

Relatório Segmentação por Área

Daniella Martins Vasconcellos¹ e Miguel Alfredo Nunes¹

¹Departamento de Ciência da Computação – Universidade Estadual de Santa Catarina (UDESC)

daniella.vasconcellos@edu.udesc.br, miguel.nunes@edu.udesc.br

Resumo. Trabalho feito para a disciplina de Processamento de Imagens (PIM0001), ministrada pelo Professor Doutor Gilmário Barbosa dos Santos. O trabalho consistia em realizar a aplicação da segmentação por área para a resolução do problema de contagem de moedas por uma foto.

1. Introdução

O enunciado proposto pelo professor foi o seguinte:

Utilize uma imagem capturada por você mesmo contendo moedas de R\$ 1,00 e R\$ 0,10 em quantidades variadas e sem sobreposição. Com base na rotulação e segmentação por área, construa uma solução que contabilize o total de reais nas moedas capturadas na imagem.

Tendo isso em mente, o seguinte trabalho se propõe a aplicar o algoritmo segmentação por área para a resolução do problema.

Este trabalho se divide nas seguintes sessões: na seção 2, será descrito a metodologia do código. A seção 3 mostrará os resultados observados. Por fim, a seção 4 tratará das conclusões do trabalho.

2. Metodologia de Desenvolvimento

O trabalho foi desenvolvido na linguagem *Python 3.8.10*, utilizando da biblioteca *Python Imaging Library (PIL)* para a manipulação de imagens.

2.1. Escolhas de desenvolvimento

O código foi escrito no seguinte arquivo: **main.py**. Ele possui as seguintes funções principais: *aplica_limiar*, *encontra_moedas*, *popula_imagem*, *BFS*, e *main*.

A função *main* carrega a imagem e inicia sua matriz de pixels. Determina-se o limiar de valores que irão ser detectados no RGB de cada pixel da matriz. Com a função *aplica_limiar*, é aplicado o limiar determinado anteriormente. Com a matriz carregada e o limiar aplicado, a função *encontra_moedas* percorre a matriz com a função *BFS* (a qual é a aplicação do algoritmo de breadth-first search) e adiciona os objetos encontrados a uma lista chamada “objetos”. Seu resultado é guardado na variável *moedasEncontradas*.

Para cada moeda encontrada na lista *moedasEncontradas*, guarda-se o tamanho de cada objeto em outro vetor chamado *tamanhos*, o qual é ordenado do maior para o menor. Então encontra-se a maior moeda (o maior número da lista de tamanhos) e guarda esse valor na variável *maiorMoeda*.

Moedas de 1 real são maiores que de 10 centavos. Logo, para filtrar todas as moedas de 1 real, deve selecionar todas as moedas que tem por volta do mesmo tamanho

(utilizou-se 10% de erro). Foi considerado que toda moeda que não é de 1 real é de 10 centavos, já que foram as únicas opções pedidas pelo enunciado. Cada moeda de 1 real detectada é inserida na lista *moedas1Real* e as de 10 centavos na lista *moedas10Centavos*. Os tamanhos dos vetores são somados na variável *valor*, a qual é apresentada ao usuário com a resposta final do problema.

3. Análise dos resultados obtidos

O algoritmo descrito acima foi aplicado com a seguinte imagem:



Figure 1. Imagem utilizada para o código (fonte: os autores)

A maior moeda que foi encontrada pelo algoritmo (e guardada na variável *maior-Moeda*) foi a seguinte:



Figure 2. Moeda de 1 real

A partir da moeda acima, o algoritmo conseguiu diferenciar os tamanhos de cada moeda, comparando e acrescentando nos seus respectivos vetores com sucesso. O resultado mostrado pelo algoritmo mostrou a correta resposta de R\$3,20.

4. Conclusões

A conclusão a ser tirada é que, apesar da lógica aplicada funcionar bem e mostrar o resultado correto, é fato que o que auxiliou o bom resultado e encontro dos objetos foi o fundo branco que resultou em bom contraste das moedas. Não foi testado a lógica com outras imagens com fundos diferentes.

Agradecimentos

Agradecimentos ao Professor Doutor Gilmário Barbosa dos Santos pela disponibilização dos materiais utilizados como base para o estudo feito nesse trabalho.