

Projeto 3

Tema: Implementação do cálculo de esqueleto de um objeto em uma imagem binária.

Alunos:

Leonardo Donderi Rodrigues, RA: 754756

Luís Felipe Corrêa Ortolan, RA: 759375

Marco Antonio Bernardi Grivol, RA: 758619

Reynold Navarro Mazo, RA: 756188

Neste trabalho foi implementado o cálculo de esqueleto de um objeto em uma imagem binária. Essa técnica é aplicada para obter uma cadeia de pixels de largura de tamanho um e que representa a topologia do objeto original. Ela é uma forma eficiente de descrever formas, já que possui menos informações que a imagem original, mas é suficiente para alguns tipos de análise.

Para se construir o esqueleto de um objeto o algoritmo deve passar por todos os pixels da imagem observando sua vizinhança. Para padronização das informações, o pixel e seus vizinhos são mapeados da seguinte forma:

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

São necessários dois passos para cada pixel para ser construído o esqueleto. Esses passos removem, tornam o valor do pixel 0, caso eles se enquadrem em algumas condições. Ambos os passos são aplicados para cada pixel até que não haja mais nenhum pixel que atenda aos critérios de ambos os passos.

O primeiro passo impõe quatro condições para remover aquele pixel:

- A soma dos vizinhos de p_1 estar entre 2 e 6.
- Na sequência ($p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_2$) existir exatamente uma transição 0 - 1.
- A multiplicação de p_2, p_4 e p_6 ser igual a zero.

d) A multiplicação de p_4 , p_6 e p_8 ser igual a zero.

Já o segundo passo, consiste em verificar as condições a) e b) do passo um e verificar se ambos os produtos p_2 , p_4 e p_8 e p_2 , p_6 e p_8 são iguais a zero. Esse algoritmo vai fazendo sucessivas erosões na imagem evitando as pontas do objeto, a quebra do objeto em mais de um componente e a remoção de um ponto com largura de tamanho um.

Em nossa implementação deste algoritmo, foram implementadas as seguintes funções para o cálculo do esqueleto:

- `get_neighbours`: É uma função que a partir de uma imagem, um inteiro que indica a coluna e outro que indica a linha, retorna uma lista com os oito vizinhos daquele ponto e o próprio p_1 . Caso seja um ponto de borda, os valores faltantes são retornados como se fossem zero.
- `N`: Esta função recebe a lista de vizinhos provida pela função anterior e devolve a soma dos vizinhos de um dado pixel.
- `T`: Essa função recebe os vizinhos de um ponto e passa pela sequência de pontos (p_2 , p_3 , p_4 , p_5 , p_6 , p_7 , p_8 , p_9 , p_2) contando o número de transições 0-1. No fim, retorna o número de transições.
- `passo1`: Essa função recebe a imagem e as coordenadas de um pixel e retorna verdadeiro se todos os critérios listados no passo um do algoritmo forem satisfeitos, se não, retorna falso.
- `passo2`: Essa função recebe a imagem e as coordenadas de um pixel e retorna verdadeiro se todos os critérios listados no passo dois do algoritmo forem satisfeitos, se não, retorna falso.
- `esqueleto`: Essa função recebe a imagem original, passa por cada um dos pixels da imagem verificando os passos um e dois do algoritmo do esqueleto, caso um deles seja verdadeiro, o pixel é removido da imagem. Além disso, ele também verifica se algum pixel foi removido para repetir os passos caso isso tenha acontecido.

Abaixo, será mostrado o resultado da aplicação do algoritmo implementado em algumas imagens, além de uma imagem para comparação do resultado de uma aplicação de esqueleto em uma biblioteca já pronta, a `skimage`.

