

ALINHAMENTO DE FOTOS POR PROCESSAMENTO DIGITAL: UMA APLICAÇÃO DE ALGORITMOS DESCRITORES E DETECTORES

SORATTO, André
Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
andre.s10@aluno.ifsc.edu.br

FABRE, Diego Tiburcio
Professor de Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
diego.fabre@ifsc.edu.br

Resumo—A aerofotogrametria é uma técnica utilizada na obtenção de informações precisas sobre o relevo e o uso do solo em áreas extensas. No entanto, para que esses produtos tenham a qualidade necessária, é fundamental que as fotos estejam devidamente alinhadas e calibradas, um processo manual que exige tempo e esforço de profissionais qualificados na área de cartografia, geologia ou agrimensura. Nesse contexto, este artigo apresenta uma pesquisa sobre algoritmos de alinhamento e o desenvolvimento de um *software* que utiliza técnicas de sistemas de visão para realizar o alinhamento de imagens automatizada implementando algoritmos de detecção de pontos de interesse nas imagens em especial, o método ORB. Após o desenvolvimento do *software*, foi realizada uma análise quantitativa dos resultados a partir de uma amostra de imagens, aplicando métodos estatísticos para identificar a precisão do sistema. Os resultados analisados indicam uma baixa variação nos ângulos das linhas de correspondência das imagens quando há pontos de interesse visíveis.

Palavras-Chave—aerofotogrametria, alinhamento de imagens, ORB, OpenCV, sistemas de visão, cartografia, precisão.

I. INTRODUÇÃO

A obtenção informações por meio de técnicas tradicionais de alinhamento de fotos coletadas por aerofotogrametria é um processo demorado, dispendioso e que demanda conhecimento técnico [1]. Isso ocorre devido à necessidade de inserção manual de pontos de controle em cada imagem, uma tarefa trabalhosa e custosa (Figura 1). Além disso, esses métodos requerem conhecimento especializado em processamento de imagem, o que restringe a disponibilidade de profissionais qualificados para conduzir essas atividades.



Fig. 1: Marcadores inseridos manualmente com folhas A3.
(Mello, Simões, [2])

Utilizar de recursos digitais automatizados para a análise e manipulação de imagens emerge como uma alternativa viável em várias áreas, incluindo a mineração a céu aberto [2], controle ambiental [3] e construção civil. A fotogrametria digital oferece benefícios comparativos à medição convencional em terra [4], possibilitando a avaliação detalhada de vastas áreas sem a necessidade de investimentos significativos em tempo e recursos. Diversas ferramentas e linguagens de programação possuem capacidade de aplicar algoritmos e processar imagens digitais.

De acordo com Silva e Costa, do Departamento de Engenharia Cartográfica de Recife [5], a coleta de imagens por meio de Veículos Aéreos Não Tripulados (VANTs), também conhecidos como drones, apresenta uma abordagem vantajosa para a obtenção de imagens em projetos de alinhamento digital. Essa metodologia permite capturar imagens de áreas extensas de forma rápida e eficiente, reduzindo significativamente o tempo necessário para coletar dados em comparação com métodos terrestres tradicionais. Além disso, a capacidade de operar em altitudes variáveis e ângulos específicos possibilita a obtenção de imagens de diferentes perspectivas, o que pode ser crucial para a precisão do alinhamento. Considera-se também a flexibilidade, uma vez que os drones podem acessar locais de difícil alcance ou áreas remotas, tornando-os uma escolha ideal para projetos em diversos cenários.

Para desenvolver um projeto eficiente de alinhamento de fotos digitais utilizando sistemas de visão, pode-se empregar uma série de tecnologias e técnicas. Em primeiro lugar, o uso de bibliotecas e frameworks de processamento de imagens, como o OpenCV [6], é uma das tecnologias estudadas para realizar a análise e manipulação das imagens. Além disso, a utilização de algoritmos de visão computacional, como detecção de características-chave (keypoints) [7] e correspondência de pontos, possibilita a identificação de pontos de interesse nas imagens. A integração de técnicas de processamento de imagem avançadas estudadas nesse artigo tem como objetivo melhorar a precisão e a produtividade no alinhamento de imagens.

Com o principal objetivo de identificar objetos correlatos em imagens distintas, na Suíça, pesquisadores da Universidade Católica de Leuven, desenvolveram um método matemático baseado no conceito de descritores em regiões de uma imagem chamado *Speeded-Up Robust Features* (SURF) [8]. O algoritmo utiliza uma área

quadrada na vizinhança do ponto chave e divide em sub-regiões. Para cada uma dessas são coletados *wavelets* (onduletas) que são utilizadas para retirar o descritor de características da região [9]. Em 2011, desenvolvedores da OpenCV Labs criaram um método ainda mais performático denominado *Oriented FAST and Rotated BRIEF* (ORB) [9], capaz de realizar as mesmas operações do SURF e com a vantagem de possuir licença aberta e funcionar com maior precisão em casos onde há rotação. Um exemplo de funcionamento de detecção e descrição de imagens pode ser visto na figura 2.

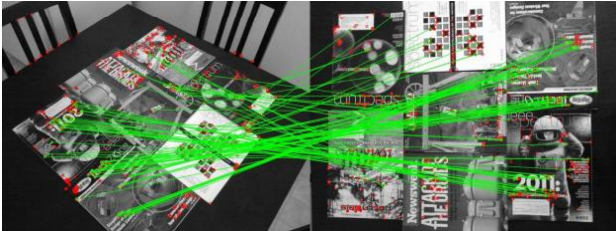


Fig. 2: Combinação entre pontos chave de duas imagens utilizando o algoritmo ORB (Rublee, E. et al [10])

O Scale Invariant Feature Transform (SIFT) é um descritor de imagem desenvolvido por David Lowe em 1999 e aprimorado em 2004 [14]. Esse descritor, juntamente com descritores relacionados, é amplamente utilizado em visão computacional para correspondência de pontos entre diferentes vistas de uma cena 3D e reconhecimento de objetos baseado em visão. O SIFT é notável por sua invariância a transformações de translação, rotação e escala na área da imagem, sendo robusto a transformações de perspectiva moderada e variações de iluminação.

Em comparação com outros métodos de identificação de pontos de interesse, o ORB se destaca quando as imagens possuem diferentes escalas. A taxa de correspondências maior (Tabela 1, *Match rate*) indica a probabilidade de que encontre uma matriz de homografia satisfatória para realizar o alinhamento. Além disso, o tempo para execução é cerca de 10 vezes menor quando comparado ao SIFT e 4 vezes menor quando comparado ao SURF.

	Tempo (seg)	Pontos1	Pontos2	Corresp.	Taxa Corresp. (%)
SIFT	0,25	248	1210	232	31,8
SURF	0,08	162	581	136	36,6
ORB	0,02	261	471	181	49,5

Tab. 1: Taxa de correspondências entre imagens de diferentes escalas entre três algoritmos descritores. (Karami, et. al. [9])

Baseando-se nessa contextualização, esse trabalho consiste no desenvolvimento de uma plataforma de intuitiva para agilizar no processo de alinhamento de imagens coletadas por levantamentos aerofotogramétricos. Desse modo, o software utiliza imagens coletadas em levantamentos aerofotogramétricos e realiza o alinhamento sem necessidade de interferência manual do usuário.

II. METODOLOGIA

A metodologia adotada inicia com a avaliação e pesquisa qualitativa sobre algoritmos descritores, a fim de selecionar as técnicas ideais para o desenvolvimento do projeto. Em

seguida, são buscadas as ferramentas a serem utilizadas no código, a captura de imagens, análise dos métodos de pré-processamento de imagem disponíveis, projeto das etapas para o desenvolvimento, como os fluxos de processamento das imagens e funcionamento do programa. Após definidas as etapas, é feita a programação do software. Os testes para validação da plataforma são realizados quantitativamente, através de análise estatística de uma amostra de imagens.

A. Avaliação e seleção da técnica de alinhamento

Através de uma pesquisa qualitativa, foram comparados os três principais métodos de identificação de pontos utilizados na literatura, denominados SIFT, SURF e ORB. A escolha do ORB foi feita ao observar uma vantagem na velocidade de execução e maior assertividade na correlação dos pontos, quando comparado com os demais algoritmos. O ORB apresentou ganho significativo na precisão principalmente em situações onde há diferença de escala entre as imagens. Outra vantagem é a sua licença de uso, enquanto SURF e SIFT são tecnologias proprietárias, o ORB possui código aberto e uso livre para aplicações acadêmicas e comerciais.

B. Interface Gráfica

Para complementar a aplicação dos algoritmos, faz-se necessário o desenvolvimento de uma interface gráfica de usuário. Ao contrário da utilização direta em código, uma interface gráfica oferece uma representação visual dos recursos e funcionalidades do sistema, simplificando a interação para usuários menos familiarizados com programação. Além disso, uma *Graphical User Interface (GUI)* facilita a configuração de parâmetros, visualização de resultados e tomada de decisões por meio de controles visuais e feedback imediato. Essa abordagem contribui significativamente para a adesão do sistema, permitindo que uma gama mais ampla de usuários, incluindo aqueles sem experiência técnica avançada, possa utilizar efetivamente as capacidades do software de alinhamento de imagens.

C. Escolha pelas ferramentas de desenvolvimento

Na escolha pela ferramenta adequada para o desenvolvimento da plataforma de alinhamento, optou-se pelo Python por possuir uma vantagem na facilidade de entendimento do código [18], que será disponibilizado no modelo *Open-Source* para futura pesquisa e complemento na área.

A interface gráfica (GUI) foi desenvolvida utilizando a plataforma PyQt5, uma ligação Python para a biblioteca Qt, que é uma estrutura de desenvolvimento de software amplamente utilizada para a criação de aplicativos com interfaces gráficas de usuário. As vantagens de usar PyQt5 incluem a facilidade de design e implementação de interfaces gráficas interativas e visualmente atraentes. Além disso, PyQt5 oferece uma ampla variedade de widgets, recursos de personalização e suporte a eventos que simplificam o processo de criação de interfaces dinâmicas e funcionais. A integração entre Python e Qt também permite uma transição suave entre a lógica de programação e a

apresentação visual, proporcionando uma experiência de desenvolvimento eficiente e flexível para o projeto.

D. Captura de Fotos com o uso de VANTs

Para uso durante o desenvolvimento da ferramenta, um banco de imagens foi coletado e providenciado pela equipe de engenharia da Universidade do Extremo Sul Catarinense – UNESC. As imagens foram coletadas utilizando um VANT DJI Phantom 4 Pro (Figura 3). Tem capacidade de coletar imagens na resolução 4K e possui sistema de GPS para geolocalização das fotos. As imagens são de uma região rural do município de Cocal do Sul, ideais para demonstração da aplicação em monitoramento de áreas de reflorestamento.



Fig. 3: DJI Phantom 4 Pro. (DJI user manual [15])

E. Fluxo de processamento das imagens.

Para minimizar a necessidade de interação com o usuário, empregou-se um fluxo de processamento linear no alinhamento das imagens, que passa por três etapas até obter um resultado. O pré-processamento é realizado com funções do *OpenCV* para facilitar na detecção de bordas, em seguida, é extraído uma coleção de, no máximo 10.000 pontos de interesse das duas imagens utilizando o método ORB. Os pontos são filtrados por um fator de proximidade baseado na taxa de confiança definida. Posteriormente, é realizada uma comparação entre essa lista de pontos de interesse para identificar correlações entre eles nas imagens. Essas correlações resultam em uma matriz de homografia, responsável por realizar as transformações de rotação, translação e redimensionamento.

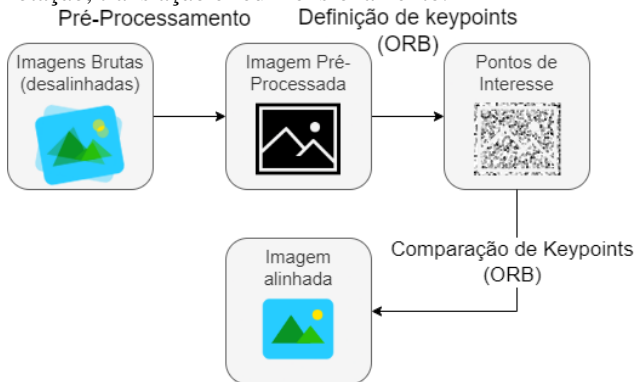


Fig. 4: Diagrama de Processamento das imagens.

A figura 4 mostra uma simplificação de como foi projetado o sistema. As imagens brutas passam por uma etapa de pré-processamento antes de serem alinhadas. Em seguida, é aplicado o ORB para definição de pontos de interesse e posterior correlação entre múltiplas imagens.

F. Fluxo de funcionamento do programa

O Funcionamento do programa consiste na interação com o usuário para determinar as imagens a serem alinhadas e um botão para inicializar o processo de alinhamento. Opcionalmente poderão ser definidos parâmetros de confiança e contraste da imagem para teste em diferentes condições. O fluxograma da figura 5 traz como entrada as imagens brutas (não processadas) e quais os resultados obtidos após o término do fluxo de processamento de imagens.

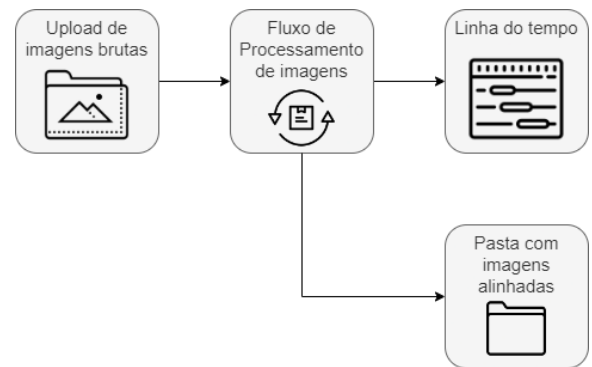


Fig. 5: Diagrama de funcionamento do software.

As saídas (resultados) do programa serão uma animação no formato GIF em ordem de upload das imagens e o resultado individual de cada imagem alinhada salvo em uma pasta “output”.

III. DESENVOLVIMENTO

O desenvolvimento do fluxo de processamento de imagens e dos algoritmos de alinhamento no contexto deste projeto foi realizado utilizando a linguagem Python 3.10.6, com o suporte do ambiente Visual Studio Code. A captura de fotos por meio de VANTs forneceu um banco de imagens para desenvolvimento e teste. O programa permite a importação em massa de imagens, com a escolha de uma imagem principal para alinhamento.

Visando melhor desempenho, o software normaliza e redimensiona imagens de alta resolução, além de empregar múltiplas *Threads*, quando disponíveis para acelerar o processo e evitar o travamento da plataforma.

A implementação dos algoritmos abrange a detecção e comparação de pontos-chave, resultando em uma matriz de homografia para o alinhamento. A interface gráfica, desenvolvida com PyQt5, proporciona uma experiência amigável, permitindo a seleção de arquivos, o acompanhamento do progresso e a visualização dos resultados. A conexão entre a interface e os algoritmos é estabelecida utilizando padrões de projeto de software comportamentais, como eventos e observadores.

A. Desenvolvimento do fluxo de processamento de imagens

Os algoritmos de Sistemas de Visão empregados nesse projeto foram implementados em conjunto com o ambiente de desenvolvimento Visual Studio Code como ferramenta na escrita, teste e correção de código.

As imagens podem ser importadas em massa selecionando uma pasta. Na tela inicial do programa, serão mostradas todas as imagens. O usuário deve selecionar qual será a imagem primária, as outras imagens então serão alinhadas com base nesta (Figura 6). Posteriormente, o programa exporta em um outro diretório uma linha do tempo na ordem das imagens carregadas e também o resultado do alinhamento de cada imagem individual.

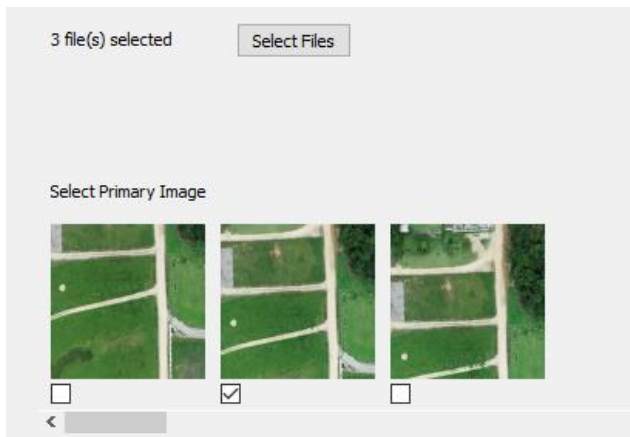


Fig. 6: Seleção da imagem primária no programa.

Para facilitar na detecção de pontos de interesse, utilizou-se uma técnica de realce de bordas (aumento de nitidez). Essa técnica consiste em subtrair da imagem original uma máscara ofuscada por um fator [11]. Para realizar essa operação, primeiro utiliza-se o método *GaussianBlur* para aplicar um desfoque gaussiano na imagem e em seguida o método *addWeighted*, que realiza a operação de subtração entre as duas imagens. O resultado desse processo pode ser observado na comparação entre as figuras 7 e 8.



Fig. 7: a) Imagem Bruta, b) Fig. 8: Imagem com filtro de Nitidez

B. Implementação dos Algoritmos de Alinhamento

Após a definição do algoritmo ideal para detecção de pontos de interesse na etapa de estudo da metodologia, utilizando a documentação da biblioteca OpenCV, encontrou-se uma forma de aplicar o algoritmo ORB e realizar a detecção e comparação dos *keypoints* (pontos-chave) nas imagens.

1) Detecção de pontos-chave de interesse

Após testes preliminares com os demais algoritmos pesquisados, decidiu-se pela utilização do ORB, por melhor correspondência de pontos, velocidade de execução e também por possuir licença de código aberto.

Conforme descrito no quadro 1, função *cv.ORB_create()* é utilizada para instanciar o detector ORB. A função *orb.detectAndCompute(...)* é utilizada para executar o modelo matemático e retorna duas listas, uma com os vetores de localização e direção dos pontos chave e outra com os descritores de cada ponto. Esses dados serão utilizados para o casamento dos pontos dentre duas imagens. O termo “*template*” na linha 3 é a imagem primária que será comparada com todas as outras selecionadas na tela apresentada na Figura 6.

```
1. orb = cv2.ORB_create(maxFeatures)
2. (kpsA, descA) = orb.detectAndCompute(img)
3. (kpsB, descB) = orb.detectAndCompute(template)
```

Quadro 1: Funções para extração de pontos de interesse e descritores.

2) Comparação entre múltiplos conjuntos de pontos de interesse

A comparação entre dois conjuntos de descritores é feita utilizando a função *matcher.match()*, que retorna uma lista de pontos-chave equivalentes nas imagens.

Em algoritmos descritores de pontos de interesse em imagens, falsos-positivos referem-se a casos em que o algoritmo erroneamente identifica correspondências entre pontos-chave em duas imagens que, na realidade, não são equivalentes. Esses falsos-positivos podem surgir devido a características semelhantes em regiões distintas das imagens, interferências visuais, variações na iluminação ou outros elementos que não representam uma correspondência genuína.

Para evitar falsos-positivos, ordena-se os pontos de interesse em comum com base nas suas distâncias. Quanto menor a distância, maior é a probabilidade de equivalência entre pontos em diferentes imagens. Após isso é possível retirar apenas os pontos com maior assertividade baseado em um fator percentual (Quadro 2, linha 3). Para testes, utiliza-se um fator de 20% dos pontos detectados. Esse fator poderá ser definido pelo usuário para suprir diferentes necessidades de precisão.

```
1. matches = matcher.match(descA, descB, None)
2. keep = int(len(matches) * keepPercent)
3. matches = matches[:keep]
4. H = find_homography(matches, kpsA, kpsB)
```

Quadro 2: Funções responsáveis pela criação da matriz de homografia.

Os pontos de interesse em comum das imagens (Figura 9) são armazenados em uma matriz de homografia (Quadro 2, Linha 4). Essa matriz consiste na transformação necessária para que todos os pontos coincidam na mesma posição em uma imagem.



Fig. 9: Correlação entre pontos de interesse.

C. Desenvolvimento da interface gráfica

Na implementação da interface gráfica, a escolha da biblioteca PyQt5 [13] proporcionou uma integração eficiente e flexível, facilitando a criação de uma interface coerente e interativa. Alguns elementos de *design* e *widgets* foram escolhidos para facilitar o entendimento do usuário.

1) PyQt5

A biblioteca PyQt5 disponibiliza uma ferramenta gratuita denominada Qt Designer, com ela, é possível utilizar diversos *widgets* disponíveis que facilitam a criação de uma interface coerente. Essa ferramenta exporta um arquivo no formato XML que pode ser convertida diretamente em código Python, sem necessidade de programação dos elementos gráficos da interface.

2) Elementos de Design

Para manter uma interface de fácil uso, foi definido um padrão de *design* simplificado (Figura 10). A tela possui um botão de seleção de arquivos, onde pode-se selecionar múltiplas imagens nos formatos PNG, JPG e BMP. Uma barra de rolagem mostra todas as imagens carregadas e permite que o usuário selecione a imagem primária (que será comparada com as demais durante o alinhamento). O botão “Align” inicia o processo que pode ser acompanhado através da barra de progresso abaixo. Quando o alinhamento for realizado, todos os arquivos estarão em uma pasta de saída que pode ser facilmente acessada pressionando o botão “Output Folder”.

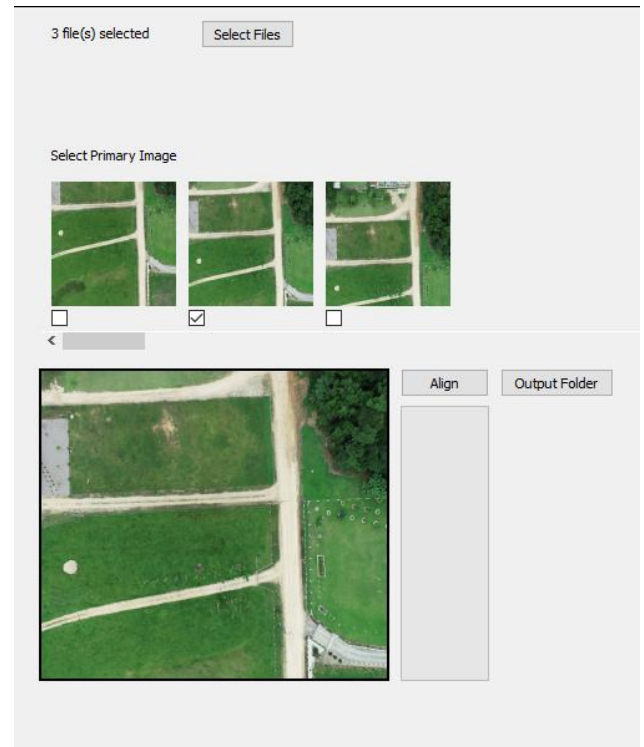


Fig. 10: Interface do Usuário.

Uma visualização rápida do resultado é demonstrada na tela no formato de sobreposição de todas as imagens alinhadas (Figura 11) ao final do processamento.

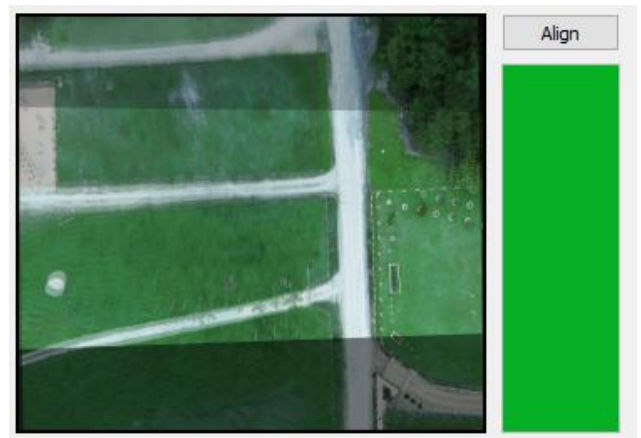


Fig. 11: Sobreposição de imagens alinhadas.

3) Conexão entre a interface de usuário e algoritmos de processamento de imagem.

A conexão entre a interface e os algoritmos ocorre principalmente nos métodos da classe *Window*, onde eventos de botões, barras de rolagem e seleção de arquivos desencadeiam a execução dos algoritmos de alinhamento.

O código-fonte contido no arquivo *main.py* segue os princípios de arquitetura de software orientado a objeto. São definidos métodos para cada ação da interface, e utilizando o método *connect()* cria-se um disparador de eventos.

4) Testes em modelos e imagens reais.

Para servir como base, foram utilizadas imagens-modelo. Esses modelos são pares de uma mesma imagem, rotacionada e escalada manualmente, com o intuito de coletar dados de situações ideais, onde todos os pontos descritores podem ser correspondidos em ambas as imagens.

Com os dados coletados para a comparação, realizou-se testes em batelada utilizando imagens reais com pequenas variações resultantes da movimentação do VANT no espaço aéreo. A amostra para avaliação quantitativa do software foi de 200 pares de imagens, totalizando 400 imagens processadas pelo algoritmo.

IV. RESULTADOS E DISCUSSÕES

Essa seção apresenta os resultados obtidos através de uma avaliação analítica de dados coletados pelo software durante a fase de testes.

Um dos pontos observados durante o desenvolvimento foi a distorção de imagens que não possuem pontos correlatos suficientes para geração de uma matriz de homografia correta. O resultado nesses casos era de uma transformação incorreta nas imagens, gerando uma saída indesejável (Figura 12). Esse problema pode ser mitigado ajustando o fator de confiança da obtenção de descritores.

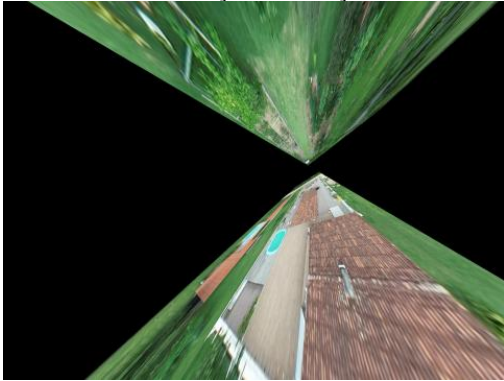


Fig. 12: Distorção resultante de uma matriz de homografia incorreta.

Como o conceito de teor de confiança pode se tornar abstrato ao usuário, foi implementado uma validação de acordo com as pesquisas de William H. Press et. al. [12], que demonstra a possibilidade de identificar falsos correlatos através do cálculo da determinante da matriz de homografia (D) e aplicando um fator limitante. Para esse projeto, identificou-se a partir de experimentação que qualquer valor $D < 0.8$, o resultado era uma imagem deformada.

Dessa forma, o programa itera diminuindo o fator de confiança (k) e repetindo o processamento até que $D \geq 0.8$ e então retorna uma imagem corretamente alinhada. Esse processo pode ser melhor entendido através do diagrama na figura 13.

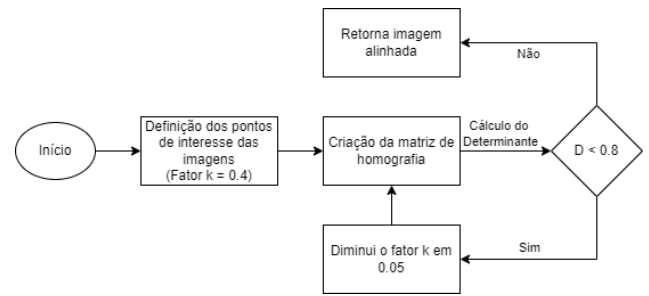


Fig. 13: Diagrama de validação da matriz de homografia

Os testes foram realizados com o banco de imagens provido e o conjunto de imagens na saída (figura 14) apresentou resultado de precisão similar em comparação com métodos manuais de alinhamento.

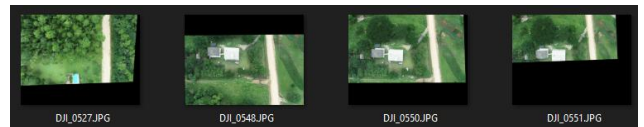


Fig. 14: Exemplo de resultados obtidos através do uso da aplicação.

Uma métrica utilizada por pesquisadores da universidade de Wuhan [16] para avaliar a assertividade do alinhamento das imagens, especialmente tratando-se de transformações geométricas, é realizada analisando os ângulos gerados entre os pontos de interesse de duas imagens. Em imagens cuja distorção não difere (obtidas com a mesma câmera), a diferença entre os ângulos obtidos não deve sofrer variação, ou seja, uma variação próxima de 0 seria o ideal para uma correlação de 100% dos pontos.

Nos testes realizados com imagens-modelo, os desvios-padrão dos ângulos permaneceram entre 0,015 e 0,020 graus, o que implica em uma falha de aproximadamente 2,23% (para uma amostra de 100 imagens).



Fig. 15: Ângulos obtidos na comparação entre imagens.

Em um resultado positivo, como na Figura 15, nota-se que os ângulos das linhas que conectam os pontos de interesse permanecem homogêneos. Comparando a diferença entre esses ângulos, é possível extrair um valor de desvio padrão entre todos os ângulos das correspondências.

A figura 16 mostra, no eixo X o ângulo θ (radianos) obtido pelas linhas de correspondência e no eixo Y a frequência f de aparição desses ângulos. A maior concentração de ângulos está em aproximadamente 87,95 graus. Essa concentração alta demonstra baixa variação em θ obtido pelo descritor.

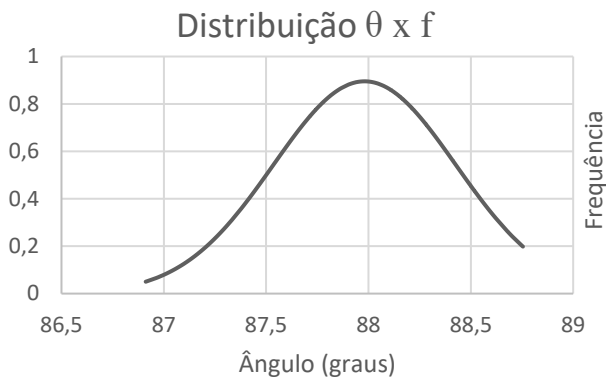


Fig. 16: Distribuição dos ângulos (graus) por probabilidade, obtidos na Figura 15.

Analisando os dados obtidos a partir das linhas na Figura 16, os valores variam de 86,51 a 88,8 graus e calcula-se um desvio padrão de aproximadamente 0,44 graus. Esse valor serve de referência para demais imagens, sendo que uma divergência significativa do desvio padrão indica excesso de deformação aplicada pela matriz de homografia. Aplicando esse mesmo cálculo em dados extraídos de duas imagens não relacionadas, obtemos um valor de desvio padrão de 125 graus, tal diferença se dá pela quantidade de falsos positivos obtidos em imagens sem semelhanças notáveis.

Além de avaliar a qualidade das transformações, é possível utilizar os valores de diferença de ângulos para filtrar as melhores correspondências de pontos de interesse. Esse filtro é realizado dividindo o conjunto de ângulos em um número de intervalos proporcional ao número de correspondências, e então identificando o intervalo com maior quantidade de ângulos, em seguida descarta-se todos os outros intervalos. Isso faz com que os falsos-positivos sejam eliminados do cálculo da matriz de homografia.

A figura 17 mostra um gráfico de distribuição de ângulos obtido a partir de uma imagem com baixa quantidade de fatores distintos. Isso gera um número mais elevado de falsos-positivos que podem ser identificados devido aos ângulos das linhas de correspondência terem grande variação.

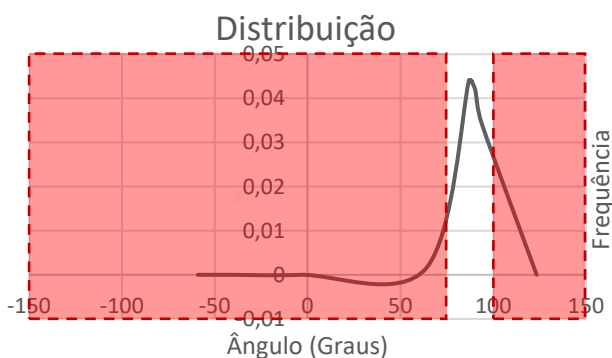


Fig. 17: Filtro por intervalo com maior correspondência.

Para a validação, foram utilizados 200 pares de imagens semelhantes, comparadas em lote e extraído seus valores de desvio padrão de ângulos. Desses valores, o maior desvio padrão foi de 9,74 graus, observado em imagens com menor quantidade de elementos distintos como construções

e estradas (Figuras 19 e 20). A média ficou em aproximadamente 6,3 graus, sendo que 41,5% dos resultados estavam abaixo dos 5 graus. Esse resultado se apresenta uma falha de, em média, 17,4% (Figura 18) para imagens com conteúdos diversos, comparado com imagens-modelo utilizando o mesmo algoritmo, que não implica na utilização para análises em situações de altíssima precisão, mas mostra-se satisfatória para visualização e acompanhamento do andamento de áreas rurais e urbanas.

Imagem	DP Máx	DP Min	% Erro
Modelo	0,20	0,15	2,23
Real	9,74	0,44	17,4

Fig. 18: Tabela de comparação de desvios-padrão entre as imagens modelo e o banco de imagens real.



Figs. 19 e 20: Imagens com poucos elementos distintos.

No caso das figuras 19 e 20, fez-se necessário a adição de marcadores durante a captura das imagens para que se obtenha maior precisão no alinhamento. Esses marcadores podem ser objetos, ou formas geométricas adicionadas manualmente às imagens.

Os resultados obtidos durante os testes utilizando o banco de imagens coletadas demonstraram a capacidade do software em realizar o alinhamento de forma satisfatória, gerando produtos cartográficos de alta precisão. A aplicação de técnicas de realce de bordas para a detecção de pontos de interesse e a implementação de algoritmos de otimização da matriz de homografia e filtragem de correspondências com análise dos ângulos foram estratégias eficazes para melhorar o desempenho e assertividade do programa.

V. CONCLUSÃO E TRABALHOS FUTUROS

O programa desenvolvido demonstra um comportamento alinhado com os estudos consolidados sobre técnicas de alinhamento de imagens. A utilização do robusto algoritmo ORB proporciona resultados confiáveis na identificação de pontos-chave, enquanto a interface gráfica facilita a interação do usuário com o processo de alinhamento. A capacidade do programa em lidar eficientemente com imagens aéreas, especialmente aquelas capturadas por VANTs, destaca sua adaptabilidade a cenários práticos. A implementação de funcionalidades como o ajuste dinâmico de parâmetros e a visualização em tempo real dos buscam otimizar a eficácia do programa, alinhando-se positivamente com os princípios fundamentais dos estudos em alinhamento de imagens.

Para trabalhos futuros é interessante a exportação desses arquivos em formatos compatíveis com sistemas de informação geográfica (SIG) para a criação de mapas. A

criação de uma documentação detalhada também poderia facilitar na ampliação do projeto e integração com outros programas.

REFERÊNCIAS

- [1] Pasetto, Giovanni Argenta et al. Uso da aerofotogrametria como ferramenta na mineração. Em: Anais do 10º Salão Internacional de Ensino, Pesquisa e Extensão - SIEPE, Universidade Federal do Pampa, Santana do Livramento, 6-8 nov. 2018.
- [2] Mello, C. C. S., & Simões, G. F. (2019). Aerofotogrametria com VANT aplicada ao monitoramento volumétrico de aterros sanitários. In Anais do IX Congresso Brasileiro de Geotecnia Ambiental (REGEO 2019) e VIII Congresso Brasileiro de Geossintéticos (Geossintéticos 2019), São Carlos, São Paulo, Brasil. São Carlos/SP, Brasil: IGS-Brasil/ABMS.
- [3] AVILA, Calinca barão de. Aplicabilidade de imagens obtidas a partir de aerofotogrametria realizada com aeronave remotamente pilotada como subsídio à gestão ambiental, Universidade Federal de Santa Maria – RS. 2019.
- [4] ITO, Toshihide et al. Obtaining digital topographic data using an automatic photogrammetry system with digital imaging in an open pit mine. *Geoinformatics*, v. 10, n. 4, p. 225-233, Nov. 1999. I. S.
- [5] Silva, D. C., & Costa, G. C. (2010). Aerofotogrametria em projetos de estradas. Universidade Federal de Pernambuco – UFPE, Departamento de Engenharia Cartográfica DECart, Recife/PE, Maia Melo Engenharia Ltda.
- [6] OpenCV. OpenCV Documentation. Disponível em: <https://docs.opencv.org/>. Acesso em: 20/04/2023.
- [7] Lindeberg, Tony. Scale invariant feature transform. KTH, School of Computer Science and Communication (CSC), Estocolmo, Suécia, 2012. Disponível em: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A480321&dsid=3372>. Acesso em: 01/04/2023.
- [8] BAY, Herbert; TUYTELAARS, Tinne; VAN GOOL, Luc. SURF: Speeded Up Robust Features. ETH Zurich, Katholieke Universiteit Leuven, 2006. Disponível em: <https://lirias.kuleuven.be/73068?limo=0>. Acesso em: 4/04/2023
- [9] Karami, E., Prasad, S., & Shehata, M. (Year). Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. Faculty of Engineering and Applied Sciences, Memorial University, Canada. Disponível em: <https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf>. Acesso em: 08/11/2023
- [10] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In IEEE International Conference on Computer Vision, ICCV 2011. DOI: 10.1109/ICCV.2011.6126544.
- [11] The GIMP Help Team, GIMP Documentation, 2023. Disponível em: <https://docs.gimp.org/2.10/en/gimp-filter-unsharp-mask.html>. Acesso em: 11/11/2023
- [12] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). Numerical Recipes 3rd Edition: The Art of Scientific Computing (3rd ed.). Publicado em 10 de setembro de 2007.
- [13] Yoon, G. (2022). Comparison Between Python and Web User Interfaces. Metropolia University of Applied Sciences. Bachelor of Engineering in Information Technology. Bachelor's Thesis. Publicado em 23 de agosto de 2022.
- [14] Lindeberg, Tony. Scale invariant feature transform. KTH, School of Computer Science and Communication (CSC), Estocolmo, Suécia, 2012. Disponível em: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A480321&dsid=3372>. Acesso em: 01/04/2023.
- [15] DJI. DJI Phantom 4 Pro/ Pro+ User manual. 2016. Disponível em: https://dl.djicdn.com/downloads/phantom_4_pro/Phantom+4+Pro+Pro+Plus+User+Manual+v1.0.pdf. Acesso em: 25/11/2023.
- [16] Li, J., Hu, Q., & Ai, M. RIFT: Multi-Modal Image Matching Based on Radiation-Variation Insensitive Feature Transform. (2020). IEEE Transactions on Image Processing, 29.
- [17] BOGDANCHIKOV, A et al 2013 Python to learn programming, IOP Science. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/423/1/012027>. Acesso em: 08/11/2023