ISSN 2447-6102

Artigo

Reconstrução 3D usando geometria trinocular

Andrade, G.C. [1], G. A. Monerat [1], Ventura J. [1], De Moura Neto, F. [1] e Fabbri, R. [1]

[1] Universidade do Estado do Rio de Janeiro, Instituto Politécnico;
***** Correspondence: gabriel.andrade@iprj.uerj.br;

Received: 18/01/2024; Accepted: 25/01/2024; Published: 31/01/2024

Resumo: Estrutura a partir de Movimento (*structure from motion*) é um problema de visão computacional que busca obter cenas tridimensionais a partir de um conjunto de imagens tiradas de diferentes pontos de vista, sem conhecimento prévio da configuração das câmeras. A abordagem mais utilizada consiste na montagem de uma base de reconstrução inicial com duas imagens, seguida da inserção de imagens uma a uma até a reconstrução completa. Apesar da robustez desta abordagem, há casos de falha que impossibilitam a inicialização da reconstrução, o que leva os sistemas a não concluírem o processo de reconstrução. Recentemente, o uso de três imagens como base geométrica tem demonstrado maior potencial de robustez por fornecer melhor confiabilidade na obtenção das correspondências e câmeras, sendo proposto como alternativa caso a inicialização com duas câmeras falhe. Apenas recentemente tais técnicas se tornaram aplicáveis, graças a avanços da tecnologia de solução, fazendo o uso de três features orientadas. Este artigo expõe os recentes avanços da inclusão do modelo de três câmeras no software de código aberto openMVG largamente utilizado e são providenciados os resultados da experiência empírica, ilustrando sua robustez na prática.

palavras-chave: Visão Computacional; Fotogrametria; Reconstrução 3D; Software Livre

3D reconstruction using trinocular geometry

Abstract: Structure from motion is a computer vision problem that seeks obtain tridimentional scenes from a set of images shot from different points of view without having any previously camera configuration knowledge. The main approach consists in building a base reconstruction with two images, followed up by insertion of images one by one until the complete reconstruction. In spite of it's robustness, there are cases that it's not possible to make the two-view camera initialization leaving SfM systems not conclude the reconstruction process. Recently it is used three image as geometric base that shows bigger robustness potential by providing better reliability in getting correspondences and cameras being proposed as an alternative when two-view initialization fails. Only recently pratical ones could being made due the improvement of solver thechnnieques. These improvements made relative camera pose estimation more robust and efficient by using three oriented SIFT features. This article exposes the recent advances in trinocular geometry into the open source software openMVG alongside provided experimental results showing it's practical robustness.

Keywords: Computer Vision, Photogrammetry, 3D Reconstruction, Free Software

---

1. Introdução

A estrutura a partir de movimento (*structure from motion* ou seu acrônimo SfM) é um problema da geometria de múltiplas vistas que tem por objetivo recriar cenas tridimensionais a partir de um conjunto de imagens sem conhecimento prévio dos parâmetros internos bem como as posições das câmeras que as compõem. Os sistemas de SfM amplamente utilizados pela comunidade como o openMVG [Moulon et al., 2016]

e o Colmap [Schönberger and Frahm, 2016, Schönberger et al., 2016] utilizam como abordagem principal a criação de uma base de reconstrução utilizando duas imagens, seguida da inserção das restantes uma a uma até a obtenção da reconstrução completa, conhecida como incremental ou sequencial.

Apesar da robustez, o modelo de duas câmeras possui falhas em casos em que existem estruturas curvas nas imagens, objetos de baixa textura ou de alta refletividade e transparência [Apple ARKit Team, 2018] e a presença de câmeras excessivamente distantes umas das outras no conjunto de dados. Tais casos de falha impedem não apenas a criação da base de reconstrução, mas também o processo seguinte devido ao fato de interferirem nos algoritmos de geração de correspondências entre as imagens, seja por meio da geração de correspondências incorretas das imagens ou, no pior dos casos, na não produção delas.

Mais especificamente, as estruturas curvas e arredondadas geram ambiguidade na identificação dos pontos de interesse clássicos de forma que mesmo que uma região seja identificada como verdadeira entre as múltiplas imagens (ou em inglês *ground-truth*), quanto à transparência e luminosidade as aplicações dos métodos podem gerar problemas não só na identificação da região de interesse pela luminosidade, mas também pode resultar em detecção incorreta do mesmo e a distância entre imagens dificulta a obtenção de uma amostragem suficientemente robusta para alimentar os algoritmos de estimação de pose como os de cinco e três pontos.

Recentemente, o uso de três imagens na base de inicialização tem demonstrado maior potencial de robustez [Fabbri et al., 2022] e tem sido considerada como uma alternativa para contornar os casos supracitados pois a câmera adicional permite maior confiabilidade na formação correspondências. Mesmo com tais vantagens, sua utilização esteve restrita apenas a estudos experimentais, nunca tendo sido posta à prova nos softwares de reconstrução 3D amplamente utilizados. As vantagens do emprego da inicialização de três câmeras residem na possibilidade da utilização de outros tipos de ponto como os do tipo orientado.

Este trabalho apresenta a primeira aplicação realística do modelo de três câmeras pelo pacote *MInimial problem NUmerical continuation Solver* (MiNuS) desenvolvido pela colaboração entre o Instituto Politécnico e a Brown University no software *openMVG*. Apresenta-se uma explanação sucinta do referencial matemático utilizado pelo MiNuS seguido da disponibilização dos resultados das correspondências obtidas pela aplicação criada em experimentos feitos a partir imagens do Capitólio de Providence, e de uma garrafa de *Absolut Vodka* com a tolerância de erro em uma área de um pixel. O experimento do Capitólio busca evidenciar a robustez do modelo enquanto o da garrafa busca a sua estabilidade.

2. Modelo de três câmeras

Apesar de parecer unicamente uma extensão do modelo de duas câmeras o atual modelo de três câmeras se distingue por usar pontos e, especialmente, suas orientações que permite a utilização completa de pontos de interesse invariante à escala [Internet, 2023b] (scale-invatiant feature transform ou *sift features*).

O problema trifocal chamado Chicago consiste em encontrar as rotações e translações da segunda e terceira câmeras a partir da primeira por meio de três pontos para cada uma das imagens e duas retas tangentes provenientes da orientação de destes pontos. As onze equações algébricas que constituem o modelo, resumidamente, são oriundas dos determinantes das matrizes essenciais para a geometria do sistema. Três equações adicionais codificam restrições provenientes do uso de quatérnios, incluindo três variáveis adicionais resultando em um sistema de quatorze equações com quatorze variáveis livres.

A seguir, iremos detalhar as equações essenciais da geometria do problema.

2.1. Geometria do problema

A ideia base do modelo é apresentada na figura 1 e pelas equações (1) e (2). Um ponto em uma câmera *v* é composto pelo produto do fator de profundidade α_v pelas coordenadas homogêneas $\mathbf{x_v}$ é obtido pelo produto da rotação $\mathbf{R_v}$ pela sua correspondência na primeira câmera seguido da soma com a translação t_v que é sintetizado por (1). A linha paramétrica é computada diferença de um ponto deslocado $\boldsymbol{Y = X + \epsilon D}$ com $\mathbf{X}$, ambos no espaço tridimensional demonstrado por [Fabbri et al., 2022] e exposto em (2) onde $\boldsymbol{\eta_v = \beta_v - \alpha_v}$, $\boldsymbol{\mu_v = \beta_v \delta_v}$, $\boldsymbol{\beta_v}$ é a profundidade da coordenada projetada $\mathbf{y_v}$ e $\boldsymbol{\delta_v}$ o deslocamento ao longo da direção $\mathbf{d_v}$. A figura 1 mostra que o problema de três câmeras consiste em obter as matrizes de rotação e translação R_2, t_2, R_3 e t_3 desconhecidas a partir do conhecimento prévio dos parâmetros internos de posicionamento da primeira, logo R_1, t_1 dela são a matriz de identidade e o vetor nulo nesta ordem.

$$\alpha_v \boldsymbol{x}_v = \boldsymbol{R}_v \alpha_1 \boldsymbol{x}_1 + \boldsymbol{t}_v \quad (1)$$

$$\eta_v \boldsymbol{x}_v + \mu_v \boldsymbol{d}_v = \boldsymbol{R}_v (\eta_1 \boldsymbol{x}_1 + \mu_1 \boldsymbol{d}_1) \quad (2)$$

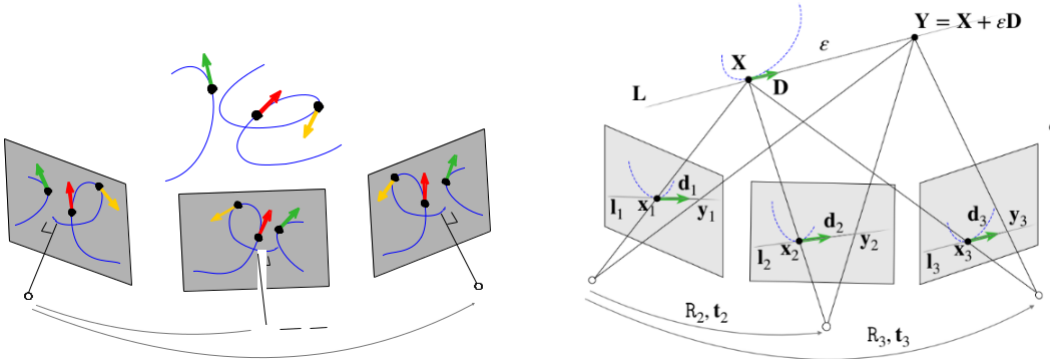

Figura 1. Representação esquemática do modelo de três câmeras. Busca-se obter os parâmetros internos da segunda e da terceira câmeras a partir das informações da primeira (esquerda). Notação [Fabbri et al., 2022] para formular o problema (direita).

2.2. Funcionamento do *solver*

A homotopia continuada é a técnica que constitui o campo da geometria algébrica que consiste em rastrear o caminho de solução entre dois sistemas de equações algébricas [Hauenstein and Sommese, 2017]. Devido as` suas características de obtenção das raízes, sistemas polinomiais são particularmente melhores para o uso deste tipo de método pois garantem a convergência global, logo o caminho passa a ser da função do problema inicial a função do problema final. Este caminho é formado por um sistema de equações diferenciais em (4) artificialmente criado pela equação diferencial parcial de Davidenko na forma matricial (3). Sejam para a descrição do problema, dois sistemas polinomiais parametrizados sob funções do mesmo tipo F (z, s^0) = ($f_s{}^1{}_0$, . . . , $f_s{}^n{}_0$) e F (z, s^*) = ($f_s{}^1 *$, . . . , $f_s{}^n *$) que simbolizam os problemas inicial e final (alvo) respectivamente, a função de homotopia é apresentada conforme (5). De acordo com [Ventura et al., 2022], os parâmetros s podem ser conectados por um segmento de reta $\varphi(t) = s^0 \cdot (1 - t) + s^* \cdot t$, $t \in [0, 1]$, como consequência H(z, t) = F (s, φ(t)), t ∈ [0, 1] e também t pode ser parametrizado em C mantendo (5) válida por meio do chamado *gamma trick* que consiste em escrever t em termos de H(z, t) mas também que os caminhos gerados por (4) sejam suaves e não se intersectem uns com os outros:

$$Jac(H)_z \frac{dz}{dt} + \frac{\partial H}{\partial t} = 0 \quad (3)$$

$$\frac{\partial H}{\partial t} = -Jac(H)_z \frac{dz}{dt} \quad (4)$$

$$H(z,t) = f(z)(1-t) + g(z)t, t \in [0,1] \quad (5)$$

Em seguida é realizado o processo de monondromia no qual selecionam-se uma das variáveis, descrevem-se as demais em termo desta, unem-se todas as equações gerando uma única equação com 312 raízes complexas. Em seguida pegam-se estas raízes e alimentam-se as outras variáveis que são usadas para inicializar o sistema de equações diferenciais de (4). A figura 2 mostra o processo dos caminhos obtidos pela homotopia continuada para cada uma das quatorze variáveis do sistema (4) que consistuem o problema Chicago.

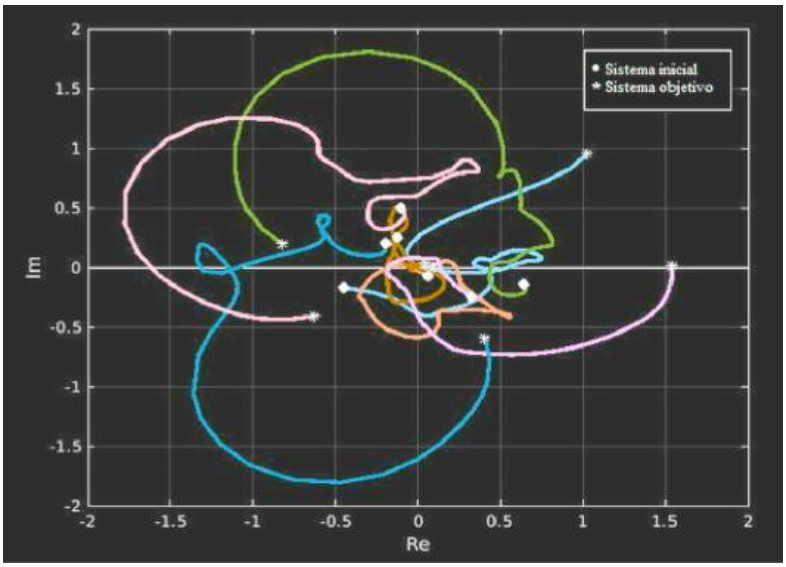

Figura 2. Representação esquemática do processo de homotopia continuada das quatorze variáveis que constituem o problema Chicago.

2.3 Estimação robusta

A estimação robusta do modelo é feita pelo algoritmo chamado de RANSAC [Internet, 2023a] que busca obter parâmetros de um modelo dado um limite de erro e um conjunto de dados. O algoritmo usa uma amostra do conjunto de dados fornecido, estima os parâmetros do modelo a partir do solver que descreve o problema, e verifica a quantidade de pontos com erro menor que o limite (inliers). No final do processo é retornado o modelo com a maior quantidade de pontos dentro deste erro. O RANSAC implementado internamente no software segue o modelo de [Hartley and Zisserman, 2004] apresentado pelo algoritimo 1.

O número N de iterações para a execução do RANSAC na prática não foi mencionada em [Fabbri et al., 2022]. A seguir, fornecemos o uma estimativa para o problema trifocal. De acordo com (6), o número N é tal que, com uma probabilidade p com pelo menos uma das amostras de três pontos é livre de outliers, onde ϵ é a máxima proporção de outliers que o algoritmo assume. Adaptando-se a equação clássica [Hartley and Zisserman, 2004],

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^3)} \quad (6)$$

Para p = 0.99 e ϵ = 50%, são necessárias apenas N = 35 iterações. Na prática, 100 iterações podem ser realizadas, fornecendo grande robustez de p = 0.999999 a ϵ = 50%.

Algoritmo 1: Algoritmo Ransac para obtenção das três câmeras

(i) Computar pontos de interesse em cada uma das três imagens
(ii) Computar correspondências entre as imagens 1 & 2 e 2
(iii) Repita para N amostras:
  (a) Selecionar uma amostra de três correspondências e compute o tensor trifocal (3 matrizes 3 × 4 representando o modelo de três câmeras). Existirão uma ou três soluções reais.
  (b) Calcular a distância entre um ponto de duas imagens reprojetada na terceira.
  (c) Computar o número de *inliers* consistente com o tensor trifocal pelo número de correspondências cuja a distância é menor que o limite de erro fornecido.
  (d) Se houverem três soluções reais para determinado tensor trifocal, computar o número de *inliers* e manter o modelo com a maior quantidade de *inliers. E*scolher o tensor trifocal com o maior número de *inliers*. Em caso de empate, escolher o que modelo com o menor desvio padrão entre *inliers*

(iv) Reestimar o tensor trifocal para todas as correspondências consideradas como *inliers*
(v) Determinar as correspondências restantes com o tensor trifocal obtido.

3. Implementação

Foi escrito um software chamado trifocal sample publicado como parte oficial do software OpenMVG, disponível em github.com/openMVG/openMVG, no ramo git develop keypoint orientation sfm. O software foi escrito C++ utilizando-se a suíte de configuração Cmake com compilador gcc 11, tendo sido desenvolvido em Linux Ubuntu 22.04 em uma máquina Intel® Xeon(R) CPU E5-2630 v4 @ 2.20GHz × 40 com 128Gb de memória RAM. O software também foi portado para Mac OS e compilador Clang/LLVM. O software consiste em bibliotecas incorporadas ao OpenMVG oficial, de um aplicativo principal, bem como de uma suíte de testes de unidade e integração. As bibliotecas foram escritas com o intuito de permitir a funcionalidade do aplicativo dentro de toda a pipeline completa de reconstrução 3D do OpenMVG, tendo sido aceita como parte oficial deste.

A implementação das bibliotecas consiste em um arquivo que utiliza os pontos de correspondência entre as imagens e executa o Ransac junto com o solver MiNuS e devolve o tensor trifocal, que consiste no conjunto das matrizes de rotação e translação do problema. A interface C++ entre o MiNuS e o OpenMVG foi escrita e testada, resultando em melhorias e depuração do MiNuS, bem como a inclusão deste como submódulo oficial do OpenMVG. Foi também escrita uma função erro e programado o Ransac para o problema trifocal utilizando-se a API do OpenMVG, bem como os testes correspondentes.

4. Resultados

Nesta seção são apresentados os resultados obtidos pela aplicação de três câmeras. São utilizados dois conjuntos de três imagens sendo uma do alto do Capitólio 3 com o intuito de verificar a robustez e de uma garrafa de *Absolut Vodka* 5 que verifica a estabilidade do modelo. Cada execução teve em média 500 microssegundos.

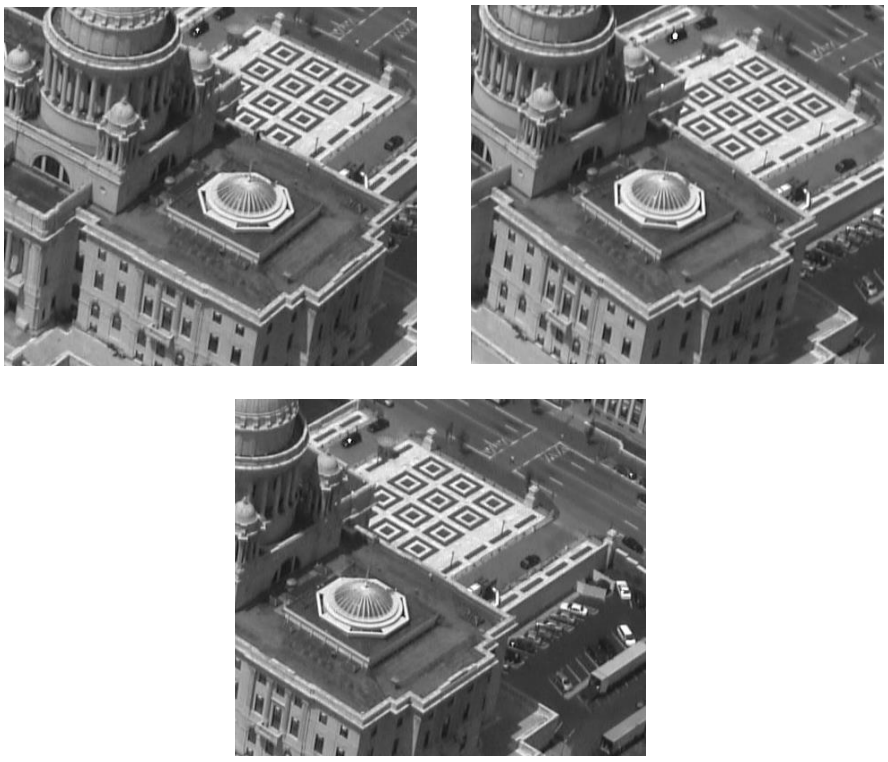

Figura 3. Imagens utilizadas para teste

O teste de robustez consiste em saber quantas correspondências são obtidas pelo melhor modelo de três

câmeras que o RANSAC do OpenMVG. Quanto mais pontos dentro dos dados fornecidos, melhor o modelo obtido. A figura 4 ilustra a robustez do modelo sendo os pontos representados pelas ligações em linhas azuis os pontos fora do melhor modelo e os pontos ligados pela cor ciano representam os que de fato são encontrados pelo mesmo. Das 199 correspondências, que representa 100% dos dados, 137 são explicadas pelo melhor modelo no pior dos casos, ou seja 68.84% dos dados são explicados pelo melhor modelo obtido pelo RANSAC a 100 iterações que demonstra robustez do modelo. Comparativamente, excluindo os pontos de ambiguidade, são 172 correspondências corretas entre as três câmeras, ou seja, foram extraídos aproximadamente 79.65% do total de correspondências corretas no pior dos casos. Com 1024 iterações, o modelo obteve entre 176 e 186 correspondências corretas no pior e melhor dos casos respectivamente que são as 172 não ambíguas mais 4 a 14 ambíguas. São consideradas correspondências aquelas nas quais os pontos tão próximos entre si de forma que a única forma de identificar a diferença entre eles é pela orientação do ponto e, no pior dos casos, não é capaz de diferenciar pela orientação.

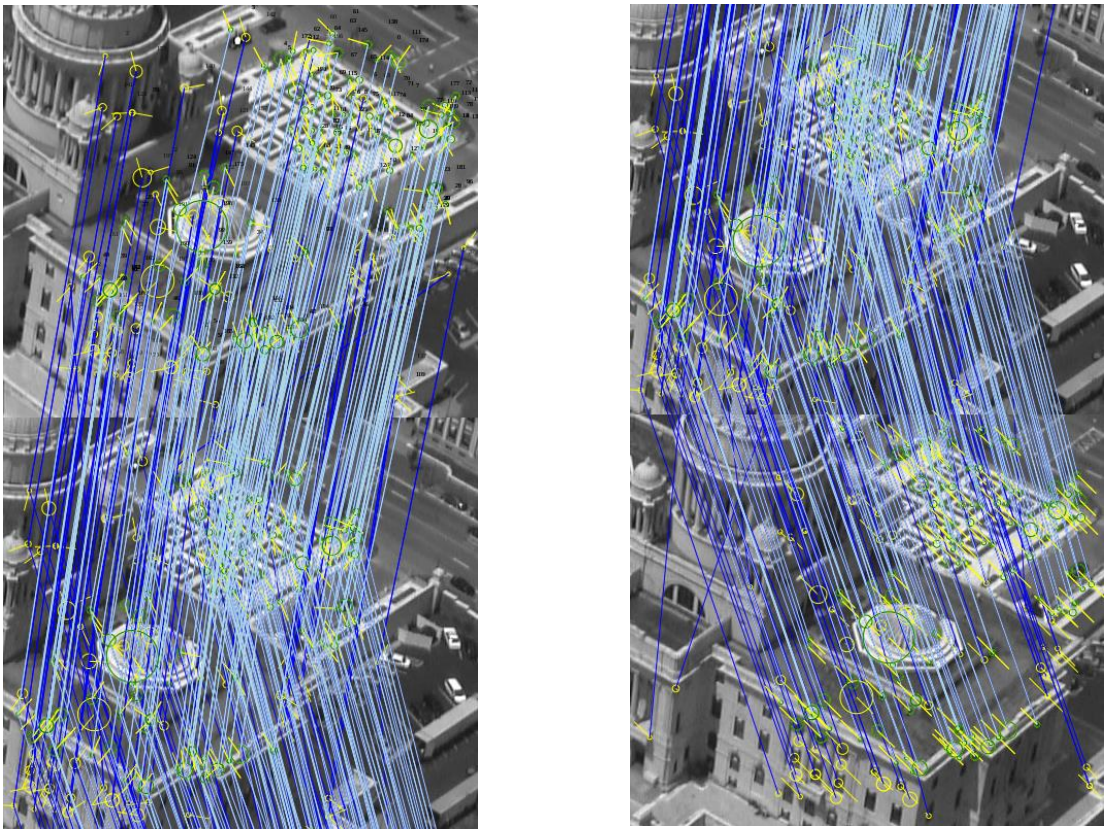

Figura 4. Correspondências entre as imagens. A primeira figura são as correspondências entre as imagens 1 e 2 são entre as imagens 2 e 3. As arestas em azul são os pontos que não explicam o modelo (outliers) enquanto as em ciano são os pontos que explicam o modelo (ou seja, que são classificados como corretas pelo modelo). Pode-se constatar que as correspondências em ciano são corretas, pois qualquer correspondência errada apareceria com uma aresta fora do padrão das demais.

A figura 5 é a garrafa de *Absolut Vodka* utilizada no experimento de estabilidade do modelo de três câmeras em situações onde a câmera não rotaciona e apenas translada, já que as equações envolvendo as tangentes (2) não restringem translação, e a configuração trifocal tradicionalmente resulta em maiores erros no caso de pontos não orientados [Faugeras and Luong, 2001]. Na prática a estabilidade é verificada quando, na ausência de rotação, é possível se estimar um modelo com o mínimo de correspondências consistentes. A figura 6 mostra que é possível obter esta quantidade, portanto o modelo é estável.

Figura 5. Imagens utilizadas para teste de estabilidade do trifocal

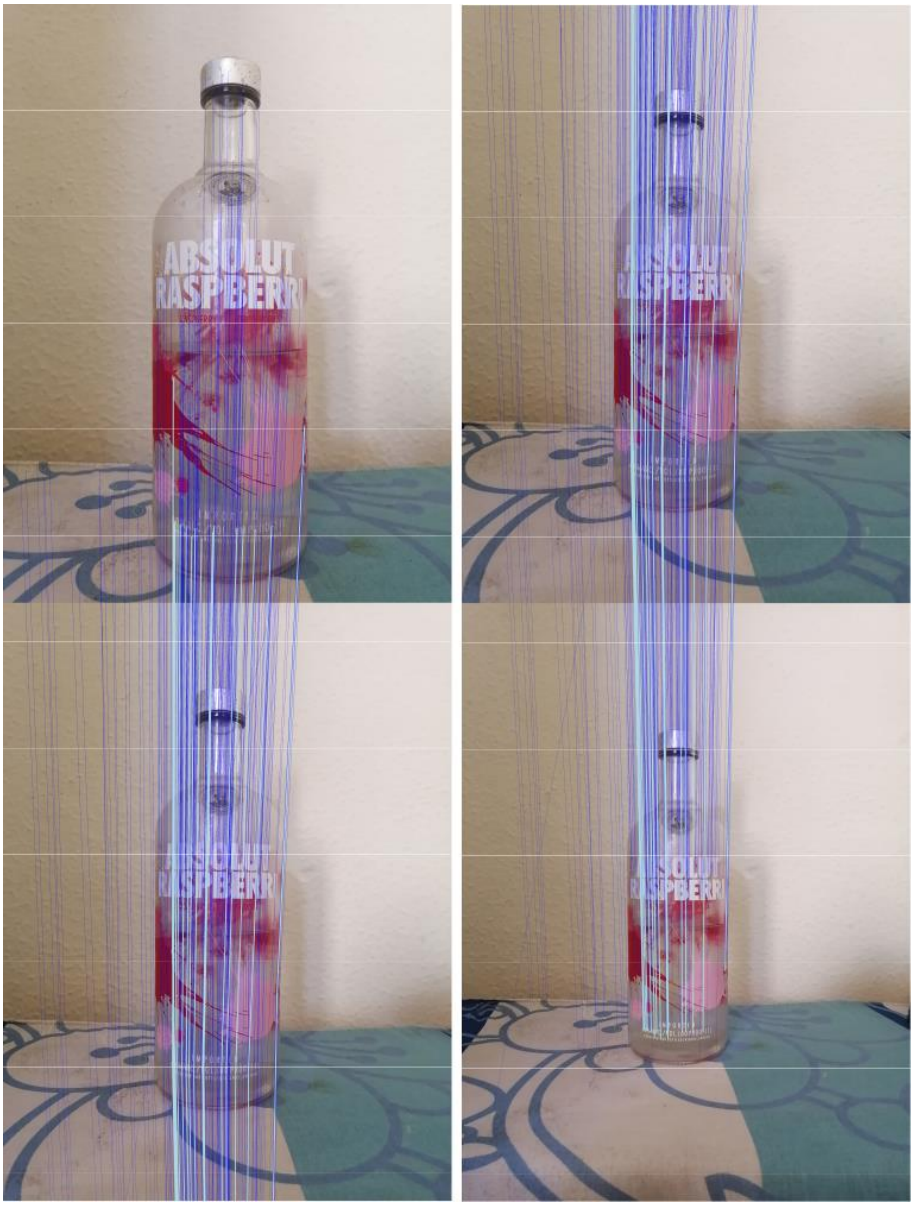

Figura 6. Correspondências do teste de estabilidade contendo apenas a translação entre as imagens. Verifica-se existe um número mínimo de correspondências corretas entre as imagens que demonstra que o modelo é estável.

5.Conclusão

Foi apresentada a primeira aplicação do modelo de três câmeras e três pontos orientados em sistemas de SfM amplamente utilizados pela comunidade. O software implementado foi aceito como parte integrante do sistema OpenMVG de reconstrução 3D a partir de múltiplas imagens. Apresentou-se o referencial teórico-matemático utilizado pela aplicação no contexto prático, juntamente com a metodologia e resultados experimentais ilustrando a robustez e estabilidade do modelo de três câmeras.

6. Agradecimentos

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001. G. A. Monerat e R. Fabbri agradecem a Universidade do Estado do Rio de Janeiro (UERJ) pela bolsa do Prociência. G. A. Monerat agradece a Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) pelo apoio financeiro parcial - Processo E-26/210.235/2022. G. C. Andrade agradece a CAPES pela bolsa de mestrado concedida.

7.Conflitos de Interesse:

Os autores do presente trabalho declaram que não possuem quaisquer conflitos de interesse em relação aos dados e experimentos do presente trabalho.

8. Referências

1. Apple ARKit Team. Understanding arkit tracking and detection. WWDC, 2018. URL https://developer.apple.com/videos/play/wwdc2018/610.
2. Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret H Regan, David da Costa de Pinho, Elias Tsigaridas, Charles W Wampler, Jonathan D Hauenstein, Peter J Giblin, Benjamin Kimia, and Tomas Pajdla. Camera pose estimation using first-order curve differential geometry. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–1, 2022.
3. Olivier Faugeras and Quang-Tuan Luong. The Geometry of Multiple Images. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262062208.
4. R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
5. Jonathan D. Hauenstein and Andrew J. Sommese. What is numerical algebraic geometry? Journal of Symbolic Computation, 79: 499–507, 2017. ISSN 0747-7171. doi: ttps://doi.org/10.1016/j.jsc.2016.07.015. URL: https://www.sciencedirect.com/science/article/pii/S0747717116300529. SI: Numerical Algebraic Geometry.
6. Internet. Ransac. https://en.wikipedia.org/wiki/Random_sample_consensus, August 2023a.
7. Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In International Workshop on Reproducible Research in Pattern Recognition, pages 60–74. Springer, 2016.
8. Johannes Lutz Schonberger¨ and Jan-Michael Frahm. Structure-from-motion revisited. In Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
9. Johannes Lutz Schonberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In European Conference on Computer Vision (ECCV), 2016.
10. Juliana Santos Barcellos Chagas Ventura, Ricardo FABBRI, and Francisco Duarte Moura NETO. Visual data science for the optimization of numerical trifocal geometry algorithms. Anais do Encontro Nacional de Modelagem Computacional, Encontro de Ciência e Tecnologia de Materiais, Conferência Sul em Modelagem Computacional e Seminário e Workshop em Engenharia Oceânica, pages 1–10, 2022.