

Construção de Panoramas a partir de Múltiplas Imagens

MC949/MO446 - Trabalho 1

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

Universidade Estadual de Campinas (UNICAMP)

3 de Setembro de 2025

O Que é um Panorama?

- **Definição:** Uma técnica de Visão Computacional que unifica múltiplas imagens sobrepostas em uma única imagem com um campo de visão ampliado.
- **Aplicações:** Realidade virtual, mapeamento, fotografia computacional e navegação autônoma.
- **Desafios Técnicos:**
 - Detecção de pontos-chave robustos.
 - Alinhamento geométrico preciso (cálculo de homografia).
 - Fusão (*blending*) suave para evitar artefatos visuais.

Multi-Band Blended Panorama (levels=6)



Etapas Principais da Pipeline

Com base no nosso objetivo, a pipeline foi estruturada em quatro etapas fundamentais:

- 1 **Seleção do Detector:** Análise comparativa para escolher o melhor detector de características para cada cenário de captura.
- 2 **Emparelhamento e Filtragem:** Encontrar correspondências consistentes entre as imagens e filtrar as ambíguas.
- 3 **Estimativa de Homografia:** Calcular a transformação geométrica que alinha as imagens usando o algoritmo RANSAC.
- 4 **Alinhamento e Fusão:** Projetar todas as imagens em um plano comum e aplicar *blending* para suavizar as transições.

Coleta de Dados: Cenários de Teste

Cena de Translação

- Exposição: 1/60 s; ISO 200;
 $f/1.8$; Dist. focal 4mm;
3024x4032



Cena de Rotação

- Exposição: 1/60 s; ISO 160;
 $f/1.8$; Dist. focal 4mm;
3024x4032



Análise de Detectores: O Desafio

- A escolha do detector de características é um passo crítico. A robustez das correspondências depende fundamentalmente dele.
- Para tomar uma decisão baseada em dados, realizamos uma análise quantitativa detalhada.

Análise de Detectores: O Desafio

- A escolha do detector de características é um passo crítico. A robustez das correspondências depende fundamentalmente dele.
- Para tomar uma decisão baseada em dados, realizamos uma análise quantitativa detalhada.

Candidatos Avaliados

- **SIFT:** Robusto a escala e rotação, mas computacionalmente caro.
- **ORB:** Rápido e eficiente, ideal para aplicações em tempo real.
- **AKAZE:** Bom equilíbrio entre robustez e velocidade.

Análise de Detectores: O Desafio

- A escolha do detector de características é um passo crítico. A robustez das correspondências depende fundamentalmente dele.
- Para tomar uma decisão baseada em dados, realizamos uma análise quantitativa detalhada.

Candidatos Avaliados

- **SIFT**: Robusto a escala e rotação, mas computacionalmente caro.
- **ORB**: Rápido e eficiente, ideal para aplicações em tempo real.
- **AKAZE**: Bom equilíbrio entre robustez e velocidade.

Métrica de Decisão Principal

O número de **inliers** após a estimativa de homografia com RANSAC. Esta métrica reflete a consistência geométrica das correspondências.

Análise de Detectores: Cena de Translação

Detector	Keypoints (média)	Matches (média)	Inliers (média)	Tempo (s)
SIFT	47525.89	6927.12	4078.88	17.8843
ORB	47315.78	9149.12	3962.62	10.777
AKAZE	6647.0	2868.38	980.25	7.0281

Conclusão

O **SIFT** foi escolhido. Apesar de ser mais lento, ele gerou mais do que o dobro de inliers, indicando uma robustez geométrica superior, essencial para lidar com a paralaxe.

Análise de Detectores: Cena de Translação

ORB (vermelho) vs SIFT (verde) - 'IMG_3187.jpg'



Análise de Detectores: Cena de Rotação

Detector	Keypoints (média)	Matches (média)	Inliers (média)	Tempo (s)
SIFT	26161.88	5025.0	2391.0	12.1025
ORB	36775.75	10802.57	4508.43	5.5895
AKAZE	6003.0	3606.29	1678.0	6.1857

Conclusão

O **ORB** foi a escolha ideal. Ele foi 10x mais rápido que o SIFT e produziu o maior número de inliers, oferecendo a melhor combinação de eficiência e precisão para este cenário.

Análise de Detectores: Cena de Rotação

ORB (vermelho) vs SIFT (verde) - 'IMG_3205.jpg'



Estratégia de Emparelhamento Otimizada

Para a Cena de Translação (com SIFT)

- **Descritores:** Vetores de ponto flutuante.
- **Matcher:** **FLANN**, otimizado para buscas eficientes em espaços de alta dimensão.
- **Métrica de Distância:** Euclidiana.

Para a Cena de Rotação (com ORB)

- **Descritores:** Strings binárias (BRIEF).
- **Matcher:** **Brute-Force (BFMatcher)**, viável para descritores binários.
- **Métrica de Distância:** Hamming.

- Este é um passo crucial para garantir a alta qualidade das correspondências.
- **Método:** Para cada descritor, encontramos os dois vizinhos mais próximos ($k = 2$).
- Uma correspondência só é aceita se a distância do melhor match (d_1) for significativamente menor que a do segundo melhor (d_2).
- **Limiar utilizado:** $d_1 < 0.75 \times d_2$.

Filtragem de Correspondências: Exemplo Visual

ORB matches (Lowe 0.75)
IMG_3205.jpg vs IMG_3206.jpg



Estimativa da Homografia

- O objetivo é encontrar a matriz de transformação projetiva (Homografia H) que alinha um par de imagens.
- Utilizamos o algoritmo **RANSAC** ('cv2.findHomography'), que é robusto a outliers (correspondências incorretas) que possam ter restado após a filtragem.
- RANSAC estima " H " iterativamente a partir de subconjuntos aleatórios de pontos e encontra o que melhor se ajusta à maioria das correspondências.

Estimativa da Homografia

```
H 0->1 estimated. Inliers: 1642/2128
H_0->1:[[ 1.015 -0.118 52.659]
 [ 0.004 1.009 -16.161]
 [ 0. 0. 1. ]]
H 1->2 estimated. Inliers: 791/1075
H_1->2:[[ 0.97 -0.214 317.767]
 [-0.012 0.985 10.607]
 [-0. 0. 1. ]]
H 2->3 estimated. Inliers: 518/881
H_2->3:[[ 1.036 -0.206 45.867]
 [ 0.012 1.021 -28.873]
 [ 0. 0. 1. ]]
H 3->4 estimated. Inliers: 371/707
H_3->4:[[ 0.992 -0.254 245.401]
 [ 0.01 0.987 13.051]
 [-0. -0. 1. ]]
```


- Para evitar o acúmulo de erros, não alinhamos as imagens em uma cadeia simples ($1 \rightarrow 2$, $2 \rightarrow 3$, etc.).
- Em vez disso, escolhemos a **imagem central** da sequência como o plano de referência.
- Todas as outras imagens são alinhadas em relação a ela. As homografias são multiplicadas em cadeia para mapear cada imagem ao plano central.
- A função 'cv2.warpPerspective' projeta cada imagem nesse plano comum.

- Após o alinhamento, as imagens são sobrepostas, mas as emendas podem ser visíveis devido a pequenas diferenças de exposição ou erros de alinhamento.
- Para criar uma transição suave, aplicamos uma técnica de *blending*.
- Uma máscara Gaussiana é usada para ponderar os pixels nas regiões de sobreposição, criando um panorama final com aparência contínua e natural.

Sobreposição e *Blending*: Comparação de abordagens



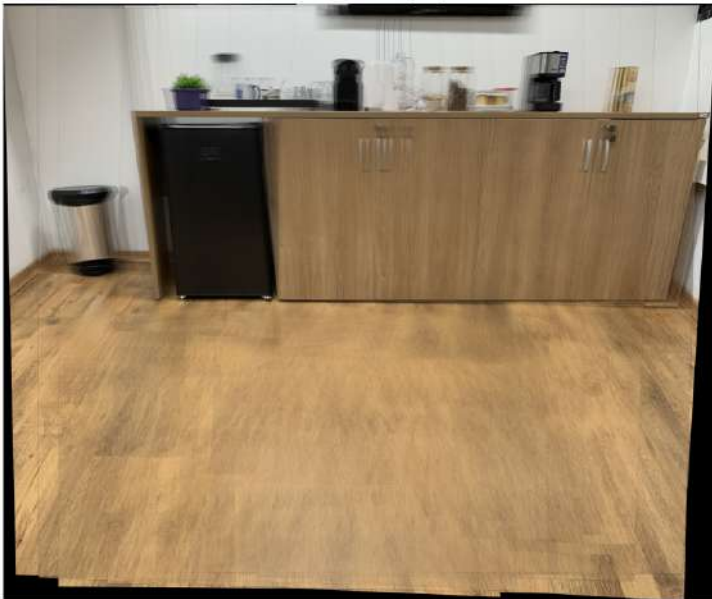
Panorama Final: Cena de Rotação

Multi-Band Blended Panorama (levels=6)



Panorama Final: Cena de Translação

Panorama (Gaussian blending, $\sigma=17$)



Discussão: O Modelo de Homografia

A grande diferença de qualidade entre os dois panoramas se deve à compatibilidade do movimento da câmera com o modelo de **homografia**.

Cenário Ideal: Rotação Pura

Uma homografia descreve **perfeitamente** a transformação entre imagens de uma câmera que apenas gira em torno de seu centro óptico. A profundidade da cena é irrelevante, resultando em um alinhamento de alta qualidade.

Discussão: O Problema da Paralaxe

Cenário Problemático: Translação

- Uma homografia assume que todos os pontos da cena estão em um **único plano**.
- O movimento de translação em uma cena 3D real causa **paralaxe**: objetos próximos se movem mais rápido na imagem do que objetos distantes.
- Uma única matriz de homografia é incapaz de modelar este efeito complexo.

Resultado

Desalinhamentos e fantasmas (*ghosting*), que não podem ser corrigidos na etapa de *blending*.

Conclusões: Não Existe “Bala de Prata”

- **Principal Aprendizado:** A construção de panoramas não possui uma solução única. A estratégia ótima é **intrinsecamente ligada ao cenário de captura**.

Conclusões: Não Existe “Bala de Prata”

- **Principal Aprendizado:** A construção de panoramas não possui uma solução única. A estratégia ótima é **intrinsecamente ligada ao cenário de captura**.
- **A Limitação Fundamental do Modelo:** O sucesso da nossa pipeline foi ditado pela adequação da **homografia** ao movimento da câmera.
 - **Para Rotação:** O modelo de homografia é matematicamente perfeito. Isso permitiu que o rápido e eficiente **ORB** produzisse um resultado excelente.
 - **Para Translação:** A homografia falha ao não conseguir modelar a **paralaxe**. Mesmo a robustez geométrica superior do **SIFT** não pôde compensar a limitação do modelo, resultando em artefatos.

Conclusões: Não Existe “Bala de Prata”

- **Principal Aprendizado:** A construção de panoramas não possui uma solução única. A estratégia ótima é **intrinsecamente ligada ao cenário de captura**.
- **A Limitação Fundamental do Modelo:** O sucesso da nossa pipeline foi ditado pela adequação da **homografia** ao movimento da câmera.
 - **Para Rotação:** O modelo de homografia é matematicamente perfeito. Isso permitiu que o rápido e eficiente **ORB** produzisse um resultado excelente.
 - **Para Translação:** A homografia falha ao não conseguir modelar a **paralaxe**. Mesmo a robustez geométrica superior do **SIFT** não pôde compensar a limitação do modelo, resultando em artefatos.
- **Conclusão Final:** A escolha de algoritmos é um trade-off governado pela física da captura. Compreender as limitações do modelo geométrico é mais crucial do que apenas selecionar o detector mais “poderoso”.

Próximos Passos para uma Solução Mais Robusta

- **Automação da Pipeline:**

- Implementar a detecção automática da ordem das imagens, eliminando a necessidade de uma sequência pré-definida.
- Otimizar a seleção de parâmetros (ex: limiar do teste da razão) de forma adaptativa.

Próximos Passos para uma Solução Mais Robusta

- **Automação da Pipeline:**

- Implementar a detecção automática da ordem das imagens, eliminando a necessidade de uma sequência pré-definida.
- Otimizar a seleção de parâmetros (ex: limiar do teste da razão) de forma adaptativa.

- **Tratamento Robusto de Paralaxe:**

- Investigar modelos de alinhamento mais sofisticados que a homografia, como deformações baseadas em malhas (*mesh-based warping*), para lidar com os efeitos de paralaxe em cenas de translação e com múltiplas profundidades.

Próximos Passos para uma Solução Mais Robusta

- **Automação da Pipeline:**

- Implementar a detecção automática da ordem das imagens, eliminando a necessidade de uma sequência pré-definida.
- Otimizar a seleção de parâmetros (ex: limiar do teste da razão) de forma adaptativa.

- **Tratamento Robusto de Paralaxe:**

- Investigar modelos de alinhamento mais sofisticados que a homografia, como deformações baseadas em malhas (*mesh-based warping*), para lidar com os efeitos de paralaxe em cenas de translação e com múltiplas profundidades.

- **Expansão para Panoramas 360°:**

- Adaptar a pipeline para lidar com o fechamento de loop (*loop closure*), onde a primeira e a última imagem da sequência se sobrepõem.
- Utilizar projeções esféricas ou cilíndricas em vez de planas para mapear as imagens e criar panoramas imersivos completos.

Obrigado!