

Universidade Tecnológica Federal do Paraná – UTFPR
 Departamento Acadêmico de Eletrônica – DAELN
 Disciplina: IF69D – Processamento Digital de Imagens
 Semestre: 2022.1 (Remoto)

RELATÓRIO

Detecção de objetos em conjunto de imagens local

Alunos:

Lucas Matheus dos Santos / 2028492

mês.ano

1. Objetivo

Implementar em matlab um algoritmo de detecção de objetos em um conjunto de imagens a partir de uma imagem de um objeto.

O algoritmo deve ter como saída uma pasta com todas as imagens que contenham o objeto especificado na entrada

2. Fundamentação Teórica

Para a identificação de objetos na imagem foram utilizados pontos de recurso obtidos com o algoritmo SIFT comparando os pontos correspondentes entre uma imagem de entrada que possui apenas o objeto e uma pasta contendo um conjunto de imagens que podem possuir o objeto buscado entre as imagens.

2.1. SIFT

O algoritmo de Sift é usado para detecção e descrição de objetos usando características locais que são invariáveis a rotação e iluminação. O algoritmo SIFT extrai os pontos de interesse da imagem e gera um conjunto de vetores que descreve numericamente as propriedades da região em volta desses pontos. Esses vetores podem ser usados posteriormente para fazer as correspondências de características[2].

O primeiro passo é a obtenção de pontos de interesse usando diferentes escalas de filtro gaussiano em oitavas, uma com a metade do tamanho da outra.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

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

Onde L é a convolução da imagem com o filtro gaussiano e σ é a escala. Após a convolução de todas as imagens é realizada a diferença do gaussiano[1].

$$L(x, y, k\sigma) - L(x, y, \sigma)$$

E $k\sigma$ é a variação de escala das imagens. Após isso é realizada uma pirâmide com os níveis de escala da diferença de gaussianos e cada ponto é comparado com seus 8 vizinhos e também seus 9 vizinhos superiores e inferiores [1].

Após isso é realizado a filtragem de pontos de baixo contraste e bordas que continuam na filtragem após a diferença de gaussianas.

Para eliminar pontos de baixo contraste é utilizado a expansão de taylor

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

Onde se obtém uma localização mais precisa dos extremos, e se a intensidade desses extremos for menor que 0.03 ele é rejeitado. Após o que é removido os vértices baseados em uma matriz 2x2 hessiana para rejeição de bordas[3].

Para geração do vetor de recurso após obter os pontos de interesse é necessário realizar a atribuição de orientação de escala aos pontos com maior magnitude, para isso é calculado a direção de cada ponto de interesse e seus vizinhos e gerado uma matriz de direções 16x16 onde é realizada a subtração da rotação do ponto chave de cada ponto vizinho assim cada orientação do gradiente é em relação ao ponto chave tornando o objeto invariável a rotação.[3]

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

$$\theta(x, y) = \tan^{-1} ((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

Onde m é a magnitude e θ é a direção, levando em consideração os pontos da imagem L . Para a obtenção das imagens de imagens com o objeto correspondente são comparados os pontos gerados nos vetores de recurso com suas vizinhanças.[3]

3. Implementação

Para a implementação foi utilizado o matlab com as bibliotecas de visão computacional para a obtenção de recursos e comparação entre as imagens o algoritmo recebe uma imagem de entrada e o caminho da pasta de imagens de comparação após isso é obtido os pontos de interesse da imagem com o método `detectSIFTFeatures` que recebe a imagem em tons de cinza e retorna o vetor de recursos da imagem, após o que , é usado a função `extractFeatures` para extrair os pontos de recursos obtidos com o sift e comparar os pontos obtidos em ambas as imagens usando `matchFeatures` para localizar os pontos extraídos de ambas as imagens é utilizado o `estimateGeometricTransform` para remover pontos discrepantes utilizando o parâmetro 'affine' que permite transformações inversas identificando os pontos rotacionados, após obter as imagens que possuam os objetos o algoritmo deve salvar as imagens contendo os objetos na saída.

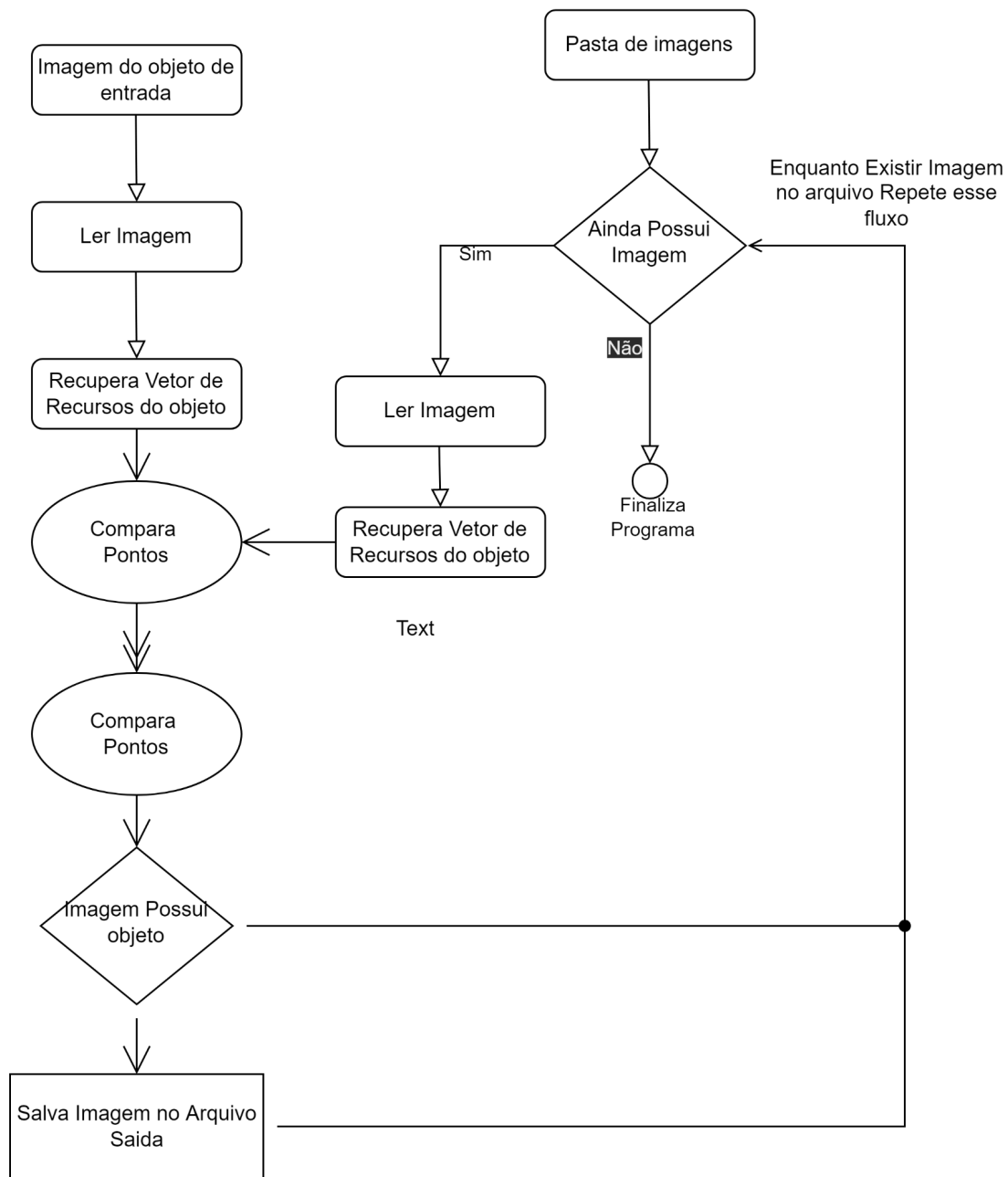


Diagrama.1 Diagrama do algoritmo para reconhecimento de objeto em imagens

4. Resultados e conclusões

O algoritmo conseguiu identificar todas as imagens que continham o objeto mesmo em meio a outros objetos e com diferentes ângulos e níveis de iluminação, mas não teve uma quantidade de pontos geométricos muito grande em algumas correspondências o que ao gerar a imagem de saída não conseguiu colocar borda corretamente nos objetos das imagens achadas.

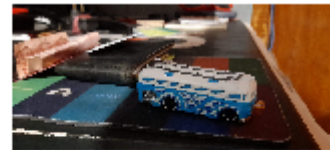


Figura.1 Imagens na pasta de entrada



Figura.2 Imagens de saída salvas no arquivo de saída.

Objeto a ser Buscado



Figura.3 Imagem com o objeto a ser procurado

Após analisar o algoritmo implementado pode ser observado que uma melhoria possível e receber mais de uma imagem do objeto de entrada a ser procurado com diferentes ângulos que possam ser feita uma comparação com cada umas das imagens no banco de busca para verificar a melhor correspondência assim tendo mais confiabilidade e robustez nos resultados e conseguindo delimitar melhor o objeto e a área do mesmo.

Referências

- [1] Bessa, Lucas Aragão., **Reconstrução esparsa em 3d a partir de múltiplas imagens**. Universidade de Brasília. 2015 .
- [2] Lowe, D. G. (1999). Object recognition from local scale-invariant features. Proceedings of the Seventh IEEE International Conference on Computer Vision, 2([8), 1150–1157. <http://doi.org/10.1109/ICCV.1999.790410>.
- [3] Tyagi , Deepanshu . Introduction to SIFT(Scale Invariant Feature Transform).Medium . 2019. disponível em : <https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>