MC949/MO446 – Trabalho 1 - Panorama

Eduardo Bouhid (RA 299223) Lucas Rodrigues Pimentel (RA 252615)

Marcelo de Souza Corumba de Campos (RA 236730)

Thiago do Carmo Rodrigues Pinto (RA 237827) Vinícius Borges Leite (RA 260716)

31 de Agosto de 2025

1 Introdução

A construção de panoramas é uma tarefa clássica em Visão Computacional que consiste em unificar múltiplas imagens sobrepostas em uma única representação de campo de visão ampliado. Com vastas aplicações, de realidade virtual a navegação autônoma, a técnica envolve superar desafios como a detecção de *keypoints* robustos a variações de escala e perspectiva, o cálculo de transformações geométricas (homografias) para o alinhamento preciso, e a fusão das imagens para minimizar artefatos visuais como "fantasmas"ou descontinuidades.

Este relatório detalha a implementação de uma *pipeline* completa para a criação de panoramas. O processo inicia com uma análise comparativa entre os detectores SIFT (*Scale-Invariant Feature Transform*) e ORB (*Oriented FAST and Rotated BRIEF*), seguido pelo emparelhamento de características com os matchers FLANN e BFMatcher, e filtragem via teste da razão de Lowe. Posteriormente, o algoritmo RANSAC (*Random Sample Consensus*) é usado para estimar homografias robustas, permitindo o alinhamento das imagens com a função `warpPerspective`. A etapa final consiste na composição do panorama com técnicas de *blending*, utilizando máscaras gaussianas para suavizar as regiões de transição e garantir a coerência do resultado.

2 Metodologia

2.1 Coleta e Pré-processamento das Imagens

O conjunto de dados utilizado neste trabalho foi capturado em um ambiente interno controlado, visando analisar o desempenho do algoritmo sob diferentes tipos de movimento da câmera. Foram coletadas duas sequências de imagens do mesmo cenário, mas com metodologias de captura distintas:

- **Cena de Translação:** Uma sequência de imagens obtida movendo o smartphone lateralmente, mantendo-o paralelo à cena (Figura 1).
- **Cena de Rotação:** Uma segunda sequência capturada ao girar o smartphone em torno de um ponto fixo (Figura 2).

Ambas as sequências são compostas por imagens registradas com um dispositivo smartphone comum, e garantem uma sobreposição parcial de aproximadamente 30-50%. A captura ocorreu sob condições de iluminação controlada e com a cena estática, buscando minimizar artefatos de "ghosting"e permitir uma avaliação focada no impacto do movimento da câmera.

A lista de imagens é ordenada alfa numericamente para assumir uma sequência lógica, embora uma detecção automática de ordem (baseada em ranqueamento de emparelhamentos) possa ser explorada como extensão.

2.2 Seleção e Extração de Características Locais – A busca pelo melhor detector

A escolha de um detector de características robusto é um passo fundamental para o sucesso da construção de panoramas. Para fundamentar essa decisão de forma empírica, foi realizada uma análise comparativa

Imagens	Tempo de Exposição	ISO	Abertura	Dist. Focal	Tamanho (px)
1–4	1/60 s	160	f/1.8	4 mm	3024×4032
5–9	1/60 s	200	f/1.8	4 mm	3024×4032

Figura 1: Cena do escritório com variação por **translação**: imagens de entrada e metadados.

quantitativa entre os detectores SIFT (*Scale-Invariant Feature Transform*), ORB (*Oriented FAST and Rotated BRIEF*) e AKAZE (*Accelerated KAZE*).

A avaliação foi conduzida por meio de um script automatizado que executou o seguinte pipeline (similar ao efetivamente utilizado para a criação do panorama) para ambos os detectores:

1. **Detecção e Extração:** Para cada imagem de entrada, os *keypoints* e descritores foram extraídos após a conversão para escala de cinza. Ambos os detectores foram configurados para extrair um máximo de 12.000 características por imagem (`nfeatures=12000`).

2. **Correspondência de Características:** As características entre pares de imagens adjacentes foram comparadas utilizando o método `knnMatch` com $k = 2$. Para o SIFT, utilizou-se o *FlannBasedMatcher*, enquanto para o ORB, o *BFMatcher* com a norma de Hamming.

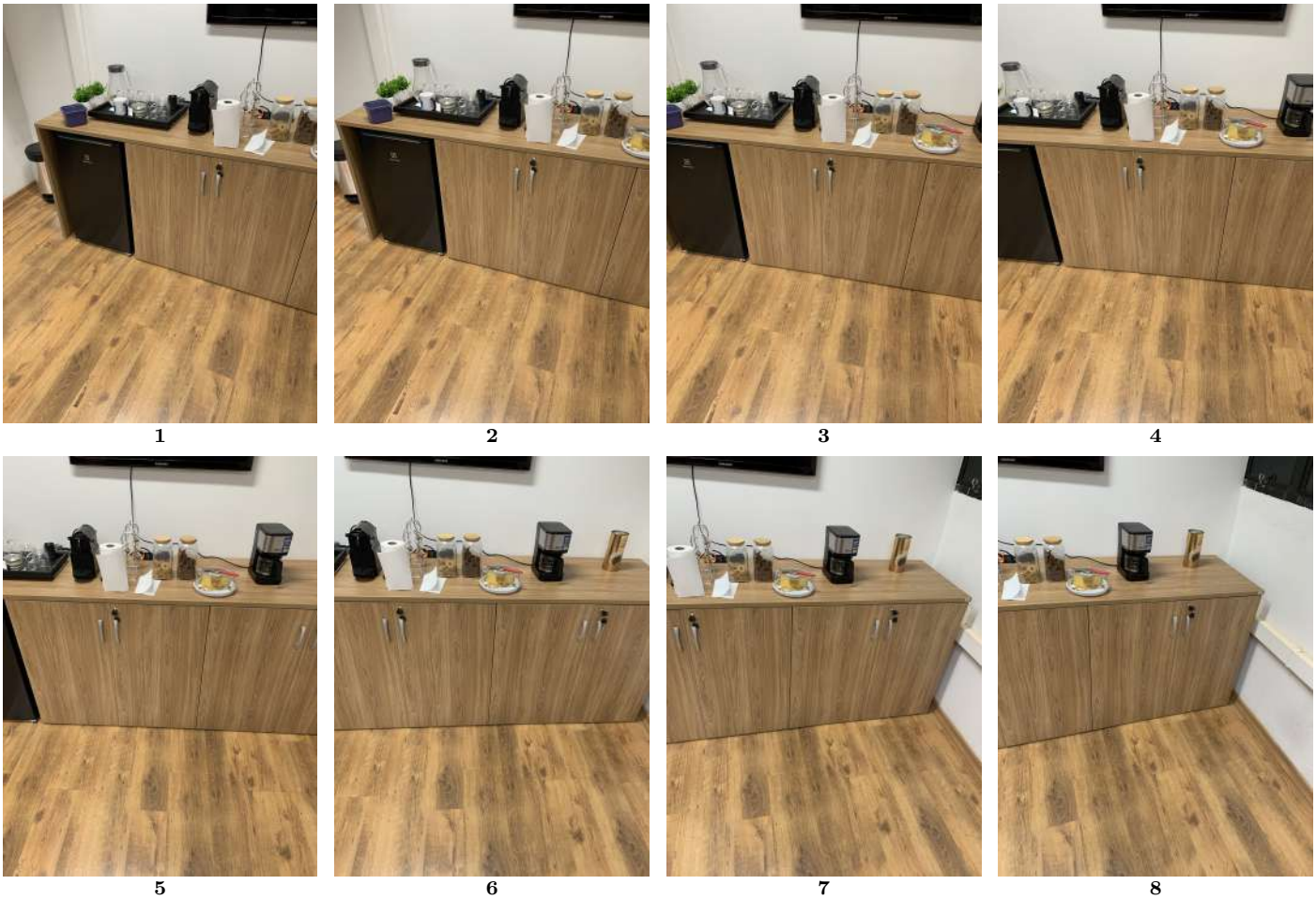
3. **Filtragem de Correspondências:** O Teste da Razão de Lowe foi aplicado para filtrar correspondências ambíguas, mantendo-se apenas aquelas cuja distância do melhor *match* (m) era significativamente menor que a do segundo melhor (n), com um limiar de $m.distance < 0.75 \times n.distance$.

4. **Estimativa de Homografia e Inliers:** A matriz de homografia entre os pares de imagens foi calculada utilizando o algoritmo RANSAC, com um limiar de reprojeção de 1.5 pixels. O número de *inliers* — correspondências consistentes com o modelo geométrico estimado — foi contabilizado como a principal métrica de robustez.

Os critérios de decisão para a seleção do detector foram, primariamente, o número médio de correspondências após o Teste da Razão e, de forma mais crítica, o número médio de *inliers* obtidos após a aplicação do RANSAC.

2.3 Emparelhamento de Características

Após a extração dos descritores, a etapa subsequente consiste em estabelecer correspondências entre pares de imagens adjacentes, um processo cuja estratégia foi customizada para as propriedades intrínsecas de cada



Metadados (constantes para todas as imagens): 1/60s, ISO 160, f/1.8, 4mm, 3024×4032px.

Figura 2: Cena do escritório com variação por **rotação**: imagens de entrada e metadados.

detector. A escolha do método de emparelhamento foi diretamente influenciada pela natureza dos descritores. Para os do SIFT, que consistem em vetores de ponto flutuante de alta dimensionalidade cuja similaridade é avaliada pela distância Euclidiana, foi empregado o matcher **FLANN** (*Fast Library for Approximate Nearest Neighbors*). Esta é uma abordagem otimizada, essencial para realizar buscas por vizinhos próximos de forma eficiente em espaços vetoriais densos. Em contrapartida, uma abordagem distinta foi necessária para os descritores do ORB, que são strings binárias (baseados no BRIEF). Neste caso, a comparação foi realizada utilizando a **distância de Hamming**, uma métrica computacionalmente leve que conta o número de bits divergentes. Dada essa característica, utilizou-se um **BFMatcher** (*Brute-Force Matcher*), que, apesar de realizar uma comparação exaustiva, mostra-se altamente performático para descritores binários.

Independentemente do matcher utilizado, a filtragem de correspondências ambíguas seguiu um critério unificado para garantir a robustez dos resultados. Foi aplicado o **Teste da Razão de Lowe**, configurando-se o processo de emparelhamento para encontrar os dois vizinhos mais próximos para cada descritor ($k = 2$). Uma correspondência somente foi validada se a distância ao melhor vizinho fosse significativamente menor que a distância ao segundo melhor, aplicando-se um limiar de 0.25 para essa razão. A eficácia deste processo foi então validada qualitativamente por meio da inspeção visual das conexões entre os pontos-chave correspondentes. Essa análise é fundamental para diagnosticar potenciais falhas, como emparelhamentos incorretos em regiões de baixa textura ou sob a influência de objetos que se moveram entre as capturas.

2.4 Estimativa de Homografia, Composição, Alinhamento e *Blending*

Para alinhar as imagens, estima-se a homografia H (matriz 3x3 de transformação projetiva) entre pares consecutivos usando RANSAC (*Random Sample Consensus*) via `cv2.findHomography`, com limiar de reprojeção de 2.0 *pixels* e confiança de 99.5%. Isso rejeita *outliers*, garantindo robustez a emparelhamentos errôneos.

As homografias são encadeadas para um *frame* de referência central (imagem do meio da sequência) para minimizar distorções acumuladas. A função `estimate_pairwise_homographies` computa H_i para cada par, e `chain_homographies_to_ref` acumula transformações $H_{\text{cum},i}$ multiplicando matrizes. Cada imagem é distorcida com `cv2.warpPerspective` para o plano de referência, considerando *offsets* para um *canvas* expandido

que acomode todas as imagens sem recortes negativos.

A composição final cria um mosaico no *canvas* estimado, projetando todas as imagens alinhadas. Para suavizar transições, aplica-se *blending* com máscaras gaussianas (σ ajustável, por padrão igual a 3.0), geradas com `cv2.GaussianBlur` sobre máscaras binárias das regiões projetadas. Isso pondera pixels nas sobreposições, reduzindo descontinuidades visíveis e aprimorando o panorama. Métricas básicas, como inspeção de gradientes nas bordas, foram usadas para validar a qualidade. O panorama resultante é visualizado com Matplotlib.

3 Resultados

3.1 Detectores de Características

Para as duas imagens, realizou-se a comparação entre os descritores SIFT, ORB e AKAZE. Os resultados indicam que a escolha do melhor detector de características/pontos-chave é influenciado pela natureza do movimento da câmera durante a captura das imagens. O SIFT se mostrou mais robusto para as cenas com movimentos de translação, enquanto o ORB ofereceu uma combinação de precisão superior e a já esperada eficiência computacional para cenas com movimentos de rotação.

Tabela 1: Comparação de Desempenho de Detectores de Keypoints para a Fig. 1.

Detector	**Nº Médio de Keypoints**	**Nº Médio de Matches**	**Nº Médio de Inliers**	**Tempo de Execução (s)**
SIFT	12000.11	2398.75	1067.75	11.4031
ORB	12000.0	2483.25	473.38	1.2575
AKAZE	6643.44	2875.5	853.38	6.0477

Tabela 2: Comparação de Desempenho de Detectores de Keypoints para a Fig. 2.

Detector	**Nº Médio de Keypoints**	**Nº Médio de Matches**	**Nº Médio de Inliers**	**Tempo de Execução (s)**
SIFT	12000.0	3170.14	1319.0	10.0058
ORB	12000.0	4435.86	1819.43	1.0033
AKAZE	5992.62	3601.43	1464.0	5.496

Para complementar a análise quantitativa, foi realizada uma inspeção visual da distribuição dos pontos-chave detectados pelos algoritmos SIFT e ORB. As Figuras 3a e 3b ilustram essa comparação para uma imagem representativa de cada cena, permitindo uma avaliação qualitativa da densidade e localização dos pontos de interesse.

(a) Cena com movimento de rotação.

(b) Cena com movimento de translação.

Figura 3: Comparação visual dos keypoints detectados pelos algoritmos SIFT (esquerda) e ORB (direita) em imagens representativas de cada tipo de cena.

3.2 Emparelhamento das características encontradas

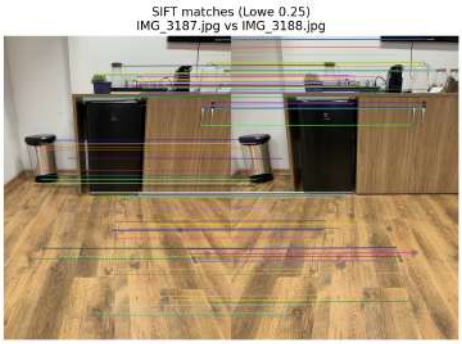
Após a seleção do detector ótimo para cada cenário, a etapa de emparelhamento de características foi executada com uma estratégia específica para cada um, visando maximizar a qualidade das correspondências.

A etapa de emparelhamento de características foi executada com estratégias distintas e otimizadas para cada par detector-cena, visando maximizar a qualidade das correspondências. Para a cena de translação, onde

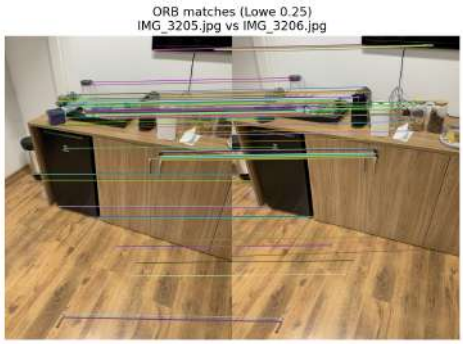
o detector SIFT foi selecionado, empregou-se um método robusto para seus descritores de ponto flutuante. O processo utilizou um *matcher* baseado em **FLANN** (*Fast Library for Approximate Nearest Neighbors*), escolhido por sua alta eficiência em encontrar correspondências em espaços de alta dimensão. A subsequente filtragem foi realizada pelo **Teste da Razão de Lowe**, no qual o método `knnMatch` foi usado para encontrar os dois vizinhos mais próximos ($k = 2$). Uma correspondência foi validada apenas se a distância do melhor vizinho fosse significativamente menor que a do segundo, com um limiar restritivo de 0.25 ($d_1 < 0.25 \times d_2$).

De forma análoga, a abordagem para a cena de rotação foi adaptada às propriedades do detector ORB e seus descritores binários. Neste caso, utilizou-se um **BFMatcher** (*Brute-Force Matcher*), cuja abordagem de comparação exaustiva é computacionalmente viável graças à eficiência do cálculo da **distância de Hamming**. A mesma estratégia de filtragem, o **Teste da Razão de Lowe**, foi então aplicada para garantir a consistência e a qualidade das correspondências encontradas.

A Figura 4 ilustra o resultado final dessas duas estratégias de emparelhamento distintas, exibindo as correspondências de maior qualidade para um par de imagens representativo de cada cena.



(a) Emparelhamento com SIFT e filtro pelo Teste da Razão para a cena de translação.



(b) Emparelhamento com ORB e filtro pelo Teste da Razão para a cena de rotação.

Figura 4: Visualização das correspondências de características encontradas com as estratégias de emparelhamento otimizadas para cada cena e detector.

4 Alinhamento e Panoramas Finais

Com os conjuntos de correspondências de características validadas, o passo seguinte é estimar a transformação geométrica entre cada imagem e uma imagem de referência, que neste trabalho foi definida como a imagem central da sequência. Para isso, calcula-se a matriz de homografia (H) utilizando o algoritmo RANSAC (*Random Sample Consensus*) disponibilizado no *OpenCV*, que é robusto a *outliers* que possam ter restado após a etapa de filtragem.

É importante frisar que todas as homografias são calculadas em relação à imagem central. As transformações para imagens não adjacentes à referência (por exemplo, a primeira imagem em uma sequência de cinco) são obtidas pelo produto acumulado das matrizes de homografia intermediárias, garantindo que todas as imagens sejam mapeadas para o mesmo plano de projeção.

Uma vez que as matrizes de homografia são conhecidas, cada imagem é projetada no plano da imagem de referência através de uma transformação de perspectiva (utilizando a função `cv2.warpPerspective`). Este processo alinha todas as imagens em um único canvas de maior dimensão.

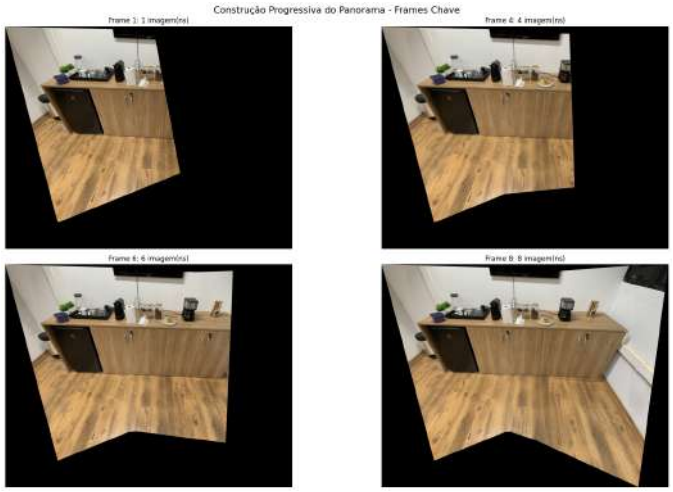
Finalmente, para suavizar as transições e emendas entre as imagens sobrepostas e evitar bordas visíveis, foi aplicado um processo de mesclagem (*blending*). Utilizou-se um filtro de kernel Gaussiano com um desvio padrão (σ) de 3 para ponderar a contribuição dos pixels nas regiões de sobreposição, resultando em um panorama final com aparência contínua e natural.

A Figura 5 demonstra visualmente a construção progressiva dos panoramas para ambas as cenas, desde o alinhamento das primeiras imagens até o resultado final.

5 Discussão

A implementação da *pipeline* de montagem de panoramas foi bem-sucedida em produzir representações espacialmente contínuas a partir de múltiplas imagens. A principal conclusão deste trabalho é a forte dependência dos resultados e das estratégias de implementação ao tipo de movimento da câmera durante a captura. Essa dependência se manifestou desde a seleção do detector de características: enquanto o ORB, com sua alta velocidade e volume massivo de *matches*, mostrou-se a escolha ideal para o cenário de rotação pura, a robustez

(a) Construção progressiva do panorama para a cena de translação (SIFT).



(b) Construção progressiva do panorama para a cena de rotação (ORB).

Figura 5: Visualização da montagem progressiva dos panoramas. Cada grade 2x2 mostra o processo de alinhamento sequencial, culminando no panorama completo, exibido no canto inferior direito de cada grade.

geométrica superior do SIFT foi indispensável para a cena de translação, onde as variações de perspectiva são mais acentuadas e exigem um maior número de *inliers* para uma estimação confiável.

Uma das observações mais significativas foi a superioridade qualitativa do panorama gerado a partir da cena de rotação em comparação ao de translação. O panorama rotacional apresentou um alinhamento satisfatório e poucos artefatos visuais, enquanto o translacional, apesar de funcional, exibiu desalinhamentos e "fantasmas"mais pronunciados. A explicação para essa diferença reside na adequação do modelo matemático utilizado — a homografia — a cada um dos cenários.

Teoricamente, uma homografia descreve **perfeitamente** a transformação entre duas imagens capturadas por uma câmera que apenas gira em torno de seu centro óptico. Nesse caso, a relação entre os planos das imagens é puramente projetiva, e a profundidade dos objetos na cena 3D é irrelevante para o alinhamento 2D. Contudo, para o movimento de translação, a homografia é um modelo limitado, pois assume que todos os pontos da cena repousam sobre um único plano no espaço 3D. Em um ambiente real, como o escritório fotografado, com objetos em diferentes profundidades, o movimento de translação induz o efeito de paralaxe: objetos mais próximos da câmera se deslocam mais rapidamente no plano da imagem do que objetos distantes. Uma única matriz de homografia é incapaz de modelar esse deslocamento múltiplo e dependente da profundidade. Esses desalinhamentos geométricos, causados pela paralaxe na cena de translação, não podem ser corrigidos na etapa de *blending*. Embora a mesclagem com kernel Gaussiano tenha sido eficaz em suavizar as emendas e variações de iluminação em ambos os cenários, ela não consegue compensar erros de alinhamento fundamentais.

Conclui-se que o objetivo de implementar uma solução foi atingido, explorando os principais conceitos e *trade-offs* da área. A partir dos resultados, pôde-se teorizar que enquanto o modelo de homografia é suficiente e eficaz para panoramas rotacionais, cenários com movimento de translação exigiriam modelos mais complexos para resultados de alta fidelidade. Reconhece-se que os resultados são qualitativamente inferiores aos de soluções comerciais modernas, que frequentemente se beneficiam de mais imagens intermediárias, modelos de alinhamento mais sofisticados (que lidam com paralaxe) e, possivelmente, técnicas de otimização baseadas em *Machine Learning*.

6 Referências

1. S. A. K. Tareen and R. H. Raza, "Potential of SIFT, SURF, KAZE, AKAZE, ORB, BRISK, AGAST, and 7 More Algorithms for Matching Extremely Variant Image Pairs,"2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2023, pp. 1-6, doi: 10.1109/iCoMET57998.2023.10099250.

2. S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK,"2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2018, pp. 1-10, doi: 10.1109/ICOMET.2018.8346440.