Universidade Federal de Juiz de Fora

Relatório Técnico do Departamento de Ciência da Computação

RelaTeDCC 002/2023

Criação de imagem panorâmica utilizando algoritmos SIFT e RANSAC

Bruno Martins Bartolomeu

Prof. Dr. Luiz Maurílio da Silva Maciel

Agosto 2023

Trabalho Prático 01

.

1. Introdução
2. SIFT e RANSAC
3. Estratégias de implementação
4. Descrição dos experimentos
5. Conclusão
6. Referências

RESUMO

A visão computacional cresceu bastante nas últimas décadas, juntamente com o poder de processamento dos computadores atuais. Neste âmbito, vários sistemas de reconhecimento de objetos e algoritmos detectores de características foram propostos. Neste relatório, apresentamos um resumo sobre os resultados obtidos em um trabalho que tinha como objetivo a construção de uma imagem panorâmica.

O algoritmo SIFT foi utilizado para encontrar a correspondência de pontos-chave entre as imagens, enquanto o RANSAC foi utilizado para encontrar a matriz de homografia entre os pares correspondentes e reduzir a quantidade de falsas correspondências entre os pontos-chave das imagens. Uma descrição resumida de ambos os algoritmos, SIFT e RANSAC, é também fornecida neste artigo.

Palavras-Chave: Visão computacional. Detecção de características. SIFT. RANSAC. Homografia. Imagem panorâmica.

ABSTRACT

Computer vision has grown a lot in recent decades, along with the power of current computer processing. In this context, several recognition systems of objects and feature detection algorithms were proposed. In this report, we present a summary of the results obtained in a work that had as The objective is to build a panoramic image.

The SIFT algorithm was used to find the correspondence of key points between the images, while RANSAC was used to find the homography matrix between the matching pairs and reduce the amount of false matches between the key points of the images. A brief description of both algorithms, SIFT and RANSAC, is also provided in this article.

Keywords: Computer vision. Feature detection. SIFT. RANSAC. Homography. Panoramic image.

1 Introdução

A correspondência de imagens é fundamental em diversas situações da Visão Computacional como reconhecimento de cenas, montagem automática de mosaicos, obtenção da estrutura 3D de múltiplas imagens, correspondência estéreo etc.

Uma abordagem para se trabalhar com correspondência de imagens é se usar descritores locais para se representar uma imagem.

Descritores são vetores de características de uma imagem ou de determinadas regiões de uma imagem e podem ser usados para se comparar regiões em imagens diferentes e geralmente são destinos, robustos à oclusão e não requerem segmentação.

Existem diversas técnicas para se descrever regiões locais em uma imagem. O mais simples descritor é um vetor com as intensidades dos pixels da imagem. A medida de correlação cruzada pode ser então usada para computar a similaridade entre duas regiões. Porém, a alta dimensionalidade de tal descritor aumenta a complexidade computacional da comparação. Então, esta técnica é principalmente usada para se encontrar correspondências ponto a ponto entre duas imagens.

2 SIFT e RANSAC

SIFT (Scale-Invariant Feature Transform) é uma técnica poderosa para correspondência de imagens que pode identificar e combinar recursos em imagens que são invariantes para dimensionamento, rotação e distorção afim . É amplamente utilizado em aplicações de visão computacional, incluindo correspondência de imagens, reconhecimento de objetos e reconstrução 3D.

Descritores obtidos com a técnica **SIFT** são altamente distintos, ou seja, um determinado ponto pode ser corretamente encontrado com alta probabilidade em um banco de dados extenso com descritores para diversas imagens.

Um aspecto importante da técnica **SIFT** é a geração de um número grande de descritores que conseguem cobrir densamente uma imagem quanto a escalas e localização.

RANSAC (RANdom SAmple Consensus) é um algoritmo iterativo para a estimativa robusta de parâmetros a partir de um subconjunto de inliers do conjunto completo de dados.

Ele é geralmente usado em visão computacional, como por exemplo, para resolver simultaneamente o problema de correspondência entre pontos de duas imagens e estimar a matriz fundamental relacionada ao par de imagens estéreo.

Uma vantagem do **RANSAC** é a sua habilidade de realizar a estimativa de parâmetros de um modelo de forma robusta, ou seja, ele pode estimar parâmetros com um alto grau de acerto mesmo quando um número significativo de outliers esteja presente nos dados analisados. Uma desvantagem do algoritmo é que não há um limite superior de tempo para que ele possa computar tais parâmetros. Quando um limite superior é usado (número máximo de iterações) a solução obtida pode não ser a melhor existente.

3 Estratégias de implementação

Este trabalho foi totalmente implementando com a linguagem **Python** e contou com as seguintes bibliotecas: **Numpy, Opencv, Randon, Matplotlib, Tensorflow e Sys.**

O ambiente de desenvolvimento utilizado foi o **Google Colab.**

Segue abaixo passo a passo da implementação de forma resumida:

- **Detecção de extremos**
- **Localização de pontos chave**
- **Definição de orientação**
- **Descritor dos pontos chaves**
- **Implantação de homografia:** Estimadas entre imagens para detectar características correspondentes através do algoritmo RANSAC.
- **Imagem blending/Output:** Fase final do processo, colagem/costura e resultado com a imagem em formato panorâmico.

Detecção de Extremos

A primeira etapa da técnica SIFT é detectar extremos em uma pirâmide da imagem convoluída com a função Diferença de Gaussiana.

A convolução de uma função f(x,y) com uma função h(x,y) é dada por:

$$f(x,y) * h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n)$$

Um filtro Gaussiano passa baixa é dado pela convolução de uma imagem I com a função G:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$
$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

A função DoG ("Difference of Gaussian") é dada pela diferença de imagens filtradas em escalas próximas separadas por uma constante k. A função DoG é definida por:

$$\text{DoG} = G(x,y,k\sigma) - G(x,y,\sigma)$$

A convolução de uma imagem com o filtro DoG é dada por:

$$D(x,y,\sigma) = \left(G(x,y,k\sigma) - G(x,y,\sigma)\right) * I(x,y)$$
$$= L(x,y,k\sigma) - L(x,y,\sigma)$$

Este filtro consegue detectar variações de intensidade na imagem, tais como contornos por exemplo.

Localização Exata de Pontos Chave

Todos os pontos detectados como extremos são possíveis pontos chave.

Deseja-se agora calcular a localização e escala Gaussiana detalhadas de cada um destes pontos. Recalcular a localização e escala interpolada dos pontos de máximo traz melhoria para a técnica. A localização dos pontos chaves como será apresentada é especialmente importante para as últimas oitavas, porque o espaçamento amostral destas representa grandes distâncias na imagem base.

O método consiste em enquadrar uma função quadrática 3D do ponto de amostragem local de modo a determinar uma localização interpolada do máximo.

Para cada ponto analisado é utilizada uma expansão de Taylor da função D transladada de modo que a origem desta expansão esteja localizada no ponto:

$$D(\overline{x}) = D + \frac{\partial D^T}{\partial \overline{x}}\overline{x} + \frac{1}{2}\overline{x}^T \frac{\partial^2 D}{\partial \overline{x}^2}\overline{x}\ldots$$

Onde:

$$D = D(x, y, \sigma)$$

$$D(\overline{x}) = D(x + x', y + y', \sigma + \sigma')$$

Esta equação deve ser entendida da seguinte maneira, D e suas derivadas são avaliadas a partir do ponto analisado e $\overline{x} = (x', y', \sigma')^T$ é o offset em relação a este ponto.

Ou seja, D é o valor da função D(x,y,) no ponto avaliado, x é o offset em relação a este ponto e é a aproximação do valor de D(x,y,) interpolado para um ponto transladado com offset x.

Os coeficientes quadráticos são computados aproximando-se as derivadas através das diferenças entre pixels das imagens já filtradas.

A localização sub-pixel / sub-escala do ponto de interesse é dada pelo extremo da função apresentada na equação anterior. Esta localização é determinada ao se fazer a derivada segunda da equação anterior com relação à x e igualando o resultado à zero. Isto é feito como a seguir:

$$\frac{\partial D(\hat{x})}{\partial \overline{x}} = \frac{\partial D^T}{\partial \overline{x}} + \frac{\partial D}{\partial \overline{x}^2}\hat{x} = 0$$

Perceba que esta derivada usa a expansão de Taylor até $\frac{\partial D}{\partial \overline{x}}$. Tem-se então a posição do extremo dada por:

$$\hat{x} = -\frac{\partial^2 D^T - 1}{\partial \overline{x}^2}\frac{\partial D}{\partial \overline{x}}$$

O resultado é um sistema linear 3x3 que pode ser resolvido com custo mínimo. Caso seja maior que 0.5 em alguma dimensão. Isto significa que o extremo se aproxima mais de outro ponto. Neste caso, o ponto é re-alocado e a interpolação é realizada para este novo ponto. O offset final é adicionado à localização do ponto analisado para se chegar à interpolação estimada da localização do extremo.

A localização estimada deverá ser usada a partir de então nos procedimentos que seguirão.

O valor da função no extremo D é utilizado para se rejeitar extremos instáveis com baixo contraste. Substituindo-se obtemos:

$$D(\hat{x}) = D + \frac{1}{2}\frac{\delta D^T}{\delta \overline{x}}\hat{x}$$

É aconselhável por Lowe que se rejeitem valores de D inferiores a um determinado valor. É aconselhado trabalhar-se com o valor 0.03. (assumindo-se que os pixels da imagem estejam entre [0,1])

Alem do procedimento apresentado para se descartar pontos, Lowe ainda aponta que a função DoG possui resposta forte ao longo de arestas, mesmo que a localização ao longo da borda seja mal determinada. Isto faz com que estes pontos sejam instáveis para ruído em até pequenas quantias.

Atribuição da Orientação dos Descritores

Ao se atribuir uma orientação para cada ponto chave, podem-se representar os descritores em relação a esta orientação, conseguindo-se assim invariância quanto à rotação. O método utilizado para se atribuir esta orientação é apresentado como se segue.

A escala Gaussiana é utilizada para se escolher a imagem filtrada L, com a escala mais próxima, e de oitava referente ao ponto avaliado. Dessa maneira, todas os cálculos passam a ser feitos com invariância à escala.

Para cada ponto de cada imagem L(x,y,) intervalar, referente às escalas e oitavas utilizadas, são calculados os gradientes. Magnitude m(x,y) e orientação (x,y) são calculados como se segue:

$$m(x,y) = \sqrt{\begin{pmatrix} (L(x+1,y) - L(x-1,y))^2 \\ +(L(x,y+1) - L(x,y-1))^2 \end{pmatrix}}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{(L(x,y+1) - L(x,y-1))}{(L(x+1,y) - L(x-11,y))}\right)$$

Observe que não aparece nas equações. Isto foi feito para simplificar, pois o processamento é feito para cada imagem L. Somente as imagens correspondentes a intervalos precisam ser processadas.

Construção do Descritor Local

Até então, para cada oitava, foram escolhidos pontos chaves para localizações, escala e orientação definidos. A etapa atual consiste em computar o descritor que represente as regiões relativas aos pontos chaves. Os procedimentos a seguir são feitos normalizados em relação à orientação definida na seção anterior para cada ponto chave.

Para cada ponto chave, a construção do descritor é feita através dos seguintes passos:

- Escolhe-se a imagem filtrada L referente à escala e oitava relativas ao ponto chave.
- De modo a se conseguir invariância, as coordenadas dos pontos vizinhos ao descritor e das orientações dos gradientes destes pontos são giradas em relação ao ponto chave de acordo com a orientação definida na seção anterior.
- Uma função gaussiana é utilizada como peso para se ajustar as magnitudes de cada ponto na vizinhança do ponto chave. ' é escolhido igual à metade da largura da janela em que será calculado o descritor.
- São definidas n x n regiões, com k x k pixels cada, ao redor da localização do ponto chave. Geralmente n = k = 4.
- Para cada região, é feito um histograma h. para 8 direções, como na 4. Este histograma é feito com as magnitudes dos pixels pertencentes a cada região. O peso referente à magnitude de cada pixel foi atenuado pela função gaussiana como já ajustado. Percebe-se que a função gaussiana não é aplicada de modo idêntico ao na seção anterior. É utilizado um peso para interpolar a direção relativa no histograma. - O descritor é então representado pelos histogramas das regiões. A Figura 5 exemplifica como fica o descritor para 2 x 2 regiões (n = 2 e k = 4);
- O descritor é representado por um vetor, onde cada valor do vetor é referente a uma das direções de um dos histogramas. Para n e k iguais a 4, o vetor tem tamanho 128.
- Para que o descritor tenha invariância à iluminação, este é normalizado. Após a normalização, todos os valores acima de um determinado limiar são ajustados para este limiar. Isto é feito para que direções com magnitude muito grande não dominem a representação do descritor. Lowe sugere usar limiar 0.2. Por fim, o vetor é normalizado novamente.

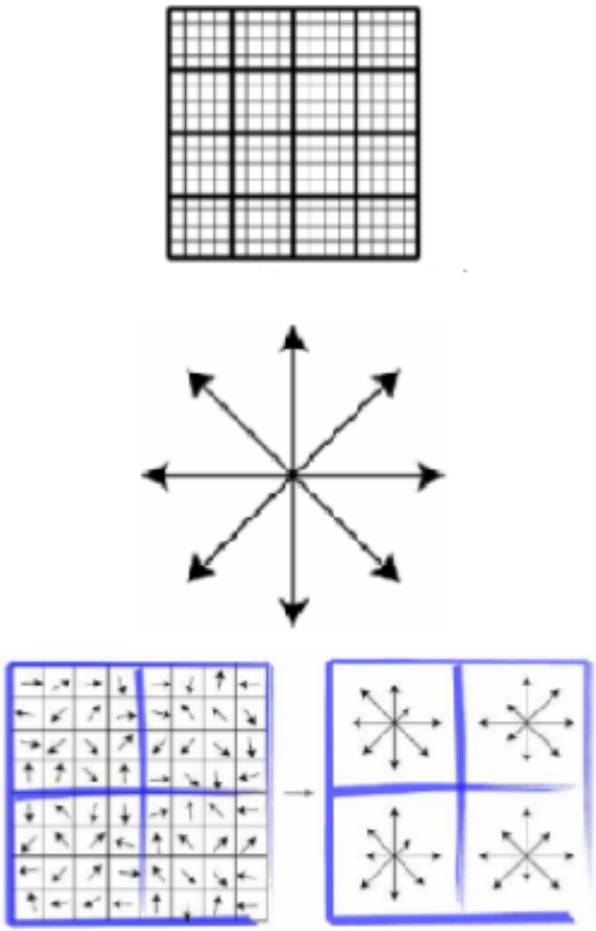

Figura 1. Imagem representativa da construção do Descritor.

Para cada imagem, são construídos diversos descritores, cada um referente a um ponto chave. Quando se aplica a técnica SIFT em uma imagem, tem-se como resultado, portanto, um conjunto de descritores. Estes descritores podem ser então, usados para se fazer a correspondência da imagem em outra imagem.

Encontrando os Pontos em Comum

Para se encontrar a correspondência entre duas imagens, devem-se encontrar pontos em comum entre as duas. Quando se trabalha com a técnica SIFT, pontos de interesse são detectados pelo método e representados em descritores. Tendo-se descritores de duas imagens, a tarefa de se encontrar a correspondência de uma imagem em outra é resumida por se encontrar entre os descritores de uma imagem, os melhores candidatos a serem seus equivalentes na outra imagem.

Portanto, dadas duas imagens "image1"e "image2", a tarefa de se encontrar a correspondência entre elas pode ser definida como se segue.

Os descritores são respectivamente definidos por di1 e dj2, onde i e j são aos índices para cada um dos descritores de cada imagem e k é o tamanho de cada descritor:

$$di_1 = (m_1 i_1, m_1 i_2, m_1 i_3 \cdots, m_1 i_k)$$
$$dj_2 = (m_2 j_1, m_2 j_2, m_2 j_3 \cdots, m_2 j_k)$$

A magnitude de cada valor dos vetores di1 e dj2 é dada por maib, onde a representa a qual imagem se refere o descritor, i é o índice do descritor e b é o índice de cada magnitude dentro do vetor.

A correspondência é feita achando-se os descritores dj2 que mais se assemelham aos descritores di1, encontrando-se as falsas equivalências e eliminando-as e por fim, encontrando-se a transformação de I1 para I2.

A tarefa de se encontrar o melhor candidato dj2 para determinado di1 é feita procurando-se o vizinho mais próximo ou “nearest neighbor” de di1 entre todos os possíveis candidatos, ou seja, para todo o índice j. Quando se procura classificar uma imagem em um extenso banco de dados de descritores para vários objetos, a busca exaustiva de vizinho mais próximo pode ser demorada e para tal existem diversas técnicas para se acelerar a busca. Porém, para o caso de se comparar duas imagens, a busca exaustiva não exige processamento pesado e, portanto, foi a escolhida.

O vizinho mais próximo de di1 para i dado é definido por dj2 que possua a menor distância euclideana em relação à di1. Ou seja, deseja se encontrar j que minimize a função:

$$|di_1 - dj_2| = \sqrt{\begin{matrix}(m_1 i_1 - m_2 j_1)^2 + (m_1 i_2 - m_2 j_2)^2 + \ldots \\ +(m_1 i_k - m_2 j_k)^2\end{matrix}}$$

Isto é feito para todo i de modo a serem encontrados todos os pares de descritores correspondentes. Perceba que muitos dos pares encontrados correspondem a falsas equivalências, portanto, as correspondências serão refinadas e falsos pares descartados.

Homografia/Output

Os métodos clássicos procuram utilizar o maior número de pontos para obter uma solução inicial e, então, eliminar os pontos inválidos. O RANSAC, ao contrário desses métodos, utiliza apenas o número mínimo e suficiente de pontos necessários para uma primeira estimativa, aumentando o conjunto com novos pontos consistentes sempre que possível.

Dado um modelo com parâmetros , deseja-se estimá-los. Para tal, é assumido:

- Os parâmetros podem ser estimados a partir de um número N de itens em um conjunto de dados conhecido;
- Existe um total de M itens no conjunto de dados;
- A probabilidade de um dado selecionado aleatoriamente fazer parte de um bom modelo é dada por pg;
- A probabilidade de que o algoritmo termine sem que se encontre um bom modelo é dada por p falha;

O algoritmo é então executado através das seguintes etapas:

- 1. N itens são escolhidos de modo aleatório;
- 2. A partir dos itens escolhidos, é estimado;
- 3. Encontra-se o número de itens que se enquadram ao modelo para determinada tolerância especificada. Este número é chamado de K;

- 4. Caso K seja grande o suficiente, para um limiar escolhido, o algoritmo termina e foi bem sucedido;
- 5. O algoritmo é repetido de 1 a 4 um número L de vezes;
- 6. Caso o algoritmo não tenha terminado após L tentativas, o algoritmo falhou;

L pode ser encontrado através das seguintes maneiras:

- pfalha = Probabilidade de L falhas consecutivas;
- pfalha = (Probabilidade de que determinado dado seja falho)*L;
- pfalha = (1 – Probabilidade de sucesso)*L;
- pfalha = (1 – (Probabilidade de que um item caiba no modelo)*N)*L;
- pfalha = (1 – (pg) *N) *L;

Para o problema específico de remoção de “outliers” na correspondência de imagens estéreo, a Matriz Fundamental (F) pode ser utilizada da seguinte forma:

- Selecionar randomicamente um subconjunto de oito pontos de correlacionados, retirados do conjunto total de pontos correlacionados (através de SIFT, por exemplo).
- Para cada subconjunto, indexado por j, calcular a matriz fundamental Fj através do algoritmo dos oito pontos.
- Para cada matriz Fj computada, determinar o número de pontos com distância até a linha epipolar, ou residual, menor que um limiar.
- Selecionar a matriz F que apresenta o maior número de pontos com residual inferior ao máximo definido.
- Recalcular a matriz F considerando todos os pontos “inliers”.

$$\tilde{m}^T F \tilde{m}' = 0$$

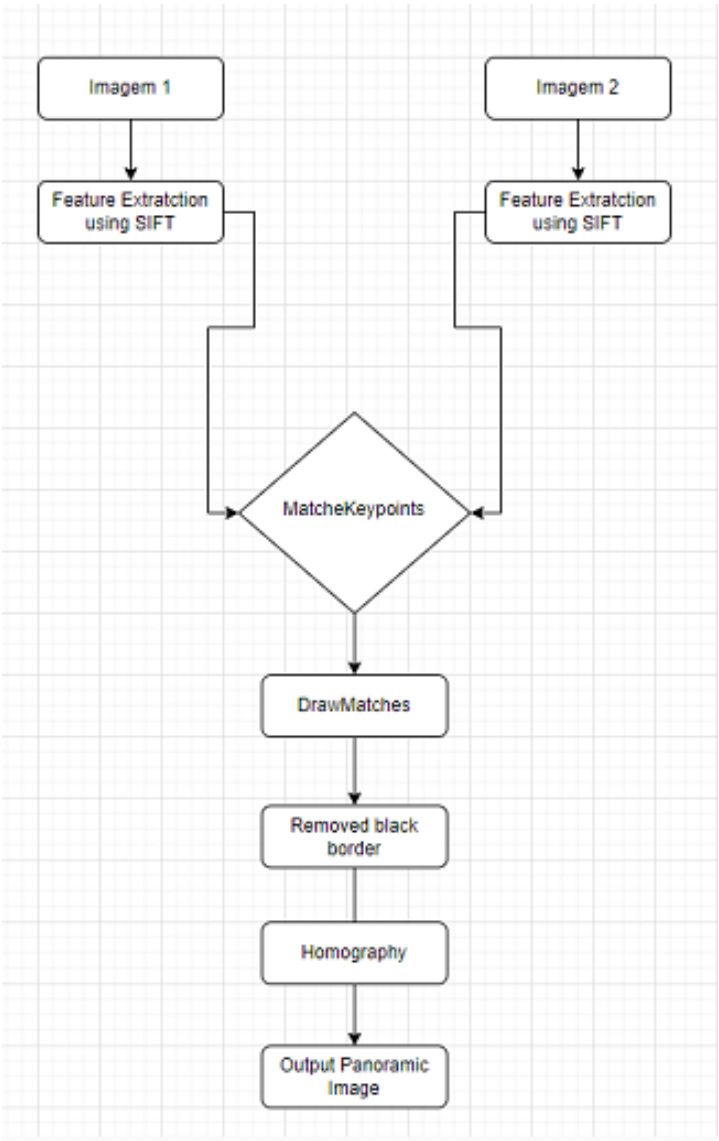

Figura 2. Imagem representativa do fluxo do processo.

4 Descrição dos experimentos

Objetivo: Criar uma imagem panorâmica através de análise de um conjunto de keypoins entre duas imagens utilizando algoritmos já conhecidos da área.

- Detecção de características – Utilização do algoritmo SIFT para a extração dos descritores de características da imagem.
- Correspondência – Utilização do RANSAC para descarte dos “outliers”, ou seja, dos falsos “matches”.

Características da imagem que teve sucesso no processo:

- Left img size (397 * 529)
- Right img size (397 * 529)
- Right img levemente com borda escura

Quantidade de imagens testadas e de onde foram obtidas:

- Foram testados 2 pares de imagens e obtidas diretamente do Google
- Uma imagem com total sucesso.(cerca de 22 segundos para performar)

Resultados obtidos de pontos correspondentes:

- 106

Resultados obtidos de inliers/outliers ao final do RANSAC:

- 97

Observações:

- Podemos observar que quanto maior for a imagem mais tempo o algoritmo leva para performar e montar a panorâmica.
- A 2º imagem apresentou o erro justamente na parte da classe "blender"que faz a colagem das imagens e gera a panorâmica. (cerca de 22 segundos para performar também)

.

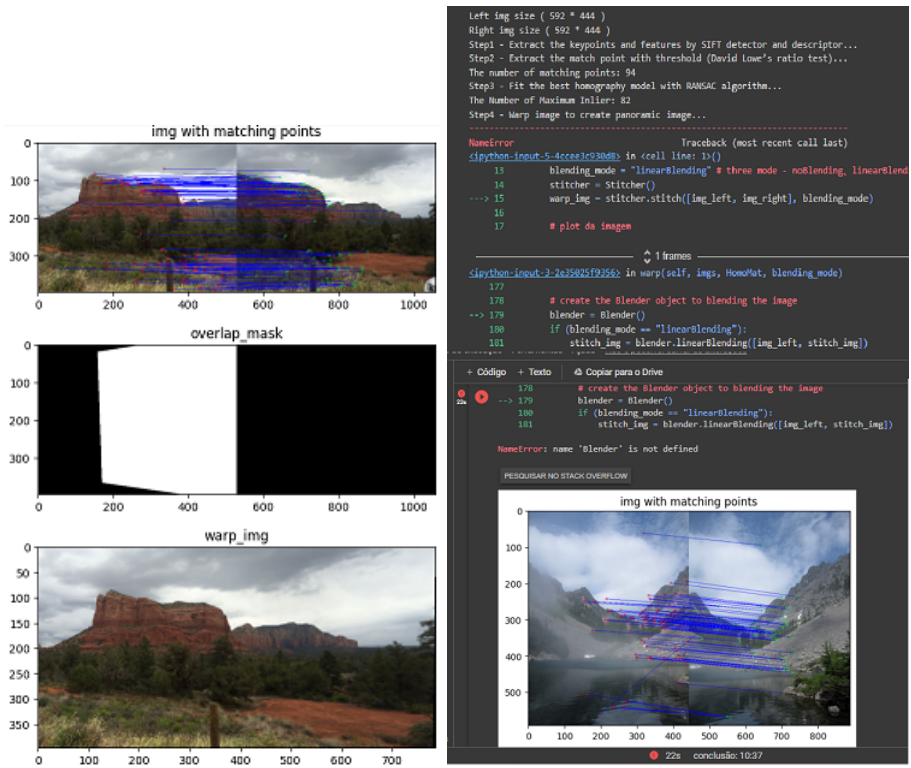

Figura 3. Resultados positivo e negativo respectivamente.

5 Conclusão

De acordo com os resultados obtidos, verifica-se a possibilidade destas ferramentas funcionarem de maneira satisfatória, com geração de resultados confiáveis e de maneira automática, auxiliando com rapidez no próposito.

A ideia também pode ser modificada para atender outras necessidades, necessitando para isso somente adaptações para a nova aplicação em questão, baseada na análise das características que se deseja analisar.

Link para repositório: `https://github.com/Bmartins25/Projeto01_UFJF_SIFT_RANSAC`

6 Referências

[1] - Feature Matching and RANSAC - https://people.cs.umass.edu/~elm/Teaching/ppt/370/370_10_RANSAC.pptx.pdf

[2] - Comparação de Algoritmos para Detecção de Pontos de Interesse - https://www.lume.ufrgs.br/bitstream/handle/10183/236205/001137526.pdf?sequence=1

[3] - Computer Vision: Algorithms and Applications, 2nd ed - https://szeliski.org/Book/?authuser=0

[4] - Mosaico de imagens com OpenCV e python - https://dev.to/tassi/mosaico-de-im

[5] - Homography Estimation - https://cseweb.ucsd.edu/classes/wi07/cse252a/homography_estimation/homography_estimation.pdf

[6] - RANSAC Panorama - https://github.com/melanie-t/ransac-panorama-stitc

[7] - Panoramic image stitching - https://github.com/joyeecheung/panoramic-image