

MC102 - Algoritmos e Programação de Computadores

Turmas QRSTWY

Instituto de Computação - Unicamp

Professores: Hélio Pedrini e Zanoni Dias

Monitores: Andre Rodrigues Oliveira, Gustavo Rodrigues Galvão, Javier Alvaro Vargas Muñoz e Thierry Pinheiro Moreira

Lab 13b - Quadtree

Prazo de entrega: 15/06/2015 às 13h59m59s

Peso: 9

Em visão computacional, segmentação se refere ao processo de dividir uma imagem digital em múltiplas regiões (conjunto de pixels) ou objetos, com o objetivo de simplificar ou alterar a representação de uma imagem para facilitar a sua análise. Segmentação de imagens é tipicamente utilizada para localizar objetos e formas (tais como linhas e contornos) em imagens.

O resultado da segmentação de imagens é um conjunto de regiões/objetos ou um conjunto de contornos extraídos da imagem. Assim, cada um dos pixels em uma mesma região é similar com referência a alguma característica ou atributo, tais como intensidade, cor, textura ou continuidade. Regiões adjacentes devem possuir diferenças significativas com respeito à(s) mesma(s) característica(s).



Imagem original



Imagem segmentada

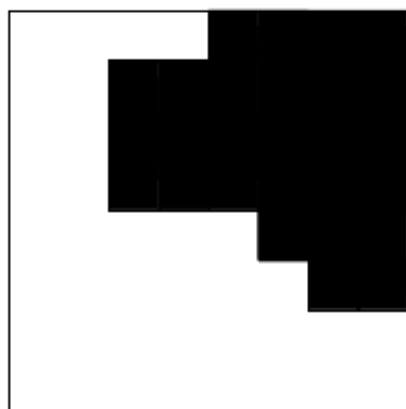
Dentre as abordagens existentes para segmentação de imagens, pode-se identificar uma família de técnicas

baseadas na *divisão de regiões*. O princípio geral destas técnicas consiste em iniciar a segmentação com regiões maiores da imagem e, recursivamente, subdividir as regiões não homogêneas em áreas menores. O processo de subdivisão termina quando todas as regiões satisfizerem o critério de homogeneidade.

Uma técnica comum de subdivisão da imagem em regiões homogêneas utiliza a representação *quadtree*, que é uma estrutura hierárquica baseada na decomposição recursiva e regular da imagem em quadrantes. Considerando apenas imagens preto-e-branco, a geração da *quadtree* segue as seguintes regras:

- No início, tem-se apenas uma região composta pela imagem inteira.
- Uma região, se não for homogênea, é dividida em quadrantes. Ou seja, a região é dividida em quatro subregiões, onde as dimensões (largura e altura) dessas subregiões são, aproximadamente, a metade da região original.
- Uma região é homogênea se for totalmente preta ou branca.
- Quando uma região é dividida em quadrantes, os quadrantes resultantes têm o mesmo tamanho e representam 4 novas regiões da imagem.
- A geração da *quadtree* termina quando todas as regiões da imagem são homogêneas.

Para ilustrar esse conceito, seja a imagem (em preto-e-branco) mostrada na figura (a), a qual é representada por uma matriz binária (0 = branco, 1 = preto) de $2^3 \times 2^3$ pixels, mostrada na figura (b). As regiões homogêneas resultantes da decomposição da *quadtree* são mostradas na figura (c).



(a)

0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	0	0	0	1	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

1	2	3	8			
	4	5				
6	7					
9			10	11	14	
			12	13		
			15		16	

(c)

A tarefa deste laboratório é desenvolver um programa que, dada uma imagem quadrada em preto-e-branco, determinar o número de regiões formadas pela *quadtree*.

Entrada

- A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 1024$) que representa as dimensões (largura e altura) da imagem, isto é, a imagem a ser processada terá um tamanho de $N \times N$ pixels. Por simplicidade, N será uma potência de 2.
- As N linhas seguintes contém N dígitos cada, separados por espaços, de tal forma que o j -ésimo dígito localizado na i -ésima linha representa a cor do pixel na posição (i,j) , conforme a seguinte convenção:
 - 0: a cor do pixel na posição (i,j) é branca.
 - 1: a cor do pixel na posição (i,j) é preta.

Saída

- Seu programa deve imprimir uma linha no formato "Numero de regioes = x", em que x é o número de regiões que serão formadas pela *quadtree*.

Exemplos

#	Entrada	Saída
1	8 0 0 0 0 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Numero de regioes = 16
2	4 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 1	Numero de regioes = 4
3	16 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Numero de regioes = 124
4	8 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0	Numero de regioes = 64
5	8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Numero de regioes = 1