

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Métodos de mosaico em imagens microscópicas

Ângela Silviane Moura Cunha

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Ângela Silviane Moura Cunha

Métodos de mosaico em imagens microscópicas

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestra em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Odemir Martinez Bruno

USP – São Carlos
Junho de 2020

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

M972m Moura Cunha, Ângela Silviane
 Métodos de mosaico em imagens microscópicas /
 Ângela Silviane Moura Cunha; orientador Odemir
 Martinez Bruno. -- São Carlos, 2020.
 111 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Ciências de Computação e Matemática
 Computacional) -- Instituto de Ciências Matemáticas
 e de Computação, Universidade de São Paulo, 2020.

 1. Mosaicos de imagens. 2. Imagens microscópicas.
 3. Correlação de fase. 4. Filtros anisotrópicos. 5.
 Plugin MosaicImage. I. Martinez Bruno, Odemir,
 orient. II. Título.

Ângela Silviane Moura Cunha

Methods of mosaic in microscopic images

Dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Odemir Martinez Bruno

USP – São Carlos
June 2020

*Este trabalho é dedicado a toda minha família e amigos, por todo auxílio e motivação.
Dedico também aos Professores Dra. Carolina Y. V. Watanabe e Dr. Odemir M. Bruno, pela
convicção, incentivo, paciência e excelente orientação.
Ao meu pai José Leônidas Costa Cunha, in memoriam.*

AGRADECIMENTOS

A Deus, acima de tudo.

Ao meu orientador, Prof. Dr. Odemir M. Bruno por todo o incentivo dado durante esses anos de mestrado, paciência e entusiasmo. Obrigada pela compreensão sobre minha ausência no Lab e pela excelente orientação.

Agradeço também a Profa. Dra. Carolina Y. V. Watanabe pela ajuda e incentivo dados ao meu ingresso e prosseguimento na pós-graduação, bem como toda a ajuda fornecida no decorrer do trabalho, e que mesmo tendo conhecimento da minha condição de portadora de TDAH, sempre acreditou na minha capacidade de aprendizagem.

A Profa. Dra. Silvia das D. Rissino e ao Prof. Dr. Paulo Roberto de O. Borges, pela grande ajuda e incentivo dados ao meu ingresso na pós-graduação. Bem como aos meus professores de graduação na Universidade Federal de Rondônia, obrigada por serem os alicerces do meu conhecimento, nunca esquecerei de vocês.

A todos os professores que tive o privilégio de conhecer e que agregaram conhecimento no Instituto de Ciências Matemáticas e de Computação ICMC - USP.

A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela concessão da bolsa de Mestrado.

A todos os meus colegas do *Scientific Computing Group*, José Paulo, Mariane, Marcela, Gisele, Wouter, Leonardo, Lucas e Rayner, pelo companheirismo e ajuda. Em especial aos meus amigos Jeaneth e Alex, pelas inúmeras vezes que me ajudaram tirando dúvidas nos códigos e textos, sempre me dando forças e se preocupando comigo.

Agradeço também aos meus amigos Otávio e Solange por sempre me ajudarem e me fazerem feliz, vocês sempre serão muito importantes em minha vida.

Ao meu querido sensei Erivan que sempre acreditou e teve orgulho de mim, mesmo eu estando longe dos treinos e da nossa academia Shidokan, obrigada sensei, Oss!

Com todo carinho, ao meu Marcos por estar comigo em todos os momentos, me ajudando e auxiliando, por ser compreensivo com a minha ausência e ter me acompanhado sempre. Eu te amo, obrigada.

Por último, mas não menos importante, aos meus filhos gatinhos pelos momentos de calma, tranquilidade e alegria que somente alguns bigodinhos podem proporcionar.

*“Eu não tenho nenhum talento especial.
Sou apenas apaixonadamente curioso.”
(Albert Einstein)*

RESUMO

CUNHA, Â. S. M. **Métodos de mosaico em imagens microscópicas**. 2020. 111 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

No estudo de imagens microscópicas ocorrem vários avanços em microscopia de luz, como registro de imagens, processamento, reconhecimento de estruturas e montagem de mosaicos que auxiliam no exame da amostra estudada. No entanto, em um ambiente real, as imagens podem passar por uma grande quantidade de degradações, como ruídos dos tipos Gaussiano, Poisson e outros. Atualmente, existe na literatura uma maior diversidade de algoritmos que realizam a filtragem das imagens, tendo como exemplo, o filtro linear da média e não-lineares como os anisotrópicos. Este trabalho analisou o desempenho desses filtros juntamente com o processo de composição de mosaicos. Dentre os algoritmos de junção de imagens, os mais significativos são o SIFT e o SURF que apresentam resultados mais satisfatórios. Entretanto, a escolha de um método depende da aplicação a que se destina, ou seja, para realizar mosaicação de imagens microscópicas, por exemplo, uma outra abordagem pode ser a mais adequada. Nesta etapa, a técnica baseada em Fourier mostrou-se mais proeminente em virtude de ser robusta a ruídos, possuir mais baixo tempo de processamento e ser invariante a translação. Dessa forma, este trabalho visou o desenvolvimento de uma metodologia que realizasse a filtragem da imagem para em seguida efetuar a junção, e dessa maneira, identificar os métodos de mosaicação mais robustos, bem como a filtragem mais adequada. Foi constatado que os filtros anisotrópicos de Perona-Malik e *Forward-Backward* mostraram os melhores resultados para os três métodos de mosaico citados. Os experimentos comprovaram que o tempo de processamento da correlação de fase é bem inferior aos demais. Sendo assim, desenvolveu-se uma ferramenta multiplataforma e de domínio público que realiza a junção de imagens microscópicas utilizando a correlação de fase juntamente como o filtro de *Forward-Backward*, visto que grande parte dos softwares disponíveis são comerciáveis e de código fechado.

Palavras-chave: Imagens microscópicas, Métodos de mosaico, Correlação de fase, Filtros anisotrópicos, *Plugin MosaicImage*.

ABSTRACT

CUNHA, Â. S. M. **Methods of mosaic in microscopic images**. 2020. 111 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Within the study of microscopic images there are several advances in light microscopy, such as image recording, processing, structure recognition, and mosaic assembly to assist in the examination of a sample being studied. However, in a real environment, the images can go through a large number of degradations, such as Gaussian, Poisson and others noises. Currently, there is a greater diversity of algorithms in the literature that perform image filtering, taking as an example the mean linear filter or the non-linear anisotropic filter. In this work, we have analyzed the performance of these filters together with the mosaic composition process. Among various image joining algorithms, SIFT and SURF are the most significant presenting more satisfactory results. However, the choice of a method depends on its application, for instance, to perform microscopic image mosaics, meantime other approaches perhaps can be more appropriate. At this stage, the Fourier-based approach proved to be more prominent due to its noise robustness, shorter processing time and translation invariant. Thus, this study aimed to develop a methodology to filter the image and then perform the junction, to identify the most robust mosaic methods, as well as the most appropriate filters. We found that Perona-Malik and Forward-Backward anisotropic filters showed the best results for the three mosaic methods previously mentioned. The experiments proved that the processing time of the phase correlation is much shorter than the others. Thus, we developed a tool to perform the junction of microscopic images using phase correlation as well as the Forward-Backward filter. Opposite to the commercial and closed source of most of the available software, we present a multiplatform and public domain tool.

Keywords: Microscopic images, Mosaic methods, Phase correlation, Anisotropic filters, MosaicImage Plugin.

LISTA DE ILUSTRAÇÕES

Figura 1 – Mosaico composto por duas imagens.	25
Figura 2 – Etapas do processo de construção do mosaico de imagens. H são as matrizes de homografia entre as imagens.	29
Figura 3 – Classificação das técnicas de mosaico com base no registro.	31
Figura 4 – Detector de canto Harris.	34
Figura 5 – Detecção de características do ponto candidato usando o algoritmo FAST com 16 <i>pixels</i> no círculo.	36
Figura 6 – Construção das imagens resultantes da diferença de Gaussianas <i>DoG</i> em cada oitava no espaço de escala.	38
Figura 7 – Magnitudes e orientações do descritor SIFT	39
Figura 8 – Passos para a construção da matriz <i>SAT</i> . Índices fora da fronteira são considerados zero.	40
Figura 9 – Aplicação da Equação 2.13 acima para o cálculo da imagem integral em um retângulo <i>ABCD</i>	40
Figura 10 – Incremento do tamanho do filtro de Caixa. Filtro D_{yy} (em cima) e D_{xy} (em baixo) para dois níveis de escalas (9×9) e (15×15).	41
Figura 11 – Exemplos de pontos de interesse detectados pelo SURF e extração de características.	42
Figura 12 – Uso do espectro de potência cruzada para detectar as transformações das imagens (a) e (b), e o deslocamento entre elas. (c) é o espectro de (a); (d) é o espectro de (b); (e) é a função que indica o deslocamento entre as imagens de origem.	45
Figura 13 – Principais passos da metodologia proposta.	50
Figura 14 – Software AxioVision e microscópio conectado ao microcomputador através de interface USB.	51
Figura 15 – (a) Imagem original. (b) Adição de 0.05% de ruído S&P.	53
Figura 16 – (a) Imagem original. (b) Imagem corrompida por ruído Gaussiano com $\bar{z} = 0$ e $\sigma^2 = 0.004$	53
Figura 17 – (a) Imagem original. (b) Imagem corrompida por ruído do tipo Poisson com duas iterações.	54
Figura 18 – (a) Imagem original. (b) Imagem corrompida por ruído do tipo multiplicativo com $\bar{z} = 0$ e $\sigma^2 = 0.02$	55
Figura 19 – Bases com ampliação de $5\times$ e $10\times$, respectivamente.	62

Figura 20 – Base adquirida com a Lupa.	62
Figura 21 – Variações do ruído SeP (% de ruído).	66
Figura 22 – Variações do ruído Gaussiano (1), ($\bar{z} = 0$ e altera a σ^2).	67
Figura 23 – Variações do ruído Gaussiano (2), somente para PC, ($\bar{z} = 0$ e altera a σ^2).	68
Figura 24 – Variações do ruído Poisson (N ^o de iterações com a imagem corrompida).	69
Figura 25 – Variações do ruído Multiplicativo (altera a σ^2).	70
Figura 26 – Filtragem de (a) imagem corrompida com 30% de ruído S&P.	71
Figura 27 – Filtragem de (a) imagem corrompida com ruído Gaussiano, com $\bar{z} = 0$ e $\sigma^2 = 0.10$	72
Figura 28 – Filtragem de (a) imagem corrompida com 10 iterações do ruído Poisson.	73
Figura 29 – Filtragem de (a) imagem corrompida com ruído Multiplicativo, com $\sigma^2 = 0.15$	74
Figura 30 – SURF: ruído S&P.	76
Figura 31 – SURF: ruído Gaussiano.	77
Figura 32 – SURF: ruído Poisson.	78
Figura 33 – SURF: ruído Multiplicativo.	79
Figura 34 – SIFT: ruído S&P.	80
Figura 35 – SIFT: ruído Gaussiano.	81
Figura 36 – SIFT: ruído Poisson.	82
Figura 37 – SIFT: ruído Multiplicativo.	83
Figura 38 – PC: ruído S&P.	84
Figura 39 – PC: ruído Gaussiano (1).	85
Figura 40 – PC: ruído Gaussiano (2).	86
Figura 41 – PC: ruído Poisson.	87
Figura 42 – PC: ruído Multiplicativo.	88
Figura 43 – Janela principal do ImageJ.	94
Figura 44 – Visão geral do código fonte do <i>MosaicImage</i>	95
Figura 45 – Janela principal do <i>MosaicImage</i>	96
Figura 46 – Submenu File do <i>plugin MosaicImage</i>	97
Figura 47 – Submenu Help do <i>plugin MosaicImage</i>	97
Figura 48 – Submenu About do <i>MosaicImage</i>	98
Figura 49 – <i>MosaicImage</i> : abrindo um diretório de imagens.	98
Figura 50 – <i>MosaicImage</i> : criando um mosaico entre duas imagens.	99
Figura 51 – <i>MosaicImage</i> : carregamento das imagens no divisão inferior da tela.	99
Figura 52 – <i>MosaicImage</i> : geração do mosaico final.	100
Figura 53 – Mosaico criado a partir de imagens com ampliação de 5×.	100
Figura 54 – Mosaico criado a partir de imagens com ampliação de 10×.	101

LISTA DE QUADROS

Quadro 1 – Comparação entre os diferentes métodos de mosaico de imagens baseados em registro.	46
Quadro 2 – Quadro resumo: melhor filtro para cada tipo de ruído.	89

LISTA DE TABELAS

Tabela 1 – Descrição das bases de imagens.	63
Tabela 2 – Parâmetros utilizados para os ruídos.	63
Tabela 3 – Parâmetros utilizados para os filtros.	64
Tabela 4 – SURF: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo S&P.	75
Tabela 5 – SURF: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Gaussiano.	76
Tabela 6 – SURF: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Poisson.	77
Tabela 7 – SURF: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Multiplicativo.	78
Tabela 8 – SIFT: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo S&P.	80
Tabela 9 – SIFT: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Gaussiano.	81
Tabela 10 – SIFT: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Poisson.	82
Tabela 11 – SIFT: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Multiplicativo.	83
Tabela 12 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo S&P.	84
Tabela 13 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Gaussiano (1).	85
Tabela 14 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Gaussiano (2).	86
Tabela 15 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Poisson.	87
Tabela 16 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Multiplicativo.	88
Tabela 17 – Comparativo entre métodos: Base de imagens com ampliação de 5×.	90
Tabela 18 – Comparativo entre métodos: Base de imagens com ampliação de 10×.	90
Tabela 19 – Comparativo entre métodos: Base de imagens da Lupa.	90
Tabela 20 – Tempo de processamento dos métodos de mosaico.	91

LISTA DE ABREVIATURAS E SIGLAS

BoW	<i>Bag-of-Words (BoW)</i>
DoG	<i>Difference of Gaussian</i>
EQM	Erro Quadrático Médio
FAST	<i>Features from Accelerated Segment Test</i>
FFT	<i>Fast Fourier Transform</i>
LoG	<i>Laplacian of Gaussian</i>
MI	<i>Mutual Information</i>
NCC	<i>Normalized Cross Correlation</i>
PC	<i>Phase Correlation</i>
PDF	<i>Probability Density Function</i>
RANSAC	<i>Randon Sample Consensus</i>
SAT	<i>Summed Area Table</i>
SCG	<i>Scientific Computing Group</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
SR/SF	Sem Ruído e Sem Filtro
SSD	<i>Sum of Squared Differences</i>
SURF	<i>Speeded Up Robust Features</i>

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação e justificativa	27
1.2	Objetivos e contribuições	27
1.3	Reprodutibilidade	28
1.4	Organização do trabalho	28
2	MÉTODOS DE CONSTRUÇÃO DE MOSAICOS	29
2.1	Mosaicos baseados em registro	31
2.1.1	<i>Domínio espacial</i>	31
2.1.1.1	<i>Correlação Cruzada Normalizada</i>	32
2.1.1.2	<i>Informação Mútua</i>	33
2.1.1.3	<i>Detector Harris</i>	34
2.1.1.4	<i>Features from Accelerated Segment Test - FAST</i>	35
2.1.1.5	<i>Scale-Invariant Feature Transform - SIFT</i>	36
2.1.1.6	<i>Speeded Up Robust Features - SURF</i>	39
2.1.1.7	<i>Contorno</i>	43
2.1.2	<i>Domínio de Frequência</i>	43
2.2	Comparativo entre as técnicas de mosaicos	45
2.3	Considerações finais	47
3	TÉCNICAS PARA APRIMORAR OS MÉTODOS DE MOSAICO	49
3.1	Passo 1: Aquisição de imagens	50
3.2	Passo 2: Adição de Ruído	51
3.2.1	<i>Ruído Sal e Pimenta ou Impulsivo</i>	52
3.2.2	<i>Ruído Gaussiano ou Térmico</i>	52
3.2.3	<i>Ruído Poisson ou Relativo à Contagem de Fótons</i>	53
3.2.4	<i>Ruído Multiplicativo ou Speckle</i>	54
3.3	Passo 3: Filtragem	55
3.3.1	<i>Filtro da Média</i>	55
3.3.2	<i>Filtro da Mediana</i>	56
3.3.3	<i>Filtro de Wiener</i>	56
3.3.4	<i>Difusão Anisotrópica por Perona-Malik</i>	57
3.3.5	<i>Regularização Forward-Backward da Difusão Anisotrópica</i>	58

3.4	Passo 4: Correspondência	59
3.5	Considerações finais	59
4	EXPERIMENTOS, RESULTADOS E DISCUSSÕES	61
4.1	Base de Imagens	61
4.1.1	<i>Bases para ruídos, filtros e correspondência</i>	<i>61</i>
4.2	Configuração dos Algoritmos	63
4.2.1	<i>Parâmetros dos ruídos</i>	<i>63</i>
4.2.2	<i>Parâmetros dos filtros</i>	<i>64</i>
4.2.3	<i>Algoritmos de mosaico</i>	<i>64</i>
4.3	Localização	64
4.4	Descrição dos experimentos	65
4.5	Experimento 1: Testes com ruídos e filtros: SURF	75
4.6	Experimento 2: Testes com ruídos e filtros: SIFT	79
4.7	Experimento 3: Testes com ruídos e filtros: PC	83
4.8	Resumo dos resultados dos testes com ruídos e filtros	88
4.9	Experimento 4: Comparativo entre os métodos de mosaico	89
4.10	Experimento 5: Análise de tempo de processamento	91
4.11	Considerações finais	91
5	FERRAMENTA DESENVOLVIDA	93
5.1	ImageJ	93
5.1.1	<i>Plugins do ImageJ</i>	<i>94</i>
5.2	MosaicImage	95
5.2.1	<i>Ferramenta Desenvolvida: Exemplos de uso</i>	<i>98</i>
5.3	Considerações finais	101
6	CONCLUSÕES	103
6.1	Contribuições do trabalho	104
6.2	Propostas para trabalhos futuros	104
	REFERÊNCIAS	105

INTRODUÇÃO

A costura de imagens ou mosaico de imagens, é o processo que combina imagens com áreas sobrepostas para formar uma grande imagem com alta resolução (WANG; YANG, 2020). A Figura 1 ilustra um exemplo de mosaico de imagens.

Figura 1 – Mosaico composto por duas imagens.



Fonte: Pandey e Pati (2015).

A criação de mosaico de imagens data de 1839 com a invenção da fotografia, quando diversas imagens eram capturadas usando um filme e logo após isso, reveladas em papel, cortadas e coladas (NEWHALL, 1949; ASSIS *et al.*, 2012). Por volta do ano de 1899 foi lançada a câmera "Al-Vista", produzida especificamente para a criação de mosaicos (LUHMANN, 2008). Essa técnica têm sido amplamente utilizada pois a visualização de uma única grande imagem é muito mais agradável do que observar essas diversas imagens separadamente. Atualmente, a composição de mosaico de imagens usa apenas software, muitas vezes incorporado à máquina

fotográfica (ASSIS *et al.*, 2012). A construção automática de mosaicos de imagens de alta resolução é uma área de pesquisa ativa nos campos da fotogrametria, visão computacional, processamento de imagens e computação gráfica. Dentre essas várias aplicações, destacam-se os mosaicos que são construídos a partir de imagens microscópicas (MA *et al.*, 2007; YONGWEI; XINKE; JIAYIN, 2009), imagens de satélite (HELMER; RUEFENACHT, 2005; BIELSKI; GRAZZINI; SOILLE, 2007) e possuem trabalhos que usam imagens aéreas (TARALLO, 2013; GUIZILINI *et al.*, 2017).

No âmbito de imagens microscópicas, salienta-se que os vários avanços em microscopia de luz advêm das câmeras fotográficas digitais e processamento de imagens computacionais. A ampliação de uma amostra se faz utilizando a luz e lentes. Dessa forma, é possível realizar o registro de diversas imagens, processamento, reconhecimento de estruturas, montagem de mosaicos, dentre outras funções que auxiliam na inspeção do objeto de estudo (TELES; ANDREANI; VALADARES, 2017). No contexto da microscopia é possível encontrar inúmeros trabalhos, dentre eles, Seo *et al.* (2019), Kose *et al.* (2017) e Chalfoun *et al.* (2017), em que os autores propõem diferentes abordagens para realizar a costura de imagens microscópicas.

Existem inúmeros métodos na literatura para a construção de mosaico de imagens, dentre eles podemos destacar os métodos baseados em:

1. **Área ou regiões:** Correlação cruzada normalizada (SZELISKI, 2006); Informação mútua (GHANNAM; ABBOTT, 2013); e Pirâmide Laplaciana Burt e Adelson (1983);
2. **Características:** Detector de canto Harris (PATIL; PATIL, 2016); FAST - *Features from Accelerated Segment Test* (JIAO; ZHAO; WU, 2011); SIFT - *Scale Invariant Feature Transform* (LOWE, 1999); SURF - *Speeded Up Robust Features* (BAY; TUYTELAARS; GOOL, 2008); e Características KAZE (ALCANTARILLA; BARTOLI; DAVISON, 2012);
3. **Contorno** (ISLAM; KABIR, 2013); e
4. **Domínio de frequência** (KUGLIN; HINES, 1975).

Para realizar a junção de duas imagens é necessário identificar a região de sobreposição entre elas, ou seja, encontrar os pontos homólogos. A realização dessa função de modo integralmente automático é uma tarefa complexa, e tem produzido diversas pesquisas na área (BASRI; JACOBS, 1997; FRIEDMAN; BENTLEY; FINKEL, 1976; ASSIS *et al.*, 2012). Os métodos baseados em área, características ou contorno conseguem realizar essa correspondência entre as imagens e posteriormente filtrar possíveis pontos incompatíveis, entretanto, são abordagens que geralmente possuem alto custo computacional. Na literatura também podem ser encontrados trabalhos com metodologias fundamentadas no domínio de frequência (BOSSERT, 2004; KIM; YOON; CHOI, 2003; PANDEY; PATI, 2015; PATIDAR; JAIN, 2011; HOGE; MAMATA; MAIER, 2003). Este método, chamado de correlação de fase, possui a vantagem de alinhar duas

imagens usando a FFT (do inglês, *Fast Fourier Transform*) e apresenta o deslocamento preciso entre as duas imagens sobrepostas, o que possibilita a correção das coordenadas da segunda imagem, com relação à primeira, chamada de imagem de referência.

1.1 Motivação e justificativa

Alguns softwares utilizados para a construção de mosaico de imagens estão disponíveis comercialmente, dentre eles o *AutoStitch* (AUTOSTITCH, 2007), o *PTGui* (PTGUI, 2011) e o *AxioVision* (AXIOVISION, 2012). Este último é usado no controle de microscópios, captura e processamento de imagens microscópicas e realização de mosaicagem. Entretanto, são softwares proprietários, ou seja, apenas uma empresa ou indivíduo têm o controle sobre as funcionalidades, correções e possíveis melhoramentos. Portanto, é necessário o desenvolvimento de um software de domínio público capaz de elaborar mosaicos das imagens microscópicas capturadas, uma vez que, esquadrihar a amostra e gerar uma única grande imagem permite a observação integral do objeto e eventual análise do problema, uma vez que a análise de imagens isoladas podem ocasionar imprecisões na tomada de decisão.

Essas ferramentas proprietárias fazem uso de diversos métodos para a geração de um mosaico final, e uma das etapas fundamentais para a junção de imagens é encontrar semelhanças entre as imagens que devem integrar o mosaico. Esse passo requer algoritmos robustos e alguns deles com custo computacional alto. Na literatura pode-se encontrar uma diversidade de abordagens baseadas no domínio espacial ou mesmo no domínio de frequência que realizam a tarefa de construção de mosaicos. Dentre os métodos de mosaico mais relevantes, estão os algoritmos SURF, SIFT e a correlação de base (que opera no domínio de frequência), sendo este último considerado muito eficaz por utilizar a FFT.

Sendo assim, é necessário elaborar uma abordagem que seja capaz de aperfeiçoar os principais algoritmos de mosaico mencionados, analisar e identificar o método de mosaico mais adequado para aplicações em imagens microscópicas, bem como desenvolver uma ferramenta de domínio público voltada para mosaicagem de imagens capturadas ao microscópio.

1.2 Objetivos e contribuições

De forma resumida, os objetivos deste trabalho foram:

1. Investigação e análise dos principais métodos da literatura de mosaico de imagens.
2. Proposta de um nova abordagem que aprimore esses principais métodos de mosaico.
3. Análise dos resultados e, a partir disso, desenvolvimento de uma ferramenta de mosaico intitulada *MosaicImage*, que tem o intuito de substituir o utilizado atualmente no estereo-

microscópio Axio-Lab A1 Zeiss V20 pertencente ao grupo de pesquisas SCG - *Scientific Computing Group* do IFSC – Instituto de Física de São Carlos.

1.3 Reprodutibilidade

Os testes de análise de ruídos e filtros foram realizados no MATLAB e o *plugin MosaicImage* foi desenvolvido em Java versão 1.8.0.212. Para fins de reprodutibilidade, os códigos estão disponíveis em um repositório Git pelo link <<https://github.com/silvianecunha/Dissertation-files.git>>.

1.4 Organização do trabalho

Esta dissertação é composta por este capítulo introdutório e mais cinco capítulos organizados da seguinte maneira:

Capítulo 2: métodos de construção de mosaicos, onde são abordadas as principais técnicas do domínio espacial e do domínio de frequência.

Capítulo 3: apresenta a metodologia desenvolvida no trabalho.

Capítulo 4: são apresentados os experimentos, resultados e discussões obtidos.

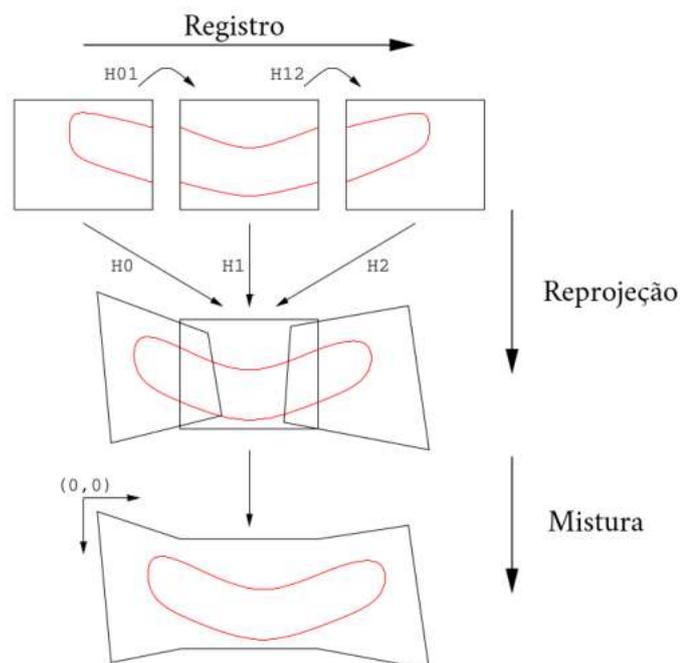
Capítulo 5: detalha a ferramenta *MosaicImage*, bem como um exemplo de uso.

Capítulo 6: apresenta as conclusões finais desta dissertação, contribuições e trabalhos futuros.

MÉTODOS DE CONSTRUÇÃO DE MOSAICOS

Um mosaico de imagens representa a união de uma determinada quantidade de imagens em uma nova e única grande imagem (SHUM; SZELISKI, 2001; TARALLO, 2013). A construção de mosaicos de imagens é descrita na literatura em vários contextos, mas para uma maior compreensão, o método envolve três etapas básicas: registro, reprojeção e mistura, como pode ser visto na Figura 2 (CAPEL, 2001).

Figura 2 – Etapas do processo de construção do mosaico de imagens. H são as matrizes de homografia entre as imagens.



Fonte: Adaptada de Capel (2001).

O **registro** de imagens é a principal etapa no processo de construção de mosaicos (BAO; XU, 1998), que é a determinação da correspondência geométrica entre um par de imagens que reproduzem uma mesma cena. Em virtude das diversas condições em que as imagens podem ser capturadas, tendo como exemplo, tempos diferentes, pontos de vista, ou mesmo por diferentes sensores. Estas imagens podem ter inúmeros tipos de distorções, ocasionando, por conseguinte, um maior ou menor grau de complexidade nas transformações a que as imagens estarão sujeitas no processo de alinhamento geométrico entre duas imagens (ZITOVÁ; FLUSSER, 2003).

A **reprojeção** se refere ao alinhamento das imagens em um sistema de coordenadas comuns usando as transformações geométricas calculadas. O objetivo neste passo é sobrepor as imagens alinhadas em uma tela maior, mesclando valores de *pixels* das partes sobrepostas e restando *pixels* onde não ocorre sobreposição. Os erros propagados por desalinhamentos geométricos e fotométricos geralmente resultam em discontinuidades indesejáveis do objeto e visibilidade dessa união na proximidade do limite entre duas imagens (GHOSH; KAABOUC, 2016).

A etapa final de **mistura**, concerne a minimização das discontinuidades na aparência global do mosaico. Um algoritmo de mistura precisa ser usado durante ou após o passo de costura das imagens. Se esse problema não for corrigido, pode dar origem a costuras anti-estéticas, gerando uma imagem que apresenta uma aparência de corte nas junções. Para melhorar esse efeito poderia ser usado, por exemplo, um método de média simples de valores de intensidade da imagem (CAPEL, 2001).

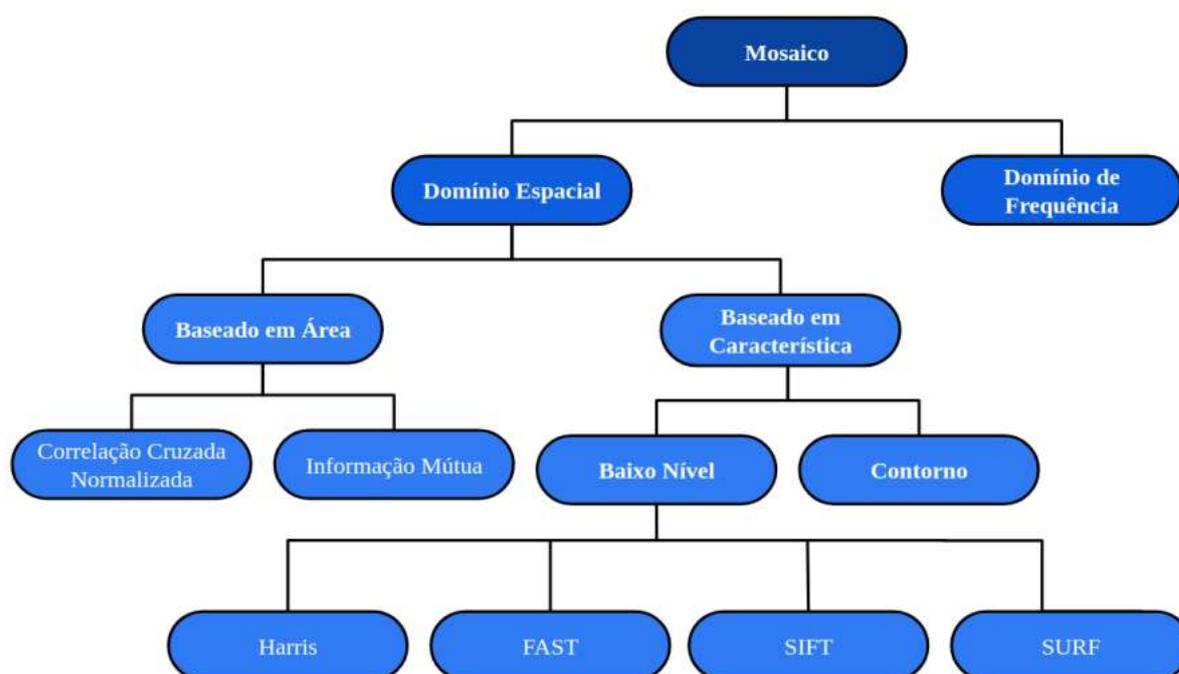
O registro de imagens e a mistura são áreas de pesquisa relevantes e que influenciam diretamente o desempenho no processo de construção de mosaicos. Torna-se quase impossível construir um algoritmo de mosaico de sucesso sem antes implementar adequadamente os algoritmos de registro e mistura. Apesar de terem sido realizados trabalhos com o objetivo de superar os erros de registro utilizando rebuscados algoritmos de mistura, a importância da etapa de registro permanece ainda inquestionável (GHOSH; KAABOUC, 2016). Dessa forma, neste capítulo serão abordadas nas próximas seções alguns conceitos das principais técnicas de mosaico de imagens que são baseadas em registro.

No processo descrito de construção de mosaicos, após encontrar os pontos de interesse em cada imagem, um algoritmo de consenso é utilizado para eliminar correspondências falsas, bem como para calcular as matrizes de homografia. O primeiro método de consenso desenvolvido chama-se RANSAC (do inglês, *Randon Sample Consensus*) proposto por (FISCHLER; BOLLES, 1981). O trabalho de Ramos (2016) apresenta alguns dos algoritmos de consenso propostos na literatura.

2.1 Mosaicos baseados em registro

O registro de imagens não é apenas um passo importante do mosaico de imagens, mas também a base disso. Conforme a [Figura 3](#), de acordo com os métodos de registro de imagens, os algoritmos podem ser fundamentados em **domínio espacial** ou em **domínio de frequência**.

Figura 3 – Classificação das técnicas de mosaico com base no registro.



Fonte: Adaptada de [Ghosh e Kaabouch \(2016\)](#).

2.1.1 Domínio espacial

Os algoritmos nesta categoria usam propriedades dos *pixels* para realizar o registro e, portanto, são os métodos mais diretos de modelagem de imagens. A maioria dos algoritmos de modelagem de imagem existentes se enquadram nesta categoria. O mosaico de imagem baseado em domínio espacial pode ser **baseado em área** ou **baseado em característica** ([GHANNAM; ABBOTT, 2013](#)).

Os métodos baseados em área, conhecidos também como métodos de correlação ou *template matching*, mesclam o passo de detecção de características com o passo de busca das mesmas. Esses métodos processam as imagens, sem tentar detectar objetos salientes, realizando, na maior parte dos casos, a busca por regiões de tamanho constante, conhecidos como *templates* ([STUCHI, 2013](#)). Os algoritmos de mosaico de imagens baseados em área mais comumente utilizados são o mosaico baseado em **correlação cruzada normalizada** ([SZELISKI, 2006](#)) e o baseado em **informação mútua** ([WOODS; CHERRY; MAZZIOTTA, 1992](#); [WOODS; MAZZIOTTA; CHERRY, 1993](#)).

Os algoritmos de mosaicos baseados em características não requerem imagens com grandes áreas sobrepostas. Estes algoritmos baseiam-se no cálculo de transformação usando um conjunto escasso de recursos de baixo nível. Incluem, por exemplo, borda, canto, *pixel*, cor ou histograma. Independentemente de qual recurso de baixo nível seja escolhido, ele deve ser distinto e espalhado por toda a imagem, e também deve ser eficientemente detectável em ambas as imagens. O algoritmo detector de recursos deve ser tal que o número de características comuns detectadas a partir de um conjunto de imagens seja suficientemente grande mesmo na presença das várias mudanças geométricas e radiométricas. Além disso, o detector deve ter alta taxa de repetibilidade, de modo que as mesmas características sejam detectadas nas regiões sobrepostas entre par de imagens (BIND, 2013). Exemplos para esses métodos de mosaico são o detector **Harris** (HARRIS; STEPHENS, 1988), **FAST** (TRAJKOVIĆ; HEDLEY, 1998), **SIFT** (LOWE, 1999), **SURF** (BAY; TUYTELAARS; GOOL, 2008) e **Contorno** (HUI; MANJUNATH; MITRA, 1995), este último baseia-se na extração de características de alto nível, por exemplo, formas ou texturas em uma imagem (ISLAM; KABIR, 2013).

Nas próximas subseções deste capítulo serão explanados de forma breve estes exemplos de métodos de registro baseados no domínio espacial, já na Seção 2.1.2 será abordado o método de registro baseado no domínio de frequência.

2.1.1.1 Correlação Cruzada Normalizada

O método de **Correlação Cruzada Normalizada** (do inglês, *Normalized Cross Correlation - NCC*) realiza o processamento da imagem no domínio do espaço. Este método, calcula a semelhança entre as “janelas” nas duas imagens e é definido como (SZELISKI, 2006):

$$NCC_{(u)} = \frac{\sum_i [I_1(x_i) - \bar{I}_1] [I_2(x_i + u) - \bar{I}_2]}{\sqrt{\sum_i [I_1(x_i) - \bar{I}_1]^2 [I_2(x_i + u) - \bar{I}_2]^2}} \quad (2.1)$$

onde

$$\bar{I}_1 = \frac{1}{N} \sum_i I_1(x_i), \quad (2.2)$$

$$\bar{I}_2 = \frac{1}{N} \sum_i I_2(x_i + u), \quad (2.3)$$

onde \bar{I}_1 e \bar{I}_2 são as imagens correspondentes às “janelas” $I_1(x, y)$ e $I_2(x, y)$, para a primeira e segunda imagem respectivamente. N é o número de *pixels* da “janela”, $x_i = (x_i, y_i)$ é a coordenada do *pixel* da “janela”; $u = (u, v)$ é o deslocamento em que o coeficiente NCC é calculado. Os valores do coeficiente NCC estão sempre dentro do intervalo $[-1, 1]$. O parâmetro de mudança correspondente ao valor máximo de NCC representa a transformação geométrica entre as duas imagens. Uma vez que as transformações geométricas entre os par de imagens são obtidas, estas

são distorcidas no quadro de referência e, finalmente, a costura/união é realizada para gerar o mosaico final. Os métodos dentro desta categoria, ou seja, baseados em área, têm a vantagem de ser computacionalmente simples, no entanto, são particularmente lentos. Além disso, estes métodos executam com precisão somente quando ocorrem sobreposições significativas entre as imagens.

2.1.1.2 Informação Mútua

As pesquisas que acabaram por conduzir à introdução da informação mútua como uma medida de registro remonta ao início dos anos de 1990 (PLUIM; MAINTZ; VIERGEVER, 2003; WOODS; CHERRY; MAZZIOTTA, 1992; WOODS; MAZZIOTTA; CHERRY, 1993), e ao contrário do *NCC*, que calcula similaridade com base em valores de intensidade da imagem, a informação mútua mede a similaridade com base na quantidade de informação compartilhada entre duas imagens. A Informação Mútua (do inglês, *Mutual Information - MI*) baseia-se em conceitos da teoria da informação, e pode ser considerada uma medida de dependência estatística entre os conjuntos de dados e é adequada principalmente para o registro de imagens de diferentes modalidades. A *MI* entre duas imagens $I_1(x, y)$ e $I_2(x, y)$ é expressa em termos de entropia como (GHOSH; KAABOUCH, 2016):

$$MI(I_1, I_2) = E(I_1) + E(I_2) - E(I_1, I_2), \quad (2.4)$$

onde $E(I_1)$ e $E(I_2)$ são as entropias de $I_1(x, y)$ e $I_2(x, y)$, respectivamente. $E(I_1, I_2)$ representa a entropia conjunta entre as duas imagens. Entropia é uma medida de variabilidade de uma variável aleatória. Assim, a variabilidade de $I_1(x, y)$ é dada por:

$$E(I_1) = - \sum_g p_{I_1}(g) \log(p_{I_1}(g)), \quad (2.5)$$

em que g são os possíveis valores de níveis de cinza de $I_1(x, y)$ e, conseqüentemente $p_{I_1}(g)$ é a função de distribuição de probabilidade de g . Do mesmo modo, a variabilidade conjunta de $I_1(x, y)$ e $I_2(x, y)$ é expressa por:

$$E(I_1, I_2) = - \sum_{g, h} p_{I_1, I_2}(g, h) \log(p_{I_1, I_2}(g, h)), \quad (2.6)$$

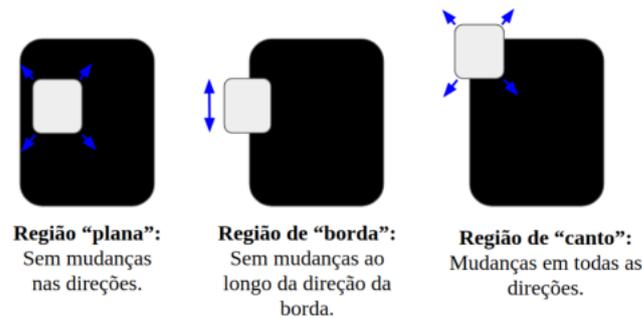
onde h indica as possibilidades de níveis de cinza de $I_2(x, y)$. E $p_{I_1, I_2}(g, h)$ é a função de distribuição de probabilidade conjunta de g e h . Normalmente, a distribuição de probabilidade conjunta entre duas imagens é medida como um histograma normalizado dos valores dos níveis de cinza. Observa-se que quanto melhor o alinhamento entre duas imagens, maior o *MI* entre elas. Dessa forma, duas imagens são alinhadas geometricamente por uma transformação se o *MI* entre elas for máximo para essa transformação. Após as transformações apropriadas serem obtidas entre os par de imagens, elas são reprojctadas e costuradas para obter o mosaico final (GHOSH; KAABOUCH, 2016).

Entretanto, semelhante aos métodos baseados em *NCC*, essa técnica têm a desvantagem de ser computacionalmente lenta e exigir um alto grau de sobreposição entre as imagens de entrada. Um dos primeiros trabalhos que propõe este método é o de [Viola e III \(1997\)](#), onde os autores descreveram uma aplicação da Informação Mútua para o registro de imagens de ressonância magnética.

2.1.1.3 Detector Harris

Este algoritmo foi desenvolvido por [Harris e Stephens \(1988\)](#) como um passo de processamento de baixo nível. Os autores estavam interessados em usar técnicas de análise de movimento para interpretar o ambiente com base em imagens de uma única câmera móvel ([PATIL; PATIL, 2016](#)). Essa técnica é invariante a rotação e variante a escala. O detetor de canto Harris, como mostra a [Figura 4](#), detecta pontos de esquina como recursos robustos de baixo nível a partir de imagens de origem.

Figura 4 – Detetor de canto Harris.



Fonte: Adaptada de [Patil e Patil \(2016\)](#).

Inicialmente, uma janela de detecção local em uma imagem é escolhida. Posteriormente, a variação de intensidade que resulta do pequeno deslocamento da janela em uma direção diferente é determinada como ([JOSHI; SINHA, 2013](#)):

$$E(u, v) = \sum_i w(x_i, y_i) [I(x_i + u, y_i + v) - I(x_i, y_i)]^2, \quad (2.7)$$

onde $w(x_i, y_i)$ é a função de janela para a janela de detecção (x_i, y_i) , $I(x_i, y_i)$ é o valor da intensidade do *pixel* localizado em (x_i, y_i) , e $I(x_i + u, y_i + v)$ é o valor de intensidade deslocada com (u, v) . A textura local em torno do *pixel* (x_i, y_i) é expressa como matriz de autocorrelação. Conforme a [Equação 2.8](#),

$$C = \sum_i w(x_i, y_i) \begin{bmatrix} I_{x_i}^2 & I_{x_i} I_{y_i} \\ I_{x_i} I_{y_i} & I_{y_i}^2 \end{bmatrix}, \quad (2.8)$$

onde I_{x_i} e I_{y_i} são as primeiras derivadas de $I(x_i, y_i)$. Dois autovalores grandes para a matriz C correspondem a um ponto de canto. O ponto central da janela é caracterizado como um

ponto de canto. Logo após os pontos de canto do Harris serem detectados a partir das imagens de origem, as correspondências são estabelecidas por *NCC* ou por qualquer outro método de Soma das Diferenças Quadráticas ou *SSD* (do inglês, *Sum of Squared Differences*). Por fim, os parâmetros de movimento geométrico são calculados e as imagens são deformadas em um quadro de referência global, a fim de realizar a junção de todas elas. O método de detecção de canto Harris original apresenta a desvantagem de que, mesmo que seja robusto para as mudanças de iluminação e rotações, é muito sensível à variação do tamanho da imagem (KHACHANE; GOHATRE, 2016).

2.1.1.4 Features from Accelerated Segment Test - FAST

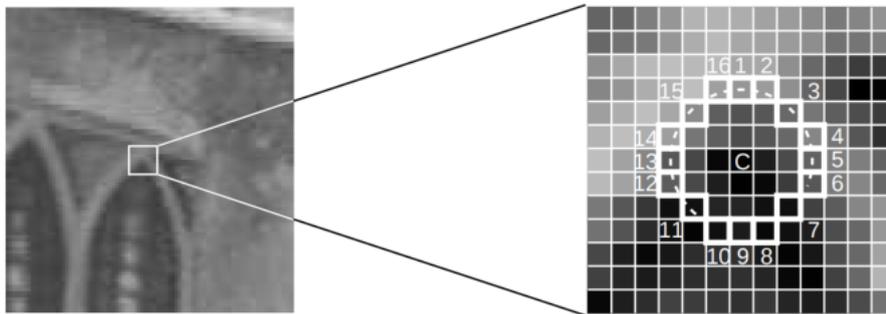
O FAST é um algoritmo baseado no critério de canto e foi apresentado por Trajković e Hedley (1998). É um detector de características de alta velocidade com propriedades de repetibilidade fortes adequadas às aplicações em tempo real. Um outro atributo do operador FAST é a sua invariância à rotação e às mudanças na escala (PATEL; GURJWAR, 2016). De acordo com os autores, o detector deve satisfazer os seguintes critérios (JOSHI; SINHA, 2013):

- As posições detectadas devem ser consistentes, insensíveis à variação de ruído e não devem se mover quando várias imagens são adquiridas da mesma cena;
- Precisão. Os cantos devem ser detectados o mais próximo possível das posições corretas;
- Rapidez. O detector de canto deve ser rápido o suficiente.

De acordo com o algoritmo FAST, o candidato é um canto se existir um conjunto de n *pixels* contíguos no círculo que são mais brilhantes do que a intensidade do *pixel* candidato mais um limiar, ou mais escuro do que a intensidade do *pixel* candidato menos o limiar, conforme mostrado na Figura 5. Para aumentar a velocidade do algoritmo FAST, é utilizada uma função de resposta de canto, que retorna o valor numérico de “cantos” de um ponto baseado na intensidade da vizinhança local da imagem (JOSHI; SINHA, 2013). Os cantos são detectados como máximos locais para a função. Após a detecção, a correspondência de ponto de canto é realizada para cada par de quadros. Às vezes, um algoritmo *Bag-of-Words* (*BoW*) é usado para representar cada imagem como um conjunto de descritores de canto para acelerar o processo de correspondência (BOTTERILL; MILLS; GREEN, 2010). Dessa forma, as matrizes de homografia podem ser calculadas e, por fim, as imagens são projetadas em uma coordenada comum para obter o mosaico final.

Devido a sua estrutura simples, o cálculo deste método é menos complexo do que os métodos gerais de detecção de cantos (JIAO; ZHAO; WU, 2011).

Figura 5 – Detecção de características do ponto candidato usando o algoritmo FAST com 16 *pixels* no círculo.



Fonte: (ROSTEN; REITMAYR; DRUMMOND, 2005)

2.1.1.5 Scale-Invariant Feature Transform - SIFT

O algoritmo SIFT foi proposto em 1999 por David G. Lowe (LOWE, 1999), é um algoritmo que detecta características de cada uma das imagens criando um conjunto de descritores de características, e cada um destes descritores são invariantes a escala, rotação e parcialmente invariante a alterações de iluminação e pontos de vista (LOWE, 2004).

As características proporcionadas pelo SIFT são bem localizadas tanto no domínio espacial como no domínio de frequência, limitando a probabilidade de não ocorrer correspondências das características por oclusão e ruído. A obtenção dos descritores é feita através de quatro etapas principais, onde as duas primeiras descrevem a parte do **detector** e as duas seguintes a formação do **descritor** (LOWE, 2004):

1. **Detecção e seleção de extremos em um espaço de escalas:** É feita a procura por todas as escalas e posições de uma imagem. Para isto é utilizada uma função conhecida como Diferença de Gaussianas, *DoG - Difference of Gaussian*, para identificação dos potenciais pontos de interesse, que são invariantes a escala e orientação. Esta é a etapa mais custosa computacionalmente.
2. **Localização de pontos chave:** Para cada local candidato, em que foi detectado um extremo (máximo ou mínimo), um modelo detalhado é ajustado para se determinar a localização e escala. Pontos chave, ou pontos de interesse, são selecionados baseando-se em suas medidas de estabilidade.
3. **Definição da orientação:** Uma ou mais orientações são atribuídas para cada ponto chave localizado, baseadas em direções do gradiente. Todas as operações posteriores são realizadas sobre os dados da imagem que foram considerados ponto chave e que foram transformados em relação à orientação, escala e localização, proporcionando invariância a estas transformações.

4. **Descritor para os pontos chave:** Os gradientes da imagem são mensurados na escala selecionada, na região ao redor de cada ponto chave, sendo criados histogramas de orientações para compor o descritor.

Etapa do Detector

O **detector** busca pontos invariantes à mudança de escala, possibilitando a detecção de pontos chave em vários níveis de aproximação do objeto de interesse. Dessa forma, procura por características estáveis em relação à escala aplicando uma função Gaussiana. Assim, o espaço escalar de uma imagem, definido como uma função $L(x, y, \sigma)$, é obtido a partir de uma operação de convolução entre a Gaussiana de escala variável com uma imagem de entrada $I(x, y)$ (LOWE, 1999):

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad (2.9)$$

em que $*$ é a operação de convolução em x e y , e

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.10)$$

corresponde a um filtro Gaussiano com um determinado valor σ .

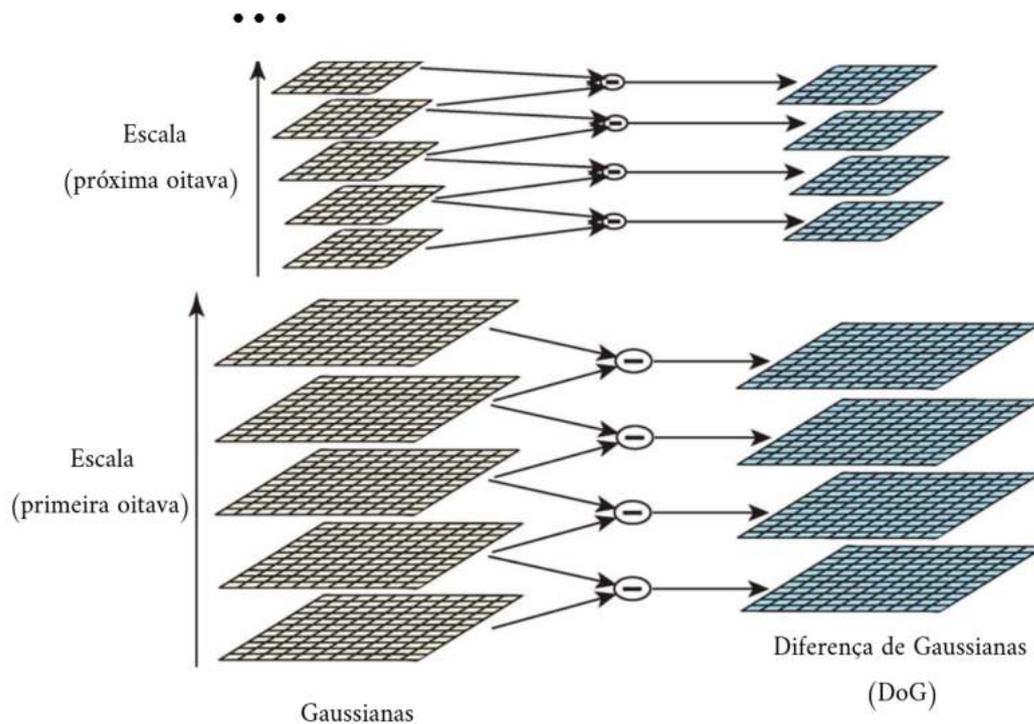
Logo após a convolução da imagem com diferentes valores para σ , torna-se possível detectar pontos de máximos/mínimos usando a *DoG*, que pode ser convoluída com uma imagem de entrada $I(x, y)$. A *DoG* é calculada como a diferença entre duas imagens filtradas (G) com escalas próximas (separadas por uma constante multiplicativa k):

$$DoG(x, y, \sigma) = (G(x, y, k \times \sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k \times \sigma) - L(x, y, \sigma). \quad (2.11)$$

Esta convolução é a diferença entre imagens suavizadas por um filtro Gaussiano em escalas σ e $k \times \sigma$. A função *DoG* é capaz de detectar variações de intensidade na imagem, como contornos, eliminar detalhes indesejados e realçar características fortes. Variando-se o σ , é possível encontrar características em diferentes escalas. O filtro *DoG* é uma aproximação do Laplaciano do Gaussiano (do inglês, *Laplacian of Gaussian - LoG*), e o *LoG* é um filtro que, ao ser aplicado em uma imagem, facilita o encontro de pontos notáveis (SOUZA, 2015). Como pode ser observado na [Figura 6](#).

As imagens resultantes do filtro Gaussiano são agrupadas por oitavas, em que cada oitava corresponde ao dobro do valor de σ , o parâmetro k é selecionado com o objetivo de atingir um número fixo de imagens para cada oitava. Este método assegura um número fixo de *DoG*'s por oitava. O autor do método SIFT, (LOWE, 1999), recomenda que a quantidade de borramentos e oitavas sejam, respectivamente 4 e 5. Os respectivos pontos de interesse são

Figura 6 – Construção das imagens resultantes da diferença de Gaussianas *DoG* em cada oitava no espaço de escala.



Fonte: Adaptada de Lowe (2004).

identificados nas regiões de máximos ou mínimos das imagens DoG em todas as escalas, cada um dos pontos é comparado com seus 8-vizinhos na mesma escala e com seus 9-vizinhos nas escalas adjacentes. Se este ponto for um máximo ou mínimo local, logo, ele é selecionado com um **ponto de interesse candidato** (RAMOS, 2016).

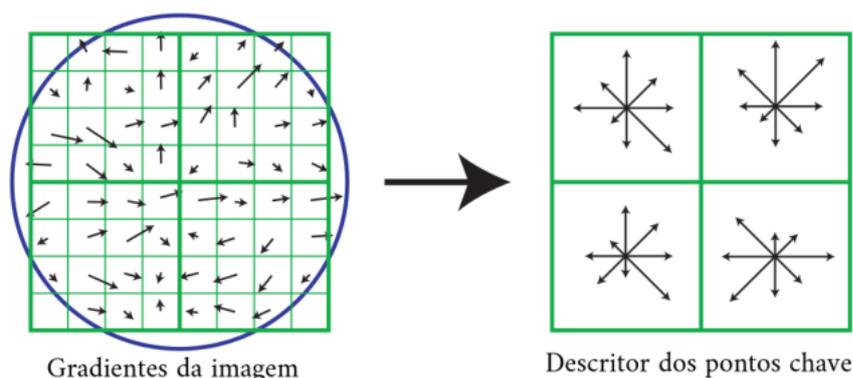
Etapas do Descritor

Logo após a identificação dos pontos de interesse, têm-se os valores das coordenadas para a representação de cada um destes pontos. Sendo assim, é possível descrever a região de cada ponto de interesse extraindo as características.

O **descritor** é construído quando uma orientação é atribuída para cada ponto de interesse. É criado um histograma das orientações para uma região vizinha ao redor deste ponto e cada ponto da sua vizinhança é adicionado ao histograma com um determinado peso. Os picos no histograma de orientações correspondem às direções dominantes dos gradientes locais e são utilizados para definir a orientação do ponto. Desta forma, cada ponto de interesse tem quatro dimensões, ou seja, sua posição (x, y) , que corresponde à magnitude e orientação. O descritor é criado calculando-se as magnitudes e orientações dos gradientes amostrados (Figura 7) ao redor da localização do ponto de interesse.

O descritor final do ponto de interesse é mostrado na Figura 7 à direita. Como pode

Figura 7 – O descritor para cada ponto de interesse ou ponto chave é feito a partir da magnitude do gradiente e a orientação ao seu redor. Esses pontos ao redor são ponderados por uma janela Gaussiana, como pode ser visto no círculo à esquerda. Então são montados histogramas de orientação que resumem o conteúdo em janelas 4×4 , como mostra à direita, onde o comprimento de cada seta corresponde à soma das magnitudes do gradiente. Nesta figura, é possível observar uma matriz de descritores 2×2 calculados a partir de uma janela 8×8 . Entretanto, o algoritmo SIFT, na prática, é implementado em uma janela 16×16 que resulta em um descritor 4×4 .



Fonte: Adaptada de [Lowe \(2004\)](#).

ser observado, ocorre que têm-se oito direções para cada janela 4×4 . O algoritmo SIFT usa uma região 16×16 para constituir um descritor 4×4 , sendo que cada janela é formada por oito orientações de gradiente. Conseqüentemente, considera-se $4 \times 4 \times 8 = 128$ características resultantes.

2.1.1.6 Speeded Up Robust Features - SURF

O SURF é um detector de característica local invariante a escala e rotação, e assim como o SIFT, esse algoritmo também se baseia em valores de espaço em escala, porém, possui um custo computacional menor, pois aproxima o *LoG* com Filtros de Caixa ou *Box Filter*, o que possibilita o uso completo da imagem ([BAY; TUYTELAARS; GOOL, 2008](#)). Os Filtros de Caixa também são conhecidos por *Summed Area Table - (SAT)* ([CROW, 1984](#)).

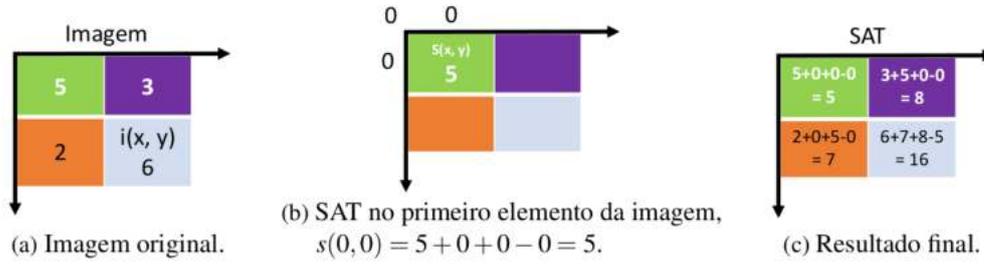
Etapa do Detector

A matriz *SAT* foi desenvolvida em 1984, porém somente no trabalho de [Viola e Jones \(2001\)](#) foi aplicada na área de visão computacional, no processo de detecção de faces. O cálculo para a construção da *SAT* de uma imagem para um *pixel* (x, y) é:

$$s(x, y) = i(x, y) + s(x - 1, y) + s(x, y - 1) - s(x - 1, y - 1), \quad (2.12)$$

onde i representa a imagem original e s a matriz *SAT*, que é a soma dos valores de linha e coluna acima do ponto (x, y) subtraído por este mesmo valor no ponto antecedente $s(x - 1, y - 1)$. O trabalho de [Ramos \(2016\)](#) exemplifica na [Figura 8](#) a [Equação 2.12](#):

Figura 8 – Passos para a construção da matriz SAT. Índices fora da fronteira são considerados zero.

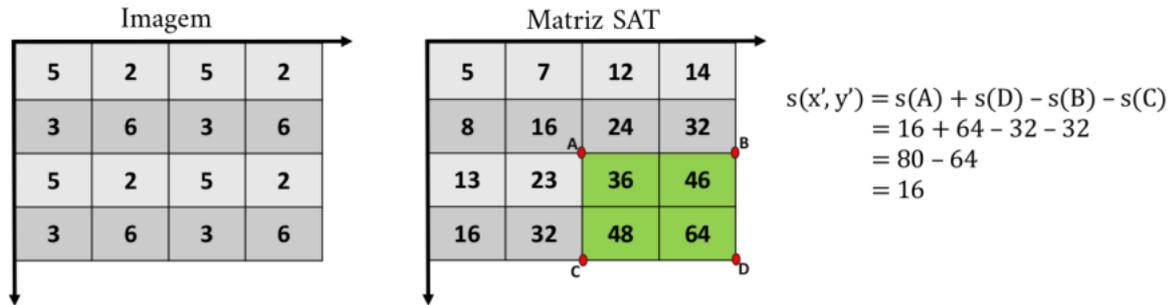


Fonte: (RAMOS, 2016)

Conforme os quatro pontos que formam o retângulo ($ABCD$), o cálculo da soma daquela região, denotado por $s(x',y')$, pode ser efetuado empregando a Equação 2.13, como pode ser visto na Figura 9.

$$s(x',y') = s(A) + s(D) - s(B) - s(C). \tag{2.13}$$

Figura 9 – Aplicação da Equação 2.13 acima para o cálculo da imagem integral em um retângulo $ABCD$.



Fonte: Adaptada de Ramos (2016).

Seguindo as etapas descritas até o momento, a matriz SAT pode, desta forma, ser utilizada para calcular, aproximadamente, as derivadas parciais de segunda ordem, os quais o algoritmo SURF usa para detectar os pontos de interesse. Consequentemente, torna-se possível calcular a matriz Hessiana:

$$H(x,y,\sigma) = \begin{pmatrix} L_{xx}(x,y,\sigma) & L_{xy}(x,y,\sigma) \\ L_{xy}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{pmatrix}, \tag{2.14}$$

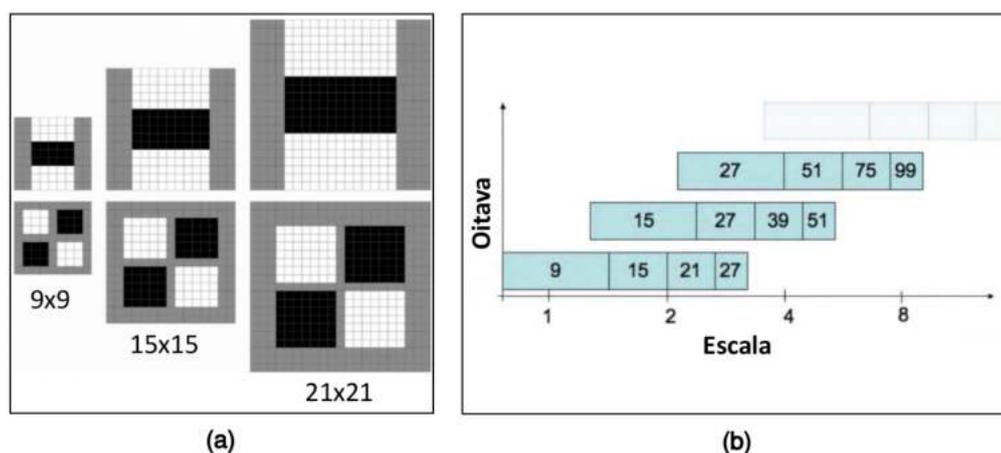
onde $L_{xx}(x,y,\sigma)$, $L_{yy}(x,y,\sigma)$, e $L_{xy}(x,y,\sigma)$ são convoluções de I no ponto (x,y) com filtros Gaussianos de segunda ordem $\frac{\partial^2}{\partial x^2}G(x,y,\sigma)$, $\frac{\partial^2}{\partial y^2}G(x,y,\sigma)$ e $\frac{\partial^2}{\partial x \partial y}G(x,y,\sigma)$, respectivamente. Os pontos de interesse são detectados usando o determinante da matriz Hessiana:

$$\det(H_{aprox}) = D_{xx}D_{yy} - (wD_{xy})^2, \tag{2.15}$$

se o valor do determinante for maior que 0, então existe um extremo, ou seja, um ponto de interesse. O w é o peso que corrige a aproximação do Gaussiano.

No processo de detecção de pontos de interesse não é realizada a minimização/maximização da imagem, e sim incremento/decremento do filtro de Caixa, conforme pode ser visto na [Figura 10a](#). Os autores do algoritmo SURF ([BAY; TUYTELAARS; GOOL, 2006](#)) recomendam o uso de três oitavas, tendo uma quarta como alternativa opcional. São aplicados quatro tamanhos diferenciados de filtros por oitava, conforme a [Figura 10b](#). O filtro de tamanho 9×9 , corresponde a um filtro Gaussiano com $\sigma = 1.2$, se o filtro aumentar, o valor do σ também aumenta. O tamanho de um filtro para outro deve ter no mínimo seis de diferença, dessa forma é possível manter a estrutura do filtro ([BAY; TUYTELAARS; GOOL, 2008](#)).

Figura 10 – Incremento do tamanho do filtro de Caixa. Filtro D_{yy} (em cima) e D_{xy} (em baixo) para dois níveis de escalas (9×9) e (15×15).



Fonte: Adaptada de [Bay, Tuytelaars e Gool \(2006\)](#).

Os pontos de interesse são calculados comparando seus 8-vizinhos mais próximos na escala original e seus 18 vizinhos em escalas adjacentes, totalizando 26 vizinhos. No SURF, um valor de limiar é definido, pontos que possuem valores menores que o limiar são descartados e pontos maiores são considerados. Sendo assim, quanto maior o valor do limiar, menos pontos de interesse são detectados, porém, esses pontos possuem maior destaque visual. Quanto menor o valor do limiar, maior a quantidade de pontos detectados. O valor do limiar é definido pelo usuário ([RAMOS, 2016](#)).

Etapa do Descritor

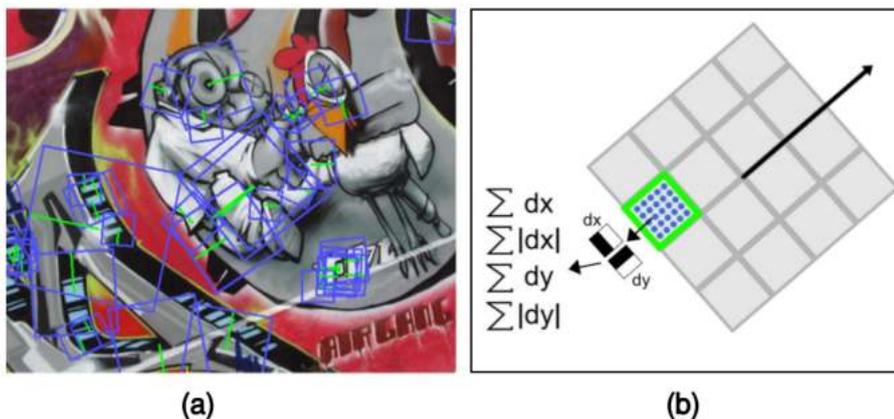
A orientação para cada ponto de interesse é calculada inicialmente em uma janela circular. Em seguida, uma janela quadrada é construída de acordo com a orientação. Assim, o extrator de características do SURF pode ser aplicado sobre esta janela quadrada ([RAMOS, 2016](#)).

O cálculo da orientação pode ser feito utilizando as respostas de *wavelet* de Haar de tamanho 4σ nas direções x e y , em um raio 6σ nos vizinhos ao redor do ponto de interesse,

sendo σ equivalente a escala na qual o ponto de interesse foi detectado (BAY; TUYTELAARS; GOOL, 2006). Após a resposta *wavelet* ser calculada, ela é ponderada por uma função Gaussiana 2.5σ , no ponto de interesse. Logo após isso, as respostas são representadas como vetores no plano cartesiano. A orientação dominante é presumida pelo cálculo da soma de todas as respostas dentro de uma janela de ângulo $\frac{\pi}{3}$, que é movida incrementalmente até cobrir todos os ângulos. Logo, a soma das duas respostas gera um novo vetor, e este é somado dentro do intervalo da janela que apresentar maior valor, sendo escolhido como orientação dominante (RAMOS, 2016).

Após a orientação de cada ponto ser encontrada, as características desse ponto podem ser extraídas. Primeiramente é construída uma janela quadrada de tamanho 20σ , sendo σ referente à escala na qual o ponto foi encontrado, de acordo com a orientação do ponto, como mostra a Figura 11a.

Figura 11 – Exemplos de pontos de interesse detectados pelo SURF e extração de características.



Fonte: Adaptada de Bay, Tuytelaars e Gool (2006).

Esta região é dividida em 4×4 sub-regiões quadradas, o que totaliza 16 regiões. Em cada uma dessas regiões são calculadas as respostas de *Haar wavelets* de tamanho 2σ , caracterizando as respostas de *wavelets* na direção x e y , como respectivamente dx e dy , conforme a Equação 2.16. Na Figura 11b pode-se observar um exemplo do processo descrito.

$$v = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy| \right]. \quad (2.16)$$

Mesmo que haja variações do SURF, o descritor padrão consiste em um vetor com $4 \times 16 = 64$ características. De acordo com Bay, Tuytelaars e Gool (2006), embora o SURF seja semelhante ao SIFT, ele é menos sensível ao ruído; outra vantagem é o tempo de processamento do SURF que se dá pelo fato de seu descritor ter apenas 64 dimensões, o que reduz o tempo computacional.

Em geral, as técnicas de mosaico baseadas em SURF são mais rápidas do que as técnicas baseadas em SIFT, porém, apresentam algumas limitações quando ocorrem, por exemplo, certas variações de cor ou iluminação.

2.1.1.7 Contorno

Estes algoritmos de mosaico de imagens baseiam-se na extração de características de alto nível, por exemplo, formas ou texturas em uma imagem. A extração de formas implica encontrar sua posição, orientação e tamanho (ISLAM; KABIR, 2013).

Normalmente, regiões de estruturas diferentes são extraídas como características de alto nível das imagens, essas características são combinadas para encontrar correspondências, que posteriormente serão usadas para calcular os parâmetros de transformação. Em seguida, técnicas diferentes são utilizadas para eliminar falsas correspondências. E finalmente, as imagens são redimensionadas e misturadas para gerar a saída do mosaico. Fazer uso das características de alto nível aumenta significativamente os cálculos para geração do mosaico. No entanto, eles são particularmente adequados para trabalhar em parâmetros de movimentos maiores e complexos (GHOSH; KAABOUC, 2016).

Algumas contribuições na área de extração de características de alto nível para geração de mosaico podem ser encontradas nos trabalhos de Xiao, Zhang e Shah (2005), Prescott *et al.* (2006) e Deshmukh e Bhosle (2011). No trabalho de Prescott *et al.* (2006), os autores apresentam um novo método para registro automático de um grande conjunto de imagens microscópicas combinando automaticamente as características de alto nível (por exemplo, regiões, pontos de referência e objetos) através da busca de estruturas cíclicas em um grafo. Em Xiao, Zhang e Shah (2005), os autores usaram um algoritmo de linha de base amplo juntamente com um método de expansão de região adaptativa para obter um registro robusto usando recursos de alto nível. Deshmukh e Bhosle (2011) propõem uma nova técnica de registro baseada em extração das bordas da imagem.

2.1.2 Domínio de Frequência

Ao contrário dos algoritmos de mosaico de imagens baseadas em domínio espacial, os métodos classificados nesta categoria requerem cálculo no domínio da frequência para encontrar os parâmetros de transformação ótimos entre um par de imagens. Esses algoritmos usam a propriedade da **correlação de fase** para registrar imagens (GHOSH; KAABOUC, 2016). Os métodos de Fourier diferem de outras estratégias de registro porque buscam a correspondência ideal de acordo com a informação no domínio da frequência (REDDY; CHATTERJI, 1996).

Se for necessária uma aceleração da velocidade computacional ou se as imagens foram adquiridas em condições variáveis ou estiverem corrompidas por ruído dependente da frequência, então os métodos de Fourier são preferidos, pois, eles exploram a representação de Fourier das imagens no domínio da frequência. O método de **correlação de fase** é baseado no Teorema de Deslocamento de Fourier (BRACEWELL, 1965) e foi originalmente proposto para o registro de imagens transladas. Ele calcula o espectro de potência cruzada das imagens detectadas e de referência e procura a localização do pico em sua inversa. O método mostra forte robustez contra

ruídos dependentes da frequência, não uniformidade e alterações na iluminação com o tempo. A economia de tempo computacional é mais significativa quando as imagens, que devem ser registradas, são grandes. (ZITOVÁ; FLUSSER, 2003).

Basicamente, calcula-se o produto espectral cruzado entre a imagem de referência e a imagem a ser alinhada e procura-se pela localização de picos na matriz resultante no domínio do espaço (PRATT, 2001). Conforme Pratt (2001), considere a imagem $F_1(x, y)$ (imagem de referência) e $F_2(x, y)$ (imagem a ser alinhada), sendo que $F_2(x, y)$ é a imagem $F_1(x, y)$ translada por um *offset* de (x_0, y_0) em relação a (x, y) (Equação 2.17):

$$F_2(x, y) = F_1(x - x_0, y - y_0). \quad (2.17)$$

A propriedade do deslocamento de Fourier afirma que um deslocamento da imagem do espaço resulta em um deslocamento na fase do sinal na frequência (Equação 2.18):

$$F(x - a, y - b) \leftrightarrow \mathcal{F}(\omega_x, \omega_y) e^{-i(\omega_x a + \omega_y b)}. \quad (2.18)$$

Dessa forma, temos no domínio da frequência (Equação 2.19):

$$\mathcal{F}_2(\omega_x, \omega_y) = \mathcal{F}_1(\omega_x, \omega_y) e^{-i(\omega_x x_0 + \omega_y y_0)}. \quad (2.19)$$

O fator de deslocamento de fase exponencial pode ser computado pelo produto espectral cruzado, que é dado por (Equação 2.20):

$$G(\omega_x, \omega_y) = \frac{\mathcal{F}_1(\omega_x, \omega_y) \mathcal{F}_2^*(\omega_x, \omega_y)}{|\mathcal{F}_1(\omega_x, \omega_y) \mathcal{F}_2^*(\omega_x, \omega_y)|} = e^{i(\omega_x x_0 + \omega_y y_0)}. \quad (2.20)$$

Temos que $\mathcal{F}_2^*(\omega_x, \omega_y)$ é o complexo conjugado de $\mathcal{F}_2(\omega_x, \omega_y)$. A normalização do denominador nesta equação é equivalente a um pré-clareamento dos sinais, tornando o método de correlação de fase intrinsecamente robusto ao ruído que está correlacionado com as imagens, como variações uniformes de iluminação ou compensações em intensidade média (HURTÓS *et al.*, 2012). O resultado obtido apresenta o deslocamento exato entre as duas imagens, possibilitando a correção das coordenadas da segunda imagem, com relação à primeira (imagem de referência) (BANDEIRA, 2005).

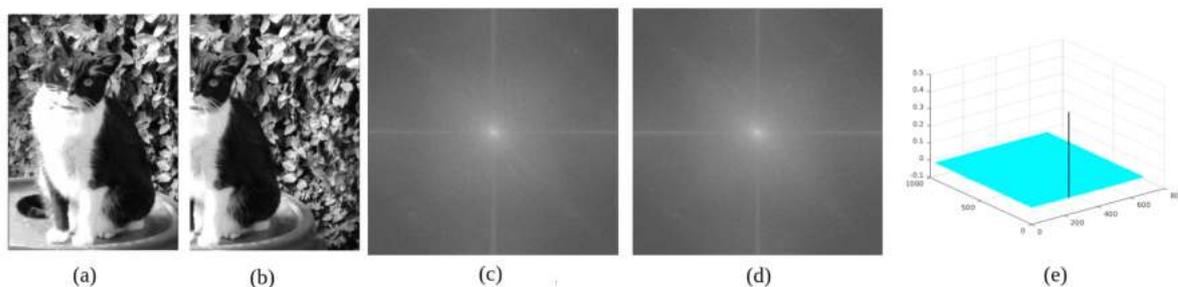
Em seguida, fazendo a transformada inversa de Fourier do produto espectral cruzado, chega-se na matriz de *offset* espacial (Equação 2.21):

$$G(x, y) = \delta(x - x_0, y - y_0). \quad (2.21)$$

Com a matriz do produto espectral no espaço (x, y) basta localizar o ponto máximo da matriz, que caracteriza o pico de correlação (STUCHI, 2013). A função $\delta(x - x_0, y - y_0)$ é

aproximadamente zero em todos os lugares exceto no deslocamento (x_0, y_0) que é necessário para registrar otimamente as duas imagens, como mostra a [Figura 12](#).

Figura 12 – Uso do espectro de potência cruzada para detectar as transformações das imagens (a) e (b), e o deslocamento entre elas. (c) é o espectro de (a); (d) é o espectro de (b); (e) é a função que indica o deslocamento entre as imagens de origem.



Fonte: Elaborada pela autora.

Na presença de ruído ou outras perturbações nas imagens, o resultado da correlação de fase será pouco satisfatório, mas enquanto contiver um pico dominante, os *offsets* podem ser recuperados ([HURTÓS *et al.*, 2012](#)).

Com os parâmetros de deslocamento (translacional), as duas imagens são ajustadas e finalmente unidas para obter um mosaico. Os algoritmos de mosaico baseados nesta técnica geralmente são eficientes devido ao uso da propriedade da transformação de Fourier e do uso da Transformada Rápida de Fourier (FFT). Além disso, para um registro preciso, muitas vezes é necessário uma sobreposição significativa entre as imagens de origem ([GHOSH; KAABOUCH, 2016](#)).

O algoritmo de correlação de fase, conforme especificado, é robusto em relação a ruídos do domínio de frequência, variações de iluminação e possui rápido processamento no domínio da frequência. No entanto, ele não consegue, inicialmente, detectar variações de escala e rotação da imagem, apenas de translação. Para resolver esse problema, o algoritmo de correlação de fase pode ser estendido para a correção de escala e rotação, utilizando coordenadas polares, conforme descrito em [Pratt \(2001\)](#) e [Stuchi \(2013\)](#). Essa transformação log-polar também é utilizada em [Yang *et al.* \(2012\)](#) para encontrar a escala e os parâmetros de translação da imagem. Outro método de extensão da correlação de fase é apresentado em [Wang, Cheng e Zhao \(2009\)](#), onde os autores sugeriram mudar os parâmetros de rotação e escala para parâmetros de translação usando a transformada de Fourier-Mellin.

2.2 Comparativo entre as técnicas de mosaicos

O [Quadro 1](#) apresenta uma visão comparativa das diferentes categorias de algoritmos de mosaico com base no registro de imagens, baseado no trabalho de [Ghosh e Kaabouch \(2016\)](#), e que foram abordadas neste trabalho.

Quadro 1 – Comparação entre os diferentes métodos de mosaico de imagens baseados em registro.

Método	Vantagens	Desvantagens
NCC	Não é necessária uma análise estrutural de alto nível e pode ser aplicada diretamente aos dados da imagem.	Boa apenas para imagens com grande sobreposição.
MI	Bons resultados para a análise multimodal e menos sensível às mudanças de iluminação e oclusão.	Processamento lento e causa erro de registro quando as imagens têm sobreposição pequena.
Harris	Cálculo rápido e preciso.	Precisa de conhecimento prévio do tamanho da janela e é bom apenas para mudanças moderadas na escala e na rotação.
FAST	Cálculo rápido e preciso.	Não é robusto quando está presente um alto nível de ruído, e é necessário um conhecimento prévio sobre limiar.
SIFT	Eficiente para imagens de alta resolução e oferece invariância a várias transformações.	Computacionalmente caro.
SURF	Cálculo rápido, bom para aplicações em tempo real.	Frac desempenho sob certas transformações (por exemplo, cor e iluminação).
Contorno	Eficiente quando envolve movimentos grandes e complicados.	Computacionalmente caro pois usa recursos de alto nível.
Domínio de Frequência	Eficaz pois utiliza a FFT.	Excessivamente sensível ao ruído e sua precisão depende de uma grande área de sobreposição.

Fonte: Adaptada de Ghosh e Kaabouch (2016).

Na literatura podem ser encontrados diversos algoritmos que utilizam descritores e fazem correspondências entre as imagens, entretanto grande parte dos trabalhos voltados para a construção de mosaicos de imagens utilizam o SIFT ou o SURF por gerarem melhores resultados. Nos trabalhos de Bae, Cho e Durand (2009), Li e Geng (2010), Kaynig *et al.* (2010), Liu *et al.* (2011), Min, Jiguo e Xusheng (2012), Song e Lu (2013) e Gui, Su e Du (2013) os autores mostram exemplos de construção de mosaicos baseados nos métodos SIFT e SURF.

Os artigos de Mikolajczyk e Schmid (2005), Bay, Tuytelaars e Gool (2008), Luo e Oubong (2009), Yang *et al.* (2012), Khan, McCane e Wyvill (2011), Bai, Ning e Wang (2012) e Ali e Hussain (2012) fazem uma comparação do SIFT com outras metodologias e concluem que o SIFT é melhor na extração de características, o que contribui para obtenção de uma boa precisão dos mosaicos, embora apresente um tempo de processamento mais lento em relação às demais metodologias (MACHADO; GONZAGA; BOAVENTURA, 2019). Entretanto, os autores do SURF (BAY; TUYTELAARS; GOOL, 2006), afirmam que o seu algoritmo é superior ao SIFT em tempo de processamento e reconhecimento de objetos, considerando-se a qualidade dos pontos de características (TARALLO, 2013). O trabalho de Tarallo (2013) apresenta um resumo dos artigos da literatura de mosaicos e comparações entre os dois métodos.

2.3 Considerações finais

Os métodos de mosaico apresentados neste capítulo são muito importantes na área de visão computacional e seu êxito depende principalmente dos métodos de registro e mistura. Entretanto, com o advento de novos métodos de mosaico torna-se difícil escolher um algoritmo que seja adequado para um propósito específico. Dessa forma, conforme o trabalho que se deseja realizar, deve-se escolher um método mais conveniente para a aplicação.

Considerando-se os métodos de registro de imagens apresentados e devido ao tipo de deslocamento encontrado entre as imagens deste trabalho, apenas movimentos de translação, decidiu-se pelo uso da técnica de correlação de fase, por fazer uso da FFT, que é invariante à translação. Assumimos que no processo de captura das imagens no microscópio todas elas se sobrepõem e que estão em um plano sem pontos de vista diferentes, ou seja, é um ambiente controlado, dessa forma não será necessário aplicar técnicas para correção de alterações espaciais (translação, escala, rotação, cor e brilho). Assim, a abordagem baseada em Fourier torna-se mais eficiente e menos custosa computacionalmente. À vista disso, com o intuito de aprimorar as principais técnicas de mosaico existentes (SIFT, SURF e Correlação de fase) uma nova abordagem fundamentada em filtros anisotrópicos é proposta e detalhada no [Capítulo 3](#).

TÉCNICAS PARA APRIMORAR OS MÉTODOS DE MOSAICO

Grande parte dos trabalhos na literatura relacionados à construção de mosaicos utilizam os descritores SIFT e SURF juntamente com o algoritmo RANSAC, que calcula a matriz homográfica e realiza a junção entre as imagens, pois geram melhores resultados. Os métodos de mosaico apresentados no [Capítulo 2](#) normalmente possuem bons resultados dependendo da aplicação a que se destinam. Neste trabalho, foram empregadas imagens de microscópio óptico (de luz) para avaliar os métodos de mosaicos. A captura de imagens é realizada de forma controlada, sendo assim, todas se sobrepõem e estão em um plano sem diferentes pontos de vista. Dessa forma, optou-se pelo emprego do método do domínio de frequência, descrito na [Subseção 2.1.2](#), pois o mesmo faz uso da correlação de fase, que é invariante à translação e utiliza a Transformada Rápida de Fourier (FFT).

O objetivo deste trabalho é analisar os principais métodos de mosaico de imagens e propor uma metodologia capaz de aprimorar essas técnicas, para isso, foram aplicados diversos tipos de ruídos e filtros nas imagens com o intuito de encontrar a melhor filtragem para as imagens de entrada.

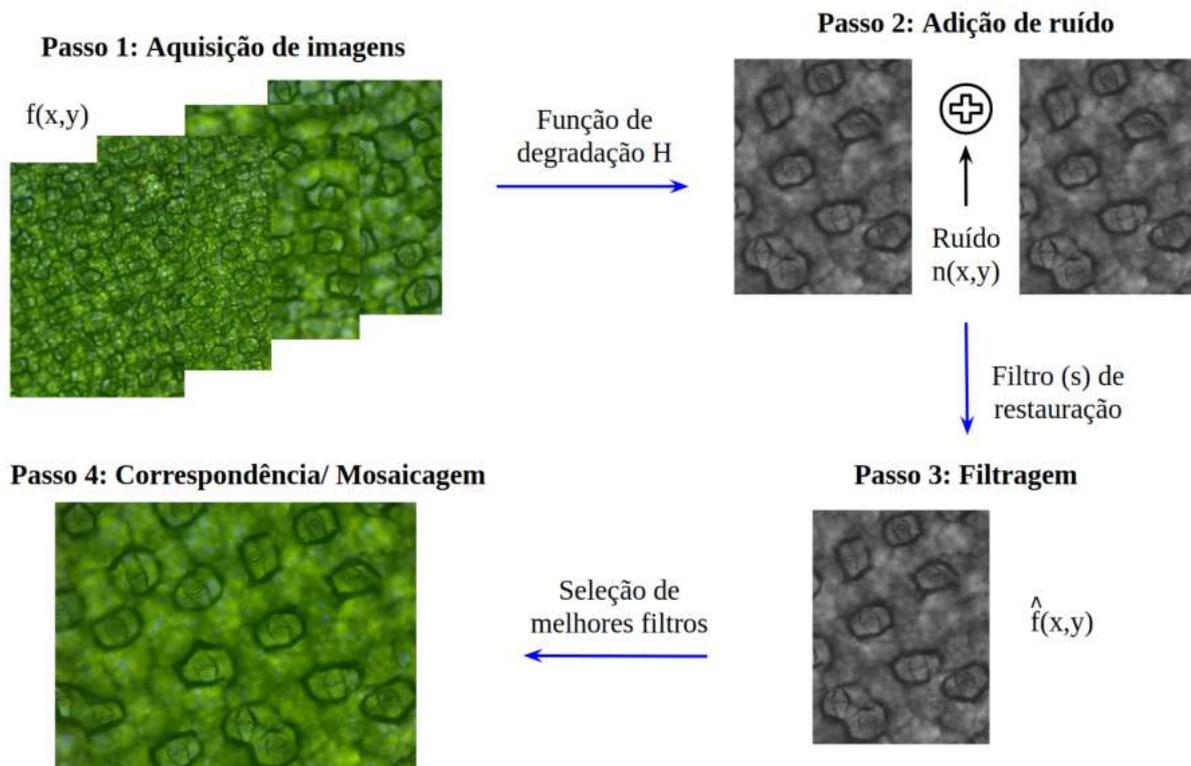
De maneira simplificada, a abordagem é construída por quatro passos principais:

- **Passo 1 - Aquisição de imagens:** são descritos os passos necessários para obter as imagens microscópicas e o hardware utilizado. Nesta primeira etapa realiza-se o corte das imagens originais capturadas, com sobreposição de 12.5% entre o par;
- **Passo 2 - Adição de ruído:** diversos ruídos são adicionados ao segundo corte da imagem. O propósito é corromper as imagens além dos padrões pré-definidos e encontrar o ruído mais problemático;

- **Passo 3 - Filtragem:** as imagens corrompidas são submetidas a vários métodos de filtragem visando constatar o melhor filtro para cada tipo de ruído;
- **Passo 4 - Correspondência/Mosaicagem:** nesta etapa final são empregados os principais métodos de mosaico para averiguar se a junção entre as imagens se deu de maneira correta.

A Figura 13 resume graficamente a metodologia utilizada no trabalho.

Figura 13 – Principais passos da metodologia proposta.



Fonte: Elaborada pela autora.

3.1 Passo 1: Aquisição de imagens

A aquisição de imagens foi efetuada através do software AxioVision 40 v.4.8.2.0 Carl Zeiss MicroImaging GmbH (AXIOVISION, 2012). O hardware utilizado é formado por uma câmera AxioCam acoplada ao microscópio e ligada a um microcomputador através de uma interface USB (Figura 14). O sistema de aquisição é constituído por:

- **Microscópio óptico:** Foi calibrado para manter as informações de foco, ampliação da objetiva ($5\times$ e $10\times$), zoom e posições dos eixos X e Y . Esses dados foram mantidos para serem reproduzidos nas demais capturas de imagens. O movimento da lâmina no plano cartesiano desse microscópio é mecânico e é acionado manualmente;

Figura 14 – Software AxioVision e microscópio conectado ao microcomputador através de interface USB.



Fonte: Elaborada pela autora.

- **Câmera AxioCamERc5s:** Captura imagens de 2560×1920 pixels com resolução de 8 bits RGB e gera um formato de arquivo de saída JPEG;
- **Microcomputador PC:** Possui um sistema operacional Windows 7 Professional 64bits. Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz, 3392 Mhz, 4 Core(s), 4GB(RAM);
- **Software AxioVision 4.8.2:** O AxioVision é um sistema modular de processamento e análise de imagens para microscopia moderna. A funcionalidade básica para aquisição de imagens e controle de microscópio, processamento de imagem e anotações, análise de imagem, documentação e configuração também pode ser expandida através da integração de módulos adicionais no sistema para tarefas específicas (AXIOVISION, 2012).

Logo após a captura das imagens microscópicas, estas imagens originais são submetidas a recortes com sobreposição de 12.5% entre o par. Esta etapa será melhor descrita na [Seção 4.1](#) que detalha as bases de imagens utilizadas no trabalho.

3.2 Passo 2: Adição de Ruído

A inserção de ruído baseia-se na ideia de que, sem condições perfeitas, não é possível garantir que as imagens estejam livres de qualquer tipo de perturbação, tais como borramento, brilho, cor, baixa iluminação, entre outras. Com o propósito de reduzir este efeito, são utilizados vários métodos de filtragem nas imagens com o objetivo de avaliar a robustez de cada filtro antes da criação do mosaico. Dessa forma, adiciona-se ruído ao segundo corte da imagem para perturbar sua informação durante o processo de correspondência ([Seção 3.4](#)). Para avaliar os métodos sob várias circunstâncias, são aplicados diferentes tipos de ruído à imagem. Uma vez que um método de mosaico é capaz de encontrar uma boa correspondência, mesmo que os dados

estejam corrompidos, este é considerado mais robusto. O propósito desta etapa da metodologia foi encontrar o ruído mais problemático e o filtro mais adequado para cada tipo de ruído.

Os valores de intensidade do ruído podem ser considerados variáveis aleatórias, caracterizadas por uma densidade de probabilidade, do inglês *Probability Density Function* - (*PDF*) (GONZALEZ; WOODS, 2010). As *PDFs* mais conhecidas são Uniforme, Gaussiano, Exponencial, Gama ou Erlang, Sal e Pimenta e Lognormal (GONZALEZ; WOODS; EDDINS, 2009). Entretanto, neste trabalho faz-se o uso de quatro ruídos aditivos, Sal e Pimenta, Gaussiano, Poisson e Multiplicativo ou *Speckle*, que são basicamente ruídos que costumam afetar imagens, e que serão detalhados nas seções seguintes.

3.2.1 Ruído Sal e Pimenta ou Impulsivo

O ruído “sal e pimenta” ou *S&P*, é modelado como sendo aditivo, independente do sinal e composto de variáveis aleatórias independentes e identicamente distribuídas (COSTA, 2009). De acordo com (GONZALEZ; WOODS; EDDINS, 2009), a *PDF* do ruído *S&P* é dada por

$$p(z) = \begin{cases} P_a & \text{para } z = a \\ P_b & \text{para } z = b \\ 0 & \text{se não.} \end{cases} \quad (3.1)$$

Se $b > a$, então a intensidade b surge como um ponto claro na imagem, e o nível a aparecerá como ponto escuro. Se P_a ou P_b tiverem valor zero, o ruído impulsivo é chamado *unipolar*. Se nenhuma probabilidade for zero, os valores do ruído se assemelharão a grãos de sal e pimenta aleatoriamente distribuídos por toda a imagem. Para uma imagem com 8 bits significa que $a = 0$ (preto) e $b = 225$ (branco).

O ruído impulsivo é encontrado em situações nas quais transientes rápidos, como problemas de chaveamento, ocorrem durante a aquisição da imagem (GONZALEZ; WOODS, 2010). A Figura 15 mostra um exemplo do ruído sal e pimenta em imagens.

Basicamente, a imagem corrompida terá aspecto de pontos brancos (sal) e pontos pretos (pimenta).

3.2.2 Ruído Gaussiano ou Térmico

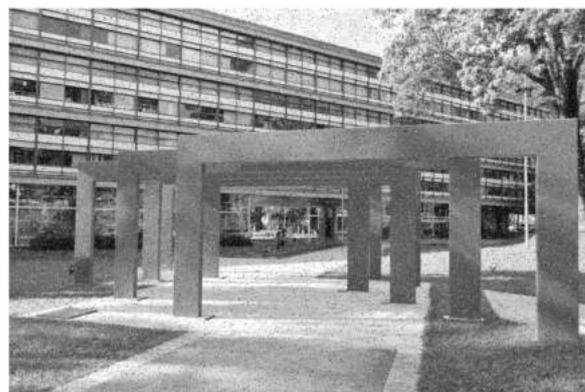
O ruído térmico é geralmente modelado como aditivo, independente do sinal e composto de variáveis aleatórias independentes e identicamente distribuídas (COSTA, 2009). Baseado em (GONZALEZ; WOODS; EDDINS, 2009), a *PDF* de uma variável aleatória Gaussiana, z é dada por

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}, \quad (3.2)$$

Figura 15 – (a) Imagem original. (b) Adição de 0.05% de ruído S&P.



(a) Imagem original



(b) Imagem corrompida

Fonte: Elaborada pela autora.

onde z representa a intensidade, \bar{z} é o valor médio de z e σ é o seu desvio padrão. O desvio padrão elevado ao quadrado, σ^2 , é chamado de variância de z . A Figura 16 mostra um exemplo da aplicação deste ruído.

Figura 16 – (a) Imagem original. (b) Imagem corrompida por ruído Gaussiano com $\bar{z} = 0$ e $\sigma^2 = 0.004$.

(a) Imagem original



(b) Imagem corrompida

Fonte: Elaborada pela autora.

O ruído Gaussiano surge em uma imagem em decorrência de fatores como, por exemplo, o ruído de circuitos eletrônicos e o ruído de sensores de imagem em virtude de iluminação insuficiente ou alta temperatura (GONZALEZ; WOODS, 2010).

3.2.3 Ruído Poisson ou Relativo à Contagem de Fótons

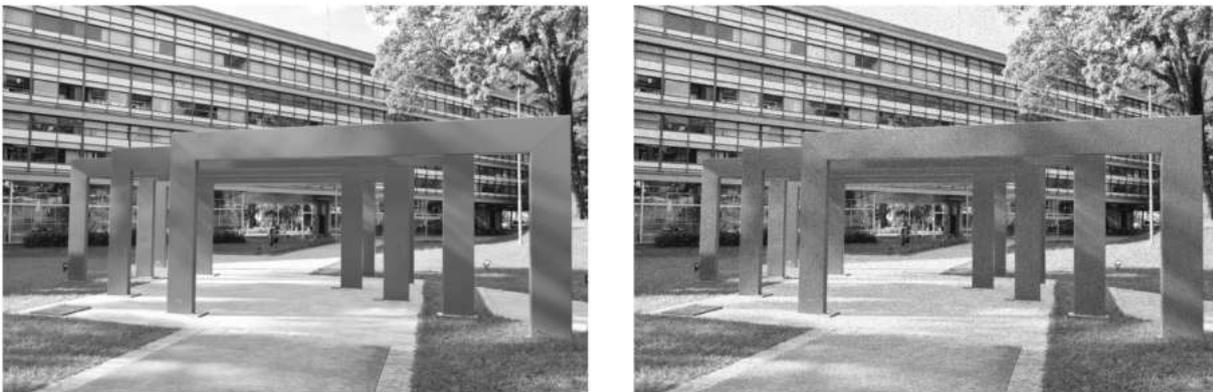
O ruído relativo à contagem de fótons é comumente modelado como sendo aditivo e dependente do sinal (COSTA, 2009). Este tipo de ruído depende dos dados da imagem e é caracterizado por ter a variância igual à média. Gera uma sequência de ruídos de números inteiros com a distribuição de probabilidade de Poisson da seguinte forma (ABOSHOSHA *et al.*,

2010):

$$P(X) = \frac{\mu^X}{X!} \cdot e^{-\mu}, \quad (3.3)$$

onde μ é o valor esperado. Este tipo de ruído é normalmente encontrado em dispositivos que dependem da contagem de fótons para obter uma imagem. Basicamente, quanto mais fótons contados em uma região, maior será o valor do *pixel*. Entretanto, devido à variação no número de fótons detectados, o valor do *pixel* é definido por um ruído cuja variância é proporcional à intensidade da luz que é capturada por esse *pixel* (COSTA, 2009). A Figura 17(b) abaixo mostra um exemplo da aplicação deste ruído para duas iterações, ou seja, a adição de ruído a partir dos dados de uma imagem anteriormente corrompida na primeira iteração.

Figura 17 – (a) Imagem original. (b) Imagem corrompida por ruído do tipo Poisson com duas iterações.



(a) Imagem original

(b) Imagem corrompida

Fonte: Elaborada pela autora.

3.2.4 Ruído Multiplicativo ou Speckle

O ruído multiplicativo ou *speckle* é definido como um ruído granular e proporcional ao nível de cinza local em uma imagem. O sinal e o ruído são estatisticamente independentes um do outro (PARIDA, 2014). De acordo com (AL-AMRI; KALYANKAR; KHAMITKAR, 2010), a distribuição do ruído multiplicativo é definida como,

$$J = I + n * 1, \quad (3.4)$$

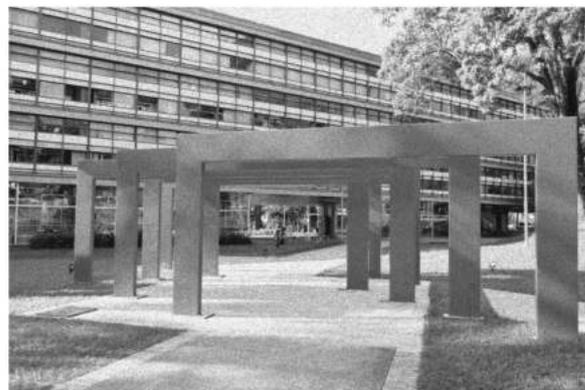
onde J é a imagem resultante, I é a imagem de entrada e n é o ruído aleatório uniformemente distribuído com $\bar{z} = 0$ e $\sigma^2 = v$.

A Figura 18(b) ilustra um exemplo de uma imagem corrompida por este tipo de ruído.

Figura 18 – (a) Imagem original. (b) Imagem corrompida por ruído do tipo multiplicativo com $\bar{z} = 0$ e $\sigma^2 = 0.02$.



(a) Imagem original



(b) Imagem corrompida

Fonte: Elaborada pela autora.

3.3 Passo 3: Filtragem

Na tentativa de melhorar a qualidade da imagem e reduzir o ruído para a correspondência de pontos no processo de construção do mosaico, este trabalho propõe a aplicação de vários métodos de difusão nas imagens antes do uso de métodos de mosaico. Sete métodos de difusão são projetados para suavizar regiões em uma imagem, entre isotrópicos e anisotrópicos, sendo que dois destes métodos são uma combinação do filtro da mediana com anisotrópicos.

3.3.1 Filtro da Média

O filtro da média é convolucional, linear e é considerado o filtro mais simples para a redução de ruído. Faz uso de uma janela deslizante $m \times n$ em que calcula a média dos valores dentro desta janela, o resultado substitui o *pixel* central da imagem (MARQUES, 2011). Sejam S_{xy} as coordenadas de uma janela $m \times n$, centrada em um ponto (x, y) , o filtro calcula o valor médio da imagem degradada $g(x, y)$ na área que foi definida por S_{xy} . A imagem restaurada \hat{f} no ponto (x, y) será a média aritmética dos valores dos *pixels* que estão dentro da janela definida por S_{xy} , conforme a equação abaixo (GONZALEZ; WOODS, 2010):

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t). \quad (3.5)$$

O fator de normalização é, em geral, igual à soma dos coeficientes da máscara, de modo a preservar o valor médio. Assim, a aplicação de filtros da média em uma região homogênea da imagem, ou seja, com níveis de cinza constantes, não sofrerá alteração de seus níveis de cinza (PEDRINI; SCHUWARTZ, 2007). Por ser um filtro passa-baixa, possui o efeito de suavização da imagem, visto que as frequências altas que correspondem às transições abruptas são atenuadas.

A suavização tende também, pelo mesmo motivo, a minimizar o efeito do ruído em imagens. Entretanto, devido ao borramento causado pela filtragem passa-baixa, detalhes finos podem ser removidos da imagem (BANDEIRA, 2009).

3.3.2 Filtro da Mediana

Em situações onde o objetivo é remoção de ruídos em imagens, o filtro da média possui como desvantagem a incapacidade de manter bordas e detalhes finos da imagem. Para evitá-la, o filtro da mediana torna-se uma técnica alternativa. Nesta técnica, o nível de cinza do *pixel* central da janela é substituído pela mediana dos *pixels* situados em sua vizinhança, por exemplo, em uma janela $m \times n$ os valores dos *pixels* são ordenados e é selecionado o valor médio do conjunto (MARQUES, 2011). Sua fórmula matemática é como se segue (GONZALEZ; WOODS, 2010):

$$\hat{f}(x, y) = \text{mediana} \{g(s, t) | (s, t) \in S_{xy}\}. \quad (3.6)$$

Este método é não-linear e apresenta um desempenho de precisão particularmente bom em situações nas quais a imagem é degradada por ruído sal e pimenta. Nas ocasiões em que o ruído é do tipo Gaussiano, seu desempenho é apenas satisfatório se for comparado com o da pela média (FILHO; NETO, 1999).

3.3.3 Filtro de Wiener

O filtro de Wiener foi desenvolvido por Norbert Wiener em 1942 e é um dos primeiros e mais conhecidos métodos para restauração de imagens lineares. O objetivo é minimizar o erro quadrático médio (EQM) entre a saída do filtro e a imagem original ainda não degradada. O filtro considera a imagem e o ruído como processos aleatórios, com o propósito de encontrar uma estimativa \hat{f} da imagem não corrompida f , de modo que o EQM entre ambas seja minimizado. Esta medida do erro é dada por

$$e^2 = E \left\{ (f - \hat{f})^2 \right\}, \quad (3.7)$$

onde E é o valor esperado do argumento. Presume-se que o ruído e a imagem não estão correlacionados, que um dos dois tenha uma média 0 e que os níveis de cinza estimados são uma função linear dos níveis na imagem degradada (GONZALEZ; WOODS; EDDINS, 2009). A solução para esta equação no domínio de frequência é definida como

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)} \right] G(u, v), \quad (3.8)$$

onde,

$H(u, v)$ é a função de degradação;

$$|H(u, v)|^2 = H^*(u, v)H(u, v);$$

$H^*(u, v)$ é o complexo conjugado de $H(u, v)$;

$S_\eta(u, v) = |N(u, v)|^2$ é o produto espectral do ruído;

$S_f(u, v) = |F(u, v)|^2$ é o produto espectral da imagem não degradada.

A imagem restaurada no domínio espacial é dada pela estimativa da transformada inversa de Fourier. Note que, caso o ruído seja zero, o filtro de Wiener passa a ser o filtro inverso (GONZALEZ; WOODS; EDDINS, 2009).

3.3.4 Difusão Anisotrópica por Perona-Malik

O método proposto por (PERONA; MALIK, 1990) tem como objetivo gerar imagens em diferentes resoluções sem prejudicar as bordas destas imagens. Torna-se possível simplificar a imagem sem perder suas características importantes. A difusão anisotrópica clássica, nome do método, controla o quão “difundida” será a região de acordo com o seu gradiente. Dada a função anisotrópica (PERONA; MALIK, 1990):

$$I_t = \text{div}(c(x, y, t)\nabla I) = c(x, y, t)\nabla^2 + \nabla c \cdot \nabla I, \quad (3.9)$$

se $c(x, y, t)$ é uma constante, a função acima se torna uma difusão isotrópica. Caso contrário, se for escolhida uma função c tal que quanto maior o valor da magnitude gradiente na região, menor é o resultado de c e vice-versa, assim temos uma difusão que obedece os três termos propostos acima.

É possível implementar a equação anisotrópica de acordo com a seguinte discretização (PERONA; SHIOTA; MALIK, 1994):

$$I_{t+1}(i, j) = I_t(i, j) + \lambda [c_N \cdot \Delta_N I + c_S \cdot \Delta_S I + c_E \cdot \Delta_E I + c_W \cdot \Delta_W I]_{i,j}^t, \quad (3.10)$$

sendo $0 \leq \lambda \leq \frac{1}{4}$ e N, S, E, W representam as posições norte, sul, leste e oeste em relação ao *pixel* i, j . O símbolo Δ representa a diferença entre a intensidade dos *pixels* vizinhos. Uma possível função c é alguma g tal que $c(x, y, t) = g(\|\nabla I(x, y, t)\|)$ onde g retorne algo proporcionalmente inverso à magnitude do gradiente de $I(x, y, t)$. Os valores de c são computados de acordo com as equações que se seguem:

$$c_{N_{i,j}}^t = g\left(\left\|\left(\nabla I\right)_{i+\frac{1}{2},j}^t\right\|\right), \quad (3.11)$$

$$c_{S_{i,j}}^t = g\left(\left\|\left(\nabla I\right)_{i-\frac{1}{2},j}^t\right\|\right), \quad (3.12)$$

$$c_{E_{i,j}}^t = g \left(\left\| (\nabla I)_{i,j+\frac{1}{2}}^t \right\| \right), \quad (3.13)$$

$$c_{W_{i,j}}^t = g \left(\left\| (\nabla I)_{i,j-\frac{1}{2}}^t \right\| \right). \quad (3.14)$$

O valor do gradiente pode ser calculado a partir de diferentes estruturas da vizinhança, alcançando diferentes compensações entre precisão e localidade. Uma escolha simples consiste em aproximar a norma de gradiente em cada local com o valor absoluto. O artigo de [Perona e Malik \(1990\)](#) apresenta duas possíveis funções para g :

$$g(\nabla I) = e^{-\left(\frac{\|\nabla I\|}{k}\right)^2}, \quad (3.15)$$

$$g(\nabla I) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{k}\right)^2}, \quad (3.16)$$

a constante k pode ser determinada pelo desenvolvedor ([NEIVA, 2016](#); [CANNY, 1986](#)).

A [Equação 3.15](#) prioriza as bordas de alto contraste, enquanto a [Equação 3.16](#) privilegia grandes regiões. Em qualquer umas destas, se a magnitude do gradiente é igual a 0, $g(\nabla I)$ retorna 1 de acordo com o que foi proposto em ([PERONA; SHIOTA; MALIK, 1994](#)), regiões de baixa frequência devem ser mais suavizadas. Neste trabalho foi fixado o uso da [Equação 3.15](#).

3.3.5 Regularização Forward-Backward da Difusão Anisotrópica

O método *Forward-Backward* é uma nova classe de modelos de difusão anisotrópica proposta para o processamento de imagens e pode ser definida como um novo tipo de regularização do modelo clássico de Perona-Malik ([PERONA; SHIOTA; MALIK, 1994](#)), descrito na subseção anterior. Os autores do método *Forward-Backward*, [Guidotti, Kim e Lambers \(2013\)](#), afirmam que o modelo proposto por [Perona, Shiota e Malik \(1994\)](#) possui a desvantagem de ser mais teórico do que prático pois geram imagens *cartonizadas*. Dessa forma, propuseram uma regularização da equação descrita anteriormente adicionando dois parâmetros $p \in (1, \infty)$ e $\delta > 0$, com o objetivo de impedir o efeito de degrau e ao mesmo tempo preservar as bordas da imagem. A equação desenvolvida por [Guidotti, Kim e Lambers \(2013\)](#) é definida como:

$$u_t = \nabla \cdot \left(\left[\frac{1}{1 + K^2 |\nabla u|^2} + \delta |\nabla u|^{p-2} \right] \nabla u \right). \quad (3.17)$$

Nos experimentos realizados em ([GUIDOTTI; KIM; LAMBERS, 2013](#)) verificou-se que um parâmetro p próximo a 1 produz os melhores resultados. A inclusão de dois novos parâmetros possibilita o crescimento do gradiente como delimitação no valor máximo de crescimento.

Dessa forma, o regime de atraso é confinado em $[1 < |\nabla u| < M(\delta, p)]$ não possibilitando o desenvolvimento de descontinuidades. Determinando a região de análise, o efeito de degrau é substituído por uma pequena rampa com degraus controlados pelos valores definidos por p e δ (NEIVA, 2016; GUIDOTTI; KIM; LAMBERS, 2013).

Foram apresentados, até o momento, cinco métodos de restauração de imagens. E torna-se importante salientar que mais dois métodos utilizados são uma combinação dos filtros da **mediana com Perona-Malik** e **mediana com *Forward-Backward***. Esta abordagem foi adotada com o intuito de realizar uma filtragem de possíveis ruídos do tipo impulsivo e posterior filtragem com anisotrópicos visando preservação de bordas e suavização da imagem.

3.4 Passo 4: Correspondência

A etapa final da metodologia proposta encontra os pontos correspondentes entre o par de imagens utilizando os dois principais métodos de mosaicos, SIFT e SURF, e a correlação de fase detalhados no [Capítulo 2](#). Dispondo de um conhecimento prévio de como o conjunto de dados foi construído, torna-se possível constatar se a correspondência entre as imagens está correta ou não, a partir do padrão ouro de recorte das imagens. Portanto, a precisão pode ser medida para calcular a qualidade do método face ao ruído e o uso de métodos de filtragem. Todas as combinações são avaliadas para analisar a robustez dos métodos em diferentes condições e serão descritas no [Capítulo 4](#).

3.5 Considerações finais

Neste capítulo, foi proposto um método para aprimorar as metodologias de mosaico de imagens mais utilizadas na literatura: SIFT e SURF, juntamente com a abordagem baseada no domínio de frequência. O método propõe realizar a degradação e restauração das imagens e em seguida efetuar a correspondência entre os par de imagens. O objetivo é analisar os métodos de mosaico sob diferentes condições e encontrar os ruídos de pior caso e o filtro mais adequado para os ruídos aplicados. Posteriormente, o filtro com resultado mais satisfatório é adicionado aos métodos de mosaico com o intuito de aprimorar as correspondências entre as imagens. A partir dos resultados obtidos é possível identificar o método de mosaico mais adequado para imagens microscópicas, em conjunto com o filtro que obteve os melhores proveitos. À vista disso, foi desenvolvida uma ferramenta, que será descrita no [Capítulo 5](#) voltada para a construção de mosaicos de imagens microscópicas.

EXPERIMENTOS, RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados dos experimentos realizados pela metodologia proposta no [Capítulo 3](#), utilizando para isso as bases de imagens descritas na [Seção 4.1](#).

4.1 Base de Imagens

As imagens foram adquiridas pelo grupo de Computação Científica ² do Instituto de Física de São Carlos (IFSC) a cargo do Prof. Dr. Odemir M. Bruno em parceria com José Paulo Artêncio Júnior e Humberto Antunes de Almeida Filho. As três bases de imagens obtidas no **Passo 1** ([Seção 3.1](#)) foram utilizadas para avaliar o potencial dos métodos de mosaico com aplicação de ruídos e filtros, bem como tempo de processamento.

Na microscopia é muito importante que a amostra a ser estudada esteja bem iluminada. É possível efetuar essa iluminação por cima ou por baixo da amostra. Nas situações em que se usa luz vinda de baixo, dizemos que a luz é transmitida, e em casos de fonte de luz vinda de cima, chamamos de luz refletida. Para a captura das imagens foram utilizados os equipamentos microscópio óptico e lupa. No microscópio a luz é transmitida, ou seja, a luz atravessa o objeto a ser estudado e atinge a objetiva. Com a lupa a luz é refletida, assim, a luz incide sob a amostra e é refletida em direção a objetiva.

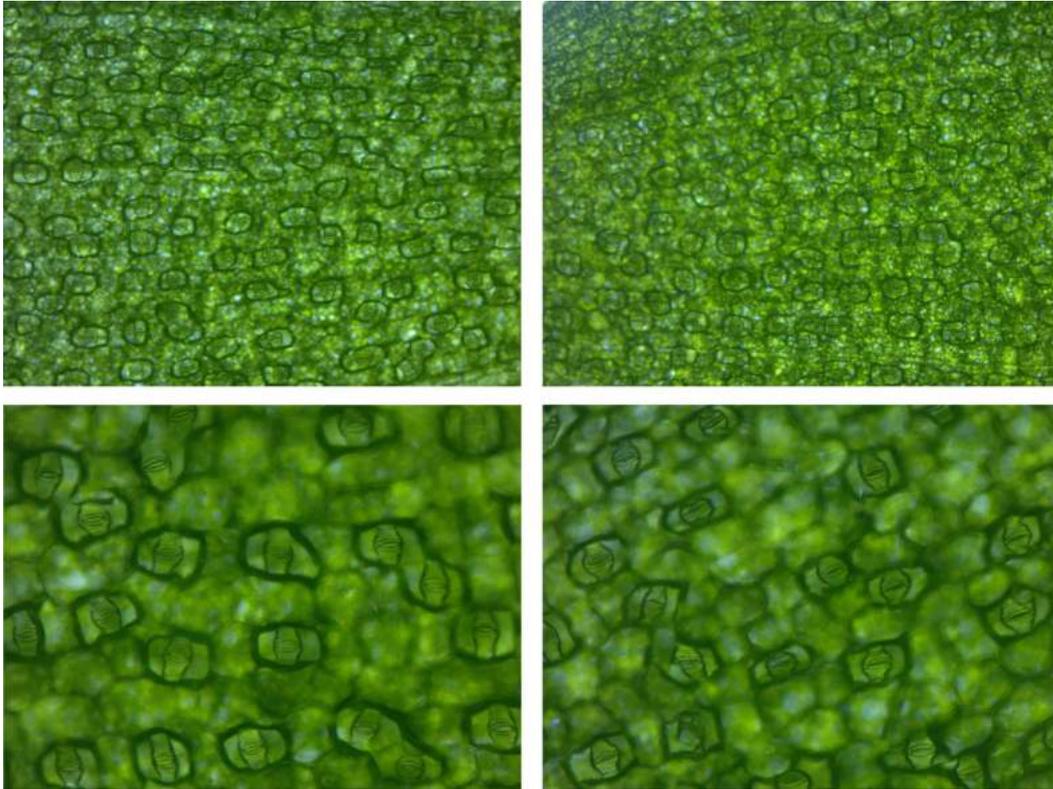
4.1.1 Bases para ruídos, filtros e correspondência

A primeira base é composta por 90 imagens microscópicas de estômatos com ampliação de $5\times$ com tamanho 2560×1920 *pixels*. A segunda base é formada por 134 imagens com aumento de $10\times$ e tamanho 2560×1920 *pixels*. E por fim, a terceira base de imagens contém

² *Scientific Computing Group (SCG)*, website: <<http://scg.ifsc.usp.br/>>.

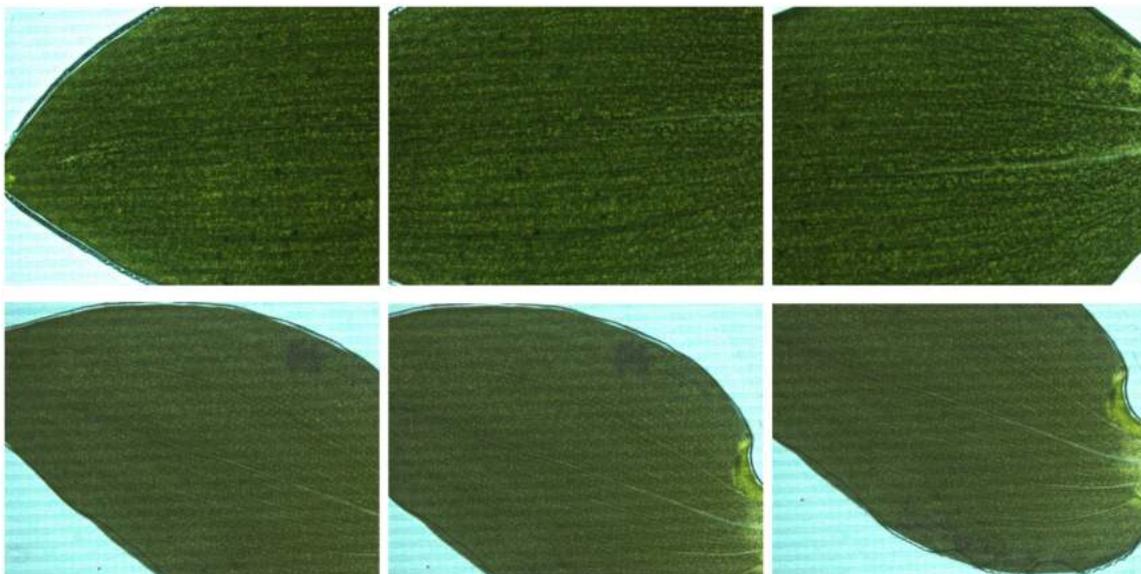
201 imagens com tamanho 1388×1038 adquiridas pelo equipamento lupa pertencente ao laboratório de pesquisas do SCG. A Figura 19 mostra alguns exemplos das imagens capturadas com ampliações de $5\times$ e $10\times$ e a Figura 20 ilustra alguns exemplos da base capturada pela lupa.

Figura 19 – Bases com ampliação de $5\times$ e $10\times$, respectivamente.



Fonte: Elaborada pela autora.

Figura 20 – Base adquirida com a Lupa.



Fonte: Elaborada pela autora.

Logo, obteve-se um total de 425 imagens, e de cada imagem original, foi criado um par, ou seja, recortes formando um conjunto de 850 pares, todas da espécie de planta *Callisia repens*. As bases com ampliação de $5\times$ e $10\times$ foram recortadas formando um par com sobreposição de 12,5% e a base obtida com a lupa possui imagens com sobreposições de 50%. Estas imagens foram submetidas às mesmas condições, foram afixados os valores de foco e iluminação de forma a possuírem as mesmas características, assim não possuem variação de escala, perspectiva e luminosidade. As imagens de todas as bases são coloridas (RGB), mas convertidas para escala de cinza no trabalho. A [Tabela 1](#) abaixo mostra resumidamente os dados das bases apresentadas.

Tabela 1 – Descrição das bases de imagens.

Bases	Nº de imagens originais	Taxa de sobreposição e tamanho	Par formado	Tamanho do par
5×	90	12,5% de sobreposição. 2560 × 1920	180	1441 × 1920
10×	134		268	
Lupa	201	50% de sobreposição. 1388 × 1038	402	1042 × 1038
Total	425		850	

Fonte: Dados da pesquisa.

4.2 Configuração dos Algoritmos

Para as diferentes bases, os métodos de ruídos e filtros foram aplicados com os mesmos parâmetros e são definidos nesta seção. Todos os algoritmos de ruídos, filtros e métodos de mosaico foram implementados em MATLAB e a ferramenta desenvolvida *MosaicImage* foi implementada em Java e será descrita no [Capítulo 5](#).

4.2.1 Parâmetros dos ruídos

Os valores de configuração padrão para cada ruído são descritos abaixo e na [Tabela 2](#):

Tabela 2 – Parâmetros utilizados para os ruídos.

Variações dos parâmetros dos ruídos					
Nível de ruído	S&P (Densidade d)	Gaussiano (1) Média = 0 e Altera a σ^2	Gaussiano (2) Média = 0 e Altera a σ^2	Poisson (Nº de iterações)	Multiplicativo Altera a σ^2
Var 1	0.03	0.001	0.01	1	0.001
Var 2	0.06	0.002	0.02	2	0.004
Var 3	0.09	0.003	0.03	3	0.006
Var 4	0.12	0.004	0.04	4	0.008
Var 5	0.15	0.005	0.05	5	0.02
Var 6	0.18	0.006	0.06	6	0.04
Var 7	0.21	0.007	0.07	6	0.06
Var 8	0.24	0.008	0.08	8	0.08
Var 9	0.27	0.009	0.09	9	0.10
Var 10	0.30	0.01	0.10	10	0.15

Fonte: Dados da pesquisa.

Com o objetivo de degradar as imagens além dos padrões pré-definidos, nos experimentos realizados os parâmetros foram modificados variando de pouco para muito ruído no intervalo de dez variações, sendo Var 1 uma imagem com pouco ruído e Var 10 uma imagem mais corrompida. Para o ruído Poisson, as variações correspondem ao número de iterações efetuadas com a imagem de entrada já ruidosa. O ruído Gaussiano (2) foi adotado somente para o método de mosaico baseado em Fourier, pois o mesmo apresenta bom desempenho com imagens ruidosas. Consequentemente tornou-se necessário realizar um aumento do nível de ruído do tipo Gaussiano.

4.2.2 Parâmetros dos filtros

Os parâmetros utilizados para todos os filtros foram definidos conforme a Tabela 3 abaixo:

Tabela 3 – Parâmetros utilizados para os filtros.

Parâmetros dos Filtros				
Média	Mediana	Wiener	Perona-Malik	Forward-Backward
$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	3×3	3×3	$t = 10$ $k = 15$ $\lambda = 0.25$ Equação 3.16	$t = 10$ $\delta = 0.1$ $p = 1.1$ $\lambda = 0.25$

Como definido anteriormente, dois métodos de filtragem foram uma junção do filtro da mediana com Perona-Malik e mediana com *Forward-Backward*. Logo, foram aplicados cinco modelos de filtros juntamente com duas combinações.

4.2.3 Algoritmos de mosaico

Todos os métodos de mosaico foram executados em MATLAB, uma vez que o software possui todas as bibliotecas necessárias. O algoritmo SURF é implementado com a função “*detectSURFFeatures*” que é baseada no artigo de Bay, Tuytelaars e Gool (2008). O SIFT funciona como uma biblioteca que pode ser instalada no MATLAB e é embasada no trabalho de Vedaldi e Fulkerson (2008)⁴. O algoritmo de correlação de fase foi implementado a partir do artigo de Kuglin e Hines (1975).

4.3 Localização

Para obter o erro médio de localização das coordenadas, aplica-se o cálculo da distância Euclidiana entre o ponto localizado e o padrão ouro. A distância Euclidiana de um ponto a outro

⁴ A biblioteca VLFeat é de código-fonte aberto, suporta Windows, Mac OS X e Linux e sua versão mais recente é .0.9.21. Está disponível para download em: <<http://www.vlfeat.org/>>.

é calculada pela Equação 4.1 e o resultado é dado em *pixels*.

$$Erro = \sqrt{(x_{po} - x_{pl})^2 + (y_{po} - y_{pl})^2}, \quad (4.1)$$

onde (x_{po}, y_{po}) é a coordenada de localização do padrão ouro e (x_{pl}, y_{pl}) é a coordenada do ponto localizado.

A partir dos dados de erro médio adquiridos foi possível calcular o desvio padrão para cada teste realizado. Além disso, o procedimento de verificação foi aplicado 20 vezes, utilizando para cada algoritmo os parâmetros definidos na Subseção 4.2.1 e na Subseção 4.2.2.

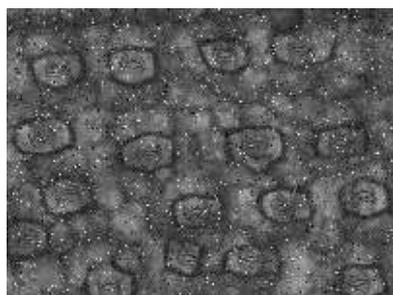
4.4 Descrição dos experimentos

As bases de imagens com ampliação de $5\times$ e $10\times$ são utilizadas em todos os experimentos listados para avaliar a robustez dos métodos de mosaico SIFT, SURF e correlação de fase (domínio de frequência), com adição de ruídos e filtros e realização de correspondência; a base de imagens obtida com a Lupa é empregada nos experimentos 4 e 5. Os resultados da pesquisa são apresentados por meio de 5 experimentos, conforme a descrição abaixo:

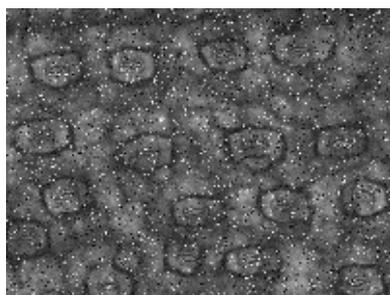
1. Experimento 1 (Seção 4.5): testes com ruídos e filtros para o método de mosaico SURF;
2. Experimento 2 (Seção 4.6): testes com ruídos e filtros para o método de mosaico SIFT;
3. Experimento 3 (Seção 4.7): testes com ruídos e filtros para o método de mosaico baseado em correlação de fase, que para fins de simplificação, neste capítulo será mencionado como PC (do inglês, *Phase Correlation*);
4. Experimento 4 (Seção 4.9): realiza uma comparação entre os métodos de mosaico sem a adição de ruídos e filtros;
5. Experimento 5 (Seção 4.10): apresenta uma análise comparativa de tempo de processamento entre os métodos de mosaico.

Para todos os experimentos, inicialmente aplicam-se os ruídos ao segundo corte da imagem, em seguida é realizada uma filtragem visando a sua respectiva restauração, por fim, a correspondência entre os dois cortes da imagem é efetuado. As imagens foram testadas com 10 variações de ruído, ou seja, de baixo nível de ruído até alto nível de ruído. Na Figura 21, Figura 22, Figura 23, Figura 24 e Figura 25 são ilustradas as imagens resultantes das aplicações dos ruídos S&P, Gaussiano 1, Gaussiano 2 (somente para a PC), Poisson e Multiplicativo, respectivamente.

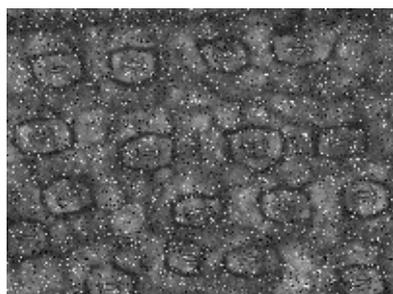
Figura 21 – Variações do ruído SeP (% de ruído).



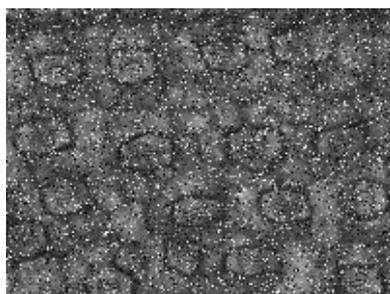
(a) 3% de ruído



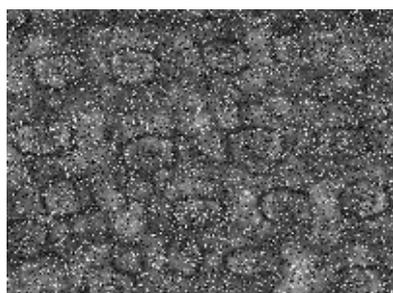
(b) 6% de ruído



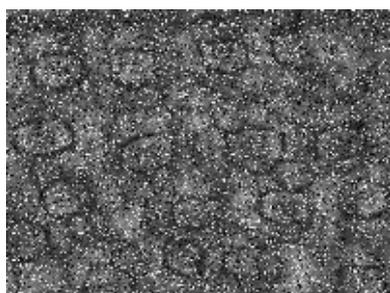
(c) 9% de ruído



(d) 12% de ruído



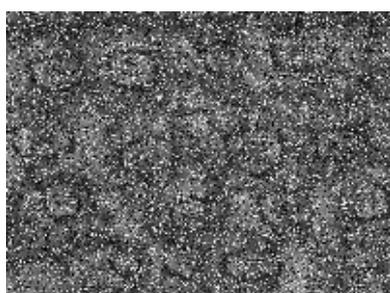
(e) 15% de ruído



(f) 18% de ruído



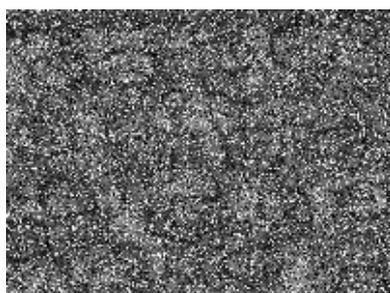
(g) 21% de ruído



(h) 24% de ruído

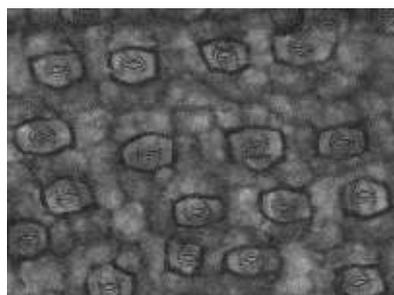
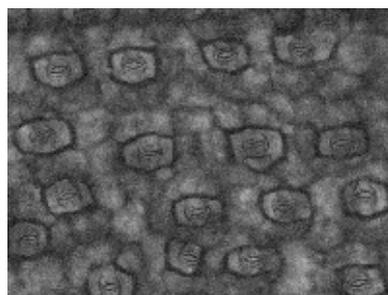
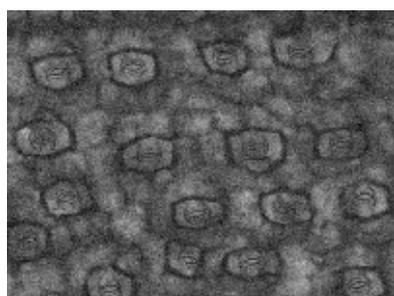
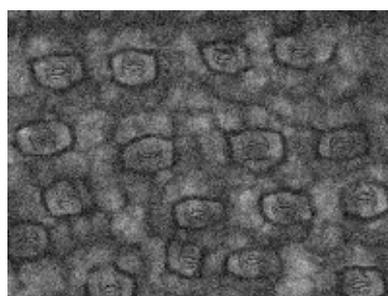
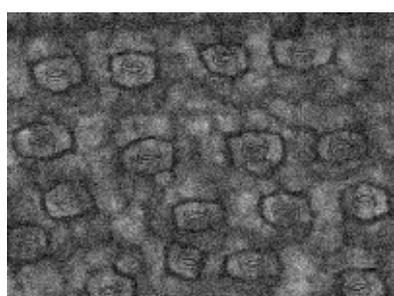
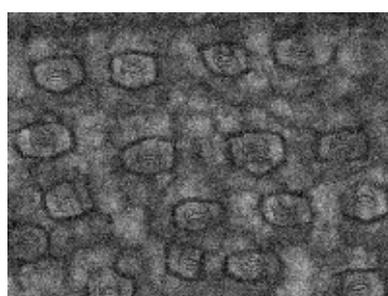
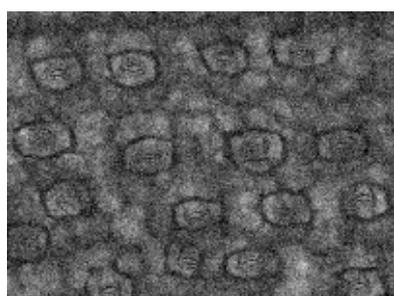
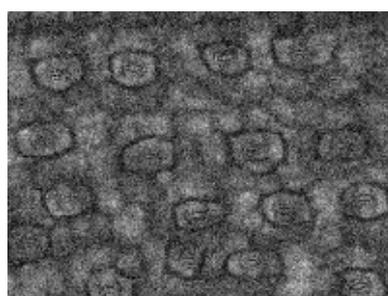
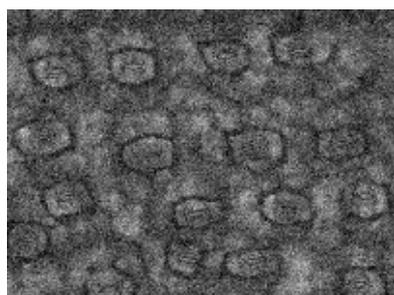
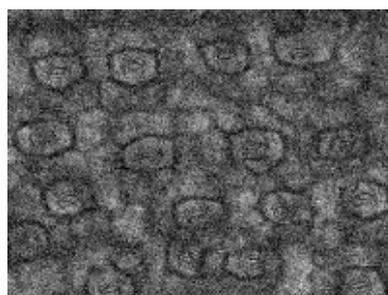


(i) 27% de ruído

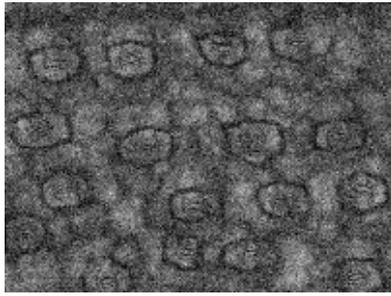
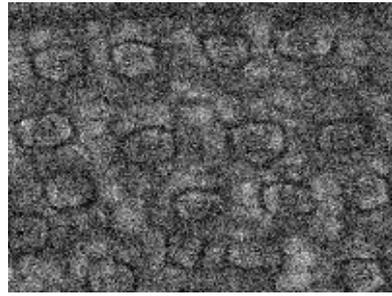
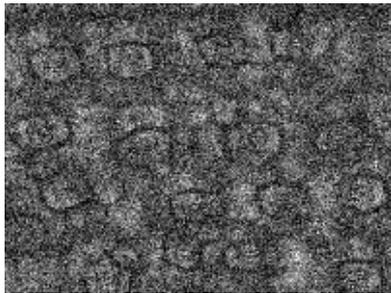
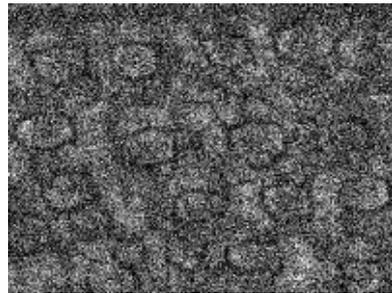
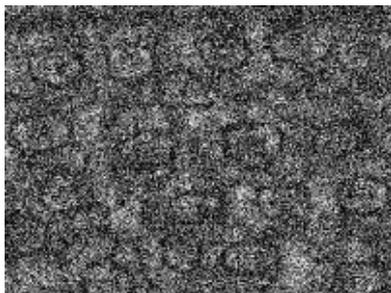
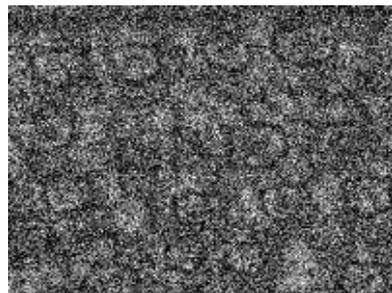
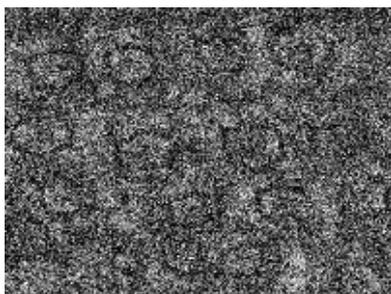
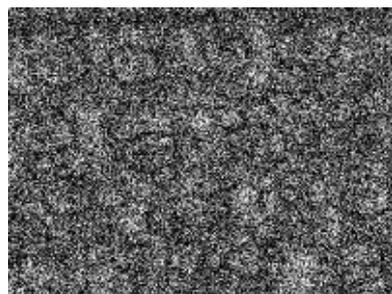
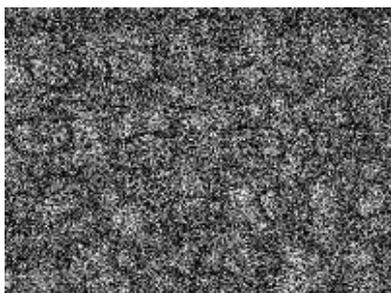
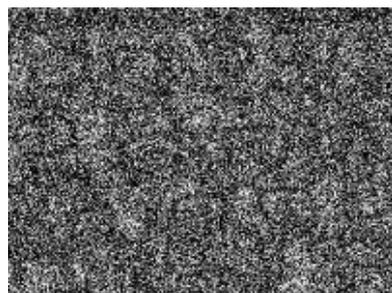


(j) 30% de ruído

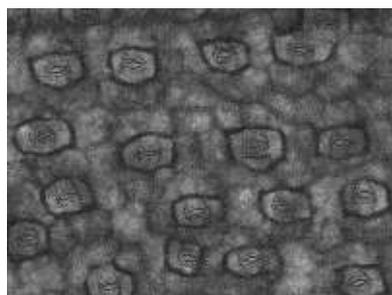
Fonte: Elaborada pela autora.

Figura 22 – Variações do ruído Gaussiano (1), ($\bar{z} = 0$ e altera a σ^2).(a) $\sigma^2 = 0.001$ (b) $\sigma^2 = 0.002$ (c) $\sigma^2 = 0.003$ (d) $\sigma^2 = 0.004$ (e) $\sigma^2 = 0.005$ (f) $\sigma^2 = 0.006$ (g) $\sigma^2 = 0.007$ (h) $\sigma^2 = 0.008$ (i) $\sigma^2 = 0.009$ (j) $\sigma^2 = 0.01$

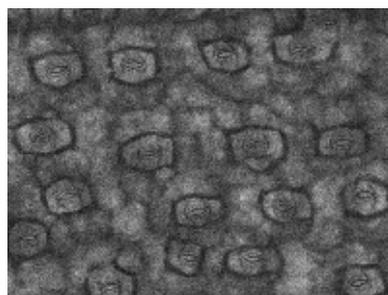
Fonte: Elaborada pela autora.

Figura 23 – Variações do ruído Gaussiano (2), somente para PC, ($\bar{z} = 0$ e altera a σ^2).(a) $\sigma^2 = 0.01$ (b) $\sigma^2 = 0.02$ (c) $\sigma^2 = 0.03$ (d) $\sigma^2 = 0.04$ (e) $\sigma^2 = 0.05$ (f) $\sigma^2 = 0.06$ (g) $\sigma^2 = 0.07$ (h) $\sigma^2 = 0.08$ (i) $\sigma^2 = 0.09$ (j) $\sigma^2 = 0.10$

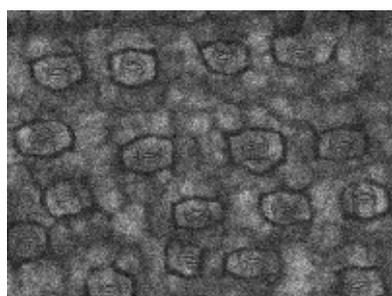
Fonte: Elaborada pela autora.

Figura 24 – Variações do ruído Poisson (N° de iterações com a imagem corrompida).

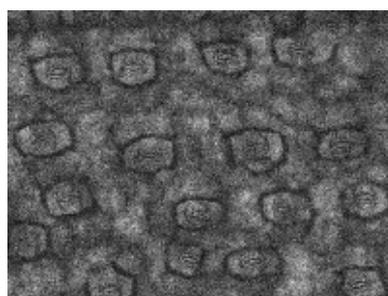
(a) 1 iteração



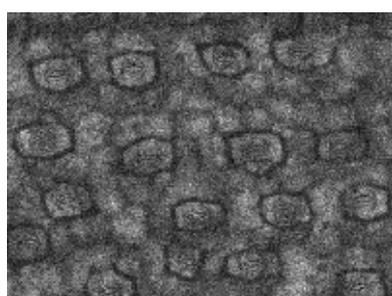
(b) 2 iterações



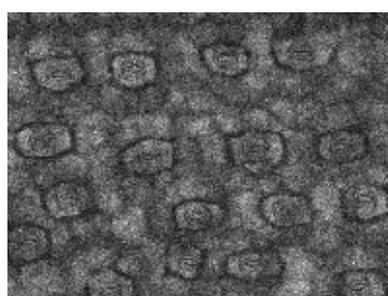
(c) 3 iterações



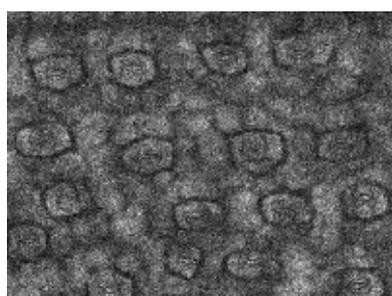
(d) 4 iterações



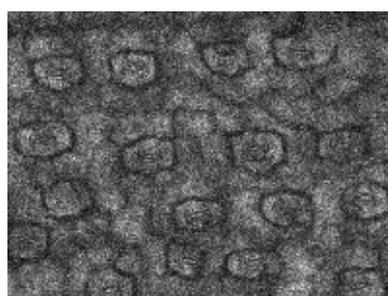
(e) 5 iterações



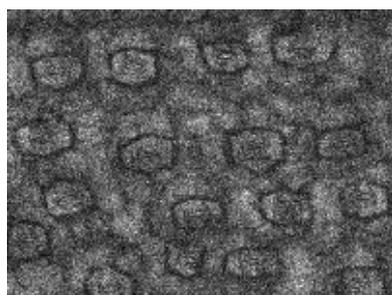
(f) 6 iterações



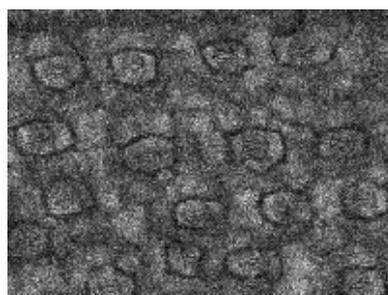
(g) 7 iterações



(h) 8 iterações

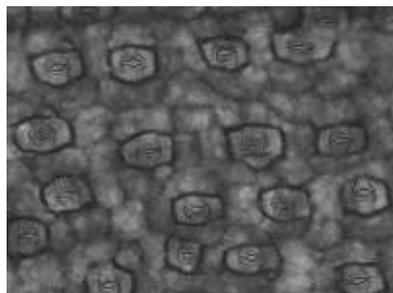
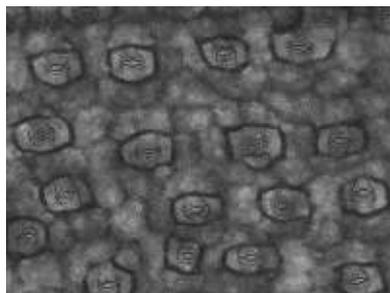
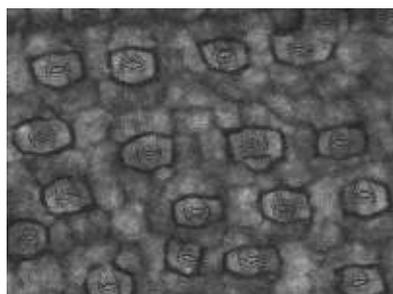
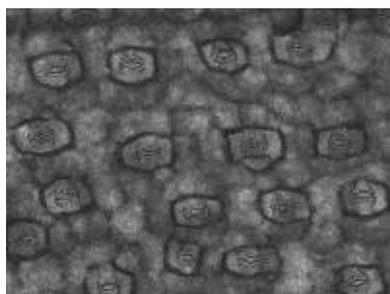
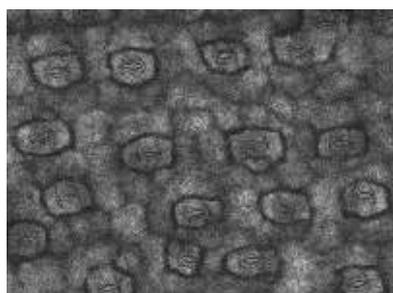
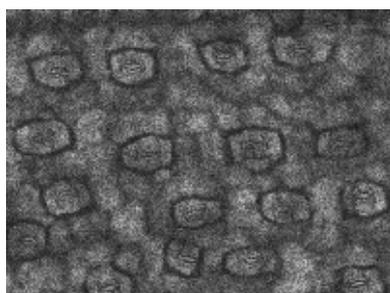
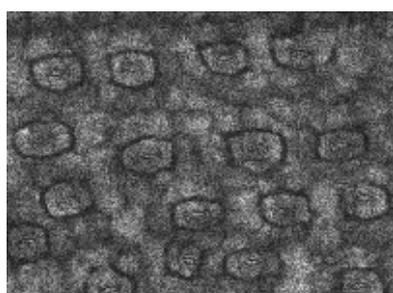
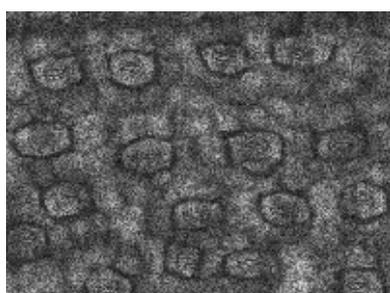
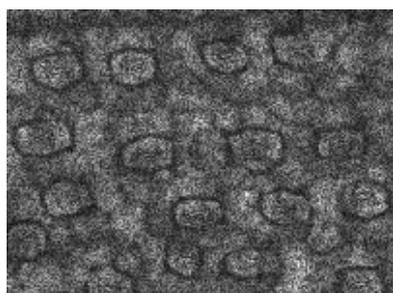
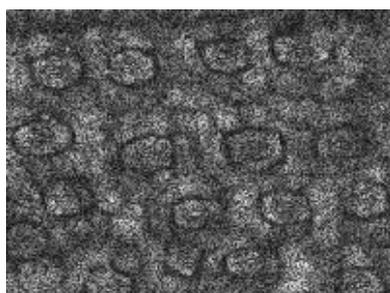


(i) 9 iterações



(j) 10 iterações

Fonte: Elaborada pela autora.

Figura 25 – Variações do ruído Multiplicativo (altera a σ^2).(a) $\sigma^2 = 0.001$ (b) $\sigma^2 = 0.004$ (c) $\sigma^2 = 0.006$ (d) $\sigma^2 = 0.008$ (e) $\sigma^2 = 0.02$ (f) $\sigma^2 = 0.04$ (g) $\sigma^2 = 0.06$ (h) $\sigma^2 = 0.08$ (i) $\sigma^2 = 0.10$ (j) $\sigma^2 = 0.15$

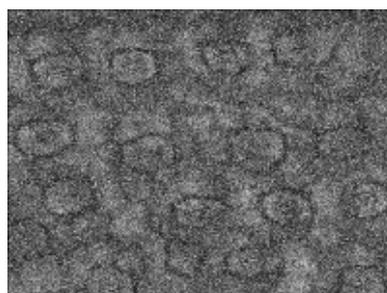
Fonte: Elaborada pela autora.

No que tange a aplicação de filtros, é do mesmo modo importante ilustrar a performance visual dos sete filtros manipulados neste trabalho. Dessa maneira, a [Figura 26](#), [Figura 27](#), [Figura 28](#) e [Figura 29](#) apresentam as imagens resultantes da filtragem utilizando para isso imagens com alto nível de ruído.

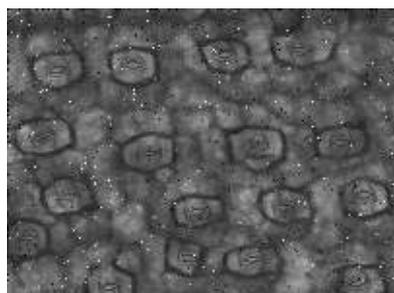
Figura 26 – Filtragem de (a) imagem corrompida com 30% de ruído S&P.



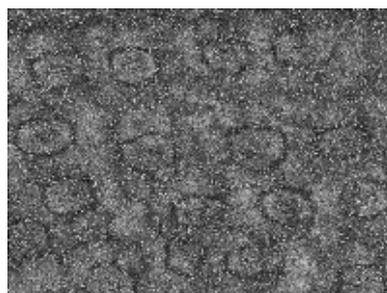
(a) Imagem com alto nível de ruído



(b) Média



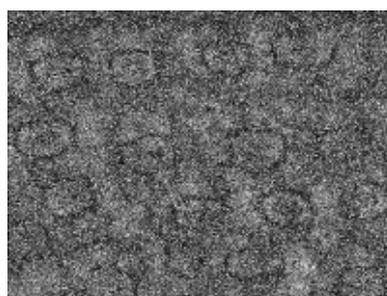
(c) Mediana



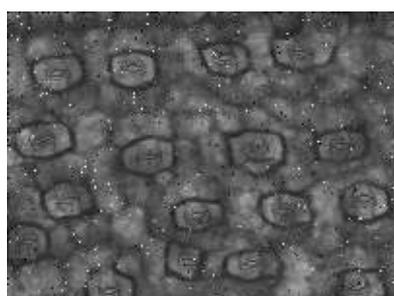
(d) Wiener



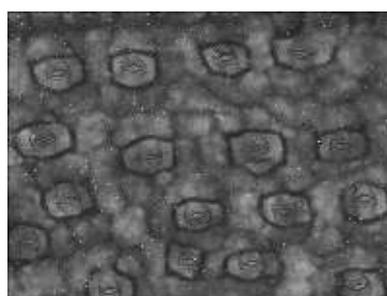
(e) Perona-Malik



(f) Forward-Backward

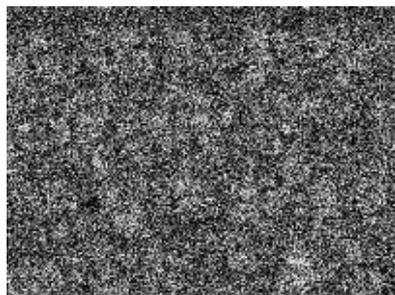


(g) Mediana com Perona-Malik

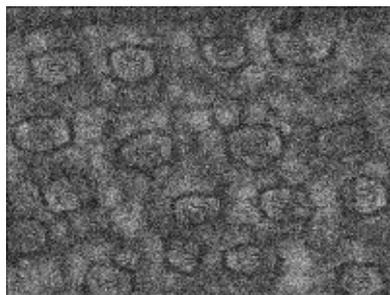


(h) Mediana com Forward-Backward

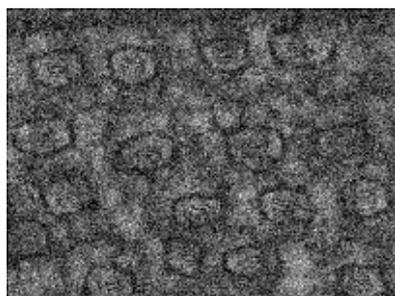
Fonte: Elaborada pela autora.

Figura 27 – Filtragem de (a) imagem corrompida com ruído Gaussiano, com $\bar{z} = 0$ e $\sigma^2 = 0.10$.

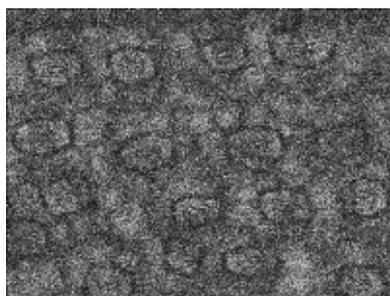
(a) Imagem com alto nível de ruído



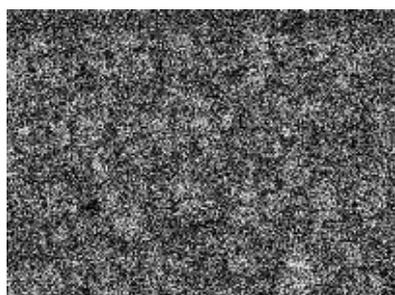
(b) Média



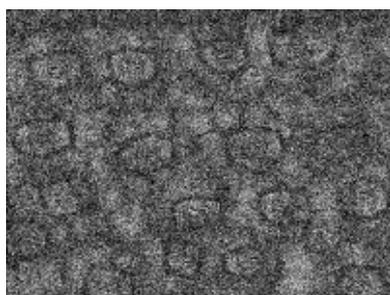
(c) Mediana



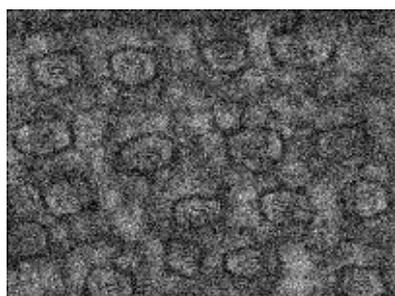
(d) Wiener



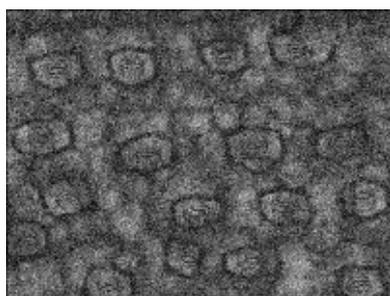
(e) Perona-Malik



(f) Forward-Backward



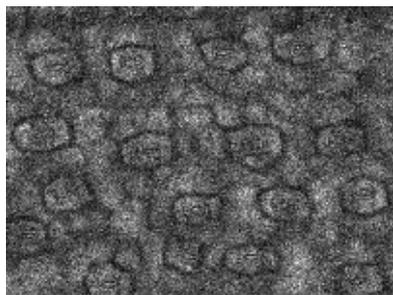
(g) Mediana com Perona-Malik



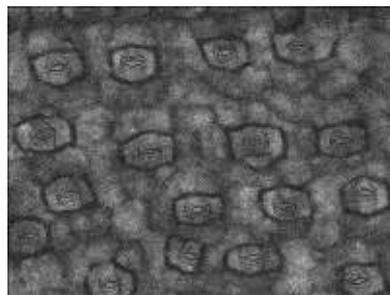
(h) Mediana com Forward-Backward

Fonte: Elaborada pela autora.

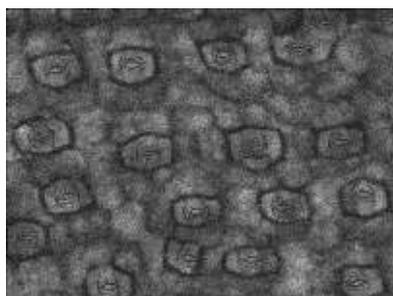
Figura 28 – Filtragem de (a) imagem corrompida com 10 iterações do ruído Poisson.



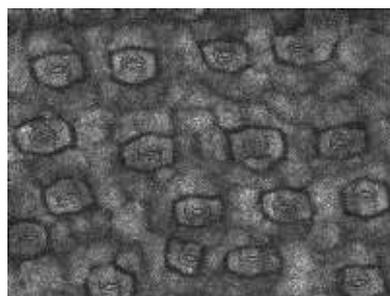
(a) Imagem com alto nível de ruído



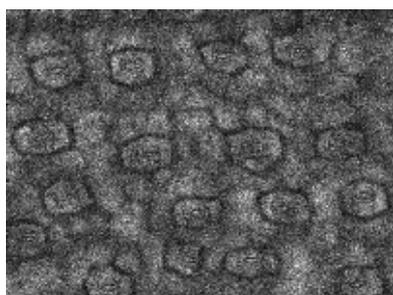
(b) Média



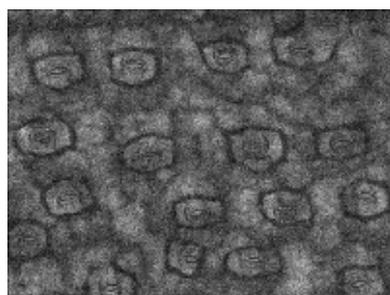
(c) Mediana



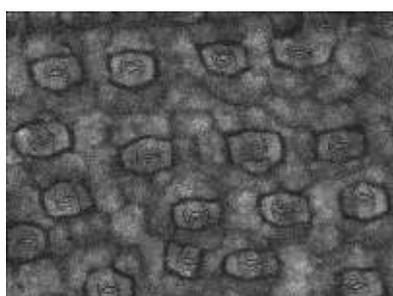
(d) Wiener



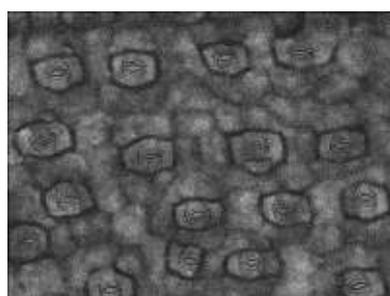
(e) Perona-Malik



(f) Forward-Backward



(g) Mediana com Perona-Malik

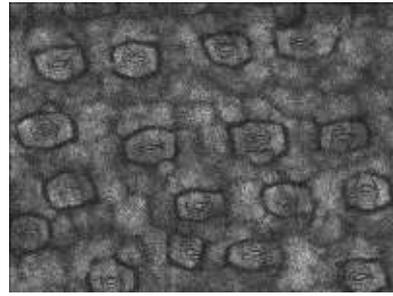


(h) Mediana com Forward-Backward

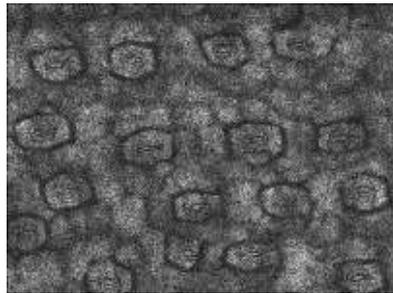
Fonte: Elaborada pela autora.

Figura 29 – Filtragem de (a) imagem corrompida com ruído Multiplicativo, com $\sigma^2 = 0.15$.

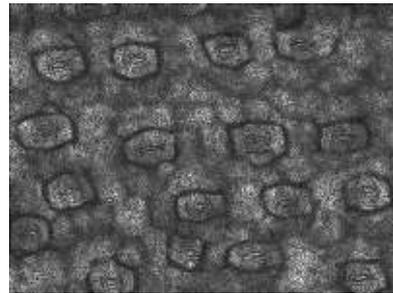
(a) Imagem com alto nível de ruído



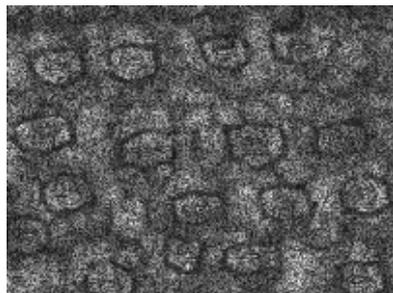
(b) Média



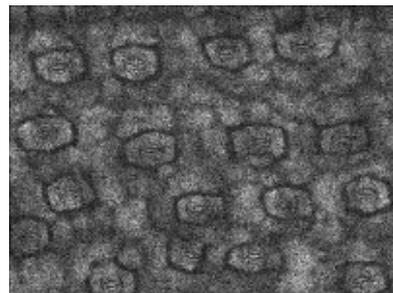
(c) Mediana



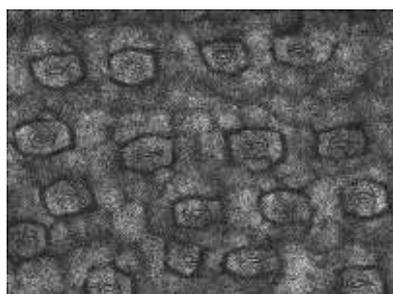
(d) Wiener



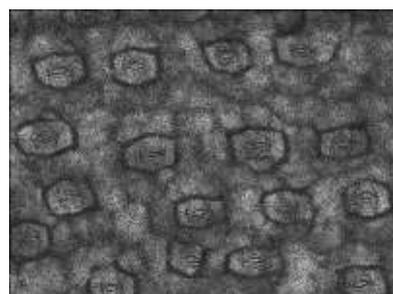
(e) Perona-Malik



(f) Forward-Backward



(g) Mediana com Perona-Malik



(h) Mediana com Forward-Backward

Fonte: Elaborada pela autora.

Os experimentos da [Seção 4.5](#), [Seção 4.6](#), [Seção 4.7](#) visam encontrar, dentre os métodos de filtragem analisados, o que produz o melhor resultado (maior taxa de acerto (%)) para cada ruído, e o ruído de pior caso considerando cada método de mosaico. A taxa de acerto é calculada a partir da contagem do número de junções ótimas entre as imagens. Logo, por exemplo, para a base com ampliação de $5\times$, o total de erros de correspondência é subtraído do total de 180 pares, resultando no número de junções ótimas. Nos experimentos citados foram analisadas

as 10 variações de cada ruído e o percentual de acerto para os 7 tipos de filtros propostos. A base de imagens com ampliação de $5\times$ possui 180 pares de imagens, e para cada variação de ruído o processo foi repetido 20 vezes, totalizando assim $180 \times 20 \times 10$ execuções para cada experimento dessa base. O conjunto de imagens com ampliação de $10\times$ possui 268 pares de imagens, assim sendo, foram realizadas $268 \times 20 \times 10$ iterações no total.

4.5 Experimento 1: Testes com ruídos e filtros: SURF

Este primeiro experimento tem como objetivo analisar o desempenho do método de mosaico SURF. Como ilustrado na [Figura 26\(c\)](#) visualmente, o filtro da mediana obteve a melhor filtragem. Considerando a [Tabela 4](#) abaixo, é possível verificar que o filtro da mediana e a sua combinação com o *Forward-Backward* foram os que produziram os melhores desempenhos mesmo quando ocorre o aumento do nível de ruído.

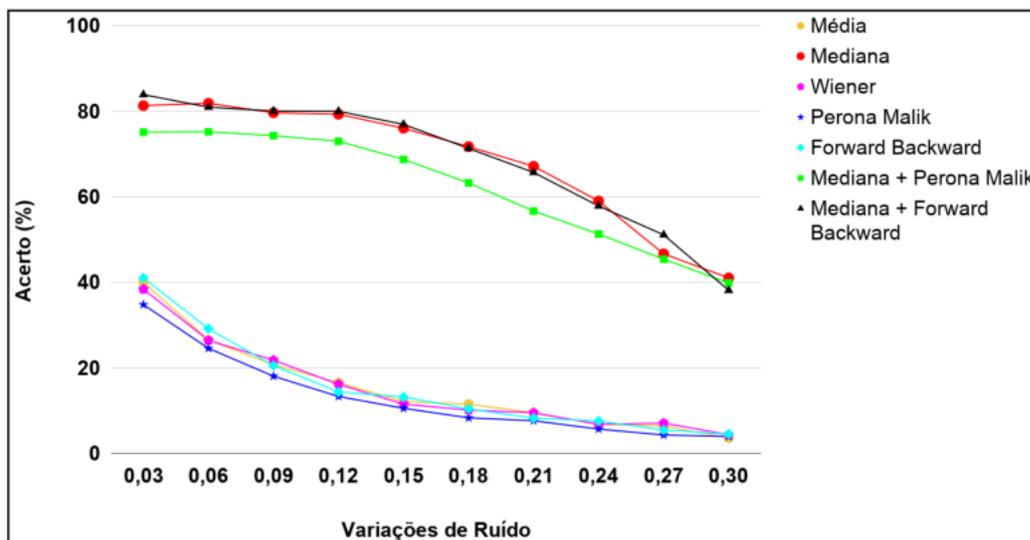
Tabela 4 – SURF: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo S&P.

Ruído S&P							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.03	40.00 (± 0.67)	81.32 (± 0.22)	38.40 (± 1.20)	34.79 (± 1.52)	41.04 (± 1.21)	75.14 (± 2.27)	83.89 (± 0.20)
0.06	26.53 (± 0.83)	81.88 (± 0.21)	26.46 (± 0.87)	24.58 (± 1.63)	29.17 (± 1.34)	75.21 (± 2.28)	80.97 (± 0.23)
0.09	20.76 (± 1.02)	79.65 (± 0.23)	21.81 (± 1.02)	18.06 (± 1.83)	20.56 (± 1.37)	74.31 (± 2.31)	80.07 (± 0.22)
0.12	16.53 (± 1.59)	79.31 (± 0.23)	16.18 (± 1.09)	13.33 (± 2.05)	14.38 (± 1.91)	72.99 (± 2.52)	80.00 (± 0.24)
0.15	12.15 (± 1.32)	76.04 (± 0.25)	11.53 (± 1.35)	10.56 (± 2.55)	13.20 (± 2.01)	68.75 (± 2.54)	76.95 (± 0.28)
0.18	11.53 (± 1.80)	71.74 (± 0.30)	10.14 (± 1.64)	8.33 (± 3.76)	10.42 (± 2.27)	63.26 (± 2.59)	71.32 (± 0.34)
0.21	9.51 (± 2.04)	67.15 (± 0.36)	9.51 (± 2.01)	7.64 (± 3.81)	8.26 (± 2.29)	56.67 (± 2.65)	65.70 (± 0.40)
0.24	6.81 (± 2.47)	59.03 (± 0.52)	6.81 (± 2.40)	5.69 (± 3.91)	7.57 (± 2.33)	51.32 (± 2.70)	57.85 (± 0.48)
0.27	6.39 (± 3.34)	46.67 (± 0.57)	7.08 (± 3.04)	4.31 (± 3.97)	5.49 (± 2.81)	45.42 (± 2.76)	51.11 (± 0.54)
0.30	3.61 (± 5.59)	41.04 (± 0.59)	4.38 (± 3.50)	3.96 (± 4.09)	4.51 (± 3.97)	39.86 (± 2.82)	38.13 (± 0.67)

Fonte: Dados da pesquisa.

Os resultados quantitativos da [Tabela 4](#) são melhores de visualizar com o gráfico da [Figura 30](#). Onde pode-se constatar que os filtros da mediana e sua combinação com anisotrópicos são os mais adequados. O filtro de pior caso para esse tipo de ruído com o SURF são média, Wiener, Perona Malik e *Forward-Backward*, com taxas de acerto abaixo de 5%.

Figura 30 – SURF: ruído S&P.



Fonte: Elaborado pela autora.

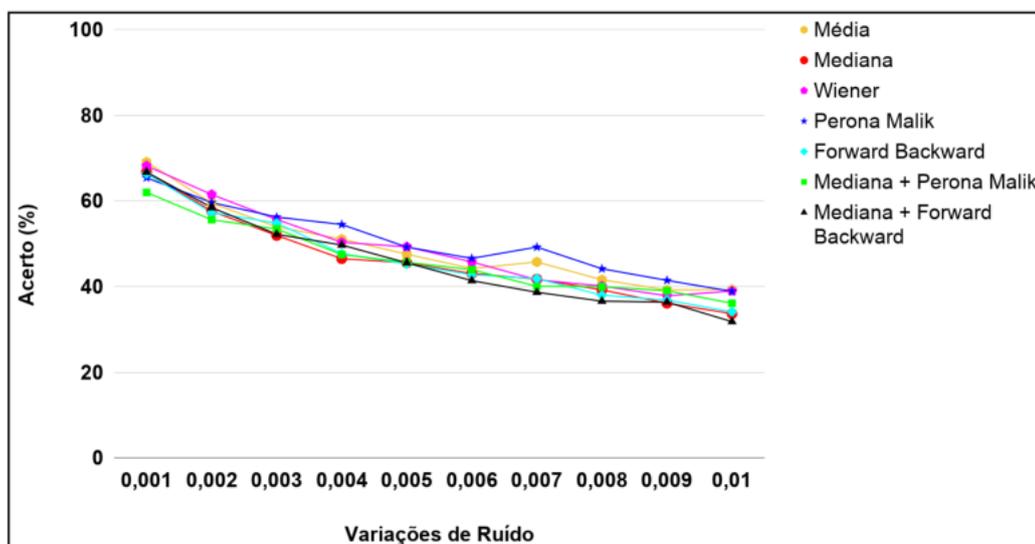
Na [Tabela 5](#) é possível verificar que o filtro de Perona Malik obteve melhores resultados na maioria dos casos, entretanto essa taxa de acerto não é muito considerável. Ao analisar o gráfico da [Figura 31](#) destaca-se que a performance de todos são semelhantes para o ruído Gaussiano. Para este tipo de ruído, os filtros apresentados são considerados inadequados.

Tabela 5 – SURF: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Gaussiano.

Ruído Gaussiano							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.001	69.10 (± 0.50)	66.81 (± 0.50)	68.20 (± 0.50)	65.42 (± 0.50)	66.39 (± 0.51)	62.01 (± 0.52)	66.74 (± 0.50)
0.002	59.51 (± 0.54)	57.64 (± 0.56)	61.53 (± 0.53)	59.65 (± 0.53)	57.29 (± 0.55)	55.62 (± 0.55)	58.47 (± 0.56)
0.003	53.96 (± 0.57)	51.95 (± 0.59)	55.69 (± 0.56)	56.25 (± 0.55)	54.86 (± 0.58)	53.47 (± 0.57)	52.22 (± 0.59)
0.004	51.11 (± 0.60)	46.53 (± 0.60)	50.35 (± 0.59)	54.51 (± 0.57)	47.71 (± 0.60)	47.50 (± 0.60)	49.72 (± 0.61)
0.005	47.57 (± 0.62)	45.56 (± 0.64)	49.31 (± 0.61)	49.24 (± 0.60)	45.28 (± 0.64)	45.69 (± 0.62)	45.56 (± 0.64)
0.006	44.24 (± 0.64)	43.06 (± 0.67)	45.76 (± 0.66)	46.60 (± 0.63)	42.78 (± 0.66)	43.96 (± 0.64)	41.39 (± 0.63)
0.007	45.76 (± 0.67)	41.74 (± 0.69)	41.60 (± 0.67)	49.24 (± 0.63)	41.88 (± 0.68)	40.07 (± 0.63)	38.68 (± 0.69)
0.008	41.60 (± 0.67)	39.24 (± 0.71)	40.14 (± 0.67)	44.17 (± 0.64)	37.99 (± 0.68)	40.00 (± 0.67)	36.60 (± 0.73)
0.009	39.24 (± 0.69)	36.11 (± 0.73)	37.85 (± 0.67)	41.46 (± 1.19)	36.88 (± 0.73)	39.03 (± 0.70)	36.39 (± 0.77)
0.01	39.31 (± 0.72)	33.68 (± 0.71)	38.96 (± 0.71)	38.89 (± 0.69)	34.10 (± 0.75)	36.11 (± 1.21)	31.81 (± 0.72)

Fonte: Dados da pesquisa.

Figura 31 – SURF: ruído Gaussiano.



Fonte: Elaborado pela autora.

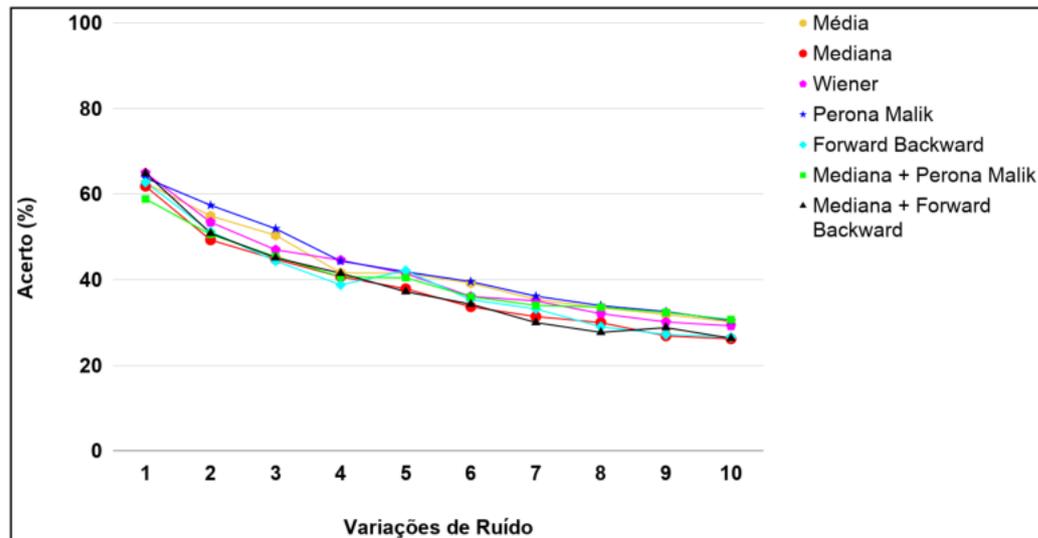
O ruído do tipo Poisson, muito comum em imagens digitais, foi melhor filtrado pelo Perona Malik. Contudo, seus resultados assim como o Gaussiano, não são muito significativos, como ilustra a Tabela 6 o gráfico da Figura 32. Todos os filtros identificam-se como inapropriados.

Tabela 6 – SURF: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Poisson.

Ruído Poisson							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
1	62.64 (±0.52)	61.87 (±0.53)	65.00 (±0.51)	63.82 (±0.52)	62.78 (±0.52)	58.89 (±0.47)	64.93 (±0.51)
2	54.93 (±0.56)	49.31 (±0.66)	53.47 (±0.57)	57.43 (±0.46)	51.25 (±0.64)	50.63 (±0.55)	50.90 (±0.65)
3	50.42 (±0.61)	44.79 (±0.71)	47.01 (±0.60)	51.95 (±0.51)	44.38 (±0.71)	45.42 (±0.60)	45.14 (±0.70)
4	41.60 (±0.63)	40.70 (±0.75)	44.58 (±0.66)	44.37 (±0.58)	38.82 (±0.76)	40.69 (±0.65)	41.53 (±0.74)
5	41.53 (±0.65)	37.92 (±0.78)	41.46 (±0.68)	41.87 (±0.61)	42.15 (±0.73)	40.49 (±0.65)	37.22 (±0.78)
6	39.24 (±0.67)	33.68 (±0.82)	36.11 (±0.69)	39.58 (±0.63)	35.35 (±0.80)	36.04 (±0.69)	34.31 (±0.81)
7	35.42 (±0.71)	31.39 (±0.84)	35.07 (±0.75)	36.18 (±0.66)	33.19 (±0.83)	34.03 (±0.71)	30.00 (±0.85)
8	33.47 (±0.71)	30.00 (±0.85)	32.08 (±0.76)	33.96 (±0.68)	29.03 (±0.87)	33.61 (±0.72)	27.71 (±0.88)
9	31.88 (±0.81)	26.88 (±0.89)	30.14 (±0.82)	32.57 (±0.69)	27.22 (±0.89)	32.36 (±0.72)	28.82 (±0.87)
10	30.14 (±0.83)	26.18 (±0.89)	29.24 (±0.99)	30.42 (±0.71)	26.60 (±0.90)	30.70 (±0.74)	26.32 (±0.89)

Fonte: Dados da pesquisa.

Figura 32 – SURF: ruído Poisson.



Fonte: Elaborado pela autora.

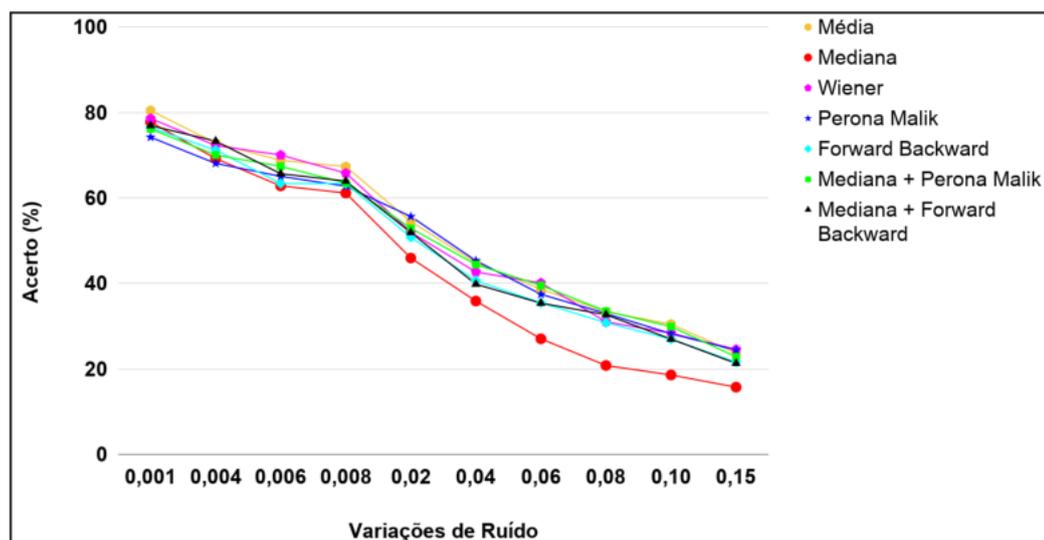
Finalmente, o último ruído analisado para o SURF foi o multiplicativo, a [Tabela 7](#) mostra que o desempenho dos filtros foi muito variado, mas média e Wiener foram os considerados, grosso modo, preferíveis. A [Figura 33](#) melhor ilustra o resultado de cada filtro. O filtro com mais baixos acertos foi da mediana.

Tabela 7 – SURF: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Multiplicativo.

Ruído Multiplicativo							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.001	80.49 (± 0.41)	77.78 (± 0.44)	78.61 (± 0.42)	74.24 (± 0.45)	76.46 (± 0.44)	76.18 (± 0.44)	76.95 (± 0.43)
0.004	72.85 (± 0.47)	69.24 (± 0.50)	72.29 (± 0.47)	68.12 (± 0.49)	71.25 (± 0.48)	70.07 (± 0.48)	73.33 (± 0.47)
0.006	68.82 (± 0.48)	62.85 (± 0.53)	70.07 (± 0.48)	65.07 (± 0.51)	63.40 (± 0.50)	67.43 (± 0.50)	65.69 (± 0.51)
0.008	67.36 (± 0.50)	61.18 (± 0.54)	65.83 (± 0.50)	62.78 (± 0.52)	63.40 (± 0.52)	63.54 (± 0.52)	63.96 (± 0.53)
0.02	54.51 (± 0.57)	45.97 (± 0.64)	52.01 (± 0.57)	55.69 (± 0.56)	50.83 (± 0.59)	52.92 (± 0.58)	51.95 (± 0.58)
0.04	44.93 (± 0.63)	35.90 (± 1.21)	42.71 (± 0.64)	45.35 (± 0.60)	40.69 (± 0.66)	44.44 (± 0.64)	39.86 (± 0.67)
0.06	38.68 (± 0.69)	27.08 (± 0.83)	40.14 (± 0.71)	37.50 (± 0.71)	35.49 (± 0.75)	39.58 (± 0.67)	35.42 (± 0.73)
0.08	33.40 (± 0.77)	20.83 (± 0.90)	31.04 (± 0.73)	32.99 (± 0.73)	30.84 (± 0.81)	33.54 (± 0.74)	32.71 (± 0.78)
0.10	30.49 (± 0.82)	18.61 (± 1.44)	28.40 (± 0.81)	28.26 (± 0.75)	27.01 (± 0.86)	29.93 (± 0.77)	27.01 (± 0.80)
0.15	23.75 (± 0.90)	15.76 (± 1.15)	24.58 (± 0.88)	24.44 (± 0.91)	21.60 (± 0.99)	22.85 (± 0.96)	21.32 (± 1.04)

Fonte: Dados da pesquisa.

Figura 33 – SURF: ruído Multiplicativo.



Fonte: Elaborado pela autora.

De modo geral, nos testes realizados para o algoritmo SURF é possível constatar que os filtros anisotrópicos foram relativamente melhores, ao passo que, o S&P é o ruído de pior caso com taxas de acerto abaixo de 5%.

4.6 Experimento 2: Testes com ruídos e filtros: SIFT

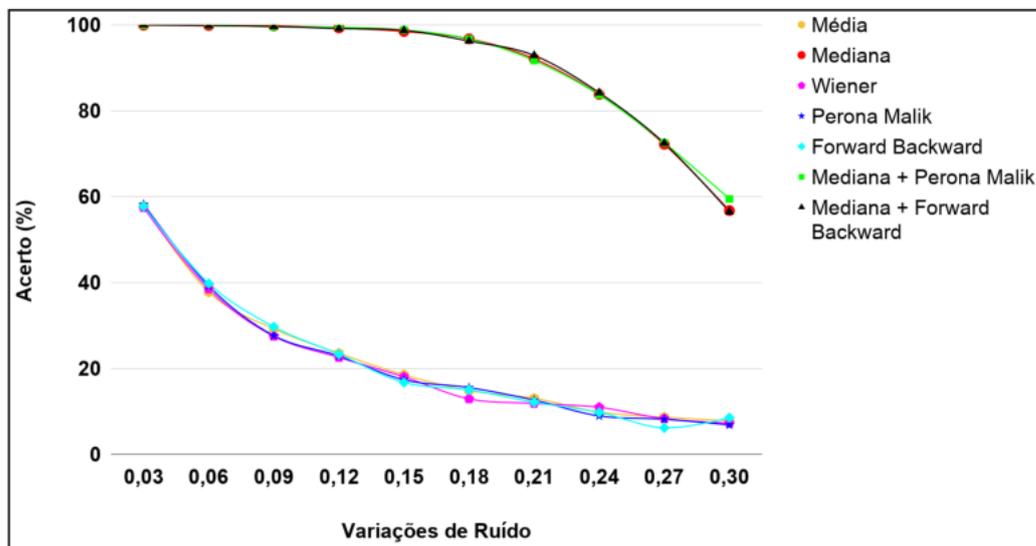
Este experimento visa encontrar dentre os métodos de filtragem analisados, o que resulta em maior percentual de acerto para todos os ruídos gerados. O SIFT evidencia resultados superiores ao SURF, levando em consideração as 4 tipos de degradações. Para o ruído do tipo impulsivo são maiores as taxas de acerto obtidas com a combinação do filtro da mediana com anisotrópicos e somente mediana, apresentado bom desempenho com elevação do ruído até 21% quando inicia o decaimento, como pode ser observado na [Tabela 8](#) e no gráfico da [Figura 34](#). Nesta etapa, os filtros da média, Wiener, Perona Malik e *Forward-Backward* são os que se identificaram como pior caso.

Tabela 8 – SIFT: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo S&P.

Ruído S&P							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.03	57.78 (± 0.60)	99.86 (± 0.04)	57.43 (± 0.60)	58.19 (± 0.61)	57.71 (± 0.59)	99.93 (± 0.03)	100.00
0.06	37.78 (± 0.72)	99.79 (± 0.05)	38.47 (± 0.73)	39.17 (± 0.73)	39.79 (± 0.68)	99.93 (± 0.03)	99.79 (± 0.05)
0.09	29.24 (± 0.81)	99.72 (± 0.05)	27.50 (± 0.80)	27.64 (± 0.81)	29.72 (± 0.82)	99.51 (± 0.07)	99.58 (± 0.06)
0.12	23.54 (± 0.93)	99.24 (± 0.09)	22.57 (± 0.93)	22.92 (± 0.87)	23.33 (± 0.91)	99.37 (± 0.08)	99.10 (± 0.09)
0.15	18.54 (± 1.02)	98.40 (± 0.13)	18.06 (± 0.98)	17.36 (± 1.01)	16.74 (± 0.97)	98.75 (± 0.11)	98.68 (± 0.11)
0.18	14.86 (± 1.06)	96.73 (± 0.18)	12.92 (± 1.12)	15.56 (± 1.10)	15.00 (± 1.10)	96.67 (± 0.18)	96.18 (± 0.19)
0.21	13.06 (± 1.21)	92.08 (± 0.28)	11.87 (± 1.15)	12.64 (± 1.22)	12.15 (± 1.18)	91.67 (± 0.29)	92.85 (± 0.27)
0.24	9.86 (± 1.26)	83.75 (± 0.39)	10.97 (± 1.30)	8.96 (± 1.27)	9.79 (± 1.30)	83.68 (± 0.41)	84.17 (± 0.38)
0.27	8.68 (± 1.47)	72.15 (± 0.50)	8.33 (± 1.40)	8.19 (± 1.46)	6.18 (± 1.58)	72.50 (± 0.51)	72.43 (± 0.51)
0.30	7.85 (± 1.51)	56.74 (± 0.59)	7.29 (± 1.50)	6.87 (± 1.48)	8.54 (± 1.48)	59.45 (± 0.60)	56.46 (± 0.61)

Fonte: Dados da pesquisa.

Figura 34 – SIFT: ruído S&P.



Fonte: Elaborado pela autora.

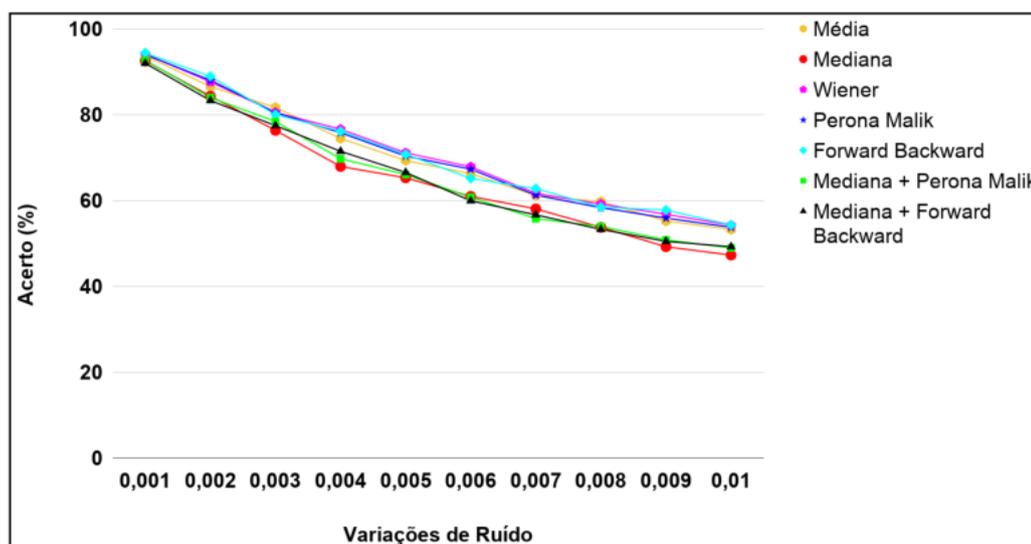
O ruído Gaussiano, assim como no SURF, apresenta decaimento em todos os resultados dos filtros, como exibe o gráfico de linhas da Figura 35, ambos os métodos de mosaico resultam em gráficos semelhantes. Porém, o SIFT possui taxas de acerto mais elevadas quando comparado ao ruído Gaussiano no SURF. Neste contexto, o filtro de *Forward-Backward* é o mais adequado, contendo mais taxas de acerto, e o menos indicado é o filtro da mediana (Tabela 9).

Tabela 9 – SIFT: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Gaussiano.

Ruído Gaussiano							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.001	93.68 (± 0.26)	92.50 (± 0.27)	94.10 (± 0.24)	93.96 (± 0.24)	94.37 (± 0.23)	92.64 (± 0.27)	91.95 (± 0.28)
0.002	86.53 (± 0.37)	84.31 (± 0.39)	87.64 (± 0.35)	87.99 (± 0.34)	88.89 (± 0.28)	83.96 (± 0.38)	83.26 (± 0.40)
0.003	81.67 (± 0.42)	76.32 (± 0.46)	80.49 (± 0.42)	80.49 (± 0.43)	80.00 (± 0.43)	78.40 (± 0.44)	77.43 (± 0.45)
0.004	74.44 (± 0.48)	67.92 (± 0.53)	76.60 (± 0.45)	75.76 (± 0.47)	76.11 (± 0.47)	69.72 (± 0.52)	71.46 (± 0.50)
0.005	69.31 (± 0.51)	65.28 (± 0.55)	71.11 (± 0.52)	70.35 (± 0.50)	70.76 (± 0.51)	66.04 (± 0.55)	66.53 (± 0.55)
0.006	66.25 (± 0.54)	60.97 (± 0.57)	67.85 (± 0.53)	67.29 (± 0.54)	65.14 (± 0.54)	60.76 (± 0.58)	59.93 (± 0.57)
0.007	61.18 (± 0.56)	58.06 (± 0.59)	61.67 (± 0.56)	61.25 (± 0.56)	62.78 (± 0.56)	55.76 (± 0.60)	56.67 (± 0.59)
0.008	59.72 (± 0.59)	53.75 (± 0.63)	59.17 (± 0.59)	58.33 (± 0.59)	58.47 (± 0.58)	53.96 (± 0.64)	53.26 (± 0.63)
0.009	55.21 (± 0.60)	49.24 (± 0.62)	56.81 (± 0.61)	55.90 (± 0.61)	57.78 (± 0.60)	50.83 (± 0.64)	50.49 (± 0.66)
0.01	53.26 (± 0.62)	47.29 (± 0.64)	54.24 (± 0.62)	53.75 (± 0.63)	54.31 (± 0.62)	48.96 (± 0.69)	49.24 (± 0.66)

Fonte: Dados da pesquisa.

Figura 35 – SIFT: ruído Gaussiano.



Fonte: Elaborado pela autora.

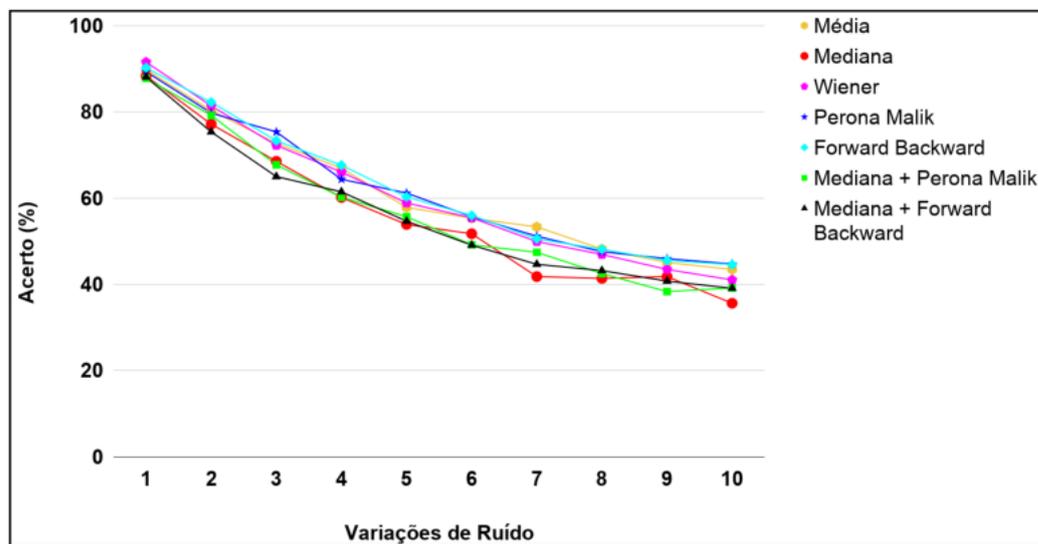
Quando ocorre perturbação na imagem resultante de ruído Poisson, os filtros anisotrópicos diferenciam-se dos demais como melhores alternativas. Analisando a Tabela 10 e o gráfico da Figura 36, ocorre uma perda do percentual de acerto semelhante para todos os métodos de filtragem, porém, com filtros de Perona Malik e *Forward-Backward* sendo sensivelmente melhores. O filtro menos adequado para ruído do tipo Poisson no SURF é o da mediana.

Tabela 10 – SIFT: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Poisson.

Ruído Poisson							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
1	89.79 (± 0.31)	88.33 (± 0.34)	91.53 (± 0.28)	89.31 (± 0.30)	90.21 (± 0.29)	87.78 (± 0.35)	88.06 (± 0.34)
2	80.28 (± 0.43)	77.15 (± 0.46)	81.39 (± 0.42)	79.72 (± 0.39)	82.22 (± 0.37)	79.17 (± 0.42)	75.35 (± 0.47)
3	72.71 (± 0.49)	68.54 (± 0.52)	72.22 (± 0.50)	75.35 (± 0.43)	73.33 (± 0.46)	67.71 (± 0.53)	65.00 (± 0.52)
4	66.94 (± 0.54)	60.14 (± 0.57)	66.04 (± 0.54)	64.37 (± 0.54)	67.64 (± 0.52)	60.21 (± 0.60)	61.46 (± 0.56)
5	57.85 (± 0.56)	53.89 (± 0.59)	58.89 (± 0.59)	61.18 (± 0.61)	60.28 (± 0.59)	55.69 (± 0.65)	54.65 (± 0.63)
6	55.35 (± 0.60)	51.74 (± 0.62)	55.42 (± 0.61)	55.69 (± 0.66)	55.97 (± 0.63)	49.17 (± 0.71)	49.10 (± 0.68)
7	53.33 (± 0.63)	41.81 (± 0.67)	49.93 (± 0.62)	51.18 (± 0.71)	50.56 (± 0.68)	47.43 (± 0.72)	44.65 (± 0.73)
8	48.19 (± 0.65)	41.39 (± 0.65)	46.94 (± 0.63)	47.57 (± 0.75)	48.13 (± 0.70)	42.50 (± 0.77)	43.20 (± 0.74)
9	45.07 (± 0.65)	41.81 (± 0.70)	43.47 (± 0.67)	45.97 (± 0.77)	45.56 (± 0.73)	38.33 (± 0.81)	40.76 (± 0.77)
10	43.47 (± 0.69)	35.63 (± 0.68)	41.04 (± 0.67)	44.72 (± 0.78)	44.58 (± 0.74)	39.17 (± 0.82)	39.10 (± 0.79)

Fonte: Dados da pesquisa.

Figura 36 – SIFT: ruído Poisson.



Fonte: Elaborado pela autora.

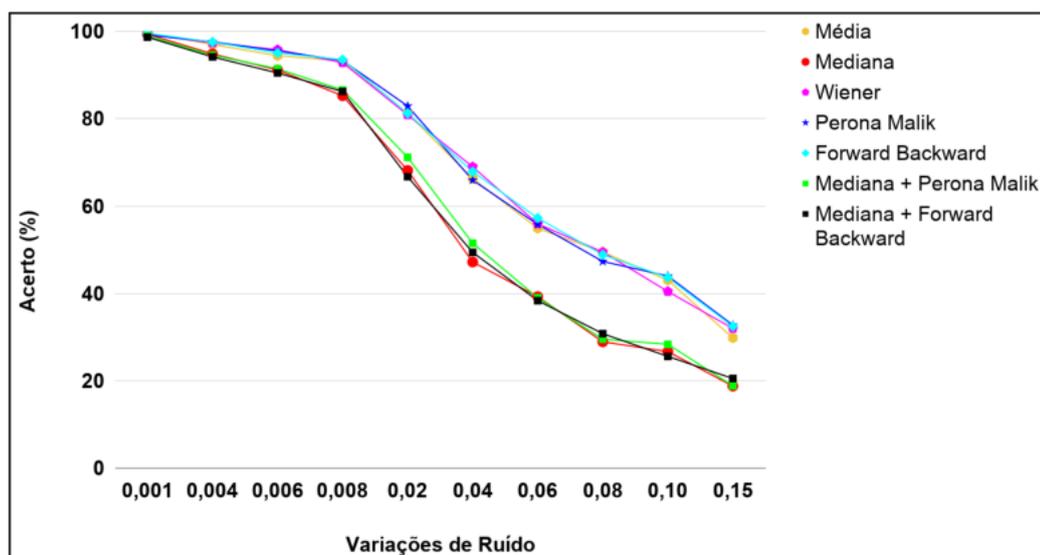
A filtragem da degradação do tipo multiplicativo expressa maior desempenho com filtros da média, anisotrópicos e Wiener. E pior taxa de acerto com filtro da mediana, como pode-se observar na análise da Tabela 11 e da Figura 37.

Tabela 11 – SIFT: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Multiplicativo.

Ruído Multiplicativo							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.001	99.65 (± 0.02)	99.37 (± 0.12)	99.37 (± 0.12)	99.17 (± 0.14)	99.58 (± 0.15)	98.96 (± 0.16)	98.68 (± 0.17)
0.004	97.01 (± 0.04)	94.86 (± 0.13)	97.36 (± 0.13)	97.64 (± 0.15)	97.57 (± 0.15)	94.58 (± 0.19)	94.17 (± 0.21)
0.006	94.44 (± 0.07)	91.18 (± 0.13)	95.83 (± 0.15)	95.55 (± 0.19)	95.07 (± 0.19)	91.46 (± 0.20)	90.49 (± 0.24)
0.008	93.26 (± 0.09)	85.28 (± 0.14)	92.85 (± 0.17)	93.33 (± 0.19)	93.54 (± 0.22)	86.60 (± 0.21)	86.25 (± 0.33)
0.02	81.25 (± 0.10)	68.19 (± 0.17)	80.90 (± 0.21)	82.92 (± 0.20)	81.25 (± 0.28)	71.18 (± 0.24)	66.74 (± 0.37)
0.04	66.32 (± 0.13)	47.22 (± 0.21)	69.03 (± 0.24)	65.97 (± 0.20)	67.92 (± 0.29)	51.53 (± 0.25)	49.44 (± 0.41)
0.06	54.93 (± 0.16)	39.31 (± 0.26)	55.97 (± 0.29)	55.84 (± 0.29)	57.29 (± 0.29)	39.03 (± 0.25)	38.40 (± 0.44)
0.08	49.51 (± 0.18)	28.96 (± 0.27)	49.51 (± 0.30)	47.36 (± 0.33)	48.89 (± 0.30)	29.58 (± 0.27)	30.83 (± 0.47)
0.10	43.06 (± 0.20)	26.74 (± 0.31)	40.49 (± 0.33)	44.03 (± 0.34)	43.68 (± 0.30)	28.40 (± 0.27)	25.63 (± 0.49)
0.15	29.86 (± 0.24)	18.82 (± 0.33)	32.01 (± 0.35)	32.78 (± 0.34)	32.50 (± 0.31)	18.89 (± 0.30)	20.56 (± 0.51)

Fonte: Dados da pesquisa.

Figura 37 – SIFT: ruído Multiplicativo.



Fonte: Elaborado pela autora.

É possível concluir a partir dos testes gerados para o SIFT que novamente o ruído de pior caso é o S&P, e que os filtros de Perona Malik, *Forward-Backward* ou mesmo suas combinações com a mediana apresentaram-se como os mais adequados para este método de mosaico.

4.7 Experimento 3: Testes com ruídos e filtros: PC

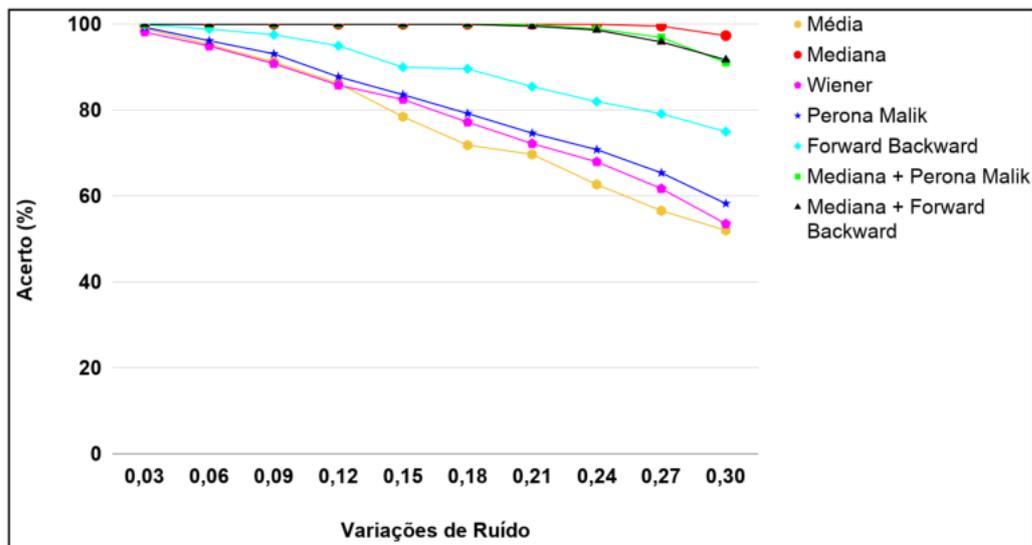
Este experimento compara o desempenho do método de correlação de fase ou PC, além disso, será exposto o quanto este é robusto na filtragem de ruídos. O ruído S&P é melhor filtrado pela mediana, porém as combinações da mediana com anisotrópicos possuem taxas de acerto de até 91% mesmo com 30% de ruído. O filtro que não alcançou bons resultados é o da média, mesmo assim, vale mencionar que até para alta taxa de ruído seu desempenho foi acima de 52%, como ilustra a [Tabela 12](#) e a [Figura 38](#).

Tabela 12 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo S&P.

Ruído S&P							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.03	99.03 (± 0.10)	100	98.19 (± 0.13)	99.31 (± 0.08)	99.93 (± 0.03)	100	100
0.06	95.28 (± 0.21)	100	94.93 (± 0.22)	96.18 (± 0.19)	98.89 (± 0.10)	100	100
0.09	91.25 (± 0.28)	100	90.83 (± 0.29)	93.13 (± 0.25)	97.57 (± 0.15)	100	100
0.12	86.25 (± 0.29)	100	85.83 (± 0.35)	87.78 (± 0.33)	95.00 (± 0.22)	100	100
0.15	78.47 (± 0.42)	100	82.50 (± 0.38)	83.61 (± 0.37)	90.00 (± 0.30)	100	100
0.18	71.87 (± 0.46)	100	77.22 (± 0.43)	79.24 (± 0.41)	89.65 (± 0.31)	100	100
0.21	69.72 (± 0.48)	100	72.22 (± 0.46)	74.65 (± 0.44)	85.49 (± 0.35)	99.93 (± 0.03)	99.58 (± 0.06)
0.24	62.71 (± 0.50)	100	67.99 (± 0.44)	70.83 (± 3.16)	82.01 (± 0.38)	98.96 (± 0.10)	98.68 (± 0.11)
0.27	56.60 (± 0.53)	99.59 (± 0.06)	61.74 (± 0.50)	65.42 (± 16.01)	79.17 (± 0.40)	96.94 (± 0.17)	95.90 (± 0.20)
0.30	52.01 (± 0.54)	97.36 (± 0.16)	53.54 (± 0.52)	58.26 (± 26.87)	75.00 (± 2.66)	91.11 (± 0.29)	91.74 (± 0.28)

Fonte: Dados da pesquisa.

Figura 38 – PC: ruído S&P.



Fonte: Elaborado pela autora.

Diferentemente dos dois métodos de mosaico vistos anteriormente, para a PC foi necessário aplicar 20 variações do ruído Gaussiano. O chamado Gaussiano(1) é o mesmo aplicado no SURF e no SIFT, todavia, como será visto a seguir, a PC mostrou-se robusta para este tipo de ruído. Dessa forma, optou-se por elevar o nível de degradação da imagem além dos pré-definidos anteriormente, é o chamado Gaussiano (2) que foi aplicado somente para a PC. Os resultados desta maior degradação foram ilustrados na [Figura 23](#).

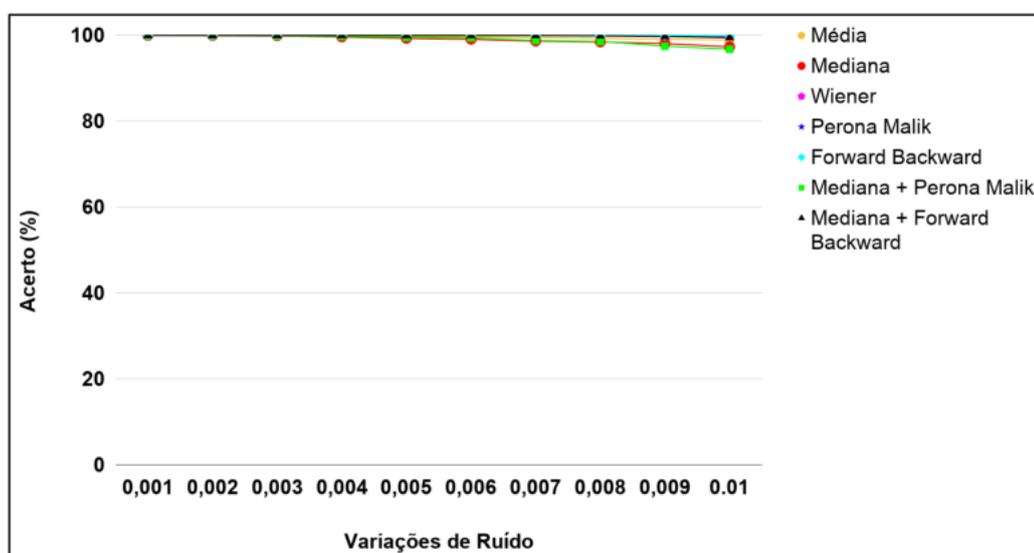
A [Tabela 13](#) abaixo ilustra os resultados do Gaussiano (1), como pode-se observar, os filtros anisotrópicos obtiveram taxas de acerto de até 100% até para a imagem mais corrompida. Isto também pode ser analisado pelo gráfico da [Figura 39](#).

Tabela 13 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Gaussiano (1).

Ruído Gaussiano (1)							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.001	100	100	100	100	100	100	100
0.002	99.93 (± 0.11)	100	100	100	100	100	100
0.003	99.93 (± 0.07)	99.93 (± 0.05)	100	100	100	99.93 (± 0.03)	100
0.004	99.86 (± 0.09)	99.65 (± 0.06)	100	100	100	99.79 (± 0.05)	100
0.005	99.79 (± 0.16)	99.31 (± 0.03)	100	100	100	99.58 (± 0.06)	100
0.006	99.65 (± 0.12)	99.10 (± 0.14)	100	100	100	99.44 (± 0.07)	100
0.007	99.58 (± 0.12)	98.68 (± 0.13)	100	100	100	98.82 (± 0.11)	100
0.008	99.31 (± 0.08)	98.47 (± 0.22)	100	100	100	98.61 (± 0.12)	99.93 (± 0.03)
0.009	99.10 (± 0.16)	98.12 (± 0.22)	99.93 (± 0.29)	100	100	97.50 (± 0.16)	99.79 (± 0.05)
0.01	98.96 (± 0.13)	97.36 (± 0.23)	99.86 (± 0.30)	100	100	96.80 (± 0.18)	99.51 (± 0.07)

Fonte: Dados da pesquisa.

Figura 39 – PC: ruído Gaussiano (1).



Fonte: Elaborado pela autora.

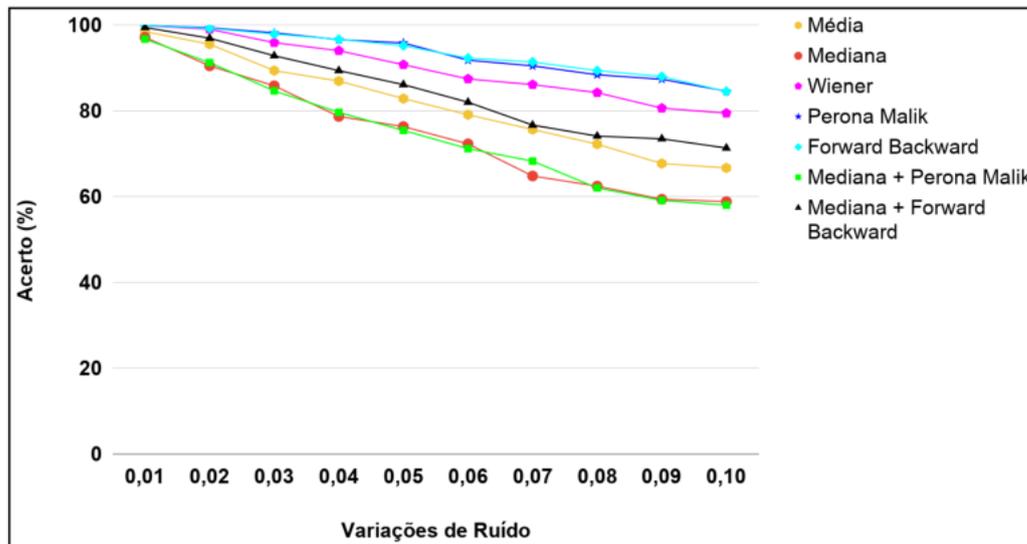
À vista disso, a Tabela 14 mostra os resultados da elevação da degradação das imagens e torna-se possível notar que ocorrem alterações no desempenho de cada filtro. O nível de ruído foi aumentado para até $\sigma^2 = 0.10$. Assim, ao realizar a análise constata-se que os filtros anisotrópicos novamente se mostraram como sendo os mais apropriados. Neste caso, o filtro de pior caso é o da mediana e, contudo, possui taxas de acerto acima de 58% para imagens com alto nível de ruído.

Tabela 14 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Gaussiano (2).

Ruído Gaussiano (2)							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.01	98.54 (± 0.13)	97.22 (± 0.15)	99.93 (± 0.02)	100	100	96.80 (± 0.21)	99.51 (± 0.05)
0.02	95.62 (± 0.22)	90.56 (± 0.35)	99.10 (± 0.06)	99.44 (± 0.10)	99.31 (± 0.08)	91.32 (± 0.28)	97.01 (± 0.16)
0.03	89.45 (± 0.31)	85.90 (± 0.37)	95.97 (± 0.18)	98.26 (± 0.12)	97.98 (± 0.16)	84.72 (± 0.34)	92.92 (± 0.28)
0.04	87.01 (± 0.33)	78.75 (± 0.42)	94.10 (± 0.23)	96.67 (± 0.21)	96.74 (± 0.19)	79.72 (± 0.42)	89.45 (± 0.31)
0.05	82.92 (± 0.35)	76.39 (± 0.44)	90.84 (± 0.30)	95.90 (± 0.23)	95.28 (± 0.39)	75.49 (± 0.44)	86.18 (± 0.33)
0.06	79.17 (± 0.39)	72.36 (± 0.45)	87.50 (± 0.39)	91.88 (± 0.25)	92.36 (± 0.41)	71.25 (± 0.47)	82.08 (± 0.39)
0.07	75.70 (± 0.42)	64.86 (± 0.49)	86.18 (± 0.44)	90.56 (± 0.31)	91.46 (± 0.43)	68.33 (± 0.54)	76.74 (± 0.41)
0.08	72.29 (± 0.48)	62.50 (± 0.52)	84.31 (± 0.49)	88.47 (± 0.34)	89.45 (± 0.48)	62.08 (± 0.57)	74.17 (± 0.45)
0.09	67.78 (± 0.49)	59.44 (± 0.57)	80.70 (± 0.55)	87.43 (± 0.37)	88.06 (± 0.52)	59.17 (± 0.61)	73.54 (± 0.49)
0.10	66.74 (± 0.55)	58.89 (± 0.63)	79.52 (± 0.67)	84.65 (± 0.57)	84.58 (± 0.39)	58.06 (± 0.73)	71.39 (± 0.51)

Fonte: Dados da pesquisa.

Figura 40 – PC: ruído Gaussiano (2).



Fonte: Elaborado pela autora.

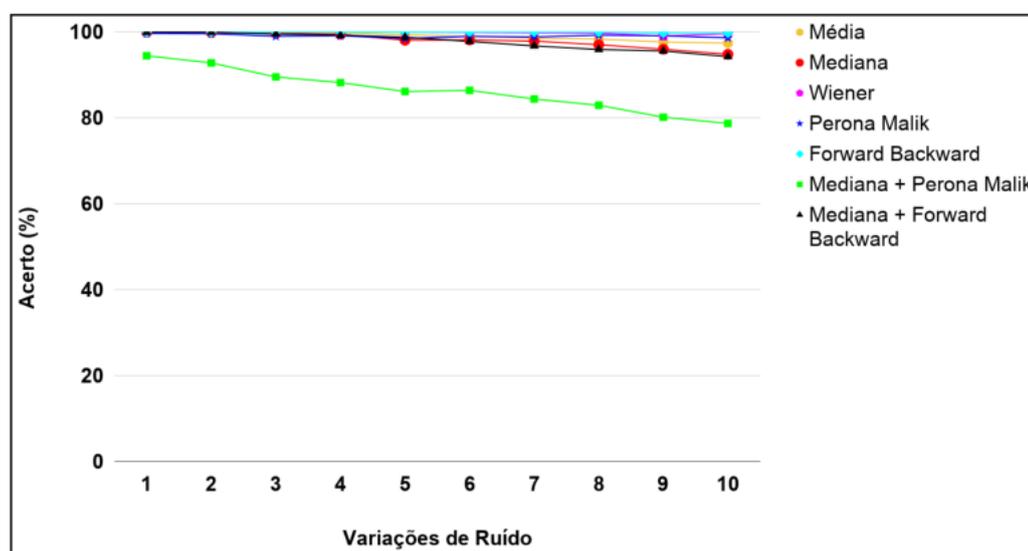
A Tabela 15 e a Figura 41 mostram que o melhor filtro para o ruído Poisson é o de *Forward-Backward* e o filtro de pior caso é a combinação da mediana com Perona Malik.

Tabela 15 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Poisson.

Ruído Poisson							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
1	100	100	100	99.72 (± 0.07)	100	94.51 (± 0.12)	100
2	100	99.93 (± 0.03)	100	99.65 (± 0.05)	100	92.85 (± 0.13)	100
3	99.93 (± 0.02)	99.79 (± 0.05)	100	99.03 (± 0.09)	100	89.58 (± 0.31)	99.58 (± 0.14)
4	99.86 (± 0.13)	99.51 (± 0.09)	100	99.17 (± 0.16)	100	88.26 (± 0.22)	99.31 (± 0.17)
5	99.44 (± 0.09)	98.12 (± 0.14)	100	98.54 (± 0.19)	100	86.18 (± 0.29)	98.82 (± 0.20)
6	99.03 (± 0.09)	98.19 (± 0.12)	99.93 (± 0.15)	99.03 (± 0.08)	100	86.46 (± 0.23)	97.85 (± 0.18)
7	98.61 (± 0.12)	97.92 (± 0.16)	99.86 (± 0.21)	98.89 (± 0.14)	100	84.44 (± 0.25)	96.80 (± 0.20)
8	98.40 (± 0.15)	97.08 (± 0.19)	99.79 (± 0.20)	99.31 (± 0.17)	100	82.99 (± 0.27)	95.97 (± 0.22)
9	97.71 (± 0.18)	96.04 (± 0.21)	99.10 (± 0.20)	99.17 (± 0.19)	99.79 (± 0.04)	80.21 (± 0.30)	95.62 (± 0.29)
10	97.43 (± 0.21)	94.86 (± 0.27)	99.65 (± 0.23)	98.68 (± 0.23)	99.79 (± 0.06)	78.75 (± 0.33)	94.31 (± 0.20)

Fonte: Dados da pesquisa.

Figura 41 – PC: ruído Poisson.



Fonte: Elaborado pela autora.

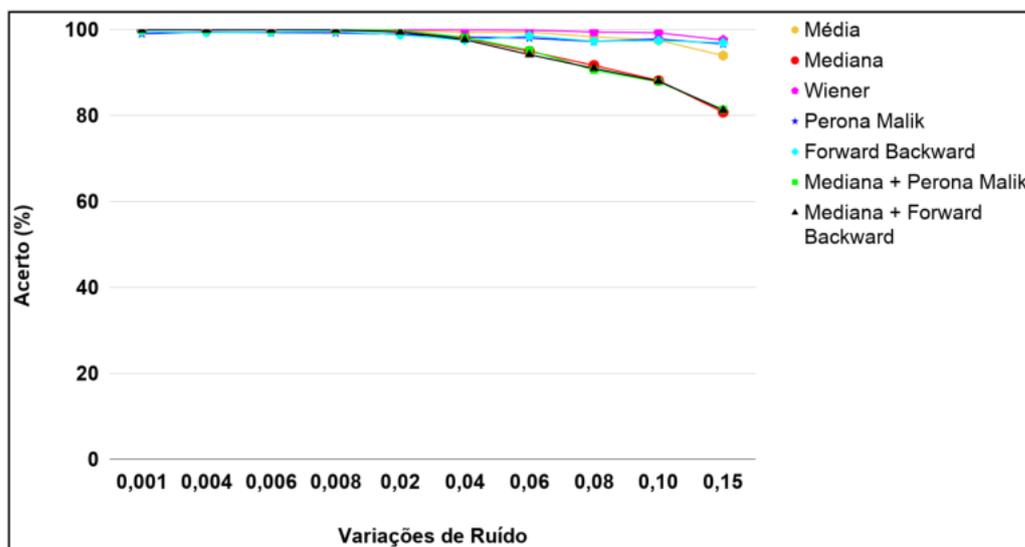
Por fim, para o multiplicativo, o filtro de Wiener alcançou os melhores resultados mas não difere muito dos demais filtros, pois em todos os casos, o percentual de acerto é acima de 80%, como apresenta a Tabela 16 e a Figura 42. O filtro menos adequado é o da mediana.

Tabela 16 – PC: Resultados obtidos para diferentes modelos de filtros com variação progressiva do nível de ruído do tipo Multiplicativo.

Ruído Multiplicativo							
Acerto (%) e Desvio Padrão							
Variações/ Filtros	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
0.001	100	100	100	99.10 (± 0.27)	99.58 (± 0.26)	100	100
0.004	100	100	100	99.51 (± 0.28)	99.37 (± 0.27)	100	100
0.006	100	100	100	99.37 (± 0.29)	99.51 (± 0.24)	100	100
0.008	100	100	100	99.31 (± 0.28)	99.65 (± 0.25)	100	100
0.02	99.93 (± 0.10)	99.58 (± 0.13)	100	98.96 (± 0.08)	98.82 (± 0.11)	99.93 (± 0.06)	99.51 (± 0.52)
0.04	99.44 (± 0.11)	98.05 (± 1.78)	100	98.33 (± 0.98)	97.57 (± 0.96)	98.26 (± 0.90)	97.71 (± 0.53)
0.06	99.44 (± 0.12)	95.00 (± 2.33)	99.93 (± 0.07)	98.12 (± 1.08)	98.61 (± 1.05)	95.28 (± 1.12)	94.24 (± 0.60)
0.08	98.40 (± 0.15)	91.74 (± 2.56)	99.51 (± 0.13)	97.29 (± 1.11)	97.36 (± 1.17)	90.63 (± 1.19)	91.04 (± 0.85)
0.10	97.57 (± 0.97)	88.20 (± 3.02)	99.31 (± 0.92)	97.85 (± 1.30)	97.50 (± 1.32)	87.92 (± 1.56)	88.06 (± 1.55)
0.15	94.03 (± 1.19)	80.83 (± 3.10)	97.64 (± 1.30)	96.73 (± 1.32)	97.01 (± 1.45)	81.46 (± 2.24)	81.32 (± 2.37)

Fonte: Dados da pesquisa.

Figura 42 – PC: ruído Multiplicativo.



Fonte: Elaborado pela autora.

O método de PC mostrou-se como o mais robusto quando adicionados os ruídos propostos. Neste experimento, os filtros anisotrópicos são os mais adequados para os ruídos do tipo Gaussiano e Poisson. Além disso, é inapropriado mencionar que os filtros da média e mediana são os de pior caso, pois os mesmos atingiram bons resultados mesmo para imagens com alto nível de ruído. Neste método de mosaico o ruído de pior caso também é considerado o S&P.

4.8 Resumo dos resultados dos testes com ruídos e filtros

Esta seção objetiva resumir os resultados vistos anteriormente. O Quadro 2 abaixo mostra quais os filtros adequados para cada tipo de ruído e método de mosaico. É possível observar que os filtros anisotrópicos são os apropriados neste estudo de imagens microscópicas.

Quadro 2 – Quadro resumo: melhor filtro para cada tipo de ruído.

Ruído/ Métodos	PC	SIFT	SURF
S&P	Mediana	Mediana + Perona Malik	Mediana e Mediana + <i>Forward-Backward</i>
Gaussiano	Perona Malik e <i>Forward-Backward</i>	<i>Forward-Backward</i>	Perona Malik
Poisson	<i>Forward-Backward</i>	Perona Malik	Perona Malik
Multiplicativo	Wiener	Perona Malik	Média e Wiener

Fonte: Dados da pesquisa.

Para o ruído S&P, todos os métodos de mosaico apresentaram melhor desempenho com o filtro da mediana ou sua combinação com os anisotrópicos; os ruídos dos tipos Gaussiano e Poisson são melhor filtrados por anisotrópicos; por fim, para o tipo multiplicativo não há um consenso do método mais adequado, sendo possível utilizar a média, Wiener ou Perona Malik. Em todos os experimentos identificou-se o ruído do tipo S&P como o mais problemático nas imagens.

Torna-se importante ressaltar que a partir dos dados e análise de todas as tabelas vistas nos 3 experimentos, nem sempre a combinação de mediana com filtros anisotrópicos se mostrou mais eficiente. Em alguns casos, comparando-se os resultados da mediana isolada com sua combinação com os anisotrópicos, a taxa de acerto para a combinação decaiu em alguns experimentos. Conjectura-se que isso decorre de borramento quando aplica-se um segundo filtro na imagem. Por outro lado, os filtros anisotrópicos foram os que geraram os melhores resultados para as bases de imagens apresentadas, aprimorando assim, os principais métodos de mosaico apresentados neste trabalho.

4.9 Experimento 4: Comparativo entre os métodos de mosaico

Neste experimento, os testes foram realizados sem o uso de ruídos, pois o objetivo é identificar os métodos mais robustos para a base de imagens microscópicas apresentadas, bem como visa constatar quais filtros são mais adequados para aprimorar cada abordagem.

Considere nas tabelas abaixo SR/SF uma abreviatura para “Sem Ruído e Sem Filtros”. A [Tabela 17](#) mostra os resultados da aplicação dos métodos de junção para a base com ampliação de $5\times$. O SURF possui uma taxa de acerto (%) de 86.67 e que, neste caso, os filtros de Perona Malik e a combinação do filtro da mediana com *Forward-Backward* são os métodos que aprimoram o resultado do SURF para 92.22%, em ambos os casos. Note que o desvio padrão de cada taxa de acerto é pequeno.

Tabela 17 – Comparativo entre métodos: Base de imagens com ampliação de 5×.

Acerto (%) e Desvio Padrão								
Base de imagens com ampliação de 5x								
Métodos/ Filtros	SR/SF	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
SURF	86.67 (±0.35)	88.89 (±0.31)	85.56 (±0.37)	86.67 (±0.34)	92.22 (±0.27)	88.89 (±0.31)	87.78 (±0.34)	92.22 (±0.29)
SIFT	100	100	100	100	100	100	100	100
PC	100	100	100	100	100	100	100	100

Fonte: Dados da pesquisa.

A Tabela 18 apresenta as deliberações para o conjunto de imagens com aumento de 10×. Neste cenário, o SURF apresenta uma queda de desempenho em comparação com a base anterior, com 77.61% de acerto e com o filtro da média alcança um melhor resultado com 88.81%.

Tabela 18 – Comparativo entre métodos: Base de imagens com ampliação de 10×.

Acerto (%) e Desvio Padrão								
Base de imagens com ampliação de 10x								
Métodos/ Filtros	SR/SF	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
SURF	77.61 (±0.43)	88.81 (±0.32)	84.33 (±0.372)	85.82 (±0.367)	79.85 (±0.46)	78.36 (±0.43)	85.07 (±0.36)	80.6 (±0.41)
SIFT	100	100	100	100	100	100	100	100
PC	100	100	100	100	100	100	100	100

Fonte: Dados da pesquisa.

Para os dois experimentos anteriores não se obteve decaimento das porcentagens no SIFT e na PC. Entretanto, para a base de imagens capturada com a Lupa, tanto o SURF quanto o SIFT apresentaram alterações. O SURF atingiu 89.55% de acerto SR/SF e com o uso do filtro de Wiener a taxa aumentou para 91.54%. O SIFT não mostrou uma melhora significativa, apenas de 0.5% com o filtro anisotrópico de *Forward-Backward*.

Tabela 19 – Comparativo entre métodos: Base de imagens da Lupa.

Acerto (%) e Desvio Padrão								
Base de imagens obtida com a Lupa								
Métodos/ Filtros	SR/SF	Média	Mediana	Wiener	Perona Malik	Forward Backward	Mediana+ Perona Malik	Mediana+ Forward Backward
SURF	89.55 (±0.31)	86.07 (±0.35)	87.56 (±0.34)	91.54 (±0.29)	90.55 (±0.28)	89.55 (±0.31)	79.60 (±0.40)	87.56 (±0.34)
SIFT	96.02 (±0.21)	93.53 (±15.69)	90.55 (±13.31)	94.53 (±13.31)	94.53 (±9.52)	96.52 (±0.20)	92.04 (±13.31)	92.54 (±13.31)
PC	100	100	100	100	100	100	100	100

Fonte: Dados da pesquisa.

Nos três experimentos anteriores o método de mosaico baseado na PC sempre apresentou um percentual de acerto de 100%, isso deve-se ao fato de que as imagens foram capturadas de forma controlada, sem alterações geométricas de escala, rotação ou translação, ou seja, todas foram submetidas às mesmas condições. É importante ressaltar que a PC apresentou bons resultados quando ocorre presença de ruído, como mostrado anteriormente, e que no caso de imagens sem degradações conjectura-se a não alteração em percentuais de acerto.

4.10 Experimento 5: Análise de tempo de processamento

A Tabela 20 apresenta uma comparação do tempo médio de execução dos algoritmos SIFT, SURF e PC. Este processo foi repetido 20 vezes para cada base, totalizando 180 execuções. Foram utilizadas as três bases de imagens disponíveis, e o tempo de todos os métodos foram obtidos com a utilização de um computador com as seguintes configurações: Linux Mint, memória de 8GB, 64-bit, processador Intel Core (TM) i7.

Tabela 20 – Tempo de processamento dos métodos de mosaico.

Tempo de processamento (s)			
	PC	SURF	SIFT
Aumento de 5x	0.3602	1.1852	12.8007
Aumento de 10x	0.3608	0.8815	7.7318
Lupa	0.2046	0.4107	4.0895
Média	0.3085	0.8258	8.2073

Fonte: Dados da pesquisa.

De acordo com a média dos tempos apresentados, é possível observar que a junção de imagens da correlação de fase mostra-se mais eficaz quando comparada com o SIFT e o SURF. O fato da construção de mosaicos ser mais rápida deve-se à utilização da transformada rápida de Fourier que calcula o deslocamento entre as imagens no domínio de frequência. Enquanto que os demais métodos trabalham no domínio espacial e possuem uma forma diferenciada para calcular os vetores de características de cada par de imagem, necessitando de um custo computacional superior à abordagem baseada em Fourier.

No entanto, é importante lembrar que a correlação de fase mostrou-se mais adequada para a aplicação em imagens microscópicas, pois as mesmas são capturadas em um ambiente controlado. Dessa forma, a implementação não exigiu uma extensão do método para tratar de mudanças de escala e rotação, tornando-se, dessa forma, um algoritmo mais simples e eficiente. À medida que o SIFT e o SURF são técnicas invariantes a certas transformações, mas são computacionalmente custosas, de uma forma geral, ou possuem fraco desempenho em alterações de cor ou iluminação na imagem.

4.11 Considerações finais

Neste capítulo foram apresentados os experimentos, resultados e discussões referentes aos métodos de mosaico SIFT, SURF e correlação de fase (PC), com a utilização de bases de imagens microscópicas. A metodologia se baseia em adição de ruídos e filtros às imagens, para posteriormente aplicar correspondência. Estes testes foram realizados com o intuito de encontrar os ruídos mais problemáticos e os filtros mais adequados para cada caso. Observou-se que para os três métodos, o ruído S&P é o mais problemático, e para realizar a filtragem, os filtros

mais apropriados são a mediana e a combinação da mediana com anisotrópicos. É interessante destacar também que os filtros de Perona Malik e *Forward-Backward* foram identificados como abordagens que obtiveram altas taxas de acerto em maior parte dos testes realizados, e que dessa forma, foram capazes de aperfeiçoar os métodos de mosaico apresentados nesta dissertação. Foi também identificado que mesmo com percentuais de acerto superiores ao SURF, o SIFT foi o que obteve maior tempo de processamento, comparando os três métodos disponíveis. A PC pode ser considerada a mais rápida em virtude de operar no domínio de frequência, e é importante ressaltar que não houve necessidade de estender o método pois o mesmo é invariante à translação, o que o tornou mais adequado para as imagens microscópicas objetos deste estudo.

FERRAMENTA DESENVOLVIDA

Este capítulo descreve a ferramenta *MosaicImage* desenvolvida a partir da metodologia proposta neste trabalho. Esse software tem como objetivo a construção automática de mosaicos de imagens microscópicas e funciona como um *plugin* para o software ImageJ, que será descrito nas próximas seções.

No mercado existem variados softwares com a função de geração de mosaico de imagens, pode-se citar, por exemplo, o *AutoStitch* e o *PTGui* que permitem a costura de fotos panorâmicas de forma automática e funcionam em diversos sistemas operacionais, todavia, são softwares pagos. O MosaicJ (THÉVENAZ; UNSER, 2007)³ é uma ferramenta livre, entretanto, a composição do mosaico se sucede de forma semi-automática, ou seja, ocorre a interação com o usuário no processo de junção das imagens. A palavra “usuário” será usada como sinônimo de “pesquisador”.

Assim sendo, a comunidade científica se beneficia do compartilhamento de conhecimento, que diferencia das aplicações tradicionais semi-automáticas ou pagas, com a geração de um *plugin* multiplataforma e concebido nos preceitos de softwares livres. O *MosaicImage* foi implementado para funcionar incorporado ao ImageJ e é voltado especificamente para imagens microscópicas, deste modo, é necessário ter um conhecimento prévio deste software.

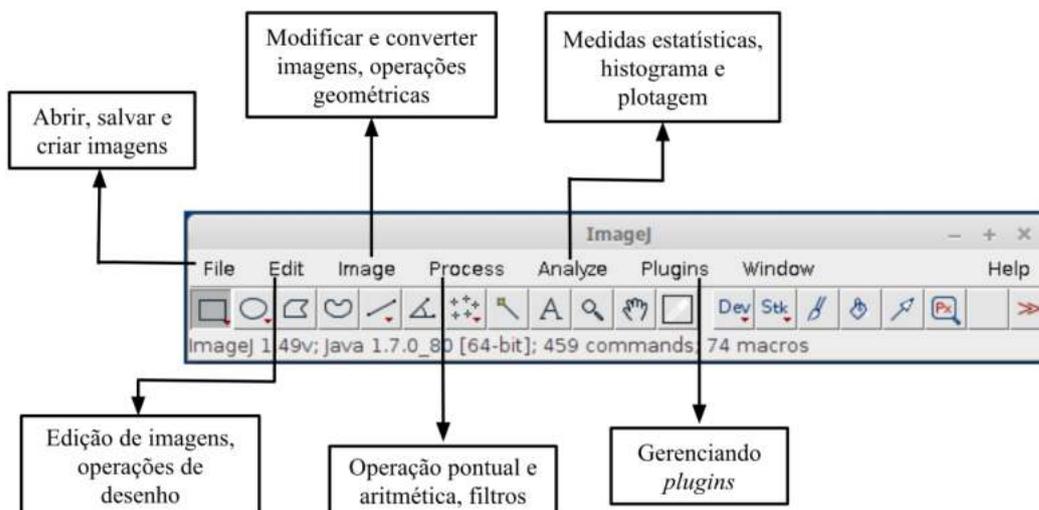
5.1 ImageJ

O ImageJ é um programa de domínio público que realiza a análise e processamento de imagens. Foi desenvolvido por Wayne Rasband no *National Institute of Mental Health*, USA, em linguagem de programação Java. Ele é executado, seja como um *applet on-line* ou como um aplicativo para desktop, em qualquer computador com uma máquina virtual Java 1.5 ou posterior. As distribuições disponíveis são para Windows, Mac OS X e Linux. Este software pode exibir,

³ *MosaicJ*. Disponível em: <<http://bigwww.epfl.ch/thevenaz/mosaicj/>>.

editar, analisar, processar, salvar e imprimir imagens de 8 bits, 16 bits e 32 bits. Pode ler muitos formatos de imagem, incluindo TIFF, GIF, JPEG, BMP, DICOM, FITS e *raw* (FERREIRA; RASBAND, 2012). A janela principal do ImageJ é ilustrada na Figura 43.

Figura 43 – Janela principal do ImageJ.



Fonte: (BURGER; BURGE, 2016)

O ImageJ inicia e tem em sua janela principal as seguintes funcionalidades (BURGER; BURGE, 2016):

- **File:** abrir, guardar e criar novas imagens;
- **Edit:** editar e desenhar em imagens;
- **Image:** modificar e converter imagens;
- **Process:** ferramentas para o processamento de imagens como filtros e ruídos;
- **Analyze:** medidas estatísticas sobre os dados presentes nas imagens, realização de histogramas, dentre outros;
- **Plugins:** editar, compilar, executar e gerenciar *plugins* definidos por usuários.

5.1.1 Plugins do ImageJ

Os *plugins* são pequenos módulos desenvolvidos em Java para estender a funcionalidade do ImageJ, e podem ser criados, editados e compilados através do menu **Plugins** na janela principal, onde podem ser adicionados aleatoriamente. Existem dois tipos principais de *plugins*:

- **Plugin:** não necessita que uma imagem seja aberta antes de ser iniciado; e
- **PlugInFilter:** a imagem que está atualmente aberta é passada para o *plugin* quando o mesmo é iniciado.

Conforme [Burger e Burge \(2016\)](#), a escolha entre as alternativas citadas depende de uma questão de preferência, tendo em vista que ambas as versões possuem suas vantagens e desvantagens.

5.2 MosaicImage

O *plugin MosaicImage* realiza o mosaico de imagens microscópicas utilizando o método de correlação de fase apresentado na [Subseção 2.1.2](#). Este método de mosaicagem foi selecionado em virtude de fazer uso da FFT, possuindo, assim, rápido processamento; ser robusto à adição de ruídos às imagens; e ser invariante à translação. O filtro de *Forward-Backward* ([Subseção 3.3.5](#)) foi adicionado ao método de correlação de fase por apresentar resultados satisfatórios para diversos tipos de ruídos, conforme pode ser observado no [Capítulo 5](#).

O *MosaicImage* foi desenvolvido em Java Swing, que é uma biblioteca gráfica oficial inclusa no JRE ou JDK, com a versão 1.8.0.212 do Java, em conjunto com a IDE NetBeans 8.0.1. A [Figura 44](#) apresenta uma visão geral da ferramenta.

Figura 44 – Visão geral do código fonte do *MosaicImage*.

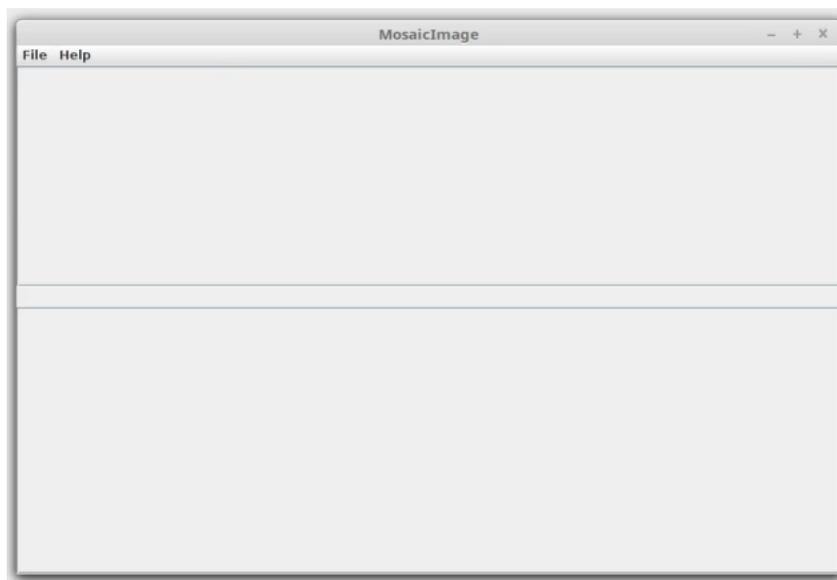
```
1 public class MosaicImage_ extends javax.swing.JFrame{
2     // Definição de atributos;
3     ...
4     // Inicia os componentes gráficos:
5     public MosaicImage_() {
6         initComponents();
7     }
8     // Método da ação que abre a janela de créditos da aplicação:
9     private void subMenuAboutActionPerformed(java.awt.event.ActionEvent evt){...}
10
11 // Método da ação ao clicar no submenu Open Image:
12 private void subMenuOpenImageActionPerformed(java.awt.event.ActionEvent evt){...}
13
14 // Método da ação ao clicar no submenu Open Image Folder:
15 private void subMenuOpenImageFolderActionPerformed(java.awt.event.ActionEvent evt){...}
16
17 // Método da ação ao clicar no submenu Clear Images:
18 private void clearImagesActionPerformed(java.awt.event.ActionEvent evt){...}
19
20 // Método da ação ao clicar no submenu Create Mosaic:
21 private void createMoisaicActionPerformed(java.awt.event.ActionEvent evt){...}
22
23 // Método da ação ao clicar no submenu Save Mosaic:
24 private void saveMoisaicActionPerformed(java.awt.event.ActionEvent evt){...}
25
26 // Método da ação ao clicar no submenu Exit:
27 private void exitPluginActionPerformed(java.awt.event.ActionEvent evt){...}
28
29 // Método principal:
30 public static void main(String args[]) {
31     public void run(){...}
32 }
33 }// Fim da classe
```

Fonte: Elaborada pela autora.

O aplicativo funciona como um *plugin* para o ImageJ, dessa forma, é necessário ter conhecimento da documentação do ImageJ nos processos de instalação de *plugins*⁴ (FERREIRA; RASBAND, 2012). Basicamente, todos os *plugins* devem ser adicionados na pasta **plugins** que é uma sub-pasta onde está instalado o ImageJ. Contudo, para que sejam exibidos automaticamente no menu “*Plugins*”, o arquivo da classe deve ser renomeado com sublinhado ao final do nome da classe.

Após adicionar o *plugin* ao ImageJ e selecionar “*MosaicImage*” na janela principal **Plugins** do ImageJ, a janela inicial do *MosaicImage* é aberta, como ilustra a Figura 45.

Figura 45 – Janela principal do *MosaicImage*.



Fonte: Elaborado pela autora.

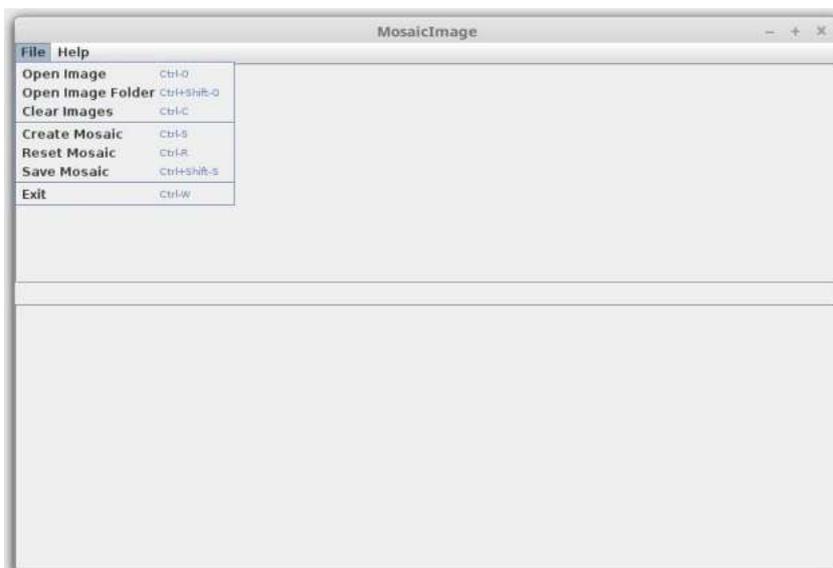
Esta tela inicial contém duas divisões, superior e inferior, implementadas com *JScrollPane* que são painéis de rolagem para exibir um objeto que seja grande ou que o tamanho possa ser alterado dinamicamente. Na divisão inferior serão exibidas as imagens carregadas pelo usuário, essas imagens serão redimensionadas pelo programa. A divisão superior é a área onde o mosaico final será exibido. Ambas as divisões possuem barras de rolagem para facilitar a visualização das imagens, quando necessário.

A barra de menus é composta por dois itens de menu, **File** e **Help**, como pode ser observado na Figura 46 e Figura 47.

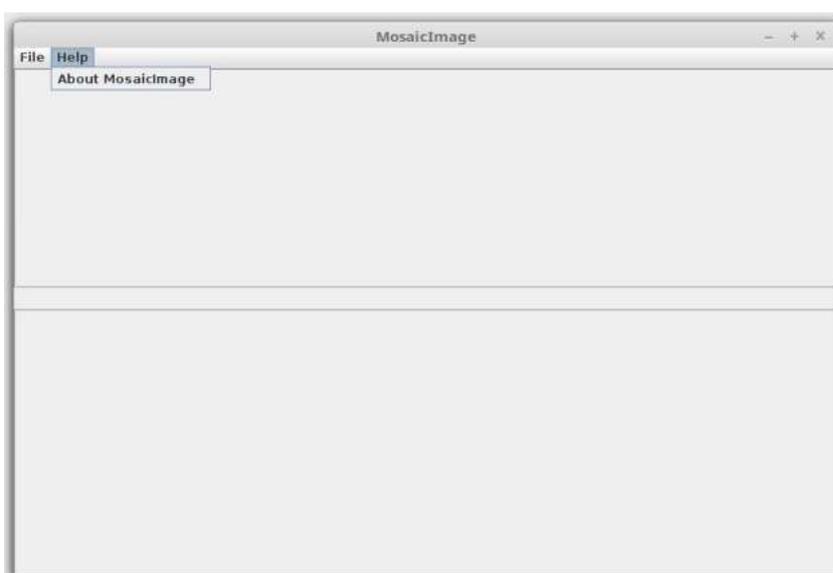
A Figura 46 com o menu **File**, abrange os seguintes itens de submenus:

- **Open Image:** permite o acesso de uma imagem por vez;
- **Open Image Folder:** abre todos os arquivos dentro de um diretório de imagens;
- **Clear Images:** permite limpar a divisão inferior da tela inicial;

⁴ Documentação do ImageJ disponível em: <<https://imagej.nih.gov/ij/docs/index.html>>.

Figura 46 – Submenu File do *plugin* MosaicImage.

Fonte: Elaborado pela autora.

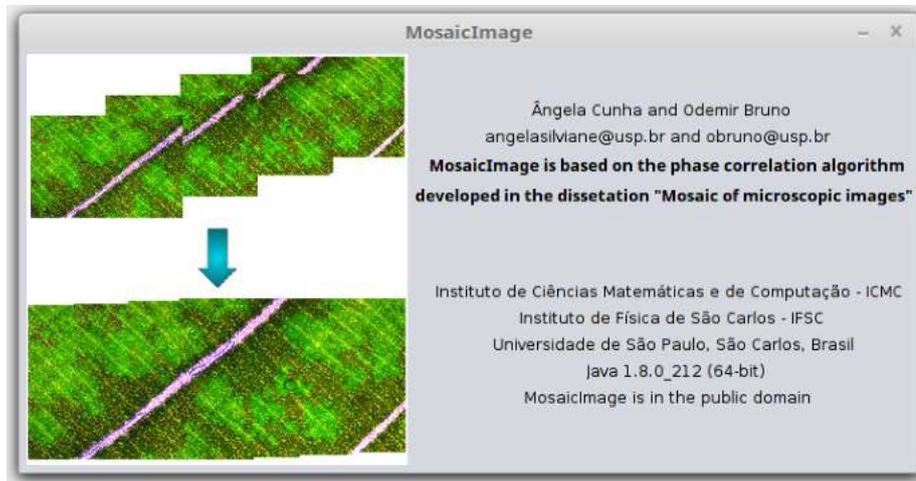
Figura 47 – Submenu Help do *plugin* MosaicImage.

Fonte: Elaborado pela autora.

- **Create Mosaic:** esta opção fica ativa quando existem imagens carregadas da divisão inferior da tela. Quando selecionado, realiza a construção automática do mosaico;
- **Reset Mosaic:** item fica ativo no momento em que existe um mosaico na divisão superior. Quando selecionado, possibilita anular o mosaico construído, ou seja, descartá-lo, e em seguida, limpar a área;
- **Save Mosaic:** esta opção fica ativa quando um mosaico está pronto. Posteriormente é possível salvar o mosaico em um diretório definido pelo usuário;
- **Exit:** é usado para sair da aplicação.

Na [Figura 47](#) quando o submenu **Help** é quando selecionado, exibe a janela de créditos da aplicação, conforme pode ser observado na [Figura 48](#).

Figura 48 – Submenu About do *MosaicImage*.



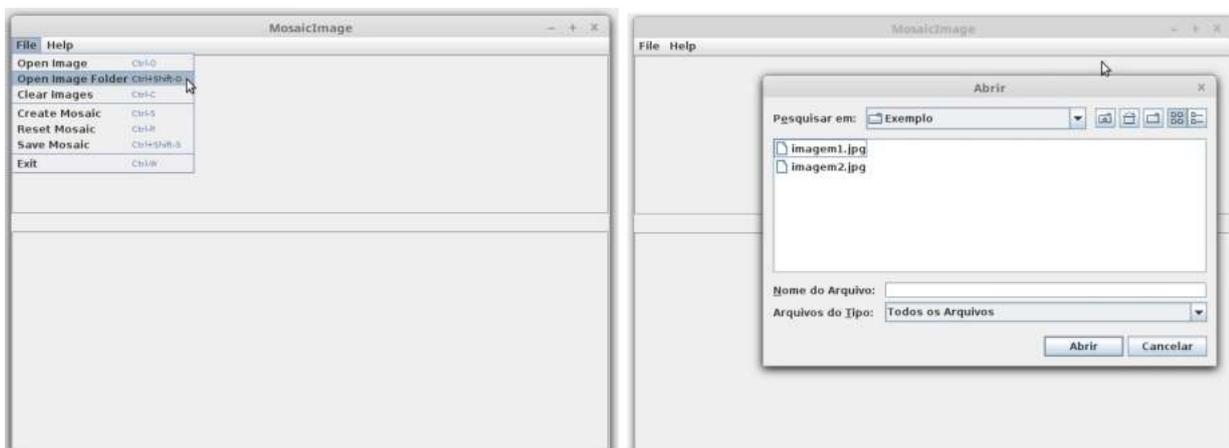
Fonte: Elaborado pela autora.

A aplicação *MosaicImage* foi desenvolvida em ambiente Linux e é definida como sendo de domínio público.

5.2.1 Ferramenta Desenvolvida: Exemplos de uso

Para mostrar a ferramenta em uso, foi definido um pequeno exemplo que realiza a fusão entre duas imagens. A partir do menu “*File*”, pode-se acessar a opção “*Open Image Folder*” que solicita ao usuário o diretório que contém as imagens que serão utilizadas, como pode ser observado na [Figura 49](#) abaixo:

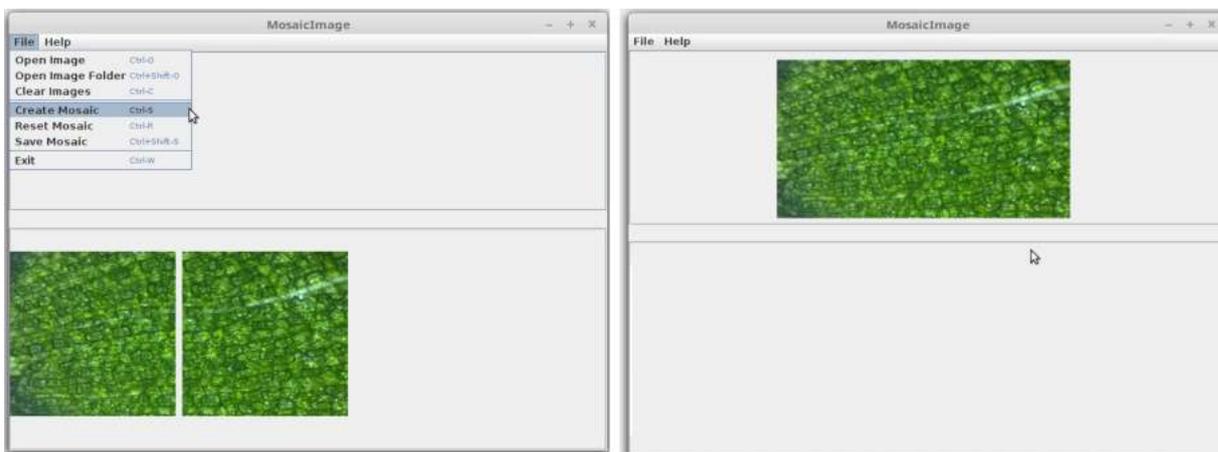
Figura 49 – *MosaicImage*: abrindo um diretório de imagens.



Fonte: Elaborado pela autora.

Como consequência, as imagens do diretório selecionado são exibidas na divisão inferior da aplicação. Em seguida, ao escolher a opção “*Create Mosaic*”, o mosaico é gerado e exibido na divisão superior da tela, conforme a [Figura 50](#):

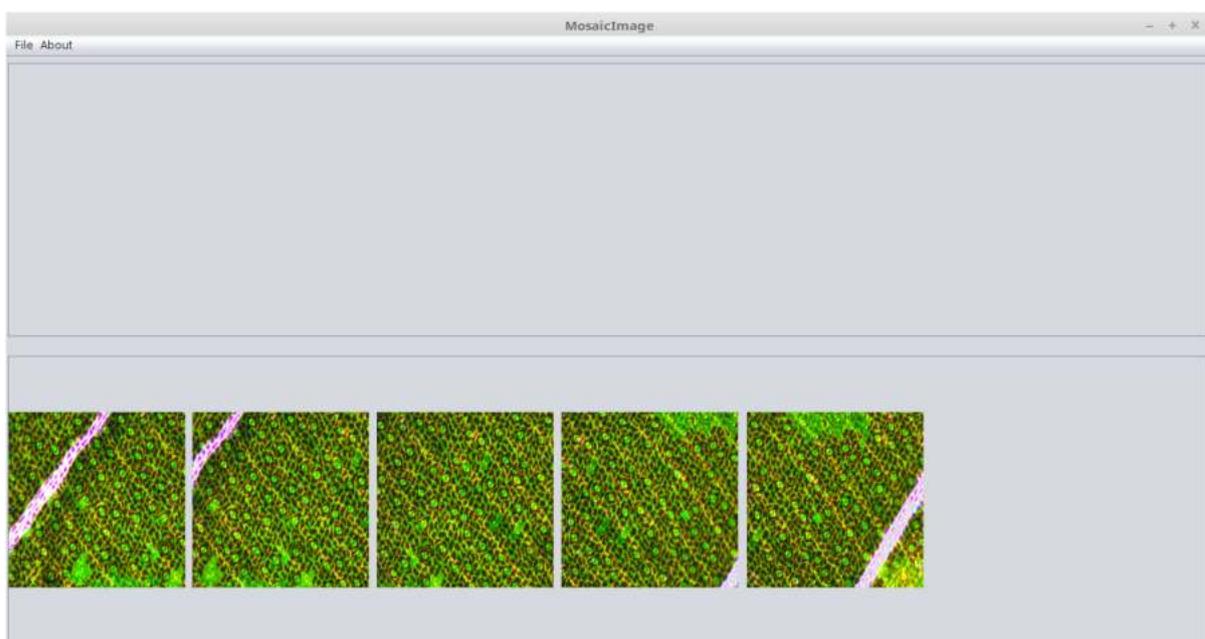
Figura 50 – *MosaicImage*: criando um mosaico entre duas imagens.



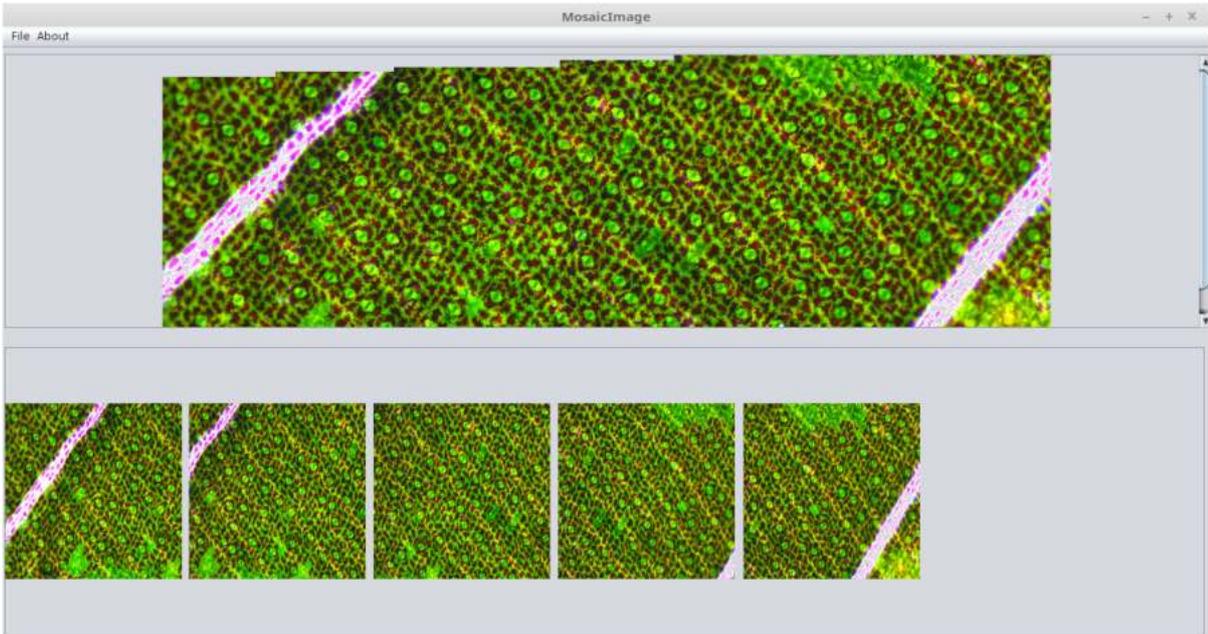
Fonte: Elaborado pela autora.

A [Figura 51](#) e a [Figura 52](#) abaixo ilustram um segundo exemplo de criação de mosaicos a partir de cinco imagens da espécie de planta *Tradescantia zebrina* que possuem ampliação de 10×:

Figura 51 – *MosaicImage*: carregamento das imagens no divisão inferior da tela.

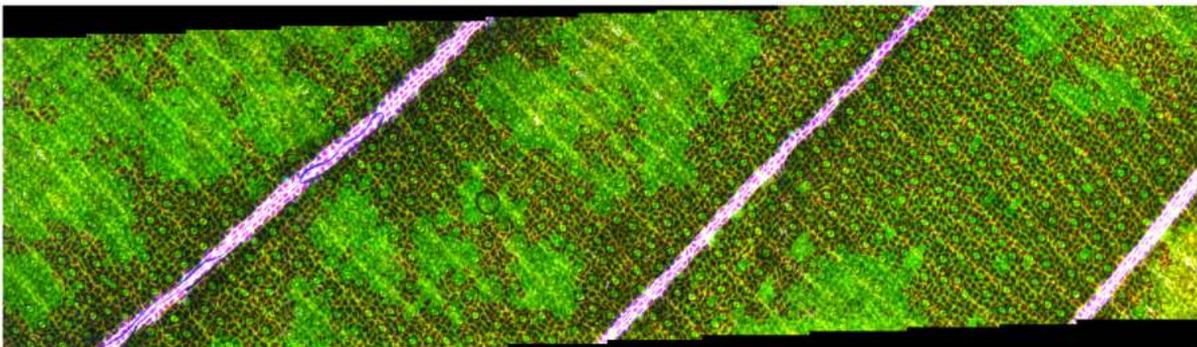


Fonte: Elaborado pela autora.

Figura 52 – *MosaicImage*: geração do mosaico final.

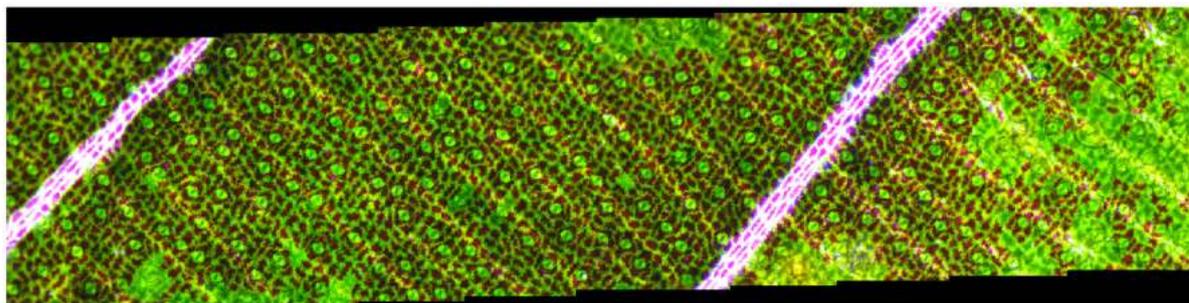
Fonte: Elaborado pela autora.

A [Figura 53](#) e a [Figura 54](#), exibe mosaicos gerados a partir de imagens da espécie de planta *Tradescantia zebrina*. Para imagens com aumento de $5\times$ é feita a junção de 9 imagens; e para as imagens com aumento de $10\times$ ocorre junção de 8 imagens.

Figura 53 – Mosaico criado a partir de imagens com ampliação de $5\times$.

Fonte: Elaborado pela autora.

Figura 54 – Mosaico criado a partir de imagens com ampliação de 10 \times .



Fonte: Elaborado pela autora.

5.3 Considerações finais

A metodologia proposta neste trabalho possibilitou o desenvolvimento da ferramenta denominada *MosaicImage*, descrita neste capítulo, que é um *plugin* que permite a construção automática de mosaicos de imagens microscópicas, desde que exista sobreposição entre as mesmas.

Optou-se por utilizar o método de correlação de fase juntamente com o filtro anisotrópico de *Forward-Backward*, em virtude de apresentar o menor tempo de processamento se comparado com os métodos SIFT e SURF, e apresentar bons resultados na presença de ruídos em imagens, conforme pode ser visto no próximo capítulo.

CONCLUSÕES

Este trabalho de mestrado teve como objetivo inicial o desenvolvimento de um *plugin* para o ImageJ, entretanto, com os variados métodos de mosaicos disponíveis na literatura, tornou-se necessário uma avaliação e experimentação das principais abordagens para identificar a mais adequada para aplicações em imagens microscópicas.

No que tange aos métodos de mosaicos, as metodologias consideradas mais robustas são o SIFT e o SURF. Alguns autores divergem sobre qual o mais satisfatório de ambos. Mas, como visto neste trabalho, a escolha entre as diversas técnicas disponíveis está sujeita à qual o tipo de imagem se queira trabalhar ou mesmo a qual aplicação se destina. A proposta de estudo desta dissertação está voltada para mosaicos de imagens microscópicas, que são imagens capturadas em um ambiente controlado e sem problemas de diferentes pontos de vista. Dessa forma, a metodologia escolhida foi a baseada no domínio de frequência, denominada correlação de fase, que é uma técnica robusta a variados tipos de ruídos, apropriada para imagens de alta resolução, possui baixo custo computacional e é invariante à translação.

Escolhida a metodologia para o desenvolvimento do *plugin*, a proposta evoluiu, e foi desenvolvida uma análise de desempenho entre as 3 principais abordagens utilizadas para junções de imagens. Como visto anteriormente, os métodos SIFT e correlação, quando aplicados em imagens sem alterações de brilho, cor ou pontos de vista, apresentaram desempenho de 100%. Porém, em situações reais, as imagens passam por alterações de luminosidade, brilho, cor, ou mesmo alterações geométricas de escala, rotação ou translação, dessa forma, aplicou-se ruídos e filtros às imagens com intuito de identificar quais os filtros mais adequados para cada ruído e ruídos de pior caso para cada método de mosaicagem.

A partir desta metodologia, foi possível identificar o método de mosaico mais robusto voltado para imagens capturadas por microscópio. Todos os experimentos foram analisados e comprovaram que os métodos fundamentados em filtragem anisotrópica de Perona Malik e *Forward-Backward* são efetivas para restauração destas bases de imagens específicas, melhorando

assim os métodos de mosaico estudados, e que o tipo de ruído que mais corrompe estas imagens são os do tipo impulsivo ou sal e pimenta. O estudo também atestou que o método de correlação de fase escolhido para o desenvolvimento da ferramenta proposta mostrou-se, de fato, a abordagem mais adequada, com taxa de acerto acima de 50% mesmo em casos de imagens muito degradadas.

Por fim, fundamentado nos experimentos realizados, a ferramenta de mosaico foi construída utilizando a correlação de fase juntamente com o filtro anisotrópico de *Forward-Backward*, pois o mesmo pode ser definido como um novo tipo de regularização do modelo clássico de Perona Malik.

6.1 Contribuições do trabalho

A principal contribuição deste trabalho está no desenvolvimento de uma abordagem que aprimora os principais métodos de mosaico estudados, SURF, SIFT e correlação de fase. Nesta etapa foi demonstrado que os filtros anisotrópicos são os mais adequados para realizar o processo de filtragem das imagens para posterior correspondência. Isto permitiu que fosse possível o desenvolvimento de uma aplicação voltada para mosaicos de imagens microscópicas multiplataforma e de domínio público, denominada *MosaicImage*.

6.2 Propostas para trabalhos futuros

Como trabalhos futuros, propõe-se o melhoramento da ferramenta *MosaicImage*, com adição de mais funcionalidades, por exemplo, subitens de menu com várias opções de filtragem antes da realização do mosaico. Outro ponto importante é aplicar ruídos coloridos nas imagens microscópicas e analisar o comportamento dos filtros. Adicionar mais filtros de imagens como as variações da média ou adaptativo da mediana, a título de exemplo, trabalhar com mais métodos de mosaico além dos 3 estudados neste trabalho, pode-se citar o FAST, Detector Harris ou métodos mais atuais que fazem o uso de *deep learning*. Aplicar outras métricas de avaliação do método como PSNR (do inglês, *Power Signal to Noise Ratio*) ou SSID (do inglês, *Structural Similarity Index*).

REFERÊNCIAS

- ABOSHOSHA, A.; SAAD, M.; ASHOUR, M.; MASHADE, M. E. Image denoising based on spatial filters, an analytical study. **Proceedings - The 2009 International Conference on Computer Engineering and Systems, ICCES 09**, p. 245 – 250, 01 2010. Citado na página 54.
- AL-AMRI, S.; KALYANKAR, N.; KHAMITKAR, S. A comparative study of removal noise from remote sensing image. v. 7, n. 1, January 2010. Citado na página 54.
- ALCANTARILLA, P. F.; BARTOLI, A.; DAVISON, A. Kaze features. **Springer**, p. 214–227, 10 2012. Citado na página 26.
- ALI, S.; HUSSAIN, M. Panoramic image construction using feature based registration methods. **2012 15th International Multitopic Conference, INMIC 2012**, p. 209–214, 12 2012. Citado na página 46.
- ASSIS, F.; SILVA, D.; HIRAGA, A. K.; ARTERO, A.; STELA, M.; PAIVA, V. D.; BARBOSA, R. Um novo algoritmo para a construção de imagens panorâmicas usando os algoritmos sift e ransac. 04 2012. Citado nas páginas 25 e 26.
- AUTOSTITCH. **AutoStitch - a new dimension in automatic image stitching**. 2007. [Online; accessed 20-April-2019]. Disponível em: <<http://matthewalunbrown.com/autostitch/autostitch.html#licensing>>. Citado na página 27.
- AXIOVISION. **AxioVision User's Guide**. Jena, Germany, 2012. Citado nas páginas 27, 50 e 51.
- BAE, S.; CHO, P. S.; DURAND, F. Image-based querying of urban knowledge databases. 2009. Citado na página 46.
- BAI, X.; NING, X.; WANG, L. Analysis and comparison of feature detection and matching algorithms for rovers vision navigation. p. 66–71, 07 2012. Citado na página 46.
- BANDEIRA, C. I. R. **Análise Comparativa de Filtros Adaptativos de Ruído Speckle**. Fortaleza - CE: [s.n.], 2009. Citado na página 56.
- BANDEIRA, M. V. **Localização de imagens ao microscópio utilizando processamento digital de imagem**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 2005. Citado na página 44.
- BAO, P.; XU, D. Panoramic image mosaics via complex wavelet pyramid. In: **SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)**. [S.l.: s.n.], 1998. v. 5, p. 4614–4619 vol.5. Citado na página 30.
- BASRI, R.; JACOBS, D. W. Recognition using region correspondences. **International Journal of Computer Vision**, v. 25, p. 145–166, 1997. Citado na página 26.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. **European Conference on Computer Vision**, p. 404 – 417, 2006. Citado nas páginas 41, 42 e 46.

_____. Surf: Speeded up robust features (surf). **Comput. Vis. Image Underst., Elsevier Science Inc.**, v. 110, n. 3, p. 346–359, 2008. Citado nas páginas 26, 32, 39, 41, 46 e 64.

BIELSKI, C.; GRAZZINI, J.; SOILLE, P. Automated morphological image composition for mosaicing large image data sets. **IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)**, p. 4068–4071, 2007. Citado na página 26.

BIND, V. S. **Robust Techniques for Feature-based Image Mosaicing**. Tese (Doutorado) — Department of Electronics and Communication Engineering - National Institute of Technology Rourkela, Rourkela, 2013. Citado na página 32.

BOSSERT, O. A java software for creation of image mosaics. **Computer Methods and Programs in Biomedicine**, n. 75, p. 159–161, 2004. Citado na página 26.

BOTTERILL, T.; MILLS, S.; GREEN, R. Real-time aerial image mosaicing. **International Conference of Image and Vision Computing New Zealand (IVCNZ)**, p. 1–8, 2010. Citado na página 35.

BRACEWELL, R. N. **The Fourier Transform and Its Applications**. New York: Hardcover, 1965. Citado na página 43.

BURGER, W.; BURGE, M. J. **Digital Image Processing: An Algorithmic Introduction Using Java**. 2nd. ed. [S.l.]: Springer Publishing Company, 2016. Citado nas páginas 94 e 95.

BURT, P.; ADELSON, E. A multiresolution spline with application to image mosaics. **ACM T. Graphic.**, v. 2, 10 1983. Citado na página 26.

CANNY, J. A computacional approach to edge detection. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, n. 6, p. 679–698, 1986. Citado na página 58.

CAPEL, D. P. **Image Mosaicing and Super-resolution**. Tese (Doutorado) — University of Oxford, 2001. Citado nas páginas 29 e 30.

CHALFOUN, J.; MAJURSKI, M.; BLATTNER, T.; BHADRIRAJU, K.; KEYROUZ, W.; BAJCSY, P.; BRADY, M. Mist: Accurate and scalable microscopy image stitching tool with stage modeling and error minimization. **Scientific Reports**, v. 7, 07 2017. Citado na página 26.

COSTA, A. L. N. T. da. **Redução de Ruído em Imagens**. Dissertação (Mestrado) — Programa de Engenharia Elétrica (UFRJ/COPPE), Rio de Janeiro - RJ, 2009. Citado nas páginas 52, 53 e 54.

CROW, F. C. Summed-area tables for texture mapping. **SIGGRAPH Comput. Graph.**, ACM, New York, NY, USA, v. 18, n. 3, p. 207–212, jan. 1984. ISSN 0097-8930. Disponível em: <<http://doi.acm.org/10.1145/964965.808600>>. Citado na página 39.

DESHMUKH, M.; BHOSLE, U. A survey of image registration. **International Journal of Image Processing (IJIP)**, 2011. Citado na página 43.

FERREIRA, T.; RASBAND, W. **ImageJ User Guide**. [S.l.], 2012. Citado nas páginas 94 e 96.

FILHO, O. M.; NETO, H. V. **Processamento Digital de Imagens**. 2nd. ed. [S.l.]: Brasport, 1999. ISBN 8574520098. Citado na página 56.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. **Commun. ACM**, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, 1981. Disponível em: <<http://doi.acm.org/10.1145/358669.358692>>. Citado na página 30.

FRIEDMAN, J. H.; BENTLEY, J. L.; FINKEL, R. A. An algorithm for finding best matches in logarithmic expected time. **ACM Transactions on Mathematical Software**, v. 3, p. 209–226, 1976. Citado na página 26.

GHANNAM, S.; ABBOTT, A. L. Cross correlation versus mutual information for image mosaicing. (**IJACSA**) - **International Journal of Advanced Computer Science and Applications**, v. 4, n. 11, 2013. Citado nas páginas 26 e 31.

GHOSH, D.; KAABOUCHE, N. A survey on image mosaicing techniques. **Journal of Visual Communication and Image Representation**, p. 1–11, 2016. Citado nas páginas 30, 31, 33, 43, 45 e 46.

GONZALEZ, R.; WOODS, R. **Processamento digital de imagens**. 3rd. ed. São Paulo: Prentice Hall, 2010. Citado nas páginas 52, 53, 55 e 56.

GONZALEZ, R.; WOODS, R.; EDDINS, S. **Digital Image Processing Using MATLAB**. 2nd. ed. [S.l.]: Gatesmark Publishing, 2009. Citado nas páginas 52, 56 e 57.

GUI, Y.; SU, A.; DU, J. Point-pattern matching method using surf and shape context. **Optik - International Journal for Light and Electron Optics**, v. 124, p. 1869–1873, 07 2013. Citado na página 46.

GUIDOTTI, P.; KIM, Y.; LAMBERS, J. Image restoration with a new class of forward-backward-forward diffusion equations of perona-malik type with applications to satellite image enhancement. **SIAM Journal on Imaging Sciences**, v. 6, p. 1416–1444, 07 2013. Citado nas páginas 58 e 59.

GUIZILINI, V.; SALES, D.; LAHOUD, M.; JORGE, L. Embedded mosaic generation using aerial images. In: **2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)**. [S.l.: s.n.], 2017. p. 1–6. Citado na página 26.

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. p. 147–151, 1988. Citado nas páginas 32 e 34.

HELMER, E. H.; RUEFENACHT, B. Cloud-free satellite image mosaics with regression trees and histogram matching. **IPhotogrammetric Engineering and Remote Sensing**, p. 1079–1089, 2005. Citado na página 26.

HOGUE, W. S.; MAMATA, H.; MAIER, S. E. Efficient construction of histology slide mosaics via phase correlation registration of high resolution tiles. In: **Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)**. [S.l.: s.n.], 2003. v. 1, p. I–1117. ISSN 1522-4880. Citado na página 26.

HUI, L.; MANJUNATH, B. S.; MITRA, S. K. A contour-based approach to multisensor image registration. **IEEE Transactions on Image Processing**, v. 4, n. 3, p. 320–334, March 1995. ISSN 1057-7149. Citado na página 32.

- HURTÓS, N.; CUF, X.; PETILLOT, Y.; SALVI, J. Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing. **IEEE RSJ International Conference on Intelligent Robots and Systems**, p. 7–12, 2012. Citado nas páginas 44 e 45.
- ISLAM, B.; KABIR, J. A new feature-based image registration algorithm. **Computer Technology and Application**, p. 79–84, 2013. Citado nas páginas 26, 32 e 43.
- JIAO, J.; ZHAO, B.; WU, S. A speed-up and robust image registration algorithm based on fast. **IEEE International Conference on Computer Science and Automation Engineering**, p. 1–8, 2011. Citado nas páginas 26 e 35.
- JOSHI, H.; SINHA, M. K. A survey on image mosaicing techniques. **International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)**, v. 2, 2013. Citado nas páginas 34 e 35.
- KAYNIG, V.; FISCHER, B.; MÜLLER, E.; BUHMANN, J. Fully automatic stitching and distortion correction of transmission electron microscope images. **Journal of structural biology**, v. 171, p. 163–73, 08 2010. Citado na página 46.
- KHACHANE, N.; GOHATRE, P. Image mosaic using phase correlation and harris operator. **International Journal For Emerging Trends in Engineering and Management Research (IJETEMR)**, v. 2, 2016. Citado na página 35.
- KHAN, N.; MCCANE, B.; WYVILL, G. Sift and surf performance evaluation against various image deformations on benchmark dataset. **2011 International Conference on Digital Image Computing: Techniques and Applications**, p. 501–506, 12 2011. Citado na página 46.
- KIM, D.-H.; YOON, Y.-I.; CHOI, J.-S. An efficient method to build panoramic image mosaics. **Pattern Recognition Letters**, v. 24, p. 2421–2429, 2003. Citado na página 26.
- KOSE, K.; GOU, M.; YÉLAMOS, O.; CORDOVA, M.; ROSSI, A.; NEHAL, K.; FLORES, E.; CAMPS, O.; DY, J.; BROOKS, D.; RAJADHYAKSHA, M. Automated video-mosaicking approach for confocal microscopic imaging in vivo: An approach to address challenges in imaging living tissue and extend field of view. **Scientific Reports**, v. 7, 12 2017. Citado na página 26.
- KUGLIN, C. D.; HINES, D. C. The phase correlation image alignment method. **IEEE International Conference on Cybernetics and Society**, p. 163 – 165, 1975. Citado nas páginas 26 e 64.
- LI, L.-n.; GENG, N. Algorithm for sequence image automatic mosaic based on sift feature. 08 2010. Citado na página 46.
- LIU, G.-H.; LI, Z.-Y.; ZHANG, L.; XU, Y. Image retrieval based on micro-structure descriptor. **Pattern Recognition**, v. 44, p. 2123–2133, 09 2011. Citado na página 46.
- LOWE, D. G. Object recognition from local scale-invariant features. **In: International Conference on Computer Vision**, p. 1150–1157, 1999. Citado nas páginas 26, 32, 36 e 37.
- _____. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, p. 91–110, 2004. Citado nas páginas 36, 38 e 39.

LUHMANN, T. A historical review on panorama photogrammetry. **International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, v. 34, 07 2008. Citado na página 25.

LUO, J.; OUBONG, G. A comparison of sift, pca-sift and surf. **International Journal of Image Processing**, v. 3, 10 2009. Citado na página 46.

MA, B.; ZIMMERMANN, T.; ROHDE, M.; WINKELBACH, S.; HE, F.; LINDENMAIER, W.; DITTMAR, K. E. Use of autostitch for automatic stitching of microscope images. In **ELSEVIER: The International Research and Review Journal for Microscopy (Micron)**, p. 492–499, 2007. Citado na página 26.

MACHADO, W. R. S.; GONZAGA, A.; BOAVENTURA, M. Estudo e proposta de adaptação do algoritmo sift em relação ao problema de iluminação em imagens. 06 2019. Citado na página 46.

MARQUES, O. **Practical Image and Video Processing Using MATLAB**. New York: Wiley, 2011. Citado nas páginas 55 e 56.

MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. **IEEE transactions on pattern analysis and machine intelligence**, v. 27, p. 1615–30, 11 2005. Citado na página 46.

MIN, Z.; JIGUO, Z.; XUSHENG, X. Panorama stitching based on sift algorithm and levenberg-marquardt optimization. **Physics Procedia**, v. 33, 12 2012. Citado na página 46.

NEIVA, M. B. **Métodos de pré-processamento de texturas para otimizar o reconhecimento de padrões**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação (ICMC/ USP), São Carlos - SP, 2016. Citado nas páginas 58 e 59.

NEWHALL, B. **The history of photography from 1839 to the present day**. New York: Museum of Modern Art, 1949. Citado na página 25.

PANDEY, A.; PATI, U. C. An improved dct-based phase correlation method for image mosaicing. In: . [S.l.: s.n.], 2015. p. 265–270. Citado nas páginas 25 e 26.

PARIDA, S. **Denoising Of Satellite Images**. Tese (Doutorado) — National Institute of Technology Rourkela, Índia, 2014. Citado na página 54.

PATEL, T. S.; GURJWAR, R. A study on feature extraction techniques for image mosaicing system. **Internation Journal Of Advance Research And Innovative Ideas In Education**, v. 2, 2016. Citado na página 35.

PATIDAR, D.; JAIN, A. Automatic image mosaicing: An approach based on fft. **Int J Sci Eng Technol**, v. 1, p. 01–04, 01 2011. Citado na página 26.

PATIL, A. A.; PATIL, P. S. P. Harris corner detection for image mosaic. **International Journal for Research in Technological Studies**, v. 3, 2016. Citado nas páginas 26 e 34.

PEDRINI, H.; SCHUWARTZ, W. **Análise de Imagens Digitais Princípios, Algoritmos e Aplicações**. [S.l.]: Thomson Learning, 2007. Citado na página 55.

PERONA, P.; MALIK, J. Scale-space and edge detection using anisotropic diffusion. **IEEE, IEEE Transactions on**, v. 12, n. 7, p. 629–639, 1990. Citado nas páginas 57 e 58.

- PERONA, P.; SHIOTA, T.; MALIK, J. Anisotropic diffusion. **Geometry-driven diffusion in computer vision**, Springer, p. 73–92, 1994. Citado nas páginas 57 e 58.
- PLUIM, J. P. W.; MAINTZ, J. B. A.; VIERGEVER, M. A. Mutual-information-based registration of medical images: a survey. **IEEE Transactions on Medical Imaging**, v. 22, n. 8, p. 986–1004, Aug 2003. ISSN 0278-0062. Citado na página 33.
- PRATT, W. K. **Digital Image Processing**. Los Altos, California: A Wiley Interscience Publication, 2001. Citado nas páginas 44 e 45.
- PRESCOTT, J.; CLARY, M.; WIET, G.; PAN, T.; HUANG, K. Automatic registration of large set of microscopic images using high-level features. **IEEE International Symposium on Biomedical Imaging: Nano to Macro**, p. 1284–1287, 2006. Citado na página 43.
- PTGUI. **PTGui - Create high quality panoramas**. 2011. [Online; accessed 20-April-2019]. Disponível em: <<https://www.ptgui.com/>>. Citado na página 27.
- RAMOS, J. da S. **Algoritmos de casamento de imagens com filtragem adaptativa de outliers**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP, 2016. Citado nas páginas 30, 38, 39, 40, 41 e 42.
- REDDY, B. S.; CHATTERJI, B. An fft-based technique for translation, rotation and scale invariant image registration. **IEEE Transactions on Image Processing**, p. 1266–1271, 1996. Citado na página 43.
- ROSTEN, E.; REITMAYR, G.; DRUMMOND, T. Real-time video annotations for augmented reality. In: **Proceedings of the First International Conference on Advances in Visual Computing**. Berlin, Heidelberg: Springer-Verlag, 2005. (ISVC'05), p. 294–302. ISBN 3-540-30750-8, 978-3-540-30750-1. Disponível em: <http://dx.doi.org/10.1007/11595755_36>. Citado na página 36.
- SEO, J.-H.; YANG, S.; KANG, M.-S.; HER, N.-G.; NAM, D.-H.; CHOI, J.-H.; KIM, M. Automated stitching of microscope images of fluorescence in cells with minimal overlap. **Micron**, v. 126, p. 102718, 08 2019. Citado na página 26.
- SHUM, H.-Y.; SZELISKI, R. Construction of panoramic image mosaic. **Springer-Verlag**, p. 227–268, 2001. Citado na página 29.
- SONG, F.; LU, B. An automatic video image mosaic algorithm based on sift feature matching. **Advances in Intelligent Systems and Computing**, v. 181, p. 879–886, 01 2013. Citado na página 46.
- SOUZA, A. Desempenho de algoritmos detectores de keypoints para um sistema de navegação visual de robôs baseados em smartphones. In: . [S.l.: s.n.], 2015. Citado na página 37.
- STUCHI, J. A. **Registro de Imagens por Correlação de Fase para Geração de Imagens Coloridas em Retinógrafos Digitais Utilizando Câmera CCD Monocromática**. Dissertação (Mestrado) — Escola de Engenharia de São Carlos (EESC), Universidade de São Paulo, São Carlos, 2013. Citado nas páginas 31, 44 e 45.
- SZELISKI, R. Image alignment and stitching: A tutorial. **Foundations and Trends in Computer Graphics and Vision**, v. 2, n. 1, p. 1–104, 2006. Citado nas páginas 26, 31 e 32.

- TARALLO, A. S. **Construção Automática de Mosaicos de Imagens Digitais Aéreas Agrícolas Utilizando a Transformada SIFT e Processamento Paralelo**. Tese (Doutorado) — Escola de Engenharia de São Carlos (EESC), Universidade de São Paulo, São Carlos, 2013. Citado nas páginas 26, 29 e 46.
- TELES, V. C.; ANDREANI, L.; VALADARES, L. F. **Uso de Microscopia de Luz e Eletrônica como Técnicas de Análise Morfológica**. Circular Técnica Embrapa, 2017. [Online; accessed on 05-Abril-2019]. Disponível em: <<https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/1085307/1/CIT15CNPAE.pdf>>. Citado na página 26.
- THÉVENAZ, P.; UNSER, M. User-friendly semiautomated assembly of accurate image mosaics in microscopy. **Microscopy Research and Technique**, v. 70, n. 2, p. 135–146, February 2007. Citado na página 93.
- TRAJKOVIĆ, M.; HEDLEY, M. Fast corner detection. **Image and Vision Computing**, v. 16, n. 2, p. 75 – 87, 1998. ISSN 0262-8856. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0262885697000565>>. Citado nas páginas 32 e 35.
- VEDALDI, A.; FULKERSON, B. **VLFeat: An Open and Portable Library of Computer Vision Algorithms**. 2008. <<http://www.vlfeat.org/>>. Citado na página 64.
- VIOLA, P.; III, W. M. W. Alignment by maximization of mutual information. **International Journal of Computer Vision**, v. 24, n. 2, p. 137–154, Sep 1997. ISSN 1573-1405. Disponível em: <<https://doi.org/10.1023/A:1007958904918>>. Citado na página 34.
- VIOLA, P.; JONES, M. Robust real-time object detection. **International Journal of Computer Vision - IJCV**, v. 57, 01 2001. Citado na página 39.
- WANG, C.; CHENG, Y.; ZHAO, C. Robust subpixel registration for image mosaicing. **Chinese Conference on Pattern Recognition**, p. 1–5, 2009. Citado na página 45.
- WANG, Z.; YANG, Z. Review on image-stitching techniques. **Multimedia Systems**, 03 2020. Citado na página 25.
- WOODS, R. P.; CHERRY, S. R.; MAZZIOTTA, J. C. Rapid automated algorithm for aligning and reslicing pet images. **Jornal of Computer Assisted Tomography**, v. 16, n. 4, p. 620–633, 1992. Citado nas páginas 31 e 33.
- WOODS, R. P.; MAZZIOTTA, J. C.; CHERRY, S. R. Mri-pet registration with automated algorithm. **Jornal of Computer Assisted Tomography**, v. 17, n. 4, p. 536–546, 1993. Citado nas páginas 31 e 33.
- XIAO, J.; ZHANG, Y.; SHAH, M. Adaptive region-based video registration. **IEEE Workshops on Application of Computer Vision**, p. 215–220, 2005. Citado na página 43.
- YANG, F.; WEI, L.; ZHANG, Z.; TANG, H. Image mosaic based on phase correlation and harris operator. **Journal of Computational Information Systems**, p. 2647–2655, 2012. Citado nas páginas 45 e 46.
- YONGWEI, M.; XINKE, G.; JIAYIN, X. A new method of microscopic images automatic mosaicing. **IEEE International Conference on Bioinformatics and Biomedical Engineering (ICBBE)**, 2009. Citado na página 26.
- ZITOVÁ, B.; FLUSSER, J. Image registration methods: a survey. **Image and Vision Computing** 21, p. 977–1000, 2003. Citado nas páginas 30 e 44.

