

Instituto Federal de Educação, Ciência e Tecnologia do Ceará

Engenharia de Computação

Visão Computacional

Prof. Pedro Pedrosa

Relatório Limiarização e Watershed

Gabriela de Carvalho Barros Bezerra

Fortaleza

Abril de 2018

Sumário

1	Introdução	2
2	Desenvolvimento	2
3	Resumo Geral	4

1 Introdução

Este relatório tem como objetivo principal falar sobre a implementação e utilização do openCV em python para o estudo do algoritmos de segmentação Limiarização (Threshold) e Watershed.

2 Desenvolvimento

Watershed

Qualquer imagem em preto e branco pode ser vista como uma superfície topográfica onde altas intensidades são considerados "picos" e baixas intensidades são considerados "vales". O algoritmo consiste em preencher cada vale isolado com "águas coloridas" de cores diferentes (rótulos), e enquanto a água sobe, dependendo dos picos ao redor (gradientes) as águas de diferentes cores vindas de diferentes vales vão começar a se encontrar. Para evitar isso, é necessário construir "barreiras" nas localizações onde as águas se misturam. Esse processo continua até todas as águas terem se encontrado, então as barreiras criadas darão o resultado da segmentação.

Essa estratégia pura e simples causa uma segmentação exagerada das imagens. A implementação do algoritmo na OpenCV inclui a utilização de marcadores, onde são especificados quais são todos os vales que se encontram e quais não. É uma segmentação interativa. Os rótulos para esse estudo serão 3, um para a região de background, outro para a região que com certeza é o foreground ou objeto de interesse, e outro para a região que não sabemos

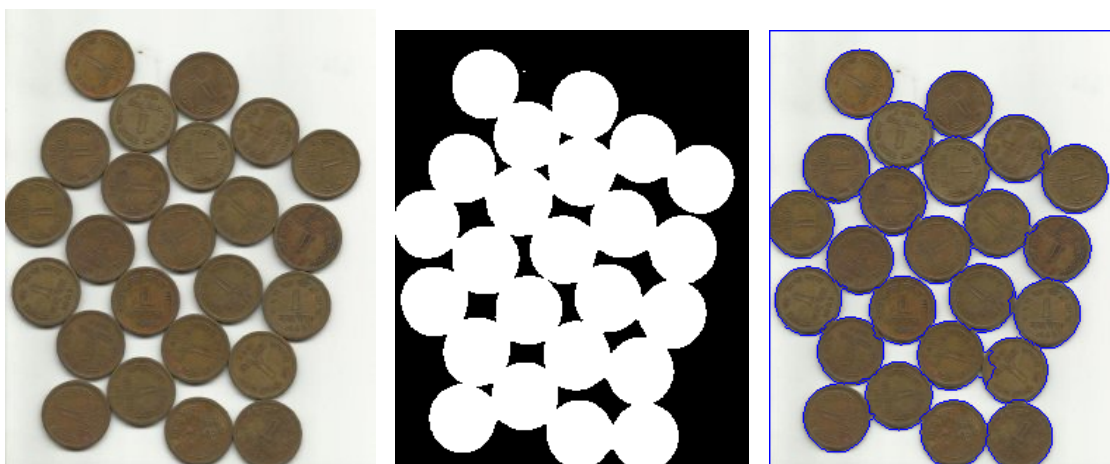
com certeza se é um objeto a ser segmentado (com valor 0). As barreiras construídas serão as bordas dos objetos segmentados, e terão rótulo com valor -1.

Limiarização

O algoritmo de limiarização consiste em analisar uma imagem, e separar nela o que for background (plano de fundo) do que for foreground, ou objetos de interesse daquela análise. É uma dos algoritmos mais simples de segmentação que existem. Geralmente, o resultado é uma imagem binária. Através de uma análise do histograma, é atribuído 0 (cor preta) aos pixels que estão em background, e 1 (cor branca) aos pixels que estão em foreground.

Estudos utilizando python e OpenCV

Comecei meus estudos fazendo uma limiarização da imagem original (a), com redução de ruídos. Dessa forma ficou mais fácil diferenciar os objetos que serão segmentados, no caso serão moedas. Esse foi o resultado (b):



(a) Imagem original

(b) Após limiarização

(c) Após Watershed

Após ser feita a limiarizacao, utilizando o openCV foram extraídos o background da imagem com a função dilated, o foreground com distanceTransform, e a região "desconhecida" utilizando o subtract para encontrar a região entre o foreground e o background. A região desconhecida tem como objetivo marcar limites mais exatos dos objetos que serão segmentados, juntamente com a região de contato entre eles.

Consegui encontrar as regiões de foreground que estão conectadas, e atribuir "markers" para elas, de forma que todo background terá valor "1" e a região desconhecida terá valor "0".

O watershed foi aplicada à imagem original (a) a partir dos marcadores, e assim obtive o seguinte resultado (c):

Cada moeda está segmentada de acordo com os marcadores da região desconhecida, e com as áreas de contato encontradas entre elas.

3 Resumo Geral

Neste trabalho foi utilizada a linguagem python e a biblioteca openCV para entendimento e utilização dos algoritmos de Threshold e Watershed. Consegui entender como ambos os algoritmos funcionam, visualizando a maneira que os dados extraídos de uma imagem são vistos, tratados e segmentados a partir desses algoritmos dentro do OpenCV. Código implementado neste trabalho está disponível no Github através deste link.

Referências

Watershed em python utilizando OpenCV..