Vetorização de Imagens
Projeto Final EA979

João Pedro Bizzi Velho RA:218711
Lucas Morandi RA: 202031
Rodrigo Santana RA:205565

Universidade Estadual de Campinas - Unicamp
FEEC

3 de agosto de 2020

Agenda

- Motivação - O que é Vetorização?
- Objetivo do Projeto
- Bibliotecas e técnicas utilizadas
- Imagens utilizadas
- Funcionamento
- Vídeo da execução de alguns exemplos
- Limitações
- Conclusão

O que é Vetorização?

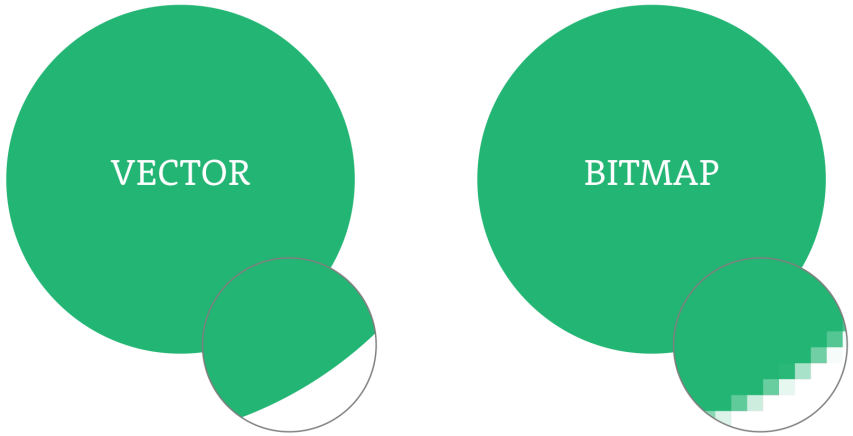


Figura: Imagem Vetorizada vs Rasterizada

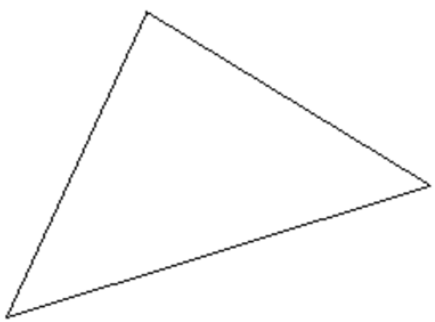
Imagens em vetor são compostas de pontos, linhas e curvas representadas por expressões matemáticas, o que permite que ser ampliadas infinitamente sem perder a qualidade. São comumente utilizadas por softwares de ilustração, impressão, arquivos PDF e muitas outras aplicações, visto sua facilidade em manipulação e tamanho compacto.

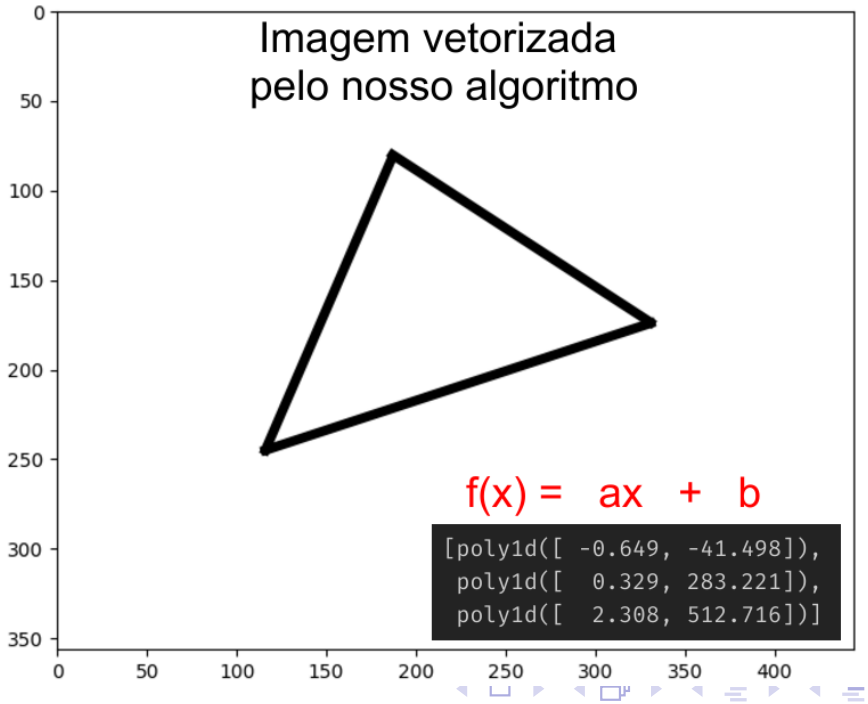
Objetivos do Projeto

Proposta

Desenvolver um software que teria como função vetorizar imagens simples como quadrados e triângulos. Utilizando algum método de segmentação apresentado na disciplina, para em seguida realizar a separação de segmentos que compôem a figura, por fim utilizando os pontos obtidos para ajusta-los à uma função matemática.

Imagem Original





Métodos e bibliotecas

As principais bibliotecas utilizadas no projeto foram

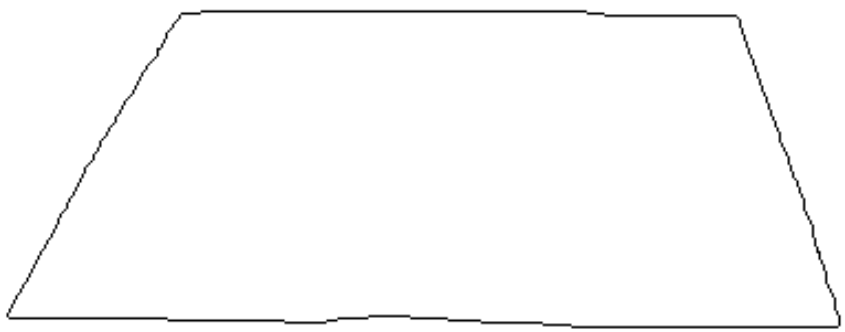
- Numpy
- Matplotlib
- PIL
- time
- Skimage

A única técnica utilizada foi a técnica de segmentação apresentada durantes as aulas da disciplina, **Watershed**.

Imagens Utilizadas

Dataset

O conjunto de imagens utilizado é composto de formas geométricas simples, regulares ou não, como quadrados, triângulos, trapézios etc.



Funcionamento

Após realizarmos segmentação por watershed na imagem, iniciamos o processo de separação em diferentes segmentos:

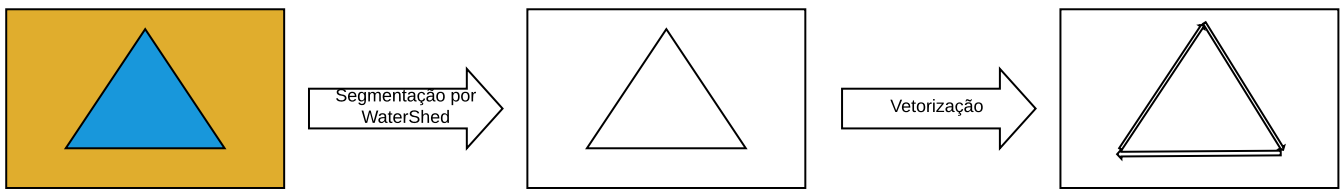


Figura: Processo Geral

Funcionamento

Vizinhança do pixel

Dividimos a vizinhança do pixel em 4 quadrantes diferentes, cada um com 3 pixels:

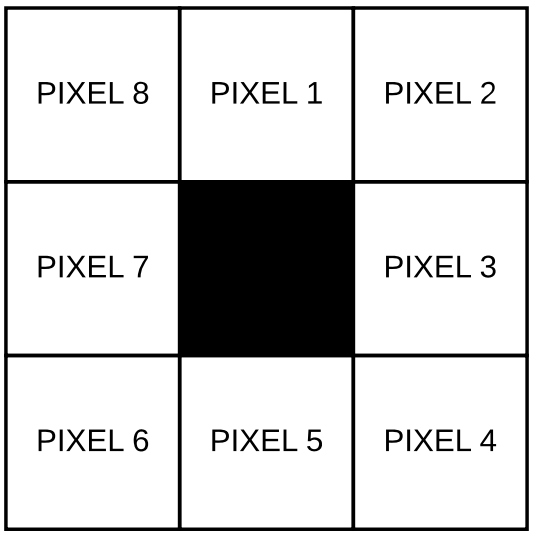
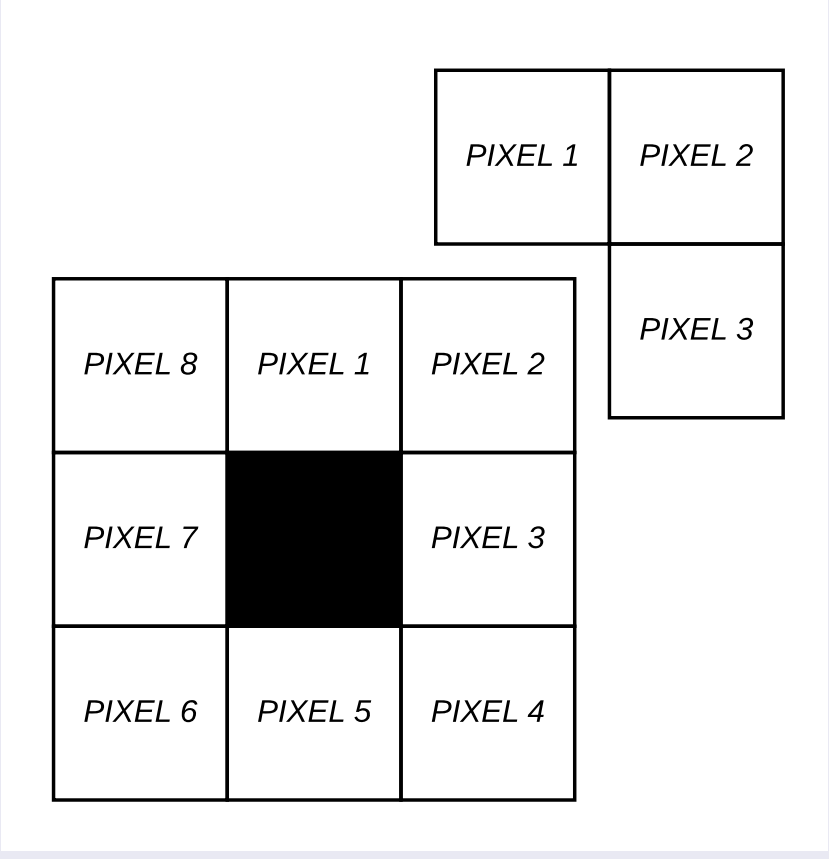


Figura: Vizinhança do pixel

Funcionamento

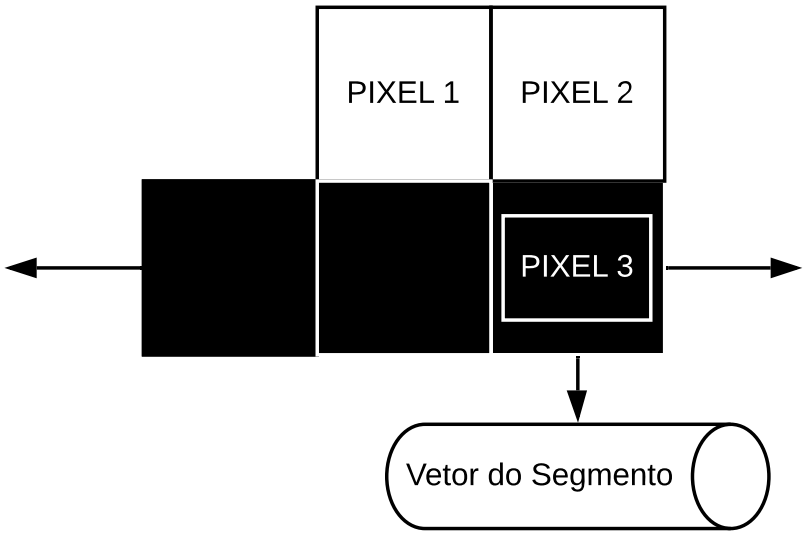
Vizinhança do pixel

Dividimos a vizinhança do pixel em 4 quadrantes diferentes, cada um com 3 pixels:

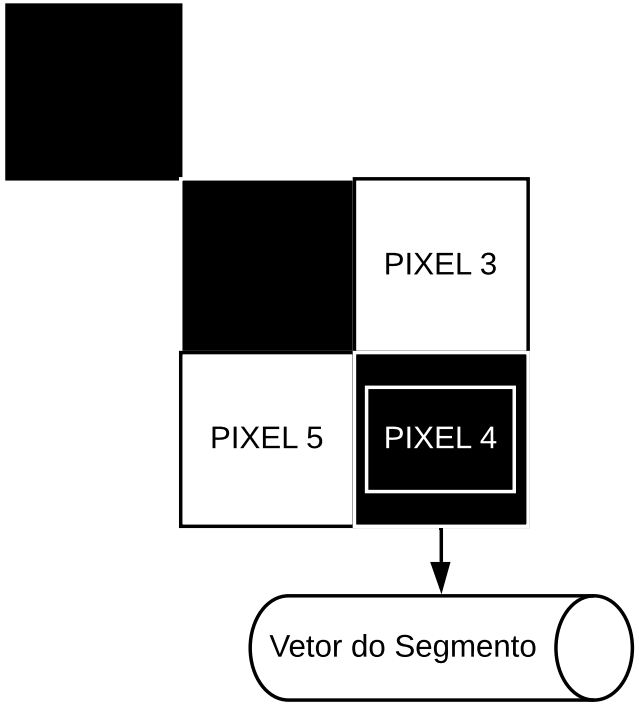


Funcionamento

Percorrendo um quadrante por vez no sentido horário



(a) Sequência Horizontal



(b) Sequência diagonal

Funcionamento

Quando o programa não encontra mais nada naquele segmento, significa que o segmento acabou:

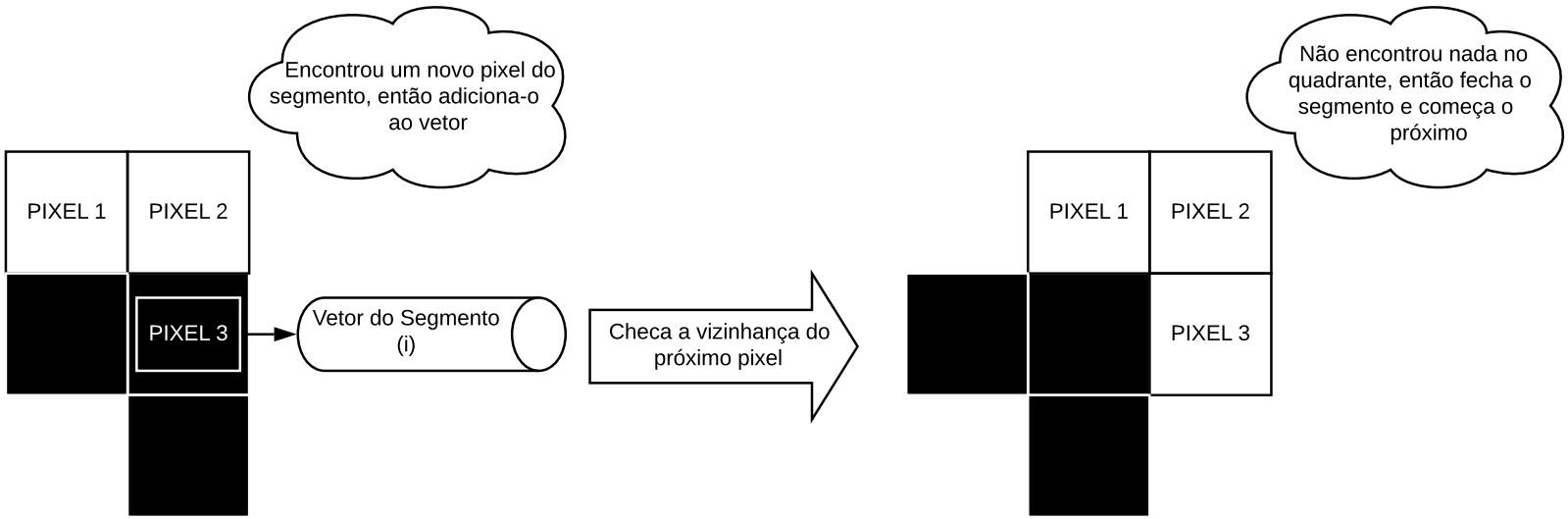


Figura: Finalização de um segmento

Funcionamento

Quando o programa não encontra mais nada naquele segmento, significa que o segmento acabou:

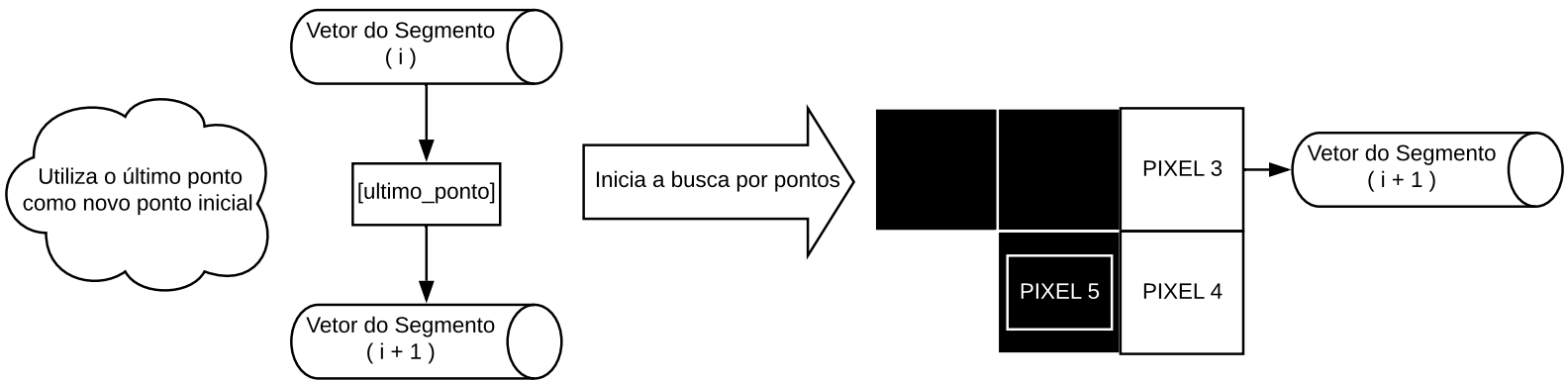
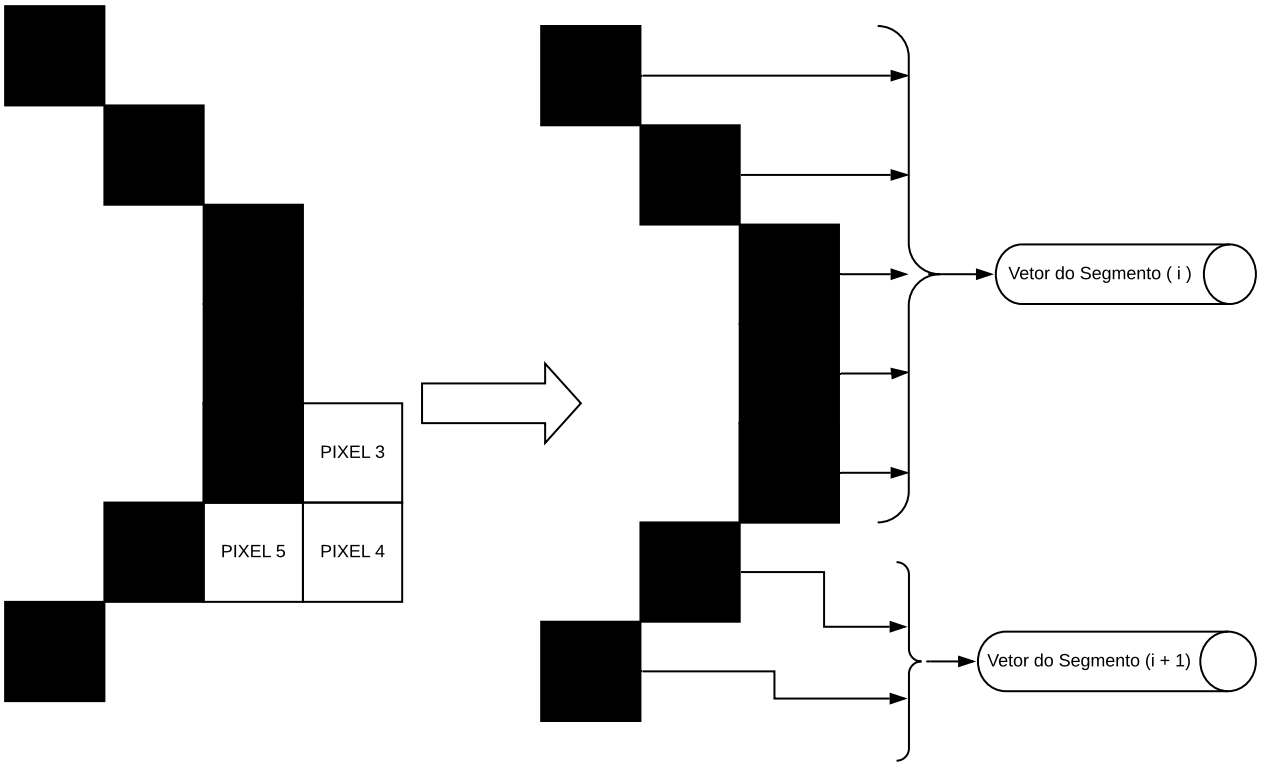


Figura: Início de um novo segmento

Execução

Vídeo do algoritmo sendo executado

Limitações



O algoritmo proposto não é capaz de detectar curvas, resultando em um único segmento, ou em dois segmentos diferentes.

Conclusão

- Mesmo com suas limitações o algoritmo é capaz de vetorizar com sucesso várias formas geométricas, estando rotacionadas ou não.
- Com um pouco mais de desenvolvimento, poderia-se implementar a lógica de detecção de curvas.