando se une múltiplas imagens, pode ocorrer erro na junção e esse erro é propagado pelo mosaico, dessa forma, ajusta-se esse erro (19). A Figura 15, representa uma panorâmica com erro de re-projeção e sem o ajuste. É notório que o erro é maior nas laterais da imagem e como se trata de uma panorâmica 360°, é evidente, que as bordas não se unirão, portanto, é necessário realizar o ajustamento.

Figura 15 – Panorâmica sem *Bundle Adjustment* Aplicado



Fonte: (19)

## 2.11 PROJEÇÃO DE IMAGENS

Nos últimos anos, a questão de escalabilidade em imagens, principalmente, em panorâmicas e a questão de se projetar imagens em uma superfície comum como um cilindro ou uma esfera tornou-se foco de concentração de alguns estudos. Luhmann discutiu a reconstrução de panoramas com base em projeção cilíndrica (83). Fangi realizou estudos utilizando a modelagem panorâmica de projeção esférica (84, 85, 86).

### 2.11.1 Projeção Cilíndrica

Os panoramas cilíndricos são comumente usados devido à sua facilidade de construção. Para construir um panorama cilíndrico, uma sequência de imagens é tirada. Se a distância focal da câmera ou campo de visão for conhecido, cada imagem em perspectiva pode ser distorcida em coordenadas cilíndricas.

A projeção cilíndrica projeta todas as imagens em um plano cilíndrico. Assim, cada plano de imagem de todas as imagens terá a mesma curvatura. A projeção em perspectiva projeta todas as imagens em um único plano de imagem plana, resultando em grande distorção em algumas das imagens. Portanto, mais imagens podem ser unidas usando a projeção cilíndrica em comparação com a projeção em perspectiva (87).

Para construir uma imagem panorâmica cilíndrica, um pixel 2D genérico de uma imagem adquirida  $[u, v]^T$ , pode ser projetado para um ponto 3D  $p = (x, y, z)$  e a partir de relações trigonométricas obter as coordenadas cilíndricas 2D  $(\theta, v)$  a partir das equações

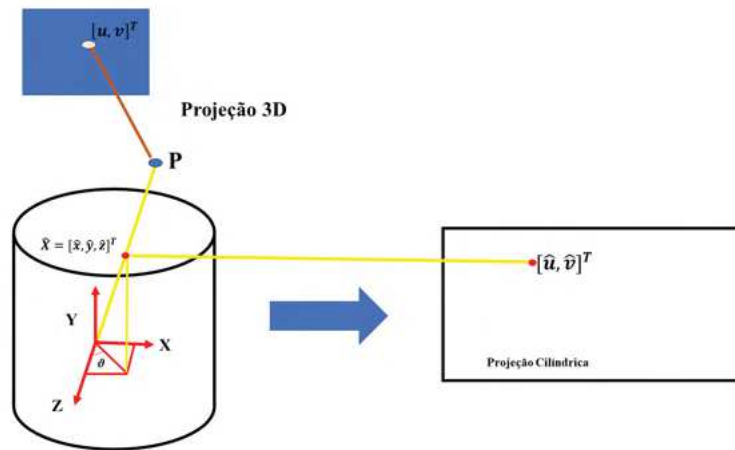
2.19 e 2.20 (88).

$$\theta = \arctan \frac{x}{z} \quad (2.19)$$

$$v = \arccos \frac{y}{\sqrt{x^2 + z^2}} \quad (2.20)$$

Onde,  $\theta$  é ângulo panorâmico e  $v$  é a linha de varredura . A Figura 16 representa a projeção cilíndrica onde um ponto 3D é projetado em um cilindro para obter coordenadas cilíndricas e, então obter , a partir dessas coordenadas, a imagem plana convertida.

Figura 16 – Projeção Cilíndrica



Fonte: Elaborado pela autora (2021)

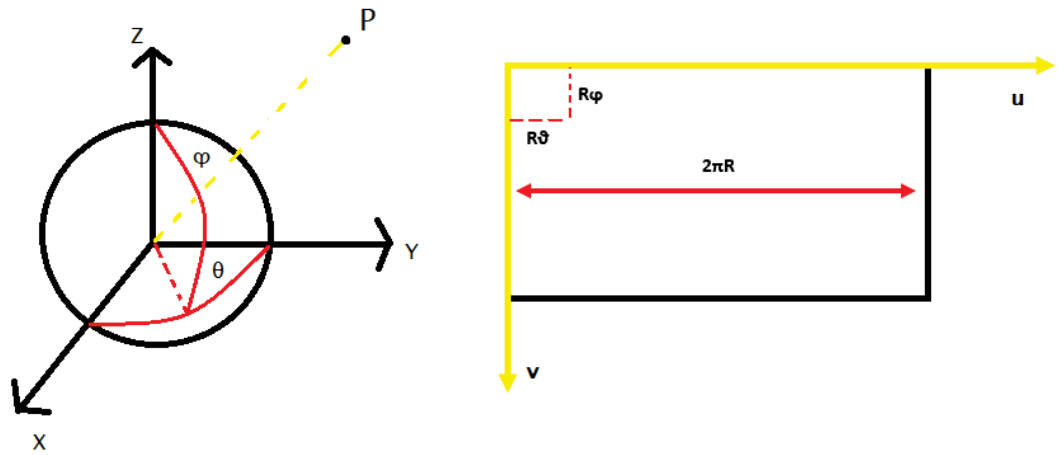
Projeções cilíndricas, muitas vezes, não são consideradas pois podem resultar em panoramas com distorções como mostrado em no trabalho de Kopf et al. (89), por exemplo.

### 2.11.2 Projeção Esférica

Um outro método adotado para criação de panorâmicas é o modelo de projeção esférica, onde cada panorama é mapeado em uma esfera a partir de um ponto e, as imagens são percorridas para cobrir o espaço. A esfera é, então, mapeada em um plano cartográfico com a chamada projeção equiretangular (90), ou latitude-longitude para relação entre coordenada de pixel e coordenadas geográficas.

As imagens são projetadas usando as relações  $[u, v]^T = R[\theta, \phi]^T$ , onde  $(\theta, \phi)$  são as direções angulares horizontal e vertical (longitude e latitude, respectivamente), e  $(u, v)$  são coordenadas de pixel (91). A projeção fornece um valor  $(u, v)$  diretamente de um valor  $(x, y, z)$ , conforme Figura 17:

Figura 17 – Projeção Esférica



Fonte: Elaborado pela autora (2021)

O ponto P representado pelas coordenadas cartesianas dados por  $(x, y, z)$  é a projeção 3D de um ponto  $(x, y)$  da imagem original. Sendo assim, as coordenadas esféricas podem ser obtidas pelas coordenadas cartesianas dados por  $(x, y, z)$ . Dessa forma, tem-se as relações mostradas em 2.21 e 2.22:

$$\theta = \arctan \frac{y}{z} \quad (2.21)$$

$$\phi = \arccos \frac{z}{\sqrt{x^2 + y^2 + z^2}} \quad (2.22)$$

Onde,  $\theta$  é a longitude e  $\phi$  é a latitude.

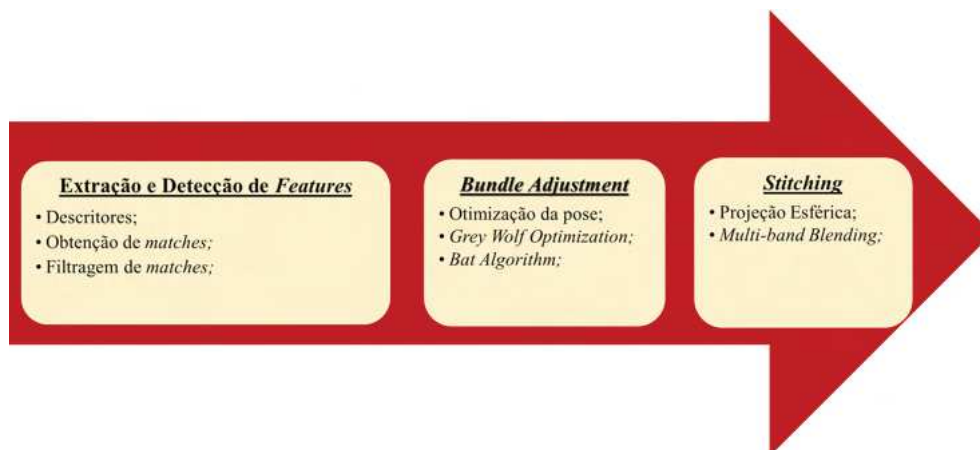
Dessa maneira, um ponto em uma esfera pode ser localizado por latitude e longitude. No formato de projeção esférica ou equiretangular, um ponto pode ser mapeado para um plano 2D simplesmente definindo longitude como valor de coordenada horizontal e latitude como vertical (92).



### 3 METOLOGIA PROPOSTA E ALGORITMOS UTILIZADOS

Nesse capítulo será abordado a evolução da pesquisa explicando as etapas de desenvolvimento, conforme Figura 18.

Figura 18 – Etapas de desenvolvimento



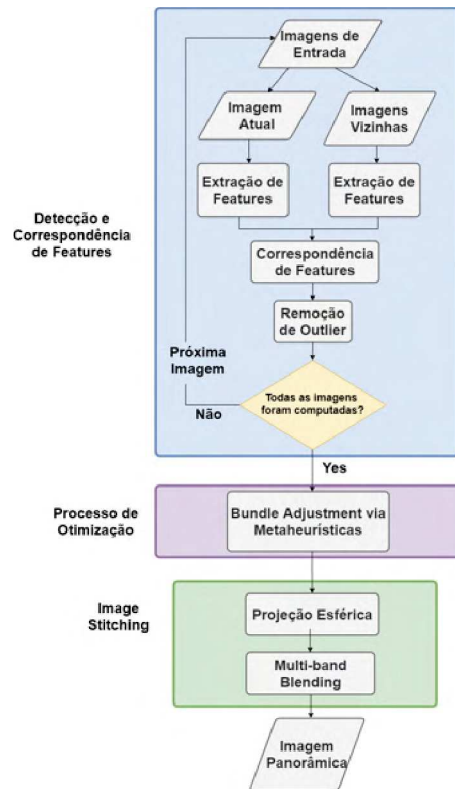
Fonte: Elaborado pela autora (2021)

A Figura 19 apresenta a arquitetura geral do método proposto. O pipeline de geração de imagem panorâmica é composto de várias etapas, incluindo extração e correspondência de *features*, *Bundle Adjustment*, projeção esférica, junção e combinação de imagens. Os pontos característicos são extraídos das imagens e combinados entre si, resultando em um conjunto de pares correspondentes. Na próxima etapa, os pares são usados para obter os parâmetros intrínsecos e extrínsecos das imagens por *Bundle Adjustment*, no qual o erro de re-projeção das correspondências é minimizado por meio de otimização utilizando técnicas de otimização bioinspiradas denominadas *Grey Wolf Optimization (GWO)* e *Bat Algorithm (BA)*, com intuito de se encontrar o ajuste ótimo da posição das câmeras. Por fim, é realizada uma projeção esférica, onde cada imagem é projetada em uma superfície esférica e todas as imagens são costuradas e mescladas para obter o panorama final. Os componentes da metodologia são descritos em seções separadas com explicação detalhada.

Para a confecção desta dissertação, a linguagem de programação utilizada para o desenvolvimento foi a C++, pelo fato de diversas bibliotecas utilizadas serem implementadas nesta linguagem e por possuir ampla documentação. No projeto utilizou-se o *Visual Studio Community 2017*<sup>1</sup> que é uma IDE gratuita completa e extensível. O motivo de ter se escolhido o Visual Studio Community é o fato de ser uma IDE de fácil utilização e acesso, pois, possui suporte em páginas e comunidades virtuais sendo possível sanar dúvidas sobre o software e sua programação.

<sup>1</sup> <https://visualstudio.microsoft.com/pt-br/vs/older-downloads/>

Figura 19 – Visão Geral da Estrutura



Fonte: Elaborado pela autora (2021)

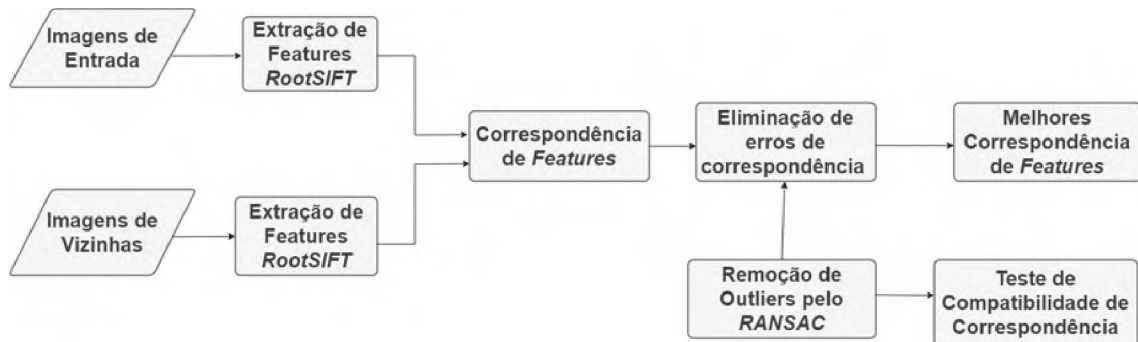
### 3.1 EXTRAÇÃO E DETECÇÃO DE *FEATURES* E *MATCHES*

O primeiro passo para a montagem das panorâmicas é extrair e detectar as *features* e os *matches* entre as imagens conforme apresentado pelo fluxograma mostrado na Figura 20. Primeiramente, realiza-se a extração das características das imagens utilizando *RootSift*, em seguida realiza-se as correspondências de *features* entre as imagens e, elimina-se os erros entre elas por meio de dois métodos distintos com intuito de aumentar a filtragem apenas de bons *matches*.

Para esses processos optou-se por utilizar o algoritmo SIFT explicado nas seções 2.5 e 2.6. Para a detecção de features utilizou-se a biblioteca do OpenCV <sup>2</sup>. Primeiramente, carrega-se as imagens e utiliza-se o detector de *features* para extrair os *Keypoints* e descritores das imagens. O algoritmo SIFT utilizado, foi originalmente projetado para ser usado com distância euclidiana (59) porém, há estudos que usando a distância euclidiana o desempenho é inferior em comparação com usar alternativas como *Hellinger*. Dessa forma, optou-se por realizar a extração de *features* e descritores utilizando o um método denominado *RootSIFT*. Arandjelovic et al. (93) sugerem uma extensão algébrica simples para o próprio descritor SIFT, que permite que os descritores sejam comparados usando

<sup>2</sup> <https://opencv.org/>

Figura 20 – Fluxograma para Extração e Detecção de *Features* e *Matches*



Fonte: Elaborado pela autora (2021)

um kernel de *Hellinger*, mas ainda utilizando a distância euclidiana.

Dessa maneira, computa-se os descritores do SIFT e considerando  $x$  e  $y$  como dois vetores de normalização euclidiana, por exemplo,  $\|x\|_2 = 1$ . Sendo assim, a relação entre a distância euclidiana ( $d_E(x, y)$ ) é dada pela equação 3.1.

$$d_E(x, y) = \|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2x^T y = 2 - 2S_E(x, y) \quad (3.1)$$

Onde,  $S_E(x, y) = x^T y$  e  $\|x\|_2^2 = \|y\|_2^2 = 1$ . Dessa maneira, substitui-se o kernel euclidiano pelo a Kernel de *Hellinger* dado pela equação 3.2:

$$H(x, y) = \sum_{i=1}^n \sqrt{x_i y_i} \quad (3.2)$$

Onde,  $x$  e  $y$  são histogramas normalizados. Sendo assim, para realizar uma medida de similaridade entre dois descritores SIFT e usando o kernel de *Hellinger*, duas operações algébricas devem ser seguidas: (i) realizar uma normalização do descritor SIFT e (ii) realizar uma operação de raiz quadrada dos elementos no vetor SIFT normalizado. Então, a Kernel Euclidiano apresenta a seguinte formulação 3.3.

$$S_E(\sqrt{x}, \sqrt{y}) = \sqrt{x^T y} = H(x, y) \quad (3.3)$$

Utilizando (*RootSIFT*) em vez do Euclidiana, melhorias significativas de desempenho podem ser obtidas (93).

Com os descritores de todas as imagens calculados, encontra-se quais são as imagens vizinhas e, em seguida, calcula-se a matriz de matches, ou seja, os pontos nas imagens e suas respectivas vizinhas que possuam seus descritores semelhantes. O *matcher* foi baseado na *Fast Approximate Nearest Neighbor Search Library* (FLANN) (94). Para cada característica encontrada pelo algoritmo escolhido, as duas melhores correspondências

candidatas (vizinhos mais próximos) foram mantidas. Para melhorar o conjunto de correspondências, inicialmente, o *ratio test* (59) é aplicado nas correspondências conforme 3.4.

$$\frac{d_1}{d_2} \leq \theta \quad (3.4)$$

Onde,  $d_1$  é a distância do vizinho mais próximo,  $d_2$  é a distância do segundo vizinho mais próximo, e  $\theta = 0,8$ . Caso a condição seja satisfeita, a correspondência é aceita; do contrário, ela é descartada. Ainda assim, nas correspondências encontradas existem *outliers*. Dessa maneira, realizou-se uma primeira filtragem, onde é possível calcular a matriz fundamental que é capaz de mapear pontos de uma imagem na outra com o algoritmo *RANSAC* com auxílio da biblioteca OpenCV. Porém, mesmo com essa técnica ainda apresenta falsas correspondências entre as imagens, como pode ser analisado na Figura 21 onde alguns *outliers* estão realçados por um quadrado amarelo.

Figura 21 – Correspondência entre Imagens Vizinhas com Destaque nos *Outliers*



Fonte: Elaborado pela autora (2021)

Dessa maneira, optou-se por realizar uma segunda filtragem nos *matches* com a metodologia apresentada em (70) para essa finalidade. Selecionou-se randomicamente três pares de correspondências  $(p_1, q_1)$ ,  $(p_2, q_2)$  e  $(p_3, q_3)$  e verifica-se se esses pares são compatíveis, analisando se a seguinte condição é atendida:

$$\forall i \neq j, \tau < \frac{\|p_i - p_j\|}{\|q_i - q_j\|} < \frac{1}{\tau} \quad (3.5)$$

Onde,  $\tau = 0.9$ . Este teste verifica se as correspondências são compatíveis e este será o conjunto usado pelo algoritmo. A Figura 22 representa as duas imagens após essa nova filtragem. Percebe-se que as falsas correspondências não são consideradas e a imagem apresenta os *inliers*, ou seja, apenas bons *matches*.

A realização da filtragem nos pontos das imagens é necessária para evitar que tais pontos prejudiquem o refinamento realizado pelo *Bundle Adjustment*.

Figura 22 – Correspondências entre Imagens após Filtragem



Fonte: Elaborado pela autora (2021)

### 3.2 ETAPA DE *BUNDLE ADJUSTMENT*

Dado um conjunto de correspondências geometricamente consistentes entre as imagens utiliza-se o *Bundle Adjustment* para resolver todos os parâmetros da câmera em conjunto. As imagens são adicionadas juntamente com suas correspondentes e são inicializadas com os valores dos *matches* de suas vizinhas, focos e centro óticos da câmera e rotação. Em seguida, os parâmetros são atualizados utilizando os métodos de otimização: *Grey wolf optimizer (GWO)* e *Bat Algorithm (BA)*.

#### 3.2.1 Modelagem matemática do problema

Um passo importante na formação de imagens panorâmicas é otimizar alguns parâmetros da câmera para determinar a posição de cada imagem no modelo final 360° e sua ocupação na esfera. Para o presente trabalho, serão utilizados, como referências os valores de *pan* e *tilt*, encontrados a partir da matriz de rotação das imagens, focos e centros óticos da câmera utilizada. Mesmo assim, será necessário encontrar de forma mais precisa esses valores para obter uma perfeita imagem panorâmica. Para isso, o intuito é o obter os melhores conjuntos de parâmetros que resultam na menor função de custo pré-definida.

Para resolução desse problema de posicionamento das câmeras, utiliza-se a abordagem que se relacionam as imagens de acordo com a matriz de homografia. As homografias de pares de imagens são utilizadas para projetá-las em um plano de referência. As transformações entre duas imagens podem ser consideradas como uma homografia especial, já que todas as imagens compartilham o mesmo centro ótico (36). A homografia que representa os pares entre as imagens *i*-ésima e *j*-ésima é dada pela Equação 3.6.

$$H_{ij} = K_i R_i R_j^T K_i^{-1} \quad (3.6)$$

Onde,  $K_i = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$  representa a matriz intrínseca,  $f_x, f_y$  os focos e  $c_x, c_y$

o centro ótico da câmera. A rotação é representada por :  $R_i = R_{\theta_i}R_{\phi_i}$ . Sendo  $\theta$  e  $\phi$ , os ângulos de *pan* e *tilt*, respectivamente.

As correspondências de pontos encontradas com SIFT podem ser utilizadas nesse processo, onde, cada *feature* é projetado em todas as imagens em que corresponde, e a soma das distâncias ao quadrado das imagens são minimizadas em relação aos parâmetros da câmera. Para cada câmera de imagem pode ser otimizado os parâmetros  $(\phi, \theta)$  que representam *tilt* e *pan*, respectivamente (95). Assim, como os valores de focos e centros óticos da câmera. Então, se a  $k$ -ésima *feature* em uma imagem corresponde à  $m$ -ésima *feature* em outra, obtêm-se o resíduo de projeção entre as imagens, conforme a Equação 3.7.

$$r_{ij}^k = u_i^k - p_{ij}^k \quad (3.7)$$

Onde,  $u_i^k$  representa a  $k$ -ésima característica na  $i$ -ésima imagem,  $r_{ij}^k$  é o resíduo da projeção da  $k$ -ésima *feature* da  $j$ -ésima imagem na  $i$ -ésima imagem, e  $p_{ij}^k$  é a projeção da imagem  $j$  para imagem  $i$  do ponto correspondente  $u_j^m$ , representado por:

$$p_{ij}^k = K_i R_i R_j^T K_i^{-1} u_j^m \quad (3.8a)$$

$$p_{ij}^k = H_{ij} u_j^m \quad (3.8b)$$

Sendo assim, cada câmera pode ser parametrizada pela rotação  $(\phi_i, \theta_i)$ , focos  $(f_i)$ , centros óticos  $(c_i)$ . Assim, a função custo é dada pelo erro representado pela soma ao quadrado de todas as imagens dos erros residuais:

$$C(\{(\phi_i, \theta_i, f_i, c_i)\}_{i=1}^N) = \sum_{i=1}^N \sum_{j \in I(i)} \sum_{k \in F(i,j)} (r_{ij}^k)^2 \quad (3.9)$$

Onde,  $N$  é o número de imagens,  $I(i)$  é o conjunto de imagens correspondentes a imagem  $i$  e  $F(i, j)$  é o conjunto de *features* entre as imagens  $i$  e  $j$ .

Os valores dos parâmetros serão encontrados utilizando os métodos de otimização baseados em algoritmos bioinspirados. O referido problema a ser otimizado consiste na minimização da equação 3.9 que correspondente à soma dos erros residuais ao quadrado de todas as imagens. O problema é composto por um conjunto de restrições sendo inequações relacionadas aos valores mínimo e máximo dos parâmetros. Dessa forma, o modelo matemático do problema de otimização pode ser descrito conforme as equações 3.10 e 3.11:

$$\text{Minimizar } FOB = C(\{(\phi_i, \theta_i, f_i, c_i)\}_{i=1}^N) \quad (3.10)$$

$$\phi_i^{min} \leq \phi_i \leq \phi_i^{max} \quad (3.11a)$$

$$\theta_i^{min} \leq \theta_i \leq \theta_i^{max} \quad (3.11b)$$

$$f_i^{min} \leq f_i \leq f_i^{max} \quad (3.11c)$$

$$c_i^{min} \leq c_i \leq c_i^{max} \quad (3.11d)$$

Onde ,

*FOB*: Função objetivo do *bundle adjustment*;

*C*: Função custo do erro de imagem *i*;

$\phi_i$ : Valor em graus de *tilt* da imagem *i*;

$\theta_i$ : Valor em graus de *pan* da imagem *i*;

$f_i$ : Focos *x* e *y* da imagem *i*;

$c_i$ : Centros óticos *x* e *y* da imagem *i*;

$\phi_i^{min}$ : Valores mínimos em graus de *tilt* da imagem *i*;

$\theta_i^{min}$ : Valores mínimos em graus de *pan* da imagem *i*;

$f_i^{min}$ : Valores mínimos dos focos da imagem *i*;

$c_i^{min}$ : Valores mínimos dos centros óticos da imagem *i*;

$\phi_i^{max}$ : Valores máximos em graus de *tilt* da imagem *i*;

$\theta_i^{max}$ : Valores máximos em graus de *pan* da imagem *i*;

$f_i^{max}$ : Valores máximos dos focos da imagem *i*;

$c_i^{max}$ : Valores máximos dos centros óticos da imagem *i*;

As equações 3.11a a 3.11d representam as limitações do problema as quais devem ser satisfeitas, a fim de proporcionar um resultado satisfatório no processo de otimização. Os limites que definem os parâmetros entre um valor mínimo e máximo, foram estabelecidos de forma empírica para essa solução. Durante a aquisição de imagens, tem-se os parâmetros intrínsecos e extrínsecos das câmeras, dessa forma, como se tem os valores reais utilizou-se esses dados para criar os limites em um range de +5 e -5 dos valores originais.

### 3.2.2 Métodos de Otimização

O computador sempre foi usado para solução de problemas de otimização utilizando, antigamente, os métodos tradicionais. Com passar do tempo e devido as suas limitações,

novas técnicas foram tomando espaço. A natureza passou a inspirar a construção de novos algoritmos de otimização para solução de problemas complexos, sendo chamados de sistemas bioinspirados ou inspirados na natureza (96).

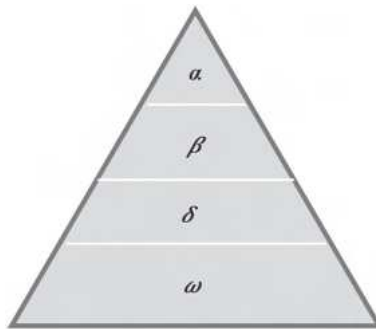
Os algoritmos baseados na natureza, segundo Mirjalili *et al.* (97), são classificados em 2 classes principais: *i*) baseado em solução única (processo de pesquisa começa com uma solução candidata) e, *ii*) baseado em população (o processo se inicia com um conjunto de soluções denominadas população). Em relação a essas aplicações percebe-se um uso em diversas aplicações das técnicas baseadas em população.

Neste trabalho optou-se a utilizar duas técnicas bioinspiradas de forma a verificar o método que encontra melhor resultado para o problema o proposto. Utilizou-se meta-heurísticas baseadas em população onde a cada iteração algumas soluções são modificadas e outras chegam até a próxima iteração. As modificações das soluções, geralmente, são feitas pelas propriedades específicas de cada algoritmo e sua população (98). Para tanto, fez-se uso do *Grey Wolf Optimizer (GWO)* e *Bat Algorithm (BA)*.

### 3.2.2.1 Grey Wolf Optimizer (GWO)

O algoritmo *Grey Wolf Optimizer*(GWO) (97) é inspirado no comportamento e hierarquia dos lobos cinzentos, conforme ilustrado na Figura 23.

Figura 23 – Hierarquia dos Lobos Cinzentos



Fonte: Elaborado pela autora (2021)

O GWO imita a liderança e o mecanismo de caça dos lobos cinzentos, a matilha possui quatro níveis diferentes. O topo da pirâmide é composto pelos líderes, chamados de lobos alfas ( $\alpha$ ), esses comandam a matilha e os outros lobos os seguem. Logo, abaixo tem-se os lobos betas ( $\beta$ ) que são os melhores candidatos a se tornarem alfas, seguidos pelos lobos deltas ( $\delta$ ) que são os responsáveis pela comida e trabalham em caso de perigo, e no nível mais baixo, tem-se os lobos ômeegas ( $\omega$ ) que são os responsabilizados pelas falhas e só se alimentam depois dos demais. Dessa maneira, a hierarquia dos lobos é formada por  $\alpha$ ,  $\beta$ ,  $\delta$  e  $\omega$ .



O comportamento dos lobos cinzentos em relação a caça de suas presas segue os principais passos:

- a) Procurando pela presa;
- b) Rastreamento, perseguindo e abordando a presa;
- c) Cercando e assediando a presa até que ela pare de se mover;
- d) Atacando a presa.

Seguindo esse padrão, o algoritmo dos lobos foi baseado nessa conduta. A Figura 24 representa o pseudocódigo do algoritmo do *GWO*.

Figura 24 – Pseudocódigo do *Grey Wolf Optimizer (GWO)*

---

```

1:      Inicialização da população de lobos:  $X_i$  ( $i = 1, 2, \dots, n$ )
2:      Inicialização  $a, A, C$ 
3:      Cálculo da fitness para cada agente:  $Fob(X_i)$ 
4:      Definição da hierarquia:  $X_\alpha, X_\beta$  e  $X_\delta$ 
5:      Enquanto não atingir critério de parada faça:
6:          Atualizar a parâmetro de busca:  $a^t$ 
7:          Loop 1: n
8:               $\vec{A} = 2 \cdot a^t \cdot \vec{r}_1 - a^t$ ,  $\vec{r}_1 \in [0,1]$ 
9:               $\vec{C} = 2 \cdot \vec{r}_2$ ,  $\vec{r}_2 \in [0,1]$ 
10:              $\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha^t - \vec{X}_i^t|$ ,  $\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta^t - \vec{X}_i^t|$ ,  $\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta^t - \vec{X}_i^t|$ 
11:              $\vec{X}_1 = \vec{X}_\alpha^t - \vec{A}_1 \cdot \vec{D}_\alpha$ ,  $\vec{X}_2 = \vec{X}_\beta^t - \vec{A}_1 \cdot \vec{D}_\beta$ ,  $\vec{X}_3 = \vec{X}_\delta^t - \vec{A}_1 \cdot \vec{D}_\delta$ 
12:              $\vec{X}_i^{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$ 
13:             Atualização da hierarquia:  $X_\alpha, X_\beta$  e  $X_\delta$ 
14:             Fim
14:      Fim

```

---

Fonte: Elaborado pela autora (2021)

O *GWO*, primeiramente, realiza a inicialização aleatória das possíveis  $n$  soluções e a partir das mesmas a obtenção da função objetivo do problema em questão. A partir dos valores obtidos da função objetivo, define-se a hierarquia dos lobos ( $X_\alpha, X_\beta$  e  $X_\delta$ ) de acordo com as melhores soluções.

Em seguida, tem-se o processo que representa a caça dos lobos, onde tem-se as equações utilizadas para atualizar a posição de cada lobo de acordo com a sua presa. Para essa atualização tem-se os coeficientes  $A$  e  $C$ , dados pelas equações 3.12 e 3.13, que ajustam a posição atual dos agentes de pesquisas do algoritmo. O componente  $a^t$  decresce

linearmente de 2 a 0 de acordo com as iterações, e os vetores  $\vec{r}_1$  e  $\vec{r}_2$  são vetores randômicos com range  $[0, 1]$ .

$$\vec{A} = 2a^t \vec{r}_1 - a^t \quad (3.12)$$

$$\vec{C} = 2\vec{r}_2 \quad (3.13)$$

Esses componentes ajudam na busca do lobo em relação a sua presa, ou seja, encontrar a solução. Se  $A < 1$  força o lobo a atacar sua presa, com  $A > 1$  o lobo irá encontrar uma nova presa. Já o componente  $C$  ajuda a presa, dificultando os lobos de encontrá-la, se tornando um peso extra na presa. Quando  $C < 1$  significa redução da importância da solução, se  $C > 1$  enfatiza o efeito da presa na definição da distância. Dessa forma, as equações que representam o comportamento dos lobos durante a caça é dada pelas equações 3.14 e 3.15.

$$\vec{D}_i^t = |\vec{C} \vec{X}_p^t - \vec{X}_i^t| \quad (3.14)$$

$$\vec{X}_i^{t+1} = \vec{X}_p^t - \vec{A} \vec{D}_i^t \quad (3.15)$$

Onde  $t$  é a iteração atual,  $\vec{X}_p$  corresponde a posição da presa,  $\vec{X}_i$  a posição do lobo cinzento. A caça é guiada pelos lobos alfa ( $\alpha$ ), beta ( $\beta$ ) e delta ( $\delta$ ) e, assume-se que os mesmos possuem o melhor conhecimento sobre a posição da presa, ou seja, a solução ótima. Entretanto, não se tem conhecimento sobre a localização da presa, e, para simular matematicamente o comportamento de caça, atualiza-se a posição dos outros lobos em relação a posição de alfa ( $\alpha$ ), beta ( $\beta$ ) e delta ( $\delta$ ) já que possuem a melhor compreensão sobre a posição da presa. As equações 3.16 representam as distâncias e as equações 3.17, os deslocamentos dos lobos. Então, a posição do lobo cinzento ( $\vec{X}_i^{t+1}$ ) é atualizada pela média dos deslocamentos dos lobos dominantes, conforme equação 3.18 .

$$\vec{D}_\alpha = |\vec{C}_1 \vec{X}_\alpha^t - \vec{X}_i^t| \quad (3.16a)$$

$$\vec{D}_\beta = |\vec{C}_2 \vec{X}_\beta^t - \vec{X}_i^t| \quad (3.16b)$$

$$\vec{D}_\delta = |\vec{C}_3 \vec{X}_\delta^t - \vec{X}_i^t| \quad (3.16c)$$

$$\vec{X}_1 = \vec{X}_\alpha^t - \vec{A}_1 \vec{D}_\alpha \quad (3.17a)$$

$$\vec{X}_2 = \vec{X}_\beta^t - \vec{A}_2 \vec{D}_\beta \quad (3.17b)$$

$$\vec{X}_3 = \vec{X}_\delta^t - \vec{A}_3 \vec{D}_\delta \quad (3.17c)$$

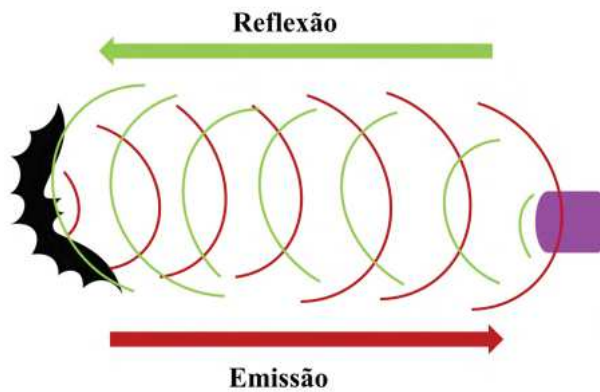
$$\vec{X}_i^{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (3.18)$$

Após a atualização das posições, realiza-se a validação no espaço de busca e calcula-se a função objetivo e, se os novos valores forem melhores que os lobos alfa ( $\alpha$ ), beta ( $\beta$ ) e delta ( $\delta$ ), atualiza-se a hierarquia com esses valores nas posições.

### 3.2.2.2 Bat Algorithm (BA)

O *Bat Algorithm (BA)* é um algoritmo de otimização, proposto por Xin-She Yang em 2010 (99) baseado no comportamento dos morcegos. O algoritmo BA tem as vantagens de um método teórico simples. A maioria das espécies de morcegos utilizam de ecolocalização para se comunicar, sentir a distâncias de suas presas e se locomoverem sem atingirem obstáculos. A ecolocalização, conforme apresentado na Figura 25, mostra a capacidade de determinar posição ou distâncias de objetos por meio de emissão e transmissão de ondas, pela análise do tempo gasto para que as ondas emitidas reflitam no objeto e voltem (100).

Figura 25 – Sistema de Ecolocalização



Fonte: Elaborado pela autora (2021)

Essa maneira de se localizar, faz com que os morcegos voem aleatoriamente com uma certa velocidade através da sua posição e com uma frequência mínima, variando o comprimento de onda e volume para procurar as presas, podendo ajustar automaticamente o comprimento de onda (ou frequência) de seus pulsos emitidos e ajustar a taxa de emissão de pulso, dependendo da proximidade de seu alvo (101). A Figura 26 representa o algoritmo baseado no sistema de ecolocalização dos morcegos.

O algoritmo inicializa com a definição da parâmetros que serão necessários para o processo de ecolocalização no qual se baseia. Inicializa-se a quantidade de morcegos

Figura 26 – Pseudocódigo do *Bat Algorithm (BA)*


---

```

1:      Inicialização dos parâmetros:  $n, \alpha, \lambda$ 
2:      Inicialização da população de morcegos:  $X_i (i = 1, 2, \dots, n)$ 
3:      Cálculo da fitness para cada morcego:  $Fob(X_i)$ 
4:      Atualização do melhor morcego  $X_*^t$ 
5:      Enquanto não atingir critério de parada faça:
6:          para  $i$  de 1 até  $n$  faça
7:               $fr_i = fr_{min} + (fr_{max} - fr_{min})\beta$   $\beta \in [0,1]$ 
8:               $V_i^{t+1} = V_i^t + (X_i^t - X_*^t) fr_i$ 
9:               $X_i^{t+1} = V_i^{t+1} + X_i^t$ 
10:             se  $rand > r_i$ ,  $rand \in [0, 1]$ , então Etapa de Busca Local
11:                  $X_i^{t+1} = round(X_i^t + \epsilon \cdot média(A_i^t))$ ,  $\epsilon \in [-1, 1]$ 
12:             fim
13:             Verificação de violação dos limites das variáveis
14:             Avaliação do morcego  $Fob(X_i^{t+1})$ 
15:             se  $rand > A_i^t$  e  $Fob(X_i^{t+1}) \leq Fob(X_i^t)$ ,  $rand \in [0, 1]$ , então
16:                  $r_i^{t+1} = 1 - (\lambda t)$ 
17:                  $A_i^{t+1} = \alpha A_i^t$  Etapa de Busca Global
18:             fim
19:             Atualização melhor morcego:  $X_*^t$ 
20:         fim
21:     fim

```

---

Fonte: Elaborado pela autora (2021)

( $n$ ), a taxa de decréscimo da amplitude sonora ( $\alpha$ ) e a taxa de aumento da emissão do pulso da onda sonora ( $\lambda$ ). Também, são gerados os valores da população ( $X_i$ ) de morcegos que representa a posição, a velocidade de cada morcego ( $V_i$ ), a frequência ( $fr_i$ ), a taxa de emissão de pulsos ( $r_i$ ) e amplitude ( $A_i$ ). Esses valores são gerados de forma aleatória, respeitando os limites da região da solução. Com a população inicial conhecida, realiza-se a avaliação da função objetivo, de acordo com cada solução e por meio do seu valor numérico determina-se a melhor solução corrente, ou seja, o melhor morcego ( $X_*$ ).

Em seguida, o processo iterativo acontece até que seu critério de parada seja alcançado. Nessa etapa, a cada iteração, o morcego voa com velocidade  $V_i^{t+1}$ , que é afetada por uma frequência  $fr_i$  pré-definida aleatoriamente. E, finalmente, o morcego localiza sua nova posição no espaço de busca  $X_i^{t+1}$ . Sendo assim, a velocidade, frequência e posição do morcego  $i$  são atualizados de acordo com as equações 3.19, 3.20 e 3.21.

$$fr_i = fr_{min} + (fr_{max} - fr_{min})\beta \quad (3.19)$$

$$V_i^{t+1} = V_i^t + (X_i^t - X_*^t)fr_i \quad (3.20)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (3.21)$$

Onde,  $fr_i$  é a frequência pertencente ao  $i$ -ésimo morcego;  $fr_{max}$  e  $fr_{min}$  são os valores de frequência máxima e mínima, respectivamente e  $\beta$  é um valor aleatório gerado entre  $[0, 1]$ .

Após a atualização dos parâmetros, o algoritmo dos morcegos propõe uma etapa de busca local em torno da melhor solução com intuito de melhorar a capacidade de busca para que o morcego possa melhorar a solução. Essa busca consiste na geração de uma nova solução, a partir das melhores soluções e adicionando um valor proveniente da amplitude sonora. Para entrar nessa etapa, compara-se a taxa de emissão de pulso ( $r_i$ ) com um valor aleatório entre  $[0, 1]$ . Caso satisfeita tal condição atualiza-se a posição ( $X_i^{t+1}$ ) em relação ao valor do melhor morcego ( $X_*^t$ ) e à média da amplitude sonora ( $A_i^t$ ) e o valor  $\epsilon$  randômico entre 0 e 1. A etapa de busca local é representada pela equação 3.22

$$X_i^{t+1} = X_*^t + \epsilon \cdot média(A^t) \quad (3.22)$$

Após a busca local, o algoritmo realiza a busca global que consiste na alteração da solução corrente, na taxa de emissão de pulsos e na amplitude sonora. Nesta etapa, avalia-se duas condições: *a*) se um valor randômico é menor que a amplitude do pulso e, *b*) se o valor da função objetivo atual é menor que o valor na iteração anterior. Se satisfeitas as condições, tem-se a atualização da taxa de emissão de pulso ( $r_i^{t+1}$ ) e da amplitude do pulso ( $A_i^{t+1}$ ) do morcego ( $X_i^{t+1}$ ). A etapa de busca global é representada pelas equações 3.23 e 3.24.

$$r_i^{t+1} = 1 - e^{\lambda t} \quad (3.23)$$

$$A_i^{t+1} = \alpha A_i^t \quad (3.24)$$

Dessa forma, avalia-se, através da função objetivo, as soluções e atualiza-se o melhor morcego com a melhor solução.

### 3.2.3 Formulação do problema

Para a formulação do problema em questão, primeiramente, exemplifica-se como os indivíduos são formados para ambos os algoritmos de otimização. A Figura 27 representa a formação dos indivíduos de acordo com os parâmetros de entrada que serão ajustados .

Dado um conjunto de  $n$  imagens, tem-se que  $p_n$  representa os valores dos ângulos em  $pan$ ,  $t_n$  são os valores dos ângulos em  $tilt$  de cada imagem;  $f_{xn}$  e  $f_{yn}$  são os focos em x e y, respectivamente e,  $C_{xn}$  e  $C_{yn}$  os centros óticos em x e y, respectivamente. Dessa

Figura 27 – Exemplificação dos Indivíduos

$t_1$	$p_1$	$f_{x1}$	$f_{y1}$	$c_{x1}$	$c_{y1}$	$t_2$	$p_2$	$f_{x2}$	$f_{y2}$	$c_{x2}$	$c_{y2}$	...	$t_n$	$p_n$	$f_{xn}$	$f_{yn}$	$c_{xn}$	$c_{yn}$
-------	-------	----------	----------	----------	----------	-------	-------	----------	----------	----------	----------	-----	-------	-------	----------	----------	----------	----------

Fonte: Elaborado pela autora (2021)

forma, para cada imagem seis parâmetros serão ajustados. Logo, ao final da otimização, considerando o número total de imagens, tem-se **6n** parâmetros otimizados.

Para realização da otimização, inicializa-se os métodos conforme os parâmetros e o tamanho da população ( $i$ ). Os indivíduos da população são gerados randomicamente dentro da região de solução. Então, todos os algoritmos meta-heurísticos são executados com o mesmo grupo inicial para evitar que o viés do ponto de partida interfira na comparação. Nesse sentido, o número total de avaliações da função objetivo por método é igual. Isso visa garantir que os algoritmos retornem uma solução final pelo menos tão boa quanto a que foi adquirida inicialmente.

A cada aquisição de imagens obtém-se o conjuntos de imagens assim como um arquivo do tipo *sfm* com os dados referentes de cada imagem. O *sfm* é dado no seguinte formato :

$$\langle camera \rangle = \langle CaminhodasImagens \rangle \langle R \rangle \langle T \rangle \langle f_x \rangle \langle f_y \rangle \langle c_x \rangle \langle c_y \rangle$$

Onde, *CaminhodasImagens* representa a localização das imagens,  $R$  os valores de cada elemento da matriz de rotação,  $T$  são os elementos do vetor de translação de cada imagem,  $f_x$  e  $f_y$  são os focos da câmera e  $c_x$  e  $c_y$  são os centros óticos da câmera. Essas informações são utilizadas durante a inicialização, onde atribui-se a um indivíduo os valores obtidos durante a captura das imagens. Os demais são gerados de forma aleatória dentro dos limites exigidos, conforme equação 3.25, onde  $[r]_0^1$  é um número aleatório entre 0 e 1,  $b_{d_{up}}$  são os limites superiores de cada parâmetro e  $b_{d_{lo}}$  os limites inferiores. Após a inicialização da população e dos parâmetros necessários para cada método, realiza-se o processo de otimização.

$$X_{i,d} = [r]_0^1 \times (b_{d_{up}} - b_{d_{lo}}) + b_{d_{lo}}; \quad (3.25)$$

Para avaliação da função objetivo em ambos métodos são necessários, além dos parâmetros a serem otimizados, os pares de *matches* das imagens que foram obtidos conforme seção 3.1. Para o cálculo do erro de re-projeção, limitou-se o número de correspondências entre as imagens, para otimizar o tempo computacional gasto. Para isso, das boas correspondências, selecionou-se aleatoriamente 50 *matches*. Logo, para cada imagem e suas respectivas vizinhas, avaliou-se o erro em 50 correspondências.

Para se avaliar as metodologias propostas utilizou-se o mesmo conjunto de corres-

pondências e valores iniciais dos indivíduos para melhor se comparar o comportamento de cada algoritmo. Os resultados serão apresentados no Capítulo 4.

Após, o ajuste das câmeras, realiza-se o processo de *stitching* das imagens para obtenção da imagem panorâmica 360° final.

### 3.3 STITCHING

Após ajuste no posicionamento das imagens, por meio do *Bundle Adjustment*, inicializa-se o processo de *stitching* das imagens. Com os valores de ângulos de *pan* e *tilt* otimizados, assim como os focos e centro óticos, realiza-se, primeiramente, a projeção das imagens.

Realiza-se a projeção esférica, conforme seção 2.11.2. Primeiramente, supõe uma esfera com raio igual 1. Também, cria-se a imagem final com a resolução desejada. Para isso, é estabelecido um *step* para definir e variar a resolução da imagem final. Sendo esses valores necessários para o decorrer da metodologia.

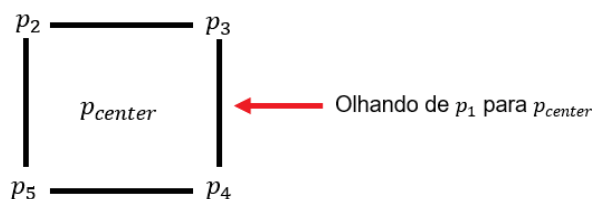
O tamanho da imagem panorâmica final será definida como apresentado nas equações 3.26a e 3.26b

$$u = \frac{360}{step} \quad (3.26a)$$

$$v = \frac{u}{2} \quad (3.26b)$$

O *step* escolhido para os experimentos foi de 0.1. Dessa forma a panorâmica final terá resolução de 3599 x 1799, considerando seu início em (0,0). Após, essa definição realiza-se a projeção das imagens. Para isso, os pixels 2D devem ser projetados para um ponto 3D. Então, considera-se um *frustum* da imagem, conforme Figura 28.

Figura 28 – Cantos do *Frustum* da Imagem



Fonte: Elaborado pela autora (2021)

Para projeção de imagem, as transformações entre as coordenadas da imagem e coordenadas da câmera são requeridas. Necessita-se da posição da câmera para que se

possa obter a relação dada pela equação 3.27.

$$p_c = R * p \quad (3.27)$$

Onde,  $p_c$  trata-se do ponto transformado em coordenadas da câmera,  $R$  é uma matriz 3x3 que representa a posição da câmera. Essa matriz de rotação é obtida com os valores de  $pan(\theta)$  e  $tilt(\phi)$  encontrados na etapa de otimização. Sendo,  $R_i = R_{\theta_i} R_{\phi_i}$ , tem-se:

$$R_{\phi_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_i & -\text{sen}\phi_i \\ 0 & \text{sen}\phi_i & \cos\phi_i \end{bmatrix} \quad (3.28a)$$

$$R_{\theta_i} = \begin{bmatrix} \cos\theta_i & 0 & \text{sen}\theta_i \\ 0 & 1 & 0 \\ -\text{sen}\theta_i & 0 & \cos\theta_i \end{bmatrix} \quad (3.28b)$$

A matriz  $R$  final será representa por 3.29:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.29)$$

Com a matriz, de posição da câmera estabelecida, pode-se encontrar os valores dos cantos do *frustum*, conforme Figura 28. Para isso, encontram-se os valores máximos e mínimos em relação a  $x$  e  $y$  da imagem.

$$\max_x = F * \frac{\text{largura}}{2 * f_x} \quad (3.30a)$$

$$\min_x = -\max_x \quad (3.30b)$$

$$\max_y = F * \frac{\text{altura}}{2 * f_y} \quad (3.30c)$$

$$\min_y = -\max_y \quad (3.30d)$$

Onde  $F$  é o foco em dimensões físicas do *frustum*, tendo valor igual ao raio da esfera e,  $f_x$  e  $f_y$  são os focos da câmera. Sendo assim, calcula-se os pontos em relação a câmera:

$$p_1 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.31a)$$



$$p_2 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} \min_x \\ \min_y \\ F \end{bmatrix} \quad (3.31b)$$

$$p_3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} \max_x \\ \min_y \\ F \end{bmatrix} \quad (3.31c)$$

$$p_4 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} \max_x \\ \max_y \\ F \end{bmatrix} \quad (3.31d)$$

$$p_5 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} \min_x \\ \max_y \\ F \end{bmatrix} \quad (3.31e)$$

$$p_{center} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \quad (3.31f)$$

A partir do *frustum*, encontra-se, então, a posição de cada ponto da imagem fonte em  $(x, y, z)$  de acordo com a resolução da imagem final. Para isso, realizou-se os seguintes cálculos:

$$hor_{step} = \frac{(p_4 - p_5)}{colunas} \quad (3.32)$$

$$ver_{step} = \frac{(p_2 - p_5)}{linhas} \quad (3.33)$$

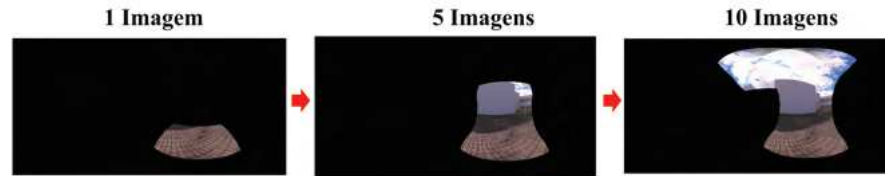
Logo, o ponto projetado se dará por meio da equação

$$ponto = p_5 + hor_{step} * j + ver_{step} * i \quad (3.34)$$

Onde,  $j$  e  $i$  são as coordenadas do pixel da imagem em  $x$  e  $y$ . E, com isso, as coordenadas esféricas são calculadas, conforme as equações 2.21 e 2.22. Dessa forma, pelas coordenadas, estima-se a posição do pixel na imagem esférica final e atribui-se a cor desse ponto referente a imagem. A figura 29 representa a forma como as imagens são projetadas. Na primeira imagem da figura tem-se apenas uma imagem, na segunda 5 imagens e na terceira a representação de 10 imagens projetadas.

A cada a projeção na imagem na resolução final, realiza-se *Multi-band Blending* para suavizar a coloração das imagens nas áreas de sobreposição entre duas imagens.

Figura 29 – Projeção Esférica de Imagens



Fonte: Elaborado pela autora (2021)

### 3.3.1 Multi-band Blending

Quando se realiza a projeção de imagens para formar uma panorâmica, haverá, em alguns casos, linhas no limite das imagens nas áreas de sobreposição devido, por exemplo, a intensidade dos pixels adjacentes diferirem (17). Dessa maneira, é necessário um algoritmo de correção para melhorar o resultado final, para que não se perceba que a imagem foi construída a partir de múltiplas imagens.

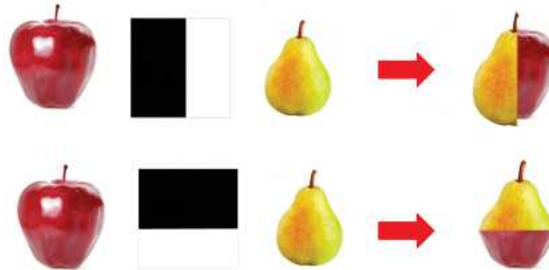
O *Multi-Band Blending* realiza a decomposição da imagem em frequência de multi-banda baseado em pirâmide. A ideia por trás da combinação de várias bandas é combinar baixas frequências em uma grande faixa espacial e altas frequências em uma faixa curta onde cada banda pode ser combinada de forma eficaz (102). Dessa forma, o *Multi-Band Blending* realiza a combinação de imagens multi-banda utilizando pirâmide Laplaciana. Usa-se Kernel Gaussiano para misturar a imagem, mantendo as características significativas no tempo médio que permite misturar tanto a cor da imagem (baixas frequências) quanto os detalhes das imagens (altas frequências). Diferentes níveis da pirâmide laplaciana são calculados com base na subtração entre o nível da imagem gaussiana e a expansão do nível inferior da pirâmide gaussiana (103).

Para iniciar o processo de mesclagem de imagens utilizando tal técnica, é necessário, inicialmente, criar máscaras binárias correspondentes a pesos associados a cada imagem de entrada localizando o conjunto de pontos para o qual cada imagem é responsável. Nas regiões para as quais uma determinada imagem de entrada é a responsável, o valor 255 é atribuído à sua respectiva máscara. Nas demais regiões (para as quais outras imagens de entrada são as responsáveis), o valor 0 é atribuído.

Como é exigido pelo método encontra-se as máscaras binárias de cada imagem a ser combinada. As máscaras são feitas a partir das áreas em comum entre as imagens de entrada. Encontra-se o *overlap* dividindo-a entre pixels pretos (0) e brancos (255) para que cada imagem fosse responsável por uma parte e ter características de ambas. Em seguida, realiza-se o processo conforme explicado anteriormente. A máscara é um fator importante para que a imagem final tenha um resultado satisfatório. A Figura 30 mostra dois tipos de máscaras para a mesma entrada de imagens e como o o resultado da mesclagem é diferente.

Dessa forma, diferentes tipos de máscaras permitem combinações mais complexas.

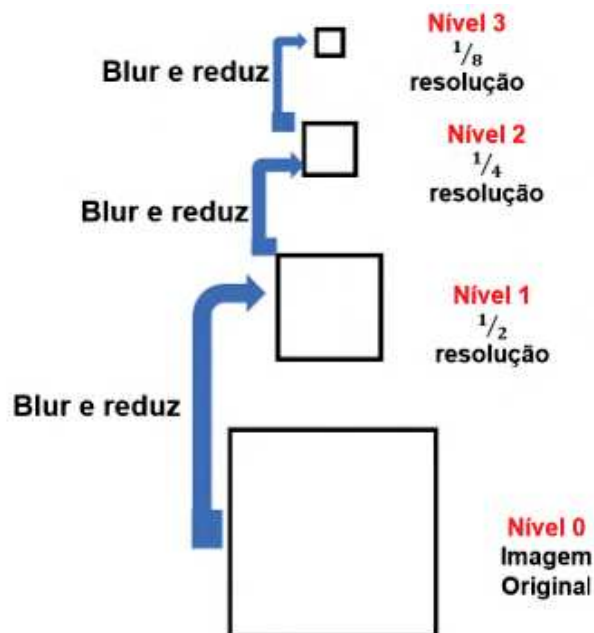
Figura 30 – Combinação e Mesclagem de Imagens com *Multi-band Blending*



Fonte: Elaborado pela autora (2021)

Em seguida, uma pirâmide Gaussiana e as imagens filtradas passa-baixa  $G_0, G_1, \dots, G_N$  podem ser obtidas por convolução repetida de uma função Gaussiana na imagem de entrada aplicando uma função de média ponderada gaussiana em cada pixel a cada iteração. Além do filtro, a imagem é sub-amostrada de forma que o resultado seja uma imagem reduzida da metade do tamanho a cada iteração, conforme mostrado na figura 31. Esse procedimento pode ser representado pela equação 3.35;

Figura 31 – Pirâmide Gaussiana



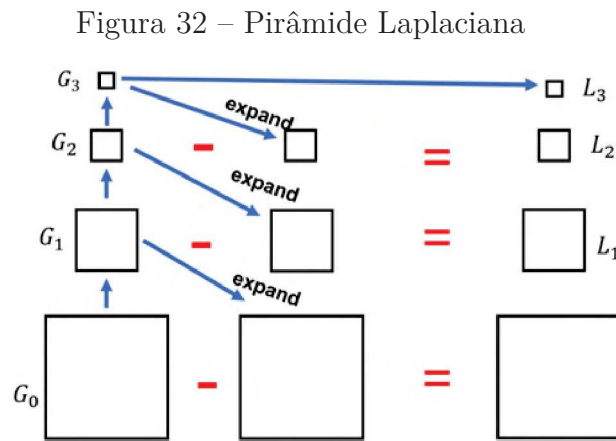
Fonte: Elaborado pela autora (2021)

$$G_i = reduce(G_{i-1}) \quad (3.35)$$

Após, cria-se uma pirâmide Laplaciana a partir da subtração da imagem original de sua versão filtrada com imagens que representam frequências espaciais no intervalo estabelecido pelo valor de desvio padrão do filtro Gaussiano. Para gerar as diferentes bandas de frequência de imagem  $L_i$  são gerados subtraindo os dois filtros passa-baixa  $G_i$  e  $G_{i-1}$ . Como  $G_{i-1}$  possui metade do valor da imagem  $G_i$ , é necessário interpolar novos pixels na imagem. Esse processo pode ser definido como:

$$G'_{i-1} = \text{expand}(G_{i-1}) \quad (3.36)$$

Dessa forma, os valores dos pixels das imagens são interpolados das amostras de forma a torná-la grande o suficiente para subtrair, conforme ilustrado na Figura 32.



Fonte: Elaborado pela autora (2021)

Onde,  $L_0, L_1, \dots, L_N$  representam a pirâmide Laplaciana e pode ser representada pela equação 3.37.

$$L_i = G_i - \text{expand}(G_{i+1}) \quad (3.37)$$

Realiza-se, então, a mistura das pirâmides laplacianas referentes as imagens. O mesmo processo para criar a pirâmide gaussiana foi aplicado nas máscaras binárias em cada nível e, dessa forma, tornam-se os pesos para se mesclar as imagens passa-banda em cada nível da pirâmide. Por fim, a imagem é recomposta adicionando todo o laplaciano e aplicando iterativamente a função *expand* e adicionando o resultado ao nível superior da pirâmide, ou seja, o mosaico da panorâmica final é obtido interpolando e somando todos os níveis da pirâmide. A equação 3.38 representa a criação da imagem mesclada.

$$L_s = G_M L_A + ((1 - G_M) L_B) \quad (3.38)$$

Onde,  $L_s$  representa a imagem mesclada,  $G_M$  a máscara binária e  $L_A$  e  $L_B$  as imagens laplacianas a serem mescladas.

Para a utilização do *multi-band blending* tem-se como entrada duas imagens a serem mescladas. A cada iteração a nova imagem mesclada serve como no algoritmo para ser mesclada com a próxima imagem, e assim segue até que todas as imagens do conjunto sejam mescladas e formem o mosaico final.

Em resumo, primeiramente obtêm-se a extração das *features* de todas as imagens e cria-se os pares de imagens vizinhas. Para cada par de imagens encontra-se com o algoritmo do SIFT os *matches* entre elas e, realiza-se uma filtragem desses dados para tratar de possíveis *outliers*. Em seguida, realiza-se o processo de *Bundle adjustment* sendo feito por dois algoritmos de otimização para tentar encontrar o melhor ajuste nas posições de todas as câmeras. Em seguida, com os resultados obtidos na otimização projeta-se as imagens e realiza-se a mesclagem das imagens duas a duas.

Dessa maneira, após esse processo obtêm-se um panorama completo 360°. Os resultados obtidos podem ser verificados na Capítulo 4.

## 4 RESULTADOS

Este capítulo apresenta os principais resultados obtidos pela aplicação da metodologia proposta. Para tal foi realizado um estudo comparativo entre os algoritmos BA e GWO visando otimizar os parâmetros intrínsecos e extrínsecos das câmeras. Também, foi realizada a comparação das imagens antes de se realizar a suavização por meio do *Multi-band blending*.

Conforme discutido anteriormente, a metodologia utilizada para a proposta deste trabalho faz uso de duas técnicas de inteligência computacional para resolução do problema. Cada uma delas possui suas particularidades e a comparação entre elas foi feita por meio de mudança nas soluções e características de convergência.

Para realizar tal comparação foram utilizados quatro conjuntos de dados distintos. Sendo três deles compostos por 56 imagens, resultando num total de 336 parâmetros otimizados. O outro conjunto é composto por 70 imagens tendo ao final 420 parâmetros otimizados.

### 4.1 PARÂMETROS UTILIZADOS

Para os algoritmos as configurações em comum utilizadas estão apresentadas na Tabela 1.

Tabela 1 – Configuração de Simulação

	<b>Valor</b>
Número de indivíduos	35
Convergência pelo número máximo de iterações	2000
Quantidade de simulações	10

Fonte: Elaborada pelo autor (2021).

Foram utilizados 35 indivíduos para cada algoritmo e o critério de parada foi o número máximo de iterações de 2000. No total realizou-se 10 simulações para cada método. Para todos os algoritmos utilizou-se as mesmas soluções iniciais, sendo uma o valor inicial vindo do robô e as 34 outras soluções geradas de forma uniforme dentro da região de solução.

Como mostrado no Capítulo 3, além das soluções iniciais e o número máximo de iterações, as metodologias apresentam outros parâmetros que foram determinados de forma empírica por meio de testes. Esses parâmetros e definições são apresentados na Tabela 2.

Tabela 2 – Valores de Parâmetros para os Algoritmos Comparativos

Algoritmo	Parâmetro	Valor	Descrição
BA	A	1	Amplitude Sonora
	r	1	Taxa de emissão
	$\lambda$	0.01	Taxa de aumento da emissão do pulso
	$\alpha$	0.995	Taxa de diminuição da amplitude
	$fr_{min}$	0	Frequência Mínima
	$fr_{max}$	100	Frequência Máxima
GWO	(a)	Redução linear de 2 a 0	Parâmetro de convergência

Fonte: Elaborada pelo autor (2021).

## 4.2 AQUISIÇÃO DE IMAGENS

A aquisição das imagens fonte de informação dos testes realizados nesse trabalho foi realizada através do sistema robótico ilustrado na Figura 33. O robô foi desenvolvido pelo Grupo de Robótica Inteligente (GRIn) da Universidade Federal de Juiz de Fora (UFJF) sendo capaz de virtualizar objetos e cenas. Trata-se de um sistema de escaneamento capaz de obter a reconstrução de todo um ambiente ou objeto 3D que podem ser inseridos em aplicações de realidade aumentada e virtual, por exemplo. Este conceito insere uma câmera e um laser em uma estrutura mecânica que é capaz de girar em condições controladas. Isso auxilia o software de reconstrução de imagens, uma vez que as posições de aquisição são previamente conhecidas.

Figura 33 – Protótipo Scanner 3D



Fonte: Elaborado pela autora (2021)

As especificações técnicas do hardware de auxílio a virtualização consistem na união de dados de imagem e profundidade da cena, obtidos por um *scanner* com processador embarcado, sendo operado remotamente pelo operador, controlando todo o processo de aquisição, organização e transmissão dos dados para processamento. É composto por um *scanner* a laser Livox MID40, uma Câmera USB Full HD C925s, dois servo motores Dynamixel AX-18A, um MPU-6050 *InvenSense Inertial Measurement Unit* (IMU), uma bateria de 12,0 V e 8400 *mAh* e um roteador sem fio com uma rede local. Possui uma placa Jetson Nano embarcada desenvolvida pela NVidia e uma CPU de quatro núcleos, 4 GB de RAM e até uma GPU incorporada.

O uso da câmera de alta resolução e do laser LIVOX MID-40 possibilitou uma aquisição de altíssima precisão para ambientes abertos ou internos e objetos. Este módulo foca no uso das imagens colhidas para reconstrução do ambiente ou objeto não só como uma imagem em 360 graus, mas como nuvem de pontos de alta precisão da cena obtida com o equipamento de virtualização. A utilização deste módulo nesse trabalho foi para obter múltiplas fotos de alta resolução para montagem de panorâmicas 360°, com dados distância e posicionamento provenientes do sensor laser. Isso será fundamental para garantir que a cena de interesse seja capturada de forma efetiva.

Os servo motores são responsáveis por movimentos de *pan* e *tilt* em uma faixa de 360 e 120 graus, respectivamente. Uma IMU dedicada foi usada para medir ângulos de rotação (*roll*) e inclinação (*pitch*) e compensar imperfeições do solo, com as medições submetidas a um filtro passa-baixo no driver. A medição do ângulo de *pan* veio do servo motor, com uma resolução de 0.088graus/*tick*.

A informação de movimentação da pose da câmera (posição e orientação no ambiente) é estimada pela câmera estéreo e capturada no mesmo instante em que a nuvem e a imagem são obtidas. Para esta dissertação, foram utilizados dados fotografados com duas configurações e cada série de conjuntos de imagens foi adquirida com resolução HD 1280 x 720 pixels em PNG.

Para criação de imagens panorâmicas deve-se ter o cuidado de obter imagens com áreas de sobreposição (*overlap*) de 15% a 30%. As configurações adotadas para aquisição foram:

- i*) Utilização de um *step* em *pan* (horizontal) de 50° e em *tilt* (vertical) de 15°, cobrindo a região em 360°. Dessa forma, adquiriu-se conjuntos compostos de 56 imagens; sendo que para valor em *pan* tem-se 8 fotos em *tilt* e, para cada valor em *tilt* tem 7 imagens em *pan* ( $8 \times 7 = 56$ );
- ii*) Utilização de um passo de 25° em valores de *pan* e mantendo a configuração anterior de *tilt*. Sendo assim, os conjuntos de dados escaneados nessa forma possuem 70 imagens no total; sendo 5 imagens na vertical para cada valor de *pan* e 14 imagens



na horizontal para cada valor de *tilt* ( $5 \times 14 = 70$ ).

### 4.3 RESULTADOS INICIAIS

Primeiramente, apresenta-se, os resultados de panorâmicas após a aquisição das imagens dos quatro conjuntos que serão utilizado, ou seja, antes do processo de otimização. As Figuras 34, 35 e 36 representam três pontos de vista diferentes da Faculdade de Artes e Design da Universidade Federal de Juiz de Fora. A Figura 37 representa uma imagem da subestação de energia da cidade de Santos Dumont.

Figura 34 – Faculdade de Artes e Design (UFJF) - Vista 1



Fonte: Elaborado pela autora (2021)

Figura 35 – Faculdade de Artes e Design (UFJF) - Vista 2



Fonte: Elaborado pela autora (2021)

Figura 36 – Faculdade de Artes e Design (UFJF) - Vista 3



Fonte: Elaborado pela autora (2021)

Figura 37 – Subestação de Santos Dumont



Fonte: Elaborado pela autora (2021)

Na Figura 34 percebe-se desalinhamentos visíveis no gerador, postes, chão e nos prédios. Já na Figura 35 há alguns desalinhamentos visíveis no chão e calçadas. A figura 36 apresenta problemas nos prédios, postes e gerador. A subestação, Figura 37, apresenta irregularidades nas calçadas, prédio, e nas linhas de transmissão. Em função disso, aplicou-se as meta-heurísticas na tentativa de resolver tais problemas.

#### 4.4 RESULTADOS DAS TÉCNICAS DE OTIMIZAÇÃO

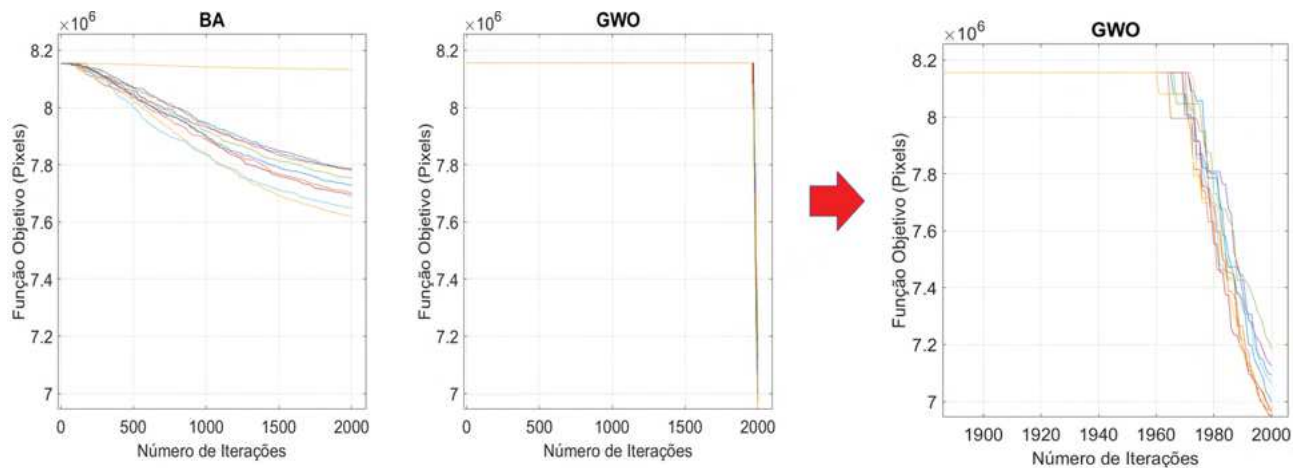
Nesta seção serão apresentados os resultados obtidos pelos algoritmos de otimização. Para isso realizou-se quatro estudos conforme os conjuntos de imagens analisadas. Os três primeiros testes foram utilizando conjuntos de imagens de pontos de vista diferentes da Faculdade de Artes e Design da Universidade Federal de Juiz de Fora. Estes conjuntos são compostos por 56 imagens. O outro teste realizado foi utilizando um conjunto de imagens tiradas da subestação de energia da cidade de Santos Dumont, sendo composto por 70 imagens no total.

Vale ressaltar que cada simulação contou com 2000 iterações e partiram dos mesmo conjunto de soluções iniciais e mesmas *features*, ou seja, para cada conjunto, encontrou-se as correspondências e as utilizou durante as 10 simulações. Como os pontos de *matches* entre as imagens é uma característica determinante para função objetivo, em todas as simulações utilizou-se os mesmos pares de correspondências entre as imagens.

#### 4.4.1 Estudo de Caso 1 - Faculdade de Artes e Design (UFJF)

Para o primeiro estudo realizado tomou-se como referência os valores da imagem original apresentada na Figura 34. Após, simular 10 vezes cada algoritmo, analisou-se o processo de convergência das meta-heurísticas que estão ilustrados pela Figura 38.

Figura 38 – Gráficos de Convergência em 10 simulações - Caso 1



Fonte: Elaborado pela autora (2021)

Os gráficos apresentados na Figura 38 representam as curvas referentes as 10 simulações realizadas. O gráfico apresentado à direita, representa a curva de convergência do GWO com *zoom* na sua etapa final de convergência para ser possível melhor analisar o comportamento do algoritmo. Diante dos resultados apresentados é possível perceber que a fase de exploração do GWO ocorre de forma mais lenta do que as apresentadas pelo BA. Verifica-se por esses dados que o *Bat Algorithm* apresenta uma etapa de busca global melhor que o *Grey wolf Optimizer*, percebe-se que a exploração da região de solução, feita no início do processo, é mais eficiente no BA, o qual, nas primeiras iterações, já encontra soluções de qualidade aceitável.

Nos dois gráficos apresentados anteriormente deve-se atentar aos valores de função objetivo. Por mais que pareçam distantes pela distribuição da figura, os dados estão na mesma ordem de grandeza e próximos. Para facilitar a análise realizou-se mais uma comparação entre os valores numéricos da função objetivo encontrados pelos algoritmos. A Tabela 3 apresenta o melhor resultado de função objetivo encontrada entre as 10 simulações e também o valor da solução original. O valor da função objetivo original, antes de ser otimizada era de  $8.250390 \times 10^6$ , após otimizada pelo *Bat Algorithm* passou a ter valor de  $7.618670 \times 10^6$  e, com *Grey Wolf Optimizer* foi de  $6.945110 \times 10^6$ .

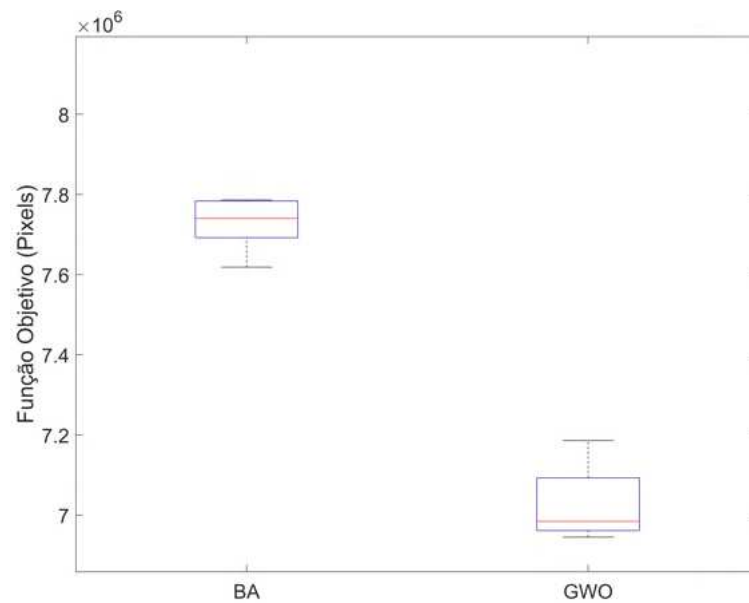
Tabela 3 – Comparação da Função Objetivo - Caso 1

	Valor
<b>Original</b>	$8.250390 \times 10^6$
<b>BA</b>	$7.618670 \times 10^6$
<b>GWO</b>	$6.945110 \times 10^6$

Fonte: Elaborada pela autora (2021).

Comparando os resultados de melhor função objetivo entre as metodologias apresentadas é possível constatar a dispersão das soluções obtidas para o Caso 1, conforme a Figura 39. Analisando o *boxplot*, verifica-se que o GWO apresenta menor dispersão de dados e menor valor de mediana se comparado com BA e desconsiderando os valores dos *outliers*. A Tabela 4 apresenta os valores máximo, mínimo e a mediana obtidos pelas técnicas, extraídos do gráfico *boxplot*.

Figura 39 – *Boxplot* dos Valores Ótimos da Função Objetivo nas 10 Simulações para o Caso 1



Fonte: Elaborado pela autora (2021)

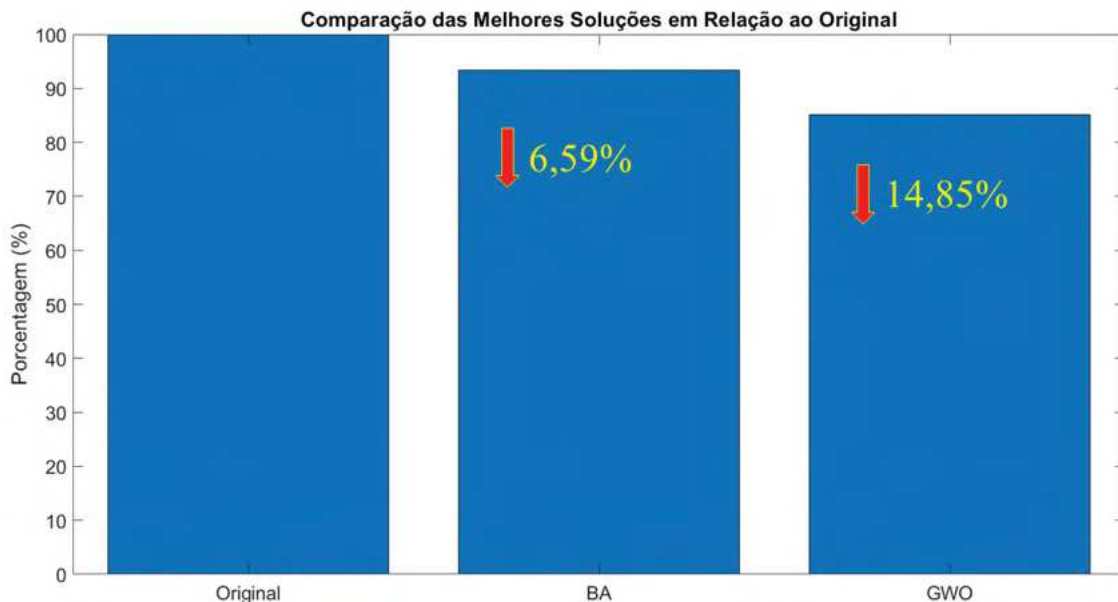
Tabela 4 – Comparação de Valores de Função Objetivo obtidos pelas Meta-heurísticas Extraídos do *Boxplot* para o Caso 1

Algoritmo	Mínimo	Máximo	Mediana
BA	$7.618670 \times 10^6$	$8.133120 \times 10^6$	$7.740220 \times 10^6$
GWO	$6.945110 \times 10^6$	$7.186040 \times 10^6$	$6.984420 \times 10^6$

Fonte: Elaborada pelo autor (2021).

Analisando em valores percentuais, a Figura 40 apresenta a relação dos valores otimizados pelo original ( $\frac{\text{otimizado}}{\text{original}}$ ) para verificar a relação de diminuição do valor da função objetivo comparada com a solução original.

Figura 40 – Comparação das Melhores soluções em Relação à Original - Caso 1



Fonte: Elaborado pela autora (2021)

O algoritmo BA encontrou uma relação de 93,41% em relação ao original. Encontrando esse resultado em porcentagem, tem-se que  $(100 - 93,41 = 6,59)$ , ou seja, houve uma diminuição em relação ao valor original de aproximadamente 6,59%. Já o algoritmo GWO possui uma relação de 85,16%. A porcentagem de diminuição da função objetivo encontrada foi, então, de aproximadamente  $(100 - 85,15 = 14,85)$  14,85%. Em valores numéricos o algoritmo GWO apresenta melhor resultado pois possui menor valor de função objetivo ao final do processo.

Em termos de tempo computacional, a Tabela 11 apresenta, em segundos, o tempo médio de cada algoritmo de otimização levando em consideração as 10 simulações. Pela tabela verifica-se que ambos algoritmos possuem tempo computacional próximos.

Tabela 5 – Tempo Médio Computacional para o Caso 1

Algoritmo	Tempo Médio (segundos)
BA	43,80
GWO	47,76

Fonte: Elaborada pela autora (2021).

Após a otimização, realiza-se a projeção desses resultados para obter a imagem panorâmica. Os resultados de imagem panorâmicas para esses valores podem ser visualizados nas Figuras 41 e 42. São apresentadas as imagens originais e as imagens otimizadas para um comparação entre os resultados.

Figura 41 – Faculdade de Artes e Design (UFJF) Vista 1 sem *Multi-band Blending - Bat Algorithm*



(a) Original



(b) BA

Figura 42 – Faculdade de Artes e Design (UFJF) Vista 1 sem *Multi-band Blending - Grey Wolf Optimizer*



(a) Original



(b) GWO

Comparando visualmente os resultados percebe-se algumas discrepâncias com relação ao valor original e valores numéricos encontrados. O problema de otimização proposto é um problema de vários mínimos locais e a função objetivo é composta de várias parcelas devido a diminuição do erro de re-projeção das *features*. Em função disso, o algoritmo encontra um valor de função objetivo menor que o original mas visualmente isso não se replica, pois devido a característica do problema, as metodologias podem melhorar

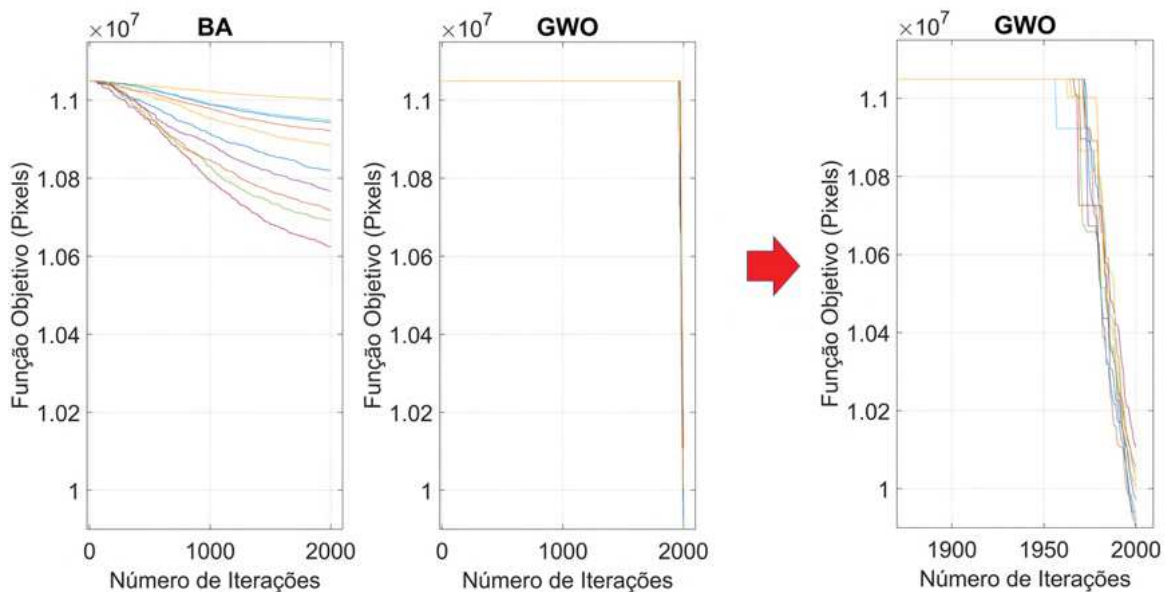
muito uma determinada parcela da função objetivo e outras parcelas nem tanto. O que ocorre é um enviesamento do algoritmo, no qual foca-se na parcela da função objetivo que gera um erro grande e, então o minimiza mas, ignora outros erros. Por isso, a função objetivo apresenta um valor de diminuição de erro mas na aparência das imagens isso não é totalmente replicado.

Verifica-se, também, comparando os dois resultados mesmo o GWO apresentando menor de função objetivo, o resultado visual é bem similar ao obtido pelo BA. Isso, se deve ao fato, também, da diferença entre os valores das funções ser de 8,26% , aproximadamente.

#### 4.4.2 Estudo de Caso 2 - Faculdade de Artes e Design (UFJF)

Verificando o comportamento dos algoritmos para um outro conjunto de imagens da Faculdade de Artes realiza-se as mesmas comparações realizadas no caso 1. Dessa forma, os resultados das curvas de convergência dos métodos podem ser verificados na Figura 43, onde é mostrado a curva de convergência do GWO com uma ampliação (figura mais à direita) para melhor visualização da característica do método.

Figura 43 – Gráficos de Convergência em 10 simulações - Caso 2



Fonte: Elaborado pela autora (2021)

Para esse conjunto de soluções verifica-se que o *Bat Algorithm* apresentou-se melhor, novamente, na etapa de exploração de solução, ou seja, na busca global, enquanto o *Grey Wolf Optimizer* obteve na etapa de busca local seu diferencial. Em relação aos valores numéricos das funções objetivos, pode-se verificar na Tabela 6 os melhores resultados de função objetivo de cada método encontrados entre as 10 simulações e, também, o



valor comparado ao da solução original. O valor da função original é de  $1.1049600 \times 10^7$ , o encontrado pelo BA foi de  $1.0624300 \times 10^7$ , e o GWO obteve um valor de  $9.899800 \times 10^6$ .

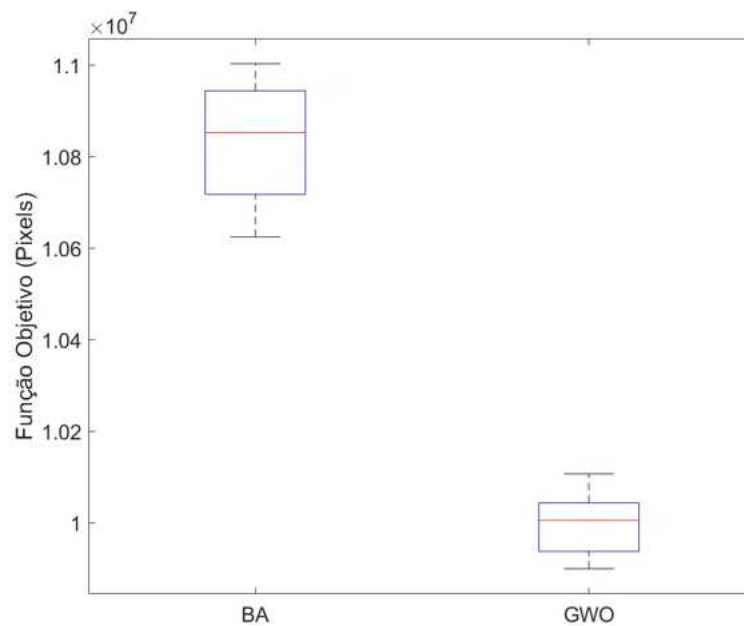
Tabela 6 – Comparação da Função Objetivo - Caso 2

	Valor
<b>Original</b>	$1.1049600 \times 10^7$
<b>BA</b>	$1.0624300 \times 10^7$
<b>GWO</b>	$9.899800 \times 10^6$

Fonte: Elaborada pela autora (2021).

Comparando os resultados pelo *boxplot* das soluções, Figura 44 e pela Tabela 7 com dados extraídos do *boxplot*, verifica-se que o algoritmo GWO possui menor mediana em comparação com o BA. E apresenta também uma menor dispersão para esse conjunto de dados.

Figura 44 – *Boxplot* dos Valores Ótimos da Função Objetivo nas 10 Simulações para o Caso 2



Fonte: Elaborado pela autora (2021)

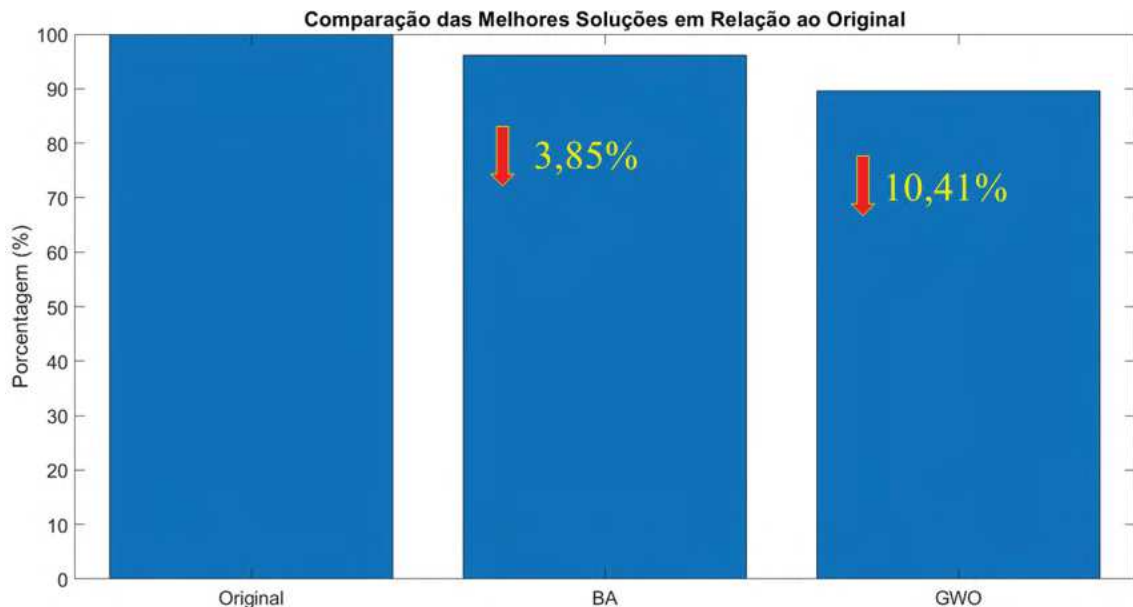
Tabela 7 – Comparação de Valores de Função Objetivo obtidos pelas Meta-heurísticas Extraídos do *Boxplot* para o Caso 2

Algoritmo	Mínimo	Máximo	Mediana
BA	$1.0624300 \times 10^7$	$1.1002500 \times 10^7$	$1.0851800 \times 10^7$
GWO	$9.899800 \times 10^6$	$1.010700 \times 10^7$	$1.0005650 \times 10^7$

Fonte: Elaborada pelo autor (2021).

Comparando as melhores soluções com a solução original, tem-se o gráfico apresentado pela Figura 45. Nele observa-se a diminuição dos valores das soluções em comparação com o original, ou seja, sem otimizar. O valor de relação encontrado pelo *Bat Algorithm* foi de 96,15%, correspondendo a 3,85% de diminuição, aproximadamente. Enquanto o algoritmo GWO apresentou uma relação de 89,59%, sendo em porcentagem, um valor de 10,41%. Comparando numericamente, para este estudo de caso, para o conjunto de imagens referentes a vista 2 da Faculdade de Artes e Design, o algoritmo GWO obteve menor valor de função objetivo. Comparando em porcentagem os resultados dos algoritmos, a diferença entre eles foi de aproximadamente 6,56% apenas.

Figura 45 – Comparação das Melhores soluções em Relação à Original - Caso 2



Fonte: Elaborado pela autora (2021)

Em termos de tempo computacional, a Tabela 8 apresenta, em segundos, o tempo médio de cada algoritmo de otimização levando em consideração as 10 simulações. Pela tabela verifica-se que ambos algoritmos possuem tempo computacional próximos, porém o BA é menor.

Tabela 8 – Tempo Médio Computacional para o Caso 2

Algoritmo	Tempo Médio (segundos)
BA	42,82
GWO	50,20

Fonte: Elaborada pela autora (2021).

Após realizada essas análises, verifica-se os resultados de imagem panorâmicas encontrados para esses valores, onde podem ser visualizados nas Figuras 46 e 47. As figuras apresentam os resultados originais e após o processo de otimização de cada algoritmo para melhor comparação.

Figura 46 – Faculdade de Artes e Design (UFJF) Vista 2 sem *Multi-band Blending - Bat Algorithm*



(a) Original



(b) BA

Figura 47 – Faculdade de Artes e Design (UFJF) Vista 2 sem *Multi-band Blending - Grey Wolf Optimizer*



(a) Original



(b) GWO

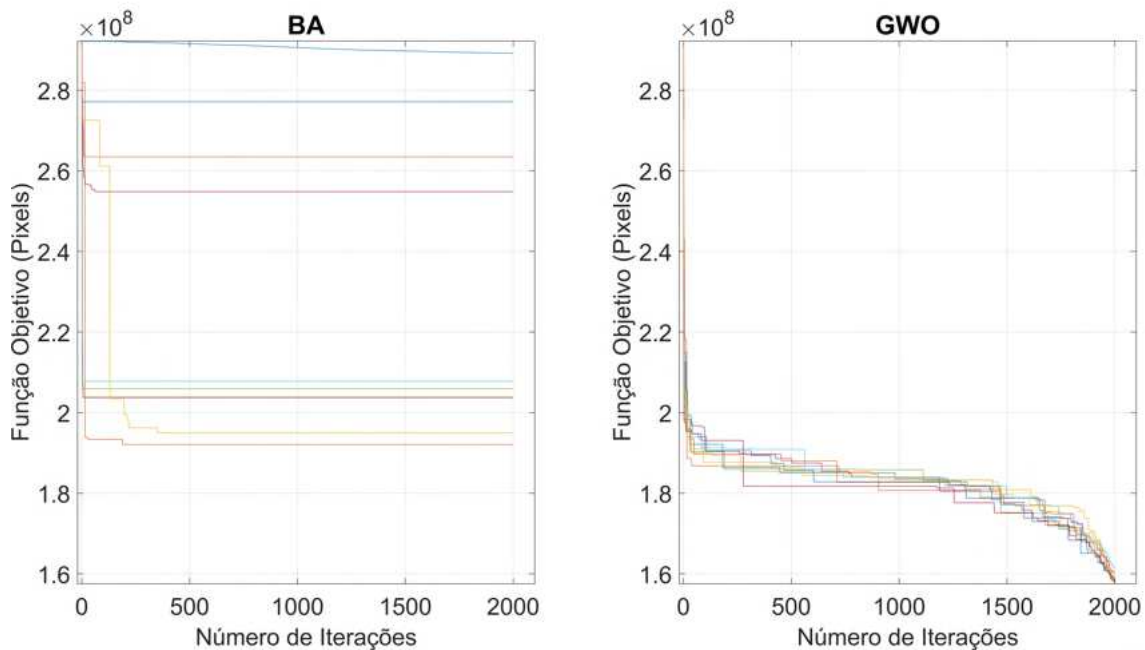
Comparando visualmente os resultados obtidos para esse conjunto, o BA possui um resultado melhor, apresentando pequenos desalinhamentos apenas mas com correção em diversas áreas da imagem. Se comparado ao original ainda apresenta alguns desalinhamentos. O mesmo ocorre para o GWO, onde apresentou melhorias em algumas regiões e desalinhamentos em outras. Como dito, no exemplo anterior, isso ocorre devido a particularidade do problema. A diminuição da função objetivo nos dois problemas

mostra que quantitativamente o algoritmo realizou sua finalidade, porém visualmente é complicado verificar essa melhoria. Eles podem estar melhorando áreas que a olho nu não consegue-se ver, e que para o algoritmo apresenta um erro maior que os visíveis. Nesse conjunto é perceptível esse comportamento, pois, em comparação com a Figura 47a original, percebe-se que houve uma melhora nos bueiros e calçadas e em alguns postes de energia.

#### 4.4.3 Estudo de Caso 3 - Faculdade de Artes e Design (UFJF)

O outro estudo foi realizado, também na Faculdade de Artes e Design, por um outro ponto de vista. Realizou-se os mesmos testes feitos anteriormente. Primeiramente, analisa-se as curvas de convergência dos algoritmos para esse conjunto de dados e como foram seus comportamentos durante as 10 simulações, conforme apresentado pela Figura 48.

Figura 48 – Gráficos de Convergência em 10 simulações - Caso 3



Fonte: Elaborado pela autora (2021)

Para esta análise ambas metodologias apresentaram-se melhores na etapa de busca global no início do processo iterativo, porém, na etapa de busca local o GWO apresentou-se melhor na busca de soluções. O algoritmo dos morcegos na etapa de busca local fica estagnado até o final das iterações não encontrando melhores resultados. A técnica, neste caso, apresenta uma dificuldade em convergir para uma solução mínima global, ficando estacionado em soluções mínimas locais.

Em seguida, verifica-se os valores numéricos da melhor solução encontrada durante as simulações, conforme a Tabela 9. O valor original de função objetivo é igual a  $2.92120000 \times 10^8$ , o valor encontrado pelo BA foi de  $1.92081000 \times 10^8$  e o valor do GWO foi de  $1.57520000 \times 10^8$ .

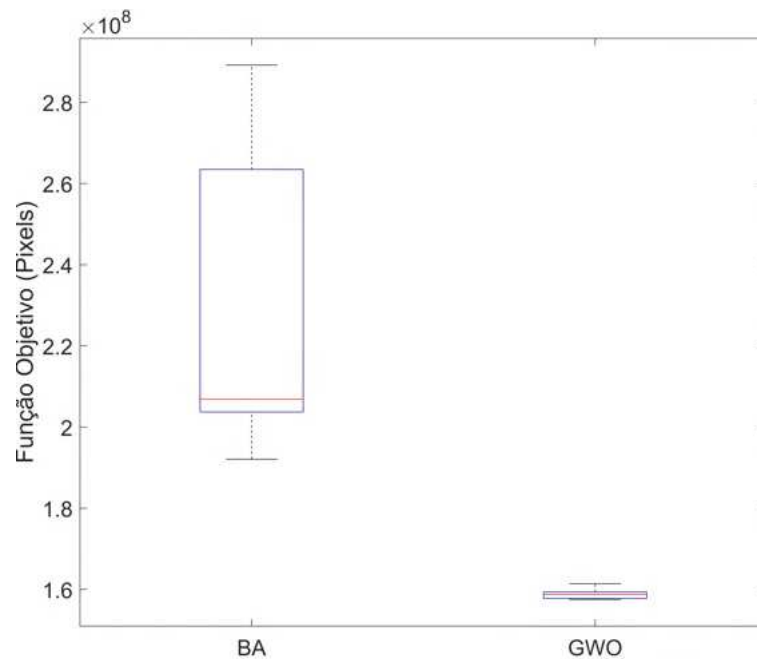
Tabela 9 – Comparação da Função Objetivo - Caso 3

	Valor
<b>Original</b>	$2.92120000 \times 10^8$
<b>BA</b>	$1.92081000 \times 10^8$
<b>GWO</b>	$1.57520000 \times 10^8$

Fonte: Elaborada pela autora (2021).

Por outro lado, comparando os resultados de melhor função objetivo entre as metodologias apresentadas é possível constatar a dispersão das soluções obtidas para o Caso 3, conforme o *boxplot* apresentado na Figura 49. Analisando o *boxplot* e a Tabela 10 verifica-se que o GWO apresenta menor dispersão dos dados em comparação com o BA, assim como menor mediana.

Figura 49 – *Boxplot* dos Valores Ótimos da Função Objetivo nas 10 Simulações para o Caso 3



Fonte: Elaborado pela autora (2021)

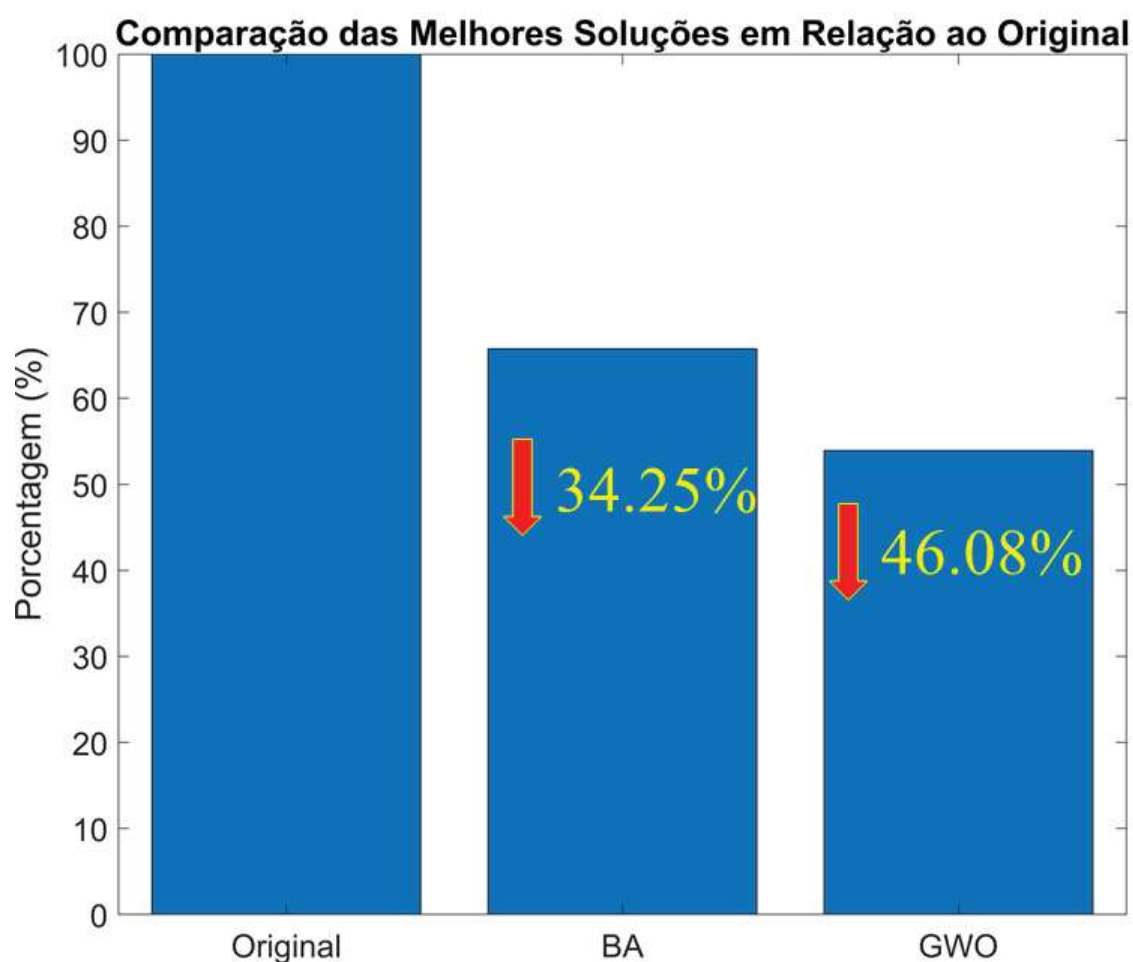
Tabela 10 – Comparação de Valores de Função Objetivo obtidos pelas Meta-heurísticas Extraídos do *Boxplot* para o Caso 3

Algoritmo	Mínimo	Máximo	Mediana
BA	$1.92081000 \times 10^8$	$2.89188000 \times 10^8$	$2.06852500 \times 10^8$
GWO	$1.57520000 \times 10^8$	$1.61412000 \times 10^8$	$1.58797500 \times 10^8$

Fonte: Elaborada pelo autor (2021).

Comparando as melhores soluções obtidas entre as 10 simulações em relação a solução original, tem-se o gráfico apresentado pela Figura 50.

Figura 50 – Comparação das Melhores soluções em Relação à Original - Caso 3



Fonte: Elaborado pela autora (2021)

A relação dada pelo valor do GWO comparado ao original foi de 65,75%, correspondendo a uma diminuição de aproximadamente 34,25%. O BA obteve relação de 53,92% sendo, então, sua diminuição em relação ao original de aproximadamente 46,08%. Logo,

por essas análises realizadas pode-se concluir que o algoritmo do GWO obteve melhor solução encontrada, devido a sua diminuição em relação ao original ser maior que o valor do BA. Dessa maneira, o GWO obteve melhor resultado numericamente. O tempo médio de execução, em segundos, de cada algoritmo é apresentado na Tabela 11. Pela tabela verifica-se que ambos algoritmos possuem tempo computacional próximos, porém o BA é ligeiramente menor.

Tabela 11 – Tempo Médio Computacional para o Caso 3

Algoritmo	Tempo Médio (segundos)
BA	43,92
GWO	46,68

Fonte: Elaborada pela autora (2021).

Aplicando as soluções encontradas nas imagens, tem-se os resultados apresentados nas Figuras 51 e 52. Analisando a panorâmica formada, verifica-se que a Figura 51 apresentou melhor resultado visualmente, mesmo analisando numericamente tendo tido pior valor de função objetivo. É perceptível muitas regiões desalinhadas. Essa característica ocorre em função da formulação da função objetivo e de como os algoritmos agem sobre ela. E, também, pela características das *features* escolhidas, isso pode ter prejudicado o resultado final.

Figura 51 – Faculdade de Artes e Design (UFJF) Vista 3 sem *Multi-band Blending - Bat Algorithm*

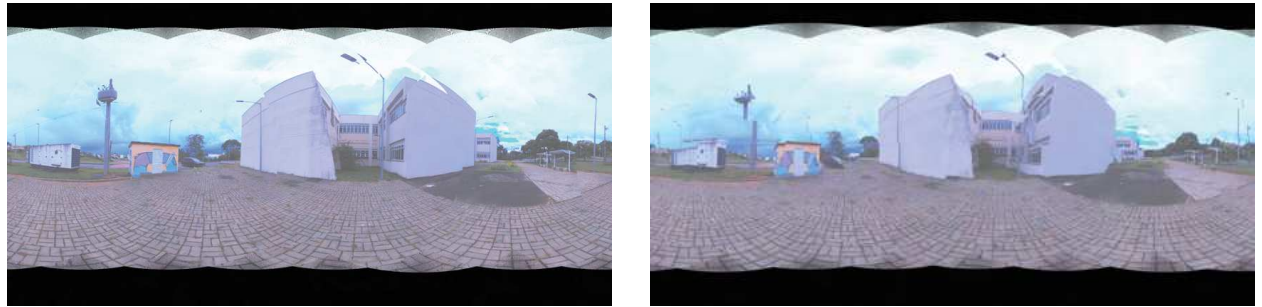


(a) Original



(b) BA

Figura 52 – Faculdade de Artes e Design (UFJF) Vista 3 sem *Multi-band Blending-Grey Wolf Optimizer*



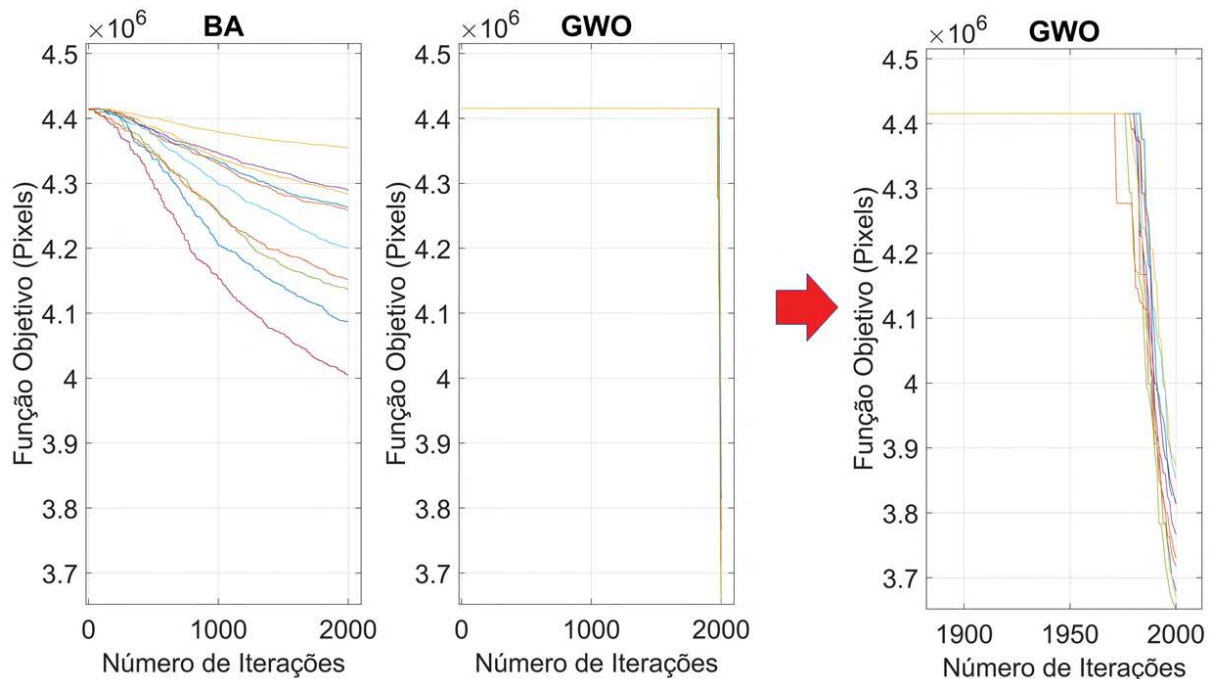
(a) Original

(b) GWO

#### 4.4.4 Estudo de Caso 4 - Subestação Santos Dumont

O último resultado analisado foi para o conjunto de dados referentes a uma subestação de energia da cidade de Santos Dumont. Os mesmos testes realizados para os conjuntos da Faculdade de Artes e Design foram repetidos para essas novas imagens. As análises de convergência podem ser conferidas na Figura 53. Percebe-se que, para esse conjunto de dados, o BA apresentou melhor etapa de exploração na região de exploração (busca global), enquanto o GWO obteve melhor resposta na etapa de busca local.

Figura 53 – Gráficos de Convergência em 10 simulações - Caso 4



Fonte: Elaborado pela autora (2021)



Realizou-se, então, a comparação numérica dos valores obtidos pelos métodos de otimização comparando com a solução original. A Tabela 12 apresenta os valores da função objetivo das melhores soluções. A solução original apresentou valor igual a  $4.415420 \times 10^6$ , o valor encontrado pelo BA foi de  $4.005020 \times 10^6$  e o valor do GWO foi  $3.651580 \times 10^6$ .

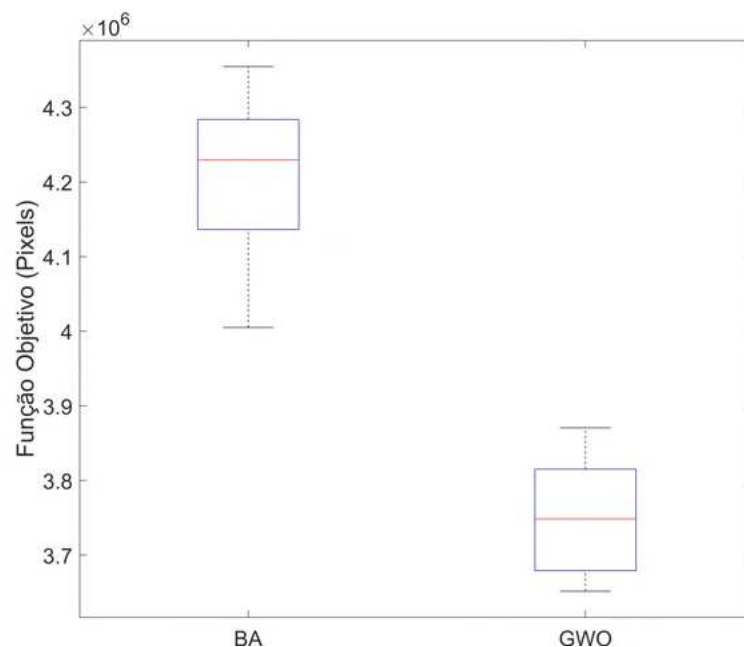
Tabela 12 – Comparação da Função Objetivo - Caso 4

	Valor
<b>Original</b>	$4.415420 \times 10^6$
<b>BA</b>	$4.005020 \times 10^6$
<b>GWO</b>	$3.651580 \times 10^6$

Fonte: Elaborada pela autora (2021).

Verificando os resultados obtidos pelas melhores soluções encontradas durante as 10 simulações dos algoritmos, criou-se um *boxplot* para verificar a dispersão das soluções para cada metodologia, conforme Figura 54 e Tabela 13. Percebe-se, então, que o algoritmo GWO apresentou menor mediana e menor dispersão das soluções em comparação com BA.

Figura 54 – *Boxplot* dos Valores Ótimos da Função Objetivo nas 10 Simulações para o Caso 4



Fonte: Elaborado pela autora (2021)

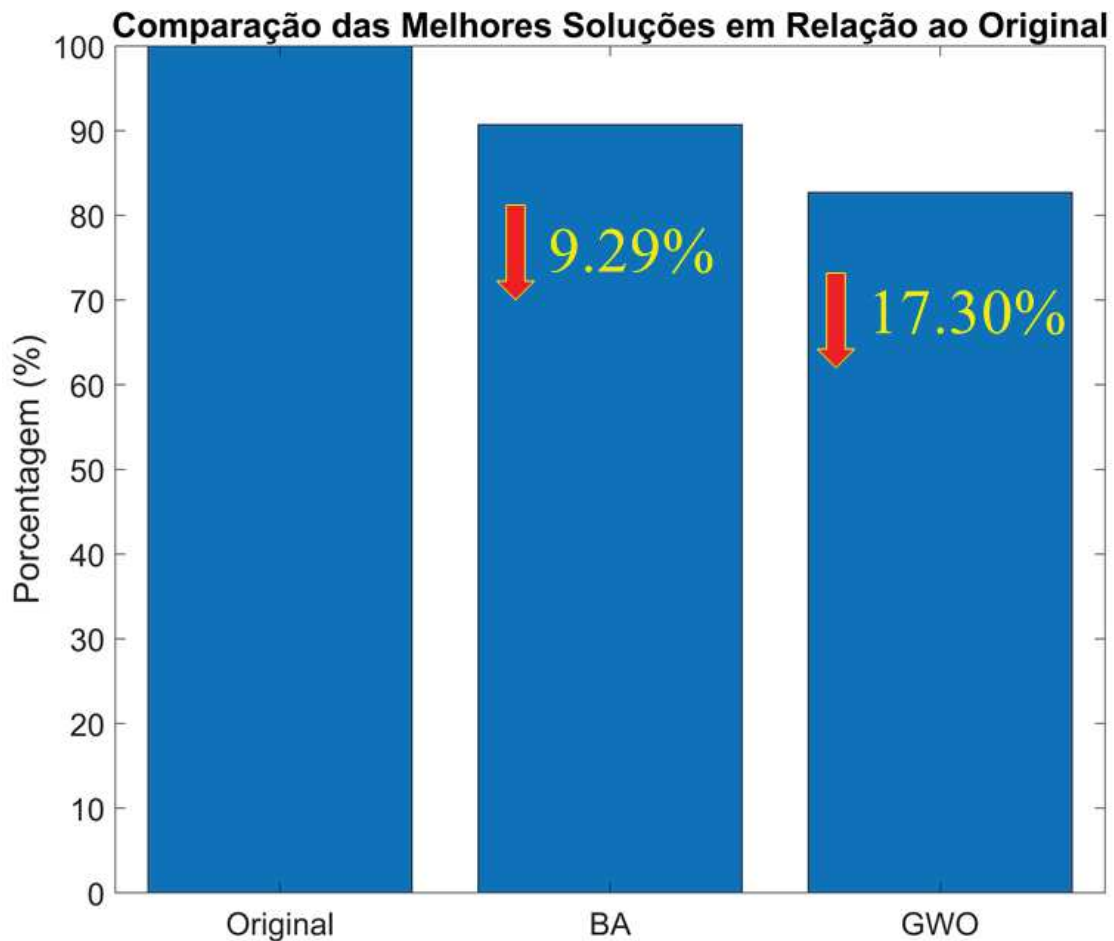
Tabela 13 – Comparação de Valores de Função Objetivo obtidos pelas Meta-heurísticas Extraídos do *Boxplot* para o Caso 4

Algoritmo	Mínimo	Máximo	Mediana
BA	$4.005020 \times 10^6$	$4.354630 \times 10^6$	$4.229450 \times 10^6$
GWO	$3.651580 \times 10^6$	$3.870550 \times 10^6$	$3.748570 \times 10^6$

Fonte: Elaborada pelo autor (2021).

A porcentagem de diminuição dos erros em relação ao valor original podem ser conferidos na Figura 55.

Figura 55 – Comparação das Melhores soluções em Relação à Original - Caso 4



Fonte: Elaborado pela autora (2021)

O BA encontrou uma relação de 90,71%, correspondendo a 9,29% de diminuição do valor da função objetivo; o GWO apresentou uma relação de 82,70%, resultando em uma porcentagem de diminuição de 17,30%, aproximadamente. Dessa forma, analisando os valores numéricos da função objetivo encontrada por cada método, conclui-se que o GWO apresentou melhor resposta visto que apresenta menor valor de função objetivo.

O tempo médio de execução, em segundos, de cada algoritmo é apresentado na Tabela 14. Pela tabela verifica-se que ambos algoritmos possuem tempo computacional próximos, porém o BA é ligeiramente menor.

Tabela 14 – Tempo Médio Computacional para o Caso 4

Algoritmo	Tempo Médio (segundos)
BA	41,97
GWO	46.45

Fonte: Elaborada pela autora (2021).

Aplicando tais valores nas imagens, tem-se as respostas ilustradas nas Figuras 57 e 56. Comparando os resultados das imagens percebe-se, que ambos os métodos apresentaram correções nas imagens em comparação ao original. Visualmente não se pode afirmar qual metodologia apresenta melhor resultado pois ambas corrigiram os desalinhamentos em algumas etapas e desalinham em outras. O *Bat Algorithm*, por exemplo, apresentou uma melhora nas calçadas mas houveram alguns desalinhamentos nas linhas de transmissão.

Figura 56 – Subestação de Energia sem *Multi-band Blending* - *Bat Algorithm*



(a) Original



(b) BA

Figura 57 – Subestação de Energia sem *Multi-band Blending* - *Grey Wolf Optimizer*



(a) Original



(b) GWO

#### 4.5 RESULTADOS OBTIDOS PELO *MULTI-BAND BLENDING*

Nesta seção serão apresentados os resultados obtidos pelo método de suavização nas imagens. Com tal metodologia espera-se que possa haver a correção de diferença de exposição nas imagens e, também em alguns desalinhamentos que ainda existiram.

Os resultados obtidos para Faculdade de Artes e Design (UFJF) , podem ser conferidos das Figuras 58 a 63. Aplicou-se a técnica para as imagens da subestação e estão representadas pelas Figuras 64 e 65.

Figura 58 – Faculdade de Artes e Design (UFJF) Vista 1 com *Multi-band Blending - Bat Algorithm*



Fonte: Elaborado pela autora (2021)

Figura 59 – Faculdade de Artes e Design (UFJF) Vista 1 com *Multi-band Blending - Grey Wolf Optimizer*



Fonte: Elaborado pela autora (2021)

Figura 60 – Faculdade de Artes e Design (UFJF) Vista 2 com  
*Multi-band Blending - Bat Algorithm*



Fonte: Elaborado pela autora (2021)

Figura 61 – Faculdade de Artes e Design (UFJF) Vista 2 com  
*Multi-band Blending - Grey Wolf Optimizer*



Fonte: Elaborado pela autora (2021)

Figura 62 – Faculdade de Artes e Design (UFJF) Vista 3 com *Multi-band Blending - Bat Algorithm*



Fonte: Elaborado pela autora (2021)

Figura 63 – Faculdade de Artes e Design (UFJF) Vista 3 com *Multi-band Blending - Grey Wolf Optimizer*



Fonte: Elaborado pela autora (2021)

Figura 64 – Subestação de Santos Dumont com *Multi-band Blending - Bat Algorithm*



Fonte: Elaborado pela autora (2021)

Figura 65 – Subestação de Santos Dumont com *Multi-band Blending - Grey Wolf Optimizer*



Fonte: Elaborado pela autora (2021)

Verifica-se que as linhas nos limites das áreas de sobreposição entre as imagens não existem mais, tornando as colorações mais uniformes. Em relação aos desalinhamentos houve alguma suavização tornando o panorama final com aspecto mais agradável.

Finalmente, pode-se concluir que o esquema de mesclagem de várias bandas efetivamente esconde as costuras, apesar das mudanças de iluminação (flash da câmera e mudança na abertura e exposição). Os quatro conjuntos testados apresentavam diferenças nas exposições e verificou-se a melhora na suavização das imagens finais, se comparadas com as originais apresentadas na Seção 4.3.

## 4.6 ANÁLISE DOS RESULTADOS

Com os resultados apresentados anteriormente pode-se concluir que os métodos de otimização utilizados para o problema apresentaram soluções que numericamente são melhores das já existentes, ou seja, encontraram soluções melhores das que se tem obtidas durante a aquisição das imagens pelo robô.

Entretanto, pelos resultados visuais observou-se que o problema em questão apresenta certas peculiaridades quanto as soluções, pois visualmente os algoritmos não otimizam a região como um todo, apresentando melhorias em algumas partes e, até piora em outras. Isso, se dá pela característica de que a função objetivo é composta pela soma ao quadrado dos resíduos do erro de re-projeção. Sendo composta por muitas parcelas, o que pode estar ocasionando em que os algoritmos em, alguns casos, melhore muito uma parcela e outra nem tanto, ou seja, acabam minimizando uma solução que não era necessária ser otimizada naquele ponto.

Outro aspecto que vale ser ressaltado é que o problema apresenta muitas variáveis, então pode ser que ele esteja caindo em um mínimo local, o que ocasiona que algumas variáveis possam influenciar mais no resultado. Isso, deve-se ao fato também que a formulação da função objetivo não está feita de forma a priorizar as características que se deseja minimizar mais, ou seja, as características onde estão os desalinhamentos nas imagens. Outros testes foram realizados aumentando-se o número de iterações assim como quantidade de indivíduos e, conclui-se que não houve melhoria significativa nos resultados.

As melhores soluções obtidas em cada estudo de caso podem ser conferidas no Apêndice 5.2.



## 5 CONCLUSÃO E TRABALHOS FUTUROS

Neste capítulo, serão apresentadas as principais conclusões acerca do trabalho desenvolvido. Além disso, também serão expostas algumas sugestões de trabalhos futuros que possam melhorar os resultados obtidos e de outras aplicações.

### 5.1 CONCLUSÃO

Nesta dissertação foi proposta a utilização de meta-heurísticas bioinspiradas para tentar solucionar o problema de alinhamento de posicionamento de imagens para formar imagens panorâmicas em 360°.

Neste trabalho, foi realizada uma revisão sobre as metodologias conhecidas para construção de imagens panorâmicas, demonstrando a sua relevância encontrando aplicações em diversas áreas, algumas delas sendo de importante valor social e econômico, como medicina e indústrias. A área de pesquisa que envolve o mosaico de imagens permanece com estudos atuais, indicando a pertinência do tema, é ampla e possui diversos métodos que podem ser utilizados para as mais variadas aplicações. Assim, são necessárias avaliações e comparações entre métodos para uma escolha apropriada à uma finalidade específica.

Com o objetivo de melhorar as técnicas já existentes, uma nova metodologia para montagem de panorâmicas 360° foi proposta nessa dissertação. Este trabalho apresentou uma metodologia para construir mosaicos automáticos de imagens de ambientes, a partir de conjuntos de imagens obtidas na Universidade Federal de Juiz de Fora obtidas por meio de uma câmera de alta resolução e um laser LIVOX MID-40 fixados em tripé, formando um sistema robótico para tal finalidade.

Ao longo do trabalho apresentou-se as técnicas detalhadas para obtenção da imagem final a partir de múltiplas imagens. Dados conjuntos de imagens, a implementação do método proposto começa com a extração das features aplicando o SIFT como algoritmo de detecção, e para eliminar os outliers existentes a combinação do método RANSAC e o teste de compatibilidade das correspondências que obtiveram um resultado satisfatório para remover as falsas correspondências. Em seguida, é apresentada a metodologia de *Bundle Adjustment* com a finalidade de se obter o ajuste ótimo imagens, visto que os valores encontrados durante a aquisição apresentaram pequenos desalinhamentos no panorama final. Para isso, foram utilizados duas técnicas de otimização baseadas em algoritmos bioinspirados: *Grey Wolf Optimizer* e *Bat Algorithm*. Ao se comparar esses algoritmos avaliando a função objetivo e curva de convergência ao final das simulações percebe-se que os métodos aplicados apresentam suas particularidades para o problema proposto. Para cada conjunto de solução obteve-se melhoria numérica quanto ao valor da função objetivo, mas como apresentado, nem sempre foi possível ser visto quanto a formação das imagens. Em termos de tempo de execução de cada método não apresenta um alto custo

computacional, executando cada algoritmo em torno de 50 segundos.

O método de suavização das imagens panorâmicas se apresentou como uma boa alternativa para a montagem dos panoramas, deixando as imagens finais com aspectos uniforme e até mesmo corrigindo alguns desalinhamentos ainda presentes.

## 5.2 TRABALHOS FUTUROS

Trabalhos futuros são vislumbrados frente aos resultados apresentados, principalmente no que se diz respeito à qualidade final das imagens panorâmicas em relação a otimização da posição das câmeras de cada imagem para formar a imagem final.

Modificações na função objetivo devem ser realizadas com o intuito de melhorar a assertividade dos resultados. Para isso, pode-se ponderar as parcelas do somatório dos erros de acordo com as *features* dos *matches*, assim os resultados de otimização poderão ser melhores verificados visualmente nas projeções das imagens. O objetivo será encontrar as características que deseja se priorizar e tratá-la de forma diferente dentro da função objetivo, aplicando na mesma uma penalidade. O interesse é realizar a otimização na região de encaixe entre as imagens, dessa forma, pode-se selecionar as correspondências relacionadas as bordas das imagens com propósito de atribuir pesos nesses pontos na função objetivo que são de maior interesse para o problema.

Considera-se interessante, também, a modelagem de um algoritmo híbrido entre o *Bat Algorithm* e *Grey Wolf Optimizer* visando agregar as principais vantagens de cada técnica. Além disso, investiga-se o uso de meta-heurísticas recentes como: *Salp Swarm Algorithm* (SSA) e *Whale Optimization Algorithm* (WOA).

## REFERÊNCIAS

- 1 RAMAMURTHY, M.; LAKSHMINARAYANAN, V.;KARLICEK, R.; SUN, C.; ZISSIS, G.; M, R. Human vision and perception. **Springer International Publishing**, Switzerland, v. 10, p. 978–3, 2015.
- 2 SOLEM, J. E. **Programming Computer Vision with Python: Tools and algorithms for analyzing images**. O’Reilly Media, Inc., 2012.
- 3 PATRÍ, D. I; RIEDER, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. **Computers and electronics in agriculture**, Elsevier, v. 153, p. 69–81, 2018.
- 4 CAVRÉ, T.; CAIROL, M. T.; CALAFELL, D. F.; RIBES, M. T.; ROCA, J. P. Project-based learning example: controlling an educational robotic arm with computer vision. **IEEE Revista Iberoamericana de Tecnologías del Aprendizaje**, IEEE, v. 8, n. 3, p. 135–142, 2013.
- 5 CRIMINISI, A.; SHOTTON, J. **Decision forests for computer vision and medical image analysis**. Springer Science & Business Media. 2013.
- 6 AYACHE, N. Medical computer vision, virtual reality and robotics. **PImage and vision computing**. Elsevier. v. 13, n.4, p. 295–313, 1995.
- 7 BERGASA, L. M.; NUEVO, J.; SOTELO, M. A.; BAREA, R.; LOPEZ, M. E. Real-time system for monitoring driver vigilance. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v.7, n. 1, p. 63–77, 2006.
- 8 BROSNAN, T.; SUN, D. Improving quality inspection of food products by computer vision -a review. **Journal of food engineering**, Elsevier, v. 61, n. 1, p. 3–16, 2004.
- 9 SPENCER, B. F.; HORKERE, V.; NARAZAKI, Y. Advances in computer vision-based civil infrastructure inspection and monitoring. **Engineering**, Elsevier, v. 4, n. 2, p. 199–222, 2019.
- 10 QUINTANA, M.; TORRES, J.; MENÉNDEZ, J. M. A simplified computer vision system for road surface inspection and maintenance. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 17, n. 3, p. 608–619, 2015.
- 11 DUAN, Y.; HAN,C.; TAO, X.; GENG, B.; DU, Y.; LU, J. Panoramic Image Generation: From 2-D Sketch to Spherical Image. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v.14, n. 1, p. 194–208, 2020.
- 12 PELEG, S.; HERMAN, J. Panoramic mosaics by manifold projection. **Proceedings of IEEE computer society conference on computer vision and pattern recognition**, IEEE, p. 338–343, 1997.
- 13 YOGAMANI, S.; HUGHES, C.; HORGAN, J.; SISTU, G.; VARLEY, P.; O’DEA, D.; URICÁR, M.; MILZ, S.; SIMON, M.; AMEND, K.; ET AL. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. **Proceedings of the IEEE/CVF International Conference on Computer Vision**, p. 9308–9318, 2019.

- 14 LEE, M.; KIM, H.; PAIK J. Correction of barrel distortion in fisheye lens images using image-based estimation of distortion parameters. **IEEE ACCESS**, IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC, v. 7, p. 45723–45733, 2019
- 15 YAO, W.; LI, Z. Instant color matching for mobile panorama imaging. **IEEE Signal Processing Letters**, IEEE, v.22, n. 1, p. 6–10, 2014.
- 16 CHA, J H.; JEON, Y-S.; MOON, Y-S.; LEE SHI-H. Seamless and fast panoramic image stitching. **2012 IEEE International Conference on Consumer Electronics (ICCE)**, IEEE, p. 29–30, 2012.
- 17 XIONG, Y.; PULLI, K. Sequential image stitching for mobile panoramas. **2009 7th International Conference on Information, Communications and Signal Processing (ICICS)**, IEEE, p. 1–5, 2009.
- 18 YE, G. Image registration and super-resolution mosaicing, 2005.
- 19 HIRAGA, A. K.; DA SILVA, F. A.; ARTERO, A. O. Algoritmos para construção de panorama de imagens 360 e visualização. **Colloquium Exactarum. ISSN: 2178-8332**, Elsevier, v. 5, n.1, p. 12–25, 2013.
- 20 JAHANSHAHI, M. R.; MARSRI, S. F.; SUKHATME, G. S. Multi-image stitching and scene reconstruction for evaluating defect evolution in structures. **Structural Health Monitoring**, Sage Publications Sage UK: London, England, v. 10, n. 6, p. 643–657, 2011.
- 21 ZHU, Z.; FU, J; YANG, J; ZHANG, X. O. Panoramic image stitching for arbitrarily shaped tunnel lining inspection. **Computer-Aided Civil and Infrastructure Engineering**, Wiley Online Library, v. 31, n.12, p. 936–953, 2016.
- 22 LI, D.; HAN, D.; ZHANG, X.; ZHANG, L. Panoramic image mosaic technology based on SIFT algorithm in power monitoring. **2019 6th International Conference on Systems and Informatics (ICSAI)**, IEEE, p. 1329–1333, 2019.
- 23 CHEN, S. E. Quicktime VR: An image-based approach to virtual environment navigation. **Proceedings of the 22nd annual conference on Computer graphics and interactive techniques**, p. 29–38, 1995.
- 24 PALUMBO, J. Grappling with Virtual Reality. *edublog.pdnonline.com*, 2017. [Online]. Acessível: <https://edublog.pdnonline.com/2017/07/grappling-with-virtual-reality.html>, [Acessado: 03 de março 2021] .
- 25 SZELISKI, R. Image alignment and stitching: A tutorial. **Foundations and Trends® in Computer Graphics and Vision**, Now Publishers Inc. Hanover, MA, USA, v. 2, n.1, p. 1–104, 2006.
- 26 MAINTZ, J. B. A.; VIERGER, M. A. A survey of medical image registration. **Medical image analysis**, Elsevier, v.2, n.1, p. 1–36, 198.
- 27 ZITOVA, B.; FLUSSER, J. Image registration methods: a survey. **Image and vision computing**, Elsevier, v. 21, n.11, p. 9–1000, 2003.

- 28 MILLS, A.; DUDEK, G. Image stitching with dynamic elements. **Image and Vision Computing**, Elsevier, v. 27, n.10, p. 1593–1602, 2009
- 29 GHOSH, D.; KAABOUCHE, N. A survey on image mosaicing techniques. **Journal of Visual Communication and Image Representation**, Elsevier, v. 34, p. 1–11, 2016.
- 30 CHEN, C.; KLETTE, R. Image stitching—Comparisons and new techniques. **International conference on computer analysis of images and patterns**, Springer, p. 615–622, 1999.
- 31 XIONG, Y.; PULLI, K. Image alignment and stitching: A tutorial. **IEEE Transactions on Consumer Electronics**, IEEE, v. 56, n.2, p. 298–306, 2010.
- 32 CHEN, J.; XU, Q.; LUO, L.; WANG, Y.; WANG, S. A robust method for automatic panoramic UAV image mosaic. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n.8, p. 1898, 2019.
- 33 ZARAGOZA, J.; CHIN, T.; BROWN, M. S.; SUTER, D. As-projective-as-possible image stitching with moving DLT. **Proceedings of the IEEE conference on computer vision and pattern recognition**, p. 2339–2346, 2013.
- 34 AGARWALA, A.; DONTCHEVA, M.; AGRAWALA, M.; DRUCKER, S.; COLBURN, A.; CURLESS, B.; SALESIN, D.; COHEN, M. Interactive digital photomontage. **ACM SIGGRAPH 2004 Papers**, p. 294–302, 2004.
- 35 EDEN, A.; UYTENDAELE, M.; SZELISKI, R. Seamless image stitching of scenes with large motions and exposure differences. **2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)**, IEEE, v. 2, p. 2498–2505, 2006.
- 36 BROWN, M.; LOWE, D. G. Automatic panoramic image stitching using invariant features. **International journal of computer vision**, Springer, v. 74, n. 1, p. 59–73, 2007.
- 37 Pérez, P.; GANGNET, M.; BLAKE, A. Poisson image editing. **ACM SIGGRAPH 2003 Papers**, p. 313–318, 2008.
- 38 TRIGGS, B.; MCLAUCHLAN, P. F.; HARTLEY, R. I.; FITZGIBBON, A. W. Bundle adjustment—a modern synthesis. **International workshop on vision algorithms**, p. 298–372, 1999.
- 39 NI, K.; STEEDLY, D.; DELLAERT, F. Out-of-core bundle adjustment for large-scale 3d reconstruction. **2007 IEEE 11th International Conference on Computer Vision**, IEEE, p. 1–8, 2007.
- 40 MCLAUCHLAN, P. F.; JAENICKE, A. Image mosaicing using sequential bundle adjustment. **Image and Vision computing**, Elsevier, v. 20, n. 9–10, p. 751–759, 2002.
- 41 MIN, Zhong, Jiguo, ZENG and Xusheng, XIE. Panorama stitching based on sift algorithm and levenberg-marquardt optimization. **Physics Procedia**, Elsevier, v. 33, p. 811–818, 2012.

- 42 BROWN, M.; LOWE, D. G.; others. Recognising panoramas. **ICCV**, v. 3, p.1218, 2003.
- 43 LOURAKIS, M. IA. A brief description of the Levenberg-Marquardt algorithm implemented by levmar. **Foundation of Research and Technology**, v. 4, p. 1–6, 2005.
- 44 ALLAWAI, Ziyad T, IBRAHEEN, Ibraheem Kasim , HUMAIDI, Amjad J. Fine-tuning meta-heuristic algorithm for global optimization. **Processes**, Multidisciplinary Digital Publishing Institute, v. 7, n. 10, p. 657, 2019.
- 45 ZHANG, X.; XU, S.; Research on Image Processing Technology of Computer Vision Algorithm. **2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)**, p. 122–124, 2020.
- 46 KAPUR, S. Computer Vision with Python 3. Packt Publishing Ltd, v. 26, 2017.
- 47 MA, Y.; SOATTO, S.; KOSECKA, J.; SASTRY, S. S. An invitation to 3-d vision: from images to geometric models. Springer Science & Business Media, v. 26, 2012.
- 48 MCANDREW, A. An introduction to digital image processing with matlab notes for scm2511 image processing. **School of Computer Science and Mathematics, Victoria University of Technology**. v. 264, n.1, p.1-264, 2004.
- 49 HARTLEY, R.; ZISSERMAN, A. Multiple view Geometry in computer vision. **Cambridge Universit Press**, 2004.
- 50 SZELISKI, R. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.
- 51 LUONG, Q.; FAUGERAS, O. D. The fundamental matrix: Theory, algorithms, and stability analysis. **International journal of computer vision**. Springer, v. 17, n.1, p.43-75, 1996.
- 52 CRIADO, R.; ROMANCE, M.; SANCHEZ, A. Interest point detection in images using complex network analysis. **Journal of Computational and Applied Mathematics**, Elsevier, v. 236, n. 12, p. 2975–2980, 2012
- 53 GOVENDER, N. **Evaluation of feature detection algorithms for structure from motion**, 2009.
- 54 LISIN, D. A.;MATTAR, M.;BLASCHKO, M. B.;LEARNED-MILLER, E. G.;BENFIELD, M. C. Combining local and global image features for object class recognition. **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops**, IEEE, p. 47–47, 2005.
- 55 TUYTELAARS, T.; MIKOLAJCZYK, K. **Local invariant feature detectors: a survey**. Now Publishers, Inc., 2008.
- 56 BUENO, L. M. **Análise dos descritores locais de imagens no contexto de detecção de semi-réplicas**. 2011.
- 57 HARRIS, C. G.; STEPHENS, M.; et al. A combined corner and edge detector. **Alvey vision conference**, Citeseer, v. 15, n. 50, p. 10–5244, 1988.

- 58 OPENCV. **Shi-Tomasi Corner Detector Good Features to Track**. Disponível em: <http://www.bdt.org.br/sma/entendendo/atual.htm>. Acesso em: 8 mar. 1999.
- 59 LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, Springer, v. 60, n. 2, p. 91–110, 2004.
- 60 SRISVASTAVA, R. **Research developments in computer vision and image processing: Methodologies and applications: Methodologies and applications**. IGI Global, 2013.
- 61 LOWE, D. G. Object recognition from local scale-invariant features. **Proceedings of the seventh IEEE international conference on computer vision**, IEEE, v. 2, p. 1150–1157, 1999.
- 62 BAY, H.; ESS, A.; TUYTELAARS, T.; VAN GOOL, L. Speeded-up robust features (SURF). **Computer vision and image understanding**, Elsevier, v. 110, p. 346–359, 2008.
- 63 RUBLEE, E.; RUBAUD, V.; KONOLIGE, K.; BRADSKI, G. ORB: An efficient alternative to SIFT or SURF. **2011 International conference on computer vision**, Ieee, p. 2564–2571, 2011.
- 64 ALCANTARILLA, P. F.; SOLUTIONS, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces. **IEEE Trans. Patt. Anal. Mach. Intell.**, TrueVision Solutions, v.34, n. 7, p. 1281–1298, 2011.
- 65 KABBAL, L.; AZAZA, A.; ABDELLAOUI, M.; DOUIK, A. Image matching based on lbp and sift descriptor. **2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15)**, IEEE, p. 1–6, 2015.
- 66 MOREL, J.; YU, G. Is SIFT scale invariant?. **Inverse Problems and Imaging**, v. 5, n. 1, p. 115–136, 2011.
- 67 MIKOLAJCZYK, K.; SCHIMID, C. A performance evaluation of local descriptors. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v.27, n.10, p. 1615–1630, 2005.
- 68 BEIS, J. S.; LOWE, D. G. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. **Proceedings of IEEE computer society conference on computer vision and pattern recognition**, IEEE, p. 1000–1006, 1997.
- 69 KITT, B.; GEIGER, A.; LATEGAHN, H. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. **2010 IEEE intelligent vehicles symposium**, IEEE, p. 486–492, 2010.
- 70 ZHOU, Q.; PARK, J.; KOLTUN, V. Fast global registration. **European Conference on Computer Vision**, Springer, p. 766–782, 2016.
- 71 FAUGERAS, O.; LUONG, Q.; PAPADOPOULOS, T. **The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications**. MIT press, 2001.

- 72 DUBRODSKY, E. Homography estimation. **Diplomová práce. Vancouver: Univerzita Britské Kolumbie**, v. 5, 2009.
- 73 ZHANG, Z. A flexible new technique for camera calibration. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v.22, n.11, p.1330–1334, 2000.
- 74 ZHANG, Z.; HANSON, A. R. 3D reconstruction based on homography mapping. **Proc. ARPA96**, p.1007–1012, 1996.
- 75 shum, H-Y., SZELISKI, R. Construction of panoramic image mosaics with global and local alignment. **Panoramic vision**, Springer, p.227–268, 2000.
- 76 GAO, J.; KIM, S. J; BROWN, M. S. Constructing image panoramas using dual-homography warping. **CVPR 2011,IEEE** p.49–56, 2011.
- 77 KUKELOVA, Z.; HELLER, J.; BUJNAK, M.; PADJLA, T. Radial distortion homography. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, p.639–647, 2015.
- 78 FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, v. 24, n. 6, p.381–395, 1981.
- 79 FÖRSTNER, W.; WROBEL, B. P. **Photogrammetric computer vision**, 2016.
- 80 LOURAKIS, M.; ARGYROS, A. A. Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment?. **Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1**, IEEE, v.2, p.1526–1531, 2005.
- 81 HEDBORG, J.; FORSSÉN, P.; FELSBURG, M.; RINGABY, E. Rolling shutter bundle adjustment. **2012 IEEE Conference on Computer Vision and Pattern Recognition,IEEE**, p.1434–1441, 2012.
- 82 ZHENG, M.; ZHOU, S.; XIONG, X.; ZHU, J. A new GPU bundle adjustment method for large-scale data. **Photogrammetric Engineering & Remote Sensing**, American Society for Photogrammetry and Remote Sensing, v.83, n.9, p.633–641, 2017.
- 83 LUHMANN, T. Panorama photogrammetry for architectural applications. **Mapping, eGeoMapping**, v. 139, p. 40–45, 2010.
- 84 FANGI, G. The Multi-image spherical Panoramas as a tool for Architectural Survey. **CIPA Heritage Documentation**, Citeseer, v. 21, p. 311–316, 2011.
- 85 FANGI, G. Further developments of the spherical photogrammetry for cultural heritage. **XXII Cipa Symposium, Kyoto** , p. 11–15, 2009.
- 86 FANGI, G. Multiscale multiresolution spherical photogrammetry with long focal lenses for architectural surveys. **International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences** v. 38 , p. 1–6, 2010.



- 87 CHEW, V. CS.; LIAN, F. Panorama stitching using overlap area weighted image plane projection and dynamic programming for visual localization. **2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)** ,IEEE, p. 250–255, 2012.
- 88 SZELISKI, M. Video mosaics for virtual environments. **IEEE Computer Graphics and Applications** , v. 16, n. 2, p. 22–30, 1996.
- 89 KOPF, J.; LISCHINSKI, D. DEUSSEN, O, COHEN-OR, D.; COHEN, M. Locally adapted projections to reduce panorama distortions. **Computer Graphics Forum** , Wiley Online Library, v. 28, n. 4, p. 1083–1089, 2009.
- 90 LIU, S.; ZHAO, L.; LI, J. Panorama Measurement Based on Spherical Projective Geometry. **International journal of database theory and application** , v. 9, n. 1, p. 81–9, 2016.
- 91 BARAZZETTI, L.; PREVITALI, M.; SCAIONI, M. O.; COHEN-OR, D.; COHEN, M. Procedures for Condition Mapping Using 360° Images. **ISPRS International Journal of Geo-Information** , Multidisciplinary Digital Publishing Institute, v. 9, n. 1, p. 34, 2020.
- 92 WANG, Y.; LI, Y.; YANG, D.; CHEN, Z. A fast intra prediction algorithm for 360-degree equirectangular panoramic video. **2017 IEEE Visual Communications and Image Processing (VCIP)** , IEEE, p. 1–4, 2017.
- 93 ARANDJELOVIĆ, R.; ZISSERMAN, A. Three things everyone should know to improve object retrieval. **2012 IEEE Conference on Computer Vision and Pattern Recognition** , IEEE, p. 2911–2918, 2012.
- 94 MUJA, M.; LOWE, D. G. Fast matching of binary features. **2012 Ninth conference on computer and robot vision** , IEEE, p. 404–410, 2012.
- 95 YONG, H.; HUANG, J.; XIANG, W.; HUA, X.; ZHANG, L. Panoramic background image generation for PTZ cameras. **IEEE Transactions on Image Processing**, IEEE, v. 28, n. 7, p. 3162–3176, 2019.
- 96 GAME, P. S.; VAZE, D.; Bio-inspired Optimization: metaheuristic algorithms for optimization. **arXiv preprint arXiv:2003.11637**, 2020.
- 97 MIRJALILI, S. ; MIRJALILI,S. M.;LEWIS,A.; Grey wolf optimizer. **Advances in engineering software**,Elsevier, v. 69,p. 46–61, 2014.
- 98 BANDARU, S. ; DEB,K.; Metaheuristic techniques. **Decision sciences: theory and practice**, v. 220,n. 4598,p. 693–750, 2016.
- 99 YANG, X. A new metaheuristic bat-inspired algorithm. **Nature inspired cooperative strategies for optimization (NICSO 2010)**, Springer, p. 65-74, 2010.
- 100 COELHO, F. C. R.; DA SILVA JUNIOR, I. C.; DIAS, B. H., MARCATO, A. L. ; Metaheurística inspirada na ecolocalização de morcegos: aperfeiçoamento e estudo de casos. **SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL,CLAIO: SBPO Rio de Janeiro**, p. 2588-2599, 2012.

- 101 YILMAZ, S.; KÜÇÜKSİLLE, E. U. A new modification approach on bat algorithm for solving optimization problems **Applied Soft Computing**, Elsevier, v. 28, p. 259-275, 2015.
- 102 BURT, P. J. ;ADELSON, E. H. ; A multiresolution spline with application to image mosaics. **ACM Transactions on Graphics (TOG)**, ACM New York, v. 2, n. 4, p.217-236, 1983.
- 103 LIU, X.; TAN, Y. H.; CHEN, B. M. Adaptive Weight Multi-Band Blending Based Fast Aerial Image Stitching and Mapping. **2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)**, IEEE, p.1997-2002, 2018.

## APÊNDICE A – Resultados Encontrados pelos Métodos de Otimização em cada Simulação

As Figuras 66 e 67 apresentam os melhores resultados das panorâmicas da Faculdade de Artes e Design (UFJF) - Vista 1 obtidas nas 10 simulações realizadas para cada método. O menor valor de função objetivo, ou seja, a melhor solução encontrada entre as 10 simulações foi a 3ª simulação para o *Bat Algorithm* e 7ª simulação para o *Grey Wolf Optimizer*.

Figura 66 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista 1 sem *Multi-band Blending* - *Bat Algorithm*



(a) 1ª Simulação



(b) 2ª Simulação



(c) 3ª Simulação



(d) 4ª Simulação



(e) 5ª Simulação



(f) 6ª Simulação



(g) 7ª Simulação



(h) 8ª Simulação



(i) 9ª Simulação



(j) 10ª Simulação

Figura 67 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF)  
Vista 1 sem *Multi-band Blending* - *Grey Wolf Optimizer*



(a) 1ª Simulação



(b) 2ª Simulação



(c) 3ª Simulação



(d) 4ª Simulação



(e) 5ª Simulação



(f) 6ª Simulação



(g) 7ª Simulação



(h) 8ª Simulação



(i) 9ª Simulação



(j) 10ª Simulação

As Figuras 68 e 69 apresentam os melhores resultados das panorâmicas da Faculdade de Artes e Design (UFJF) - Vista 2 obtidas nas 10 simulações realizadas para cada método. O melhor resultado encontrado pelo *Bat Algorithm* foi obtido na 4ª Simulação e para o *Grey Wolf Optimizer* na 2ª Simulação.

Figura 68 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista 2 sem *Multi-band Blending* - *Bat Algorithm*



(a) 1ª Simulação



(b) 2ª Simulação



(c) 3ª Simulação



(d) 4ª Simulação



(e) 5ª Simulação



(f) 6ª Simulação



(g) 7ª Simulação



(h) 8ª Simulação



(i) 9ª Simulação



(j) 10ª Simulação

Figura 69 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF)  
Vista 2 sem *Multi-band Blending - Grey Wolf Optimizer*



(a) 1ª Simulação



(b) 2ª Simulação



(c) 3ª Simulação



(d) 4ª Simulação



(e) 5ª Simulação



(f) 6ª Simulação



(g) 7ª Simulação



(h) 8ª Simulação



(i) 9ª Simulação



(j) 10ª Simulação

As Figuras 70 e 71 apresentam os melhores resultados das panorâmicas da Faculdade de Artes e Design (UFJF) - Vista 3 obtidas nas 10 simulações realizadas para cada método. Para o *Bat Algorithm* a melhor solução entre todas foi obtida pela 2ª simulação e para *Grey Wolf Optimizer* pela 8ª Simulação.

Figura 70 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF) Vista2 sem *Multi-band Blending* - *Bat Algorithm*



(a) 1ª Simulação



(b) 2ª Simulação



(c) 3ª Simulação



(d) 4ª Simulação



(e) 5ª Simulação



(f) 6ª Simulação



(g) 7ª Simulação



(h) 8ª Simulação





(i) 9ª Simulação



(j) 10ª Simulação

Figura 71 – Resultados das 10 Simulações da Faculdade de Artes e Design (UFJF)  
Vista 3 sem *Multi-band Blending - Grey Wolf Optimizer*



(a) 1ª Simulação



(b) 2ª Simulação



(c) 3ª Simulação



(d) 4ª Simulação



(e) 5ª Simulação



(f) 6ª Simulação



(g) 7ª Simulação



(h) 8ª Simulação



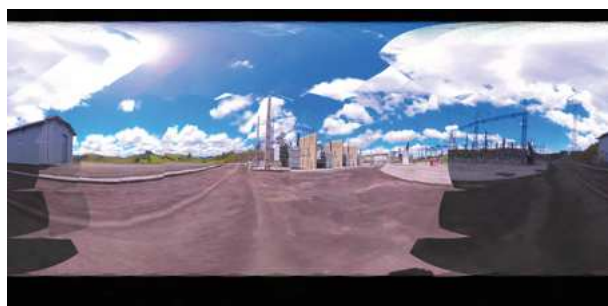
(i) 9ª Simulação



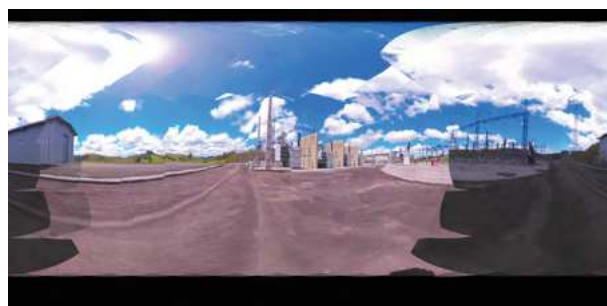
(j) 10ª Simulação

As Figuras 72 e 73 apresentam os melhores resultados das panorâmicas da Subestação de Energia da cidade de Santos Dumont obtidas nas 10 simulações realizadas para cada método. Para a *Bat Algorithm* a melhor solução foi encontrada na 7ª simulação e para *Grey Wolf Optimizer* na 5ª simulação.

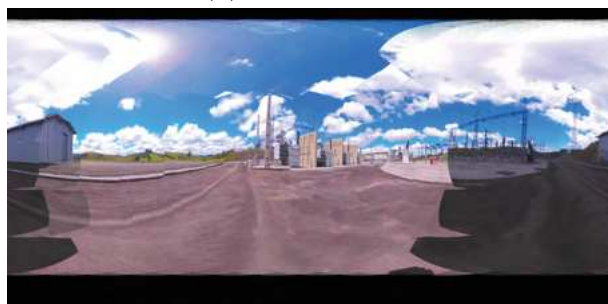
Figura 72 – Resultados das 10 Simulações da Subestação de Energia sem *Multi-band Blending - Bat Algorithm*



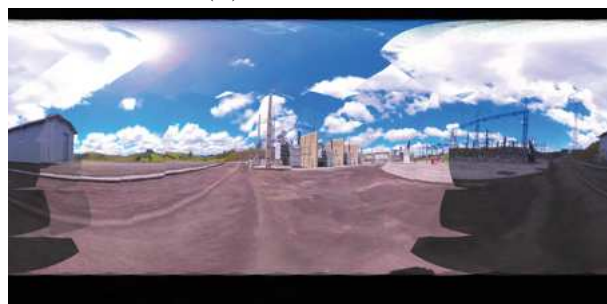
(a) 1ª Simulação



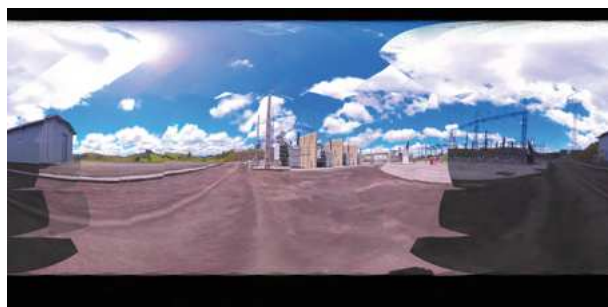
(b) 2ª Simulação



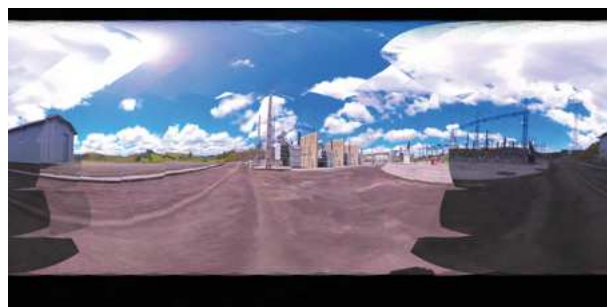
(c) 3ª Simulação



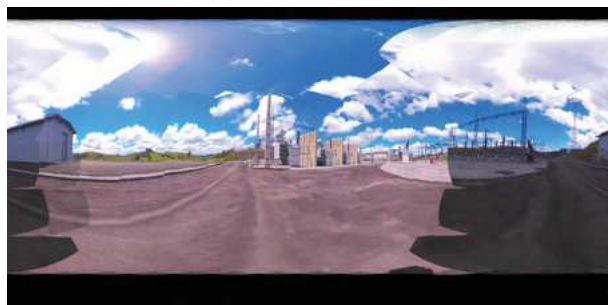
(d) 4ª Simulação



(e) 5ª Simulação



(f) 6ª Simulação



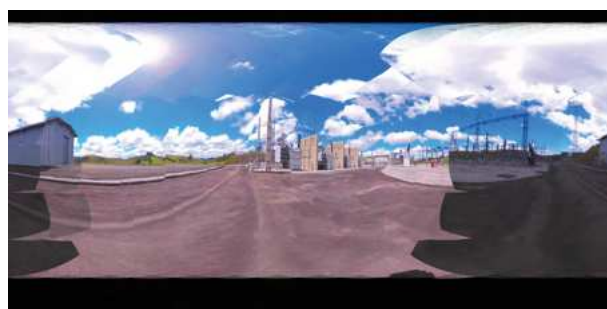
(g) 7ª Simulação



(h) 8ª Simulação



(i) 9ª Simulação



(j) 10ª Simulação

Figura 73 – Resultados das 10 Simulações da Subestação de Energia sem *Multi-band Blending - Grey Wolf Optimizer*



(a) 1ª Simulação



(b) 2ª Simulação



(c) 3ª Simulação



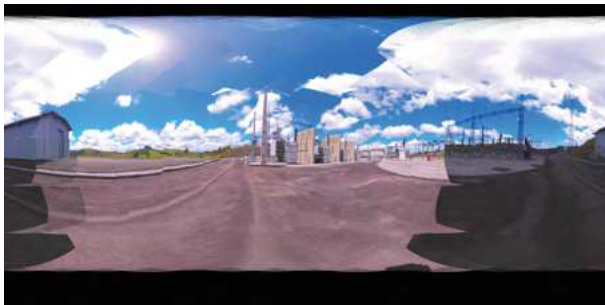
(d) 4ª Simulação



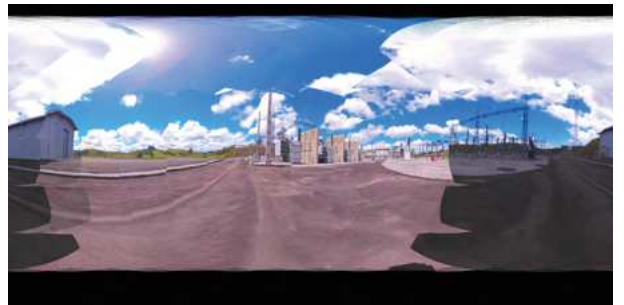
(e) 5ª Simulação



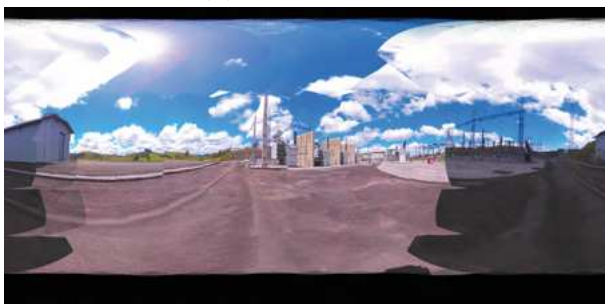
(f) 6ª Simulação



(g) 7ª Simulação



(h) 8ª Simulação



(i) 9ª Simulação



(j) 10ª Simulação