



Processamento digital de imagens

Segmentação

Region growing

Split and merge

SLIC



Segmentação



Segmentar = dividir em partes

Divisão da imagem em regiões, ou partes, compostas por pixels que de alguma maneira apresentam uma característica comum.

- O resultado ideal de uma segmentação seria, por exemplo, a demarcação dos limites espaciais dos objetos visíveis ou suas partes componentes.

Segmentação



- O grau de fragmentação da imagem em segmentos depende da aplicação pretendida.
- Geralmente, a segmentação é um passo preliminar no processo de reconhecimento de padrões, pois permite simplificar o problema isolando partes da imagem para uma segunda etapa de análise.

Métodos



- geralmente baseados em análise de descontinuidade ou similaridade de valores digitais.
- descontinuidade : a abordagem é dividir a imagem baseando-se em mudanças bruscas dos valores digitais.
- Similaridade: pixels adjacentes com propriedades similares são agrupados para formar regiões uniformes. Exemplo: crescimento de regiões e divisão e fusão de regiões.

Regras da Segmentação

- Sendo R a imagem completa, a segmentação e o processo de dividir R em n regiões R_1, R_2, \dots, R_n , tal que:
 - a) $U(R_i)=R$
 - b) R_i e uma região conexa;
 - c) $R_i \cap R_j = \emptyset$ (vazio) para todo i e j, $i \neq j$;
 - d) $P(R_i) = \text{VERDADEIRO}$ para $i = 1, 2, \dots, n$;
 - e) $P(R_i \cap R_j) = \text{FALSO}$ para $i \neq j$,
- $P(R_i)$ e o predicado lógico sobre os pontos do conjunto R_i e \emptyset e o conjunto vazio.
 - (a) indica que a segmentação deve ser completa;
 - (b) fixa a existência de regiões conexas.
 - (c) indica que as regiões devem ser disjuntas.
 - d) todos os pixels de uma mesma região tem as mesmas características
 - (e) indica que regiões diferentes são diferentes no sentido do predicado P.
 - VENTURIERI e SANTOS (1998)

Crescimento de regiões



Dado um pixel qualquer, buscar todos os vizinhos que sejam “similares” para formar a região.

Crescimento de regiões

1. Para iniciar o crescimento de regiões são necessários pixels semente.
2. A cada iteração, os pixels vizinhos à região são analisados. Se forem similares à região, então são anexados.
3. Para anexar novos pixels na fronteira da região é necessário definir um critério de similaridade. Por exemplo, distância Euclidiana entre o pixel semente e o pixel novo.

Considere a seguinte matriz (poderia ser uma pequena imagem)

99	87	78	83	98	83	87
99	87	78	83	98	83	87
67	99	9	89	71	89	99
69	87	10	83	77	83	87
92	13	11	12	98	92	13
89	87	13	83	78	83	87
09	77	78	73	98	73	77
99	87	10	83	77	83	87

A princípio não conhecemos os grupos espacialmente contínuos e uniformes, então vamos escolher um pixel aleatoriamente.

Por exemplo: linha 3 , coluna 3.

Este pixel tem valor $v(3,3)=9$ e será considerado a “semente de uma região”. Então, o valor típico da região é 9.

A seguir, analisamos a vizinhança deste pixel (3,3). Neste exemplo consideraremos a vizinhança 8, ou seja, usando uma janela 3x3.

99	87	78	83	98	83	87
99	87	78	83	98	83	87
67	99	9	89	71	89	99
69	87	10	83	77	83	87
92	13	11	12	98	92	13
89	87	13	83	78	83	87
09	77	78	73	98	73	77
99	87	10	83	77	83	87

Devemos definir o grau de “similaridade” aceitável.

$$v(i, j) - 9 \leq 20$$

Consideremos que se a diferença entre o valor do pixel vizinho e o valor da semente for menor que 20, o pixel pertence à região:

99	87	78	83	98	83	87
99	87	78	83	98	83	87
67	99	9	89	71	89	99
69	87	10	83	77	83	87
92	13	11	12	98	92	13
89	87	13	83	78	83	87
09	77	78	73	98	73	77
99	87	10	83	77	83	87

Analisando os 8 vizinhos, constata-se que um vizinho pode ser aceito, aquele localizado na mesma coluna e na linha abaixo:


Como foi incluído um vizinho, a região cresceu, logo suas fronteiras mudaram. Devemos repetir a análise, com os novos vizinhos.

Mantendo o critério $v(i, j) - 9 \leq 20$

99	87	78	83	98	83	87
99	87	78	83	98	83	87
67	99	9	89	71	89	99
69	87	10	83	77	83	87
92	13	11	12	98	92	13
89	87	13	83	78	83	87
09	77	78	73	98	73	77
99	87	10	83	77	83	87

Achamos novos vizinhos parecidos. A região cresceu, repetir a análise.

99	87	78	83	98	83	87
99	87	78	83	98	83	87
67	99	9	89	71	89	99
69	87	10	83	77	83	87
92	13	11	12	98	92	13
89	87	13	83	78	83	87
09	77	78	73	98	73	77
99	87	10	83	77	83	87



99	87	78	83	98	83	87
99	87	78	83	98	83	87
67	99	9	89	71	89	99
69	87	10	83	77	83	87
92	13	11	12	98	92	13
89	87	13	83	78	83	87
09	77	78	73	98	73	77
99	87	10	83	77	83	87

Nenhum vizinho novo com valor similar a 9 foi encontrado. Então, a região parou de crescer,

Neste ponto, todos os pixels localizados são “similares” e formam uma região contínua no espaço.

SEGMENTAÇÃO



Split-and-merge

Dividir e depois juntar

- Neste caso se parte da imagem como uma grande região única.
1. Em cada iteração e se avalia se a(s) região(ões) é uniforme. Caso não seja uniforme, divide-se a região em 4 quadrantes e se repete a análise efetuando a divisão progressiva da imagem em quadrados cada vez menores.
 2. Como a divisão é progressiva, sua evolução pode ser representada por uma árvore quaternária (quadtree)
 3. Importante: definir critério de uniformidade.

Quadtree (árvore quaternária)

“Uniformidade”: Região é uniforme se $|\text{min-max}| < 20$

1	2	3	4	3	6	2	2
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2
1	2	8	2	3	91	90	1
1	2	2	1	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	1	91	82	92	93

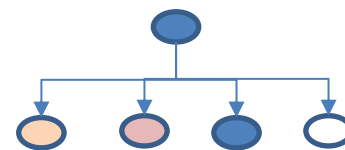
Considerando a imagem toda como uma região.

A região é uniforme?

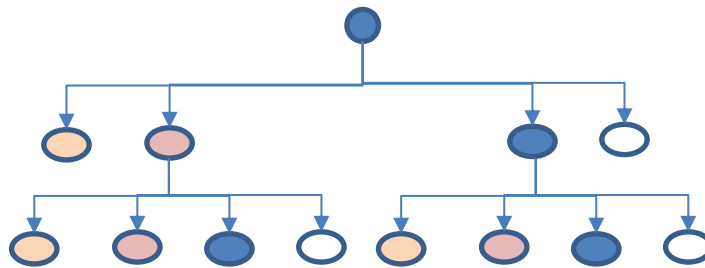
Não. Min=1: max=91. dif=90.

- Como a região não é uniforme, dividimos ela em quatro partes.
- Agora repetimos a análise considerando estas novas regiões.
- As 4 regiões são uniformes?

1	2	3	4	3	6	2	2
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2
1	2	8	2	3	91	90	1
1	2	2	1	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	1	91	82	92	93



1	2	3	4	3	6	2	c 2
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2
1	2	8	2	3	91	90	1
1	2	2	1	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	1	91	82	92	93



d- Todas são uniformes

1	2	3	4	3	6	2	2
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2
1	2	8	2	3	91	90	1
1	2	2	1	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	1	91	82	92	93

Não é necessário dividir mais as regiões. Chegamos a obter somente regiões uniformes.

Porém:

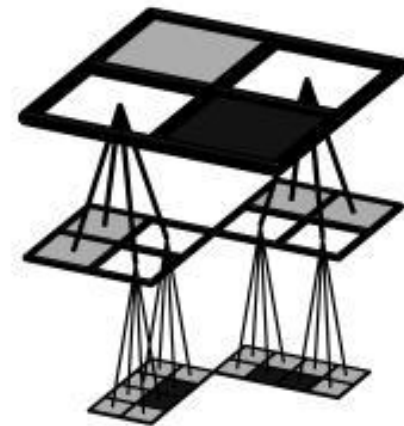
d) $P(R_i) = \text{VERDADEIRO}$ para $i = 1, 2, \dots, n$; OK!

e) $P(R_i \cup R_j) = \text{FALSO}$ para $i \neq j$, FALSO!

Existem regiões adjacentes que são similares.

e- Juntar similares/adjacentes

1	2	3	4	3	6	2	2
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2
1	2	8	2	3	91	90	1
1	2	2	1	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	1	91	82	92	93

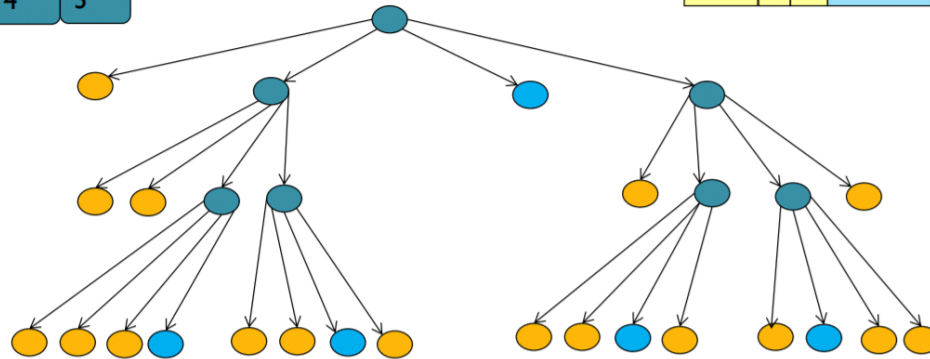
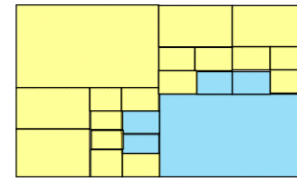


<http://graphics.cs.niu.edu/projects/regview/index.html>

1	2	3	4	3	6	2	2
1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2
1	2	8	2	3	91	90	1
1	2	2	1	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	91	91	82	92	93
1	2	2	1	91	82	92	93

quadtrees

- não uniforme
- uniforme (A)
- uniforme (B)



Algoritmo SLIC - Superpixels

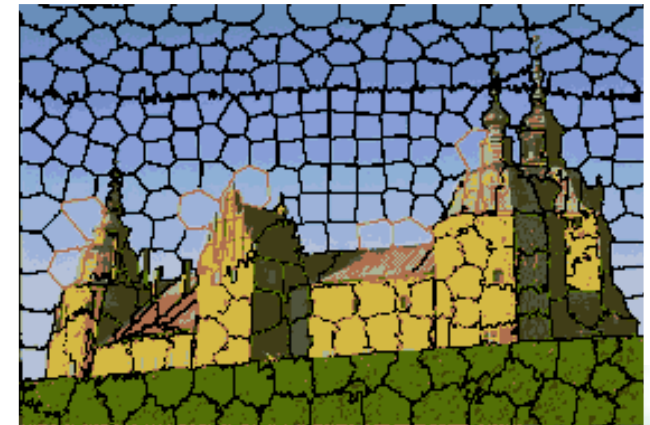
Simple Linear Iterative Clustering

El algoritmo produz superpixels agrupando pixels próximos com base na

- **similaridade em termos de cor e**
- **proximidade espacial**

Por isso, diz-se que é um agrupamento baseado em 5 dimensões

- 3 valores de cor (Não usa RGB, mas sim o espaço Lab de cores, que é mais uniforme)
- 2 coordenadas espaciais (l,c)



processo

Selecionar centros separados por uma distância “ s ” em linhas e colunas.

Depois, analisar a similaridade e proximidade de cada pixel em relação a estes centros.

Obs: uma distância “ s ” pequena produz um maior número de centros

Como a similaridade depende da cor e do espaço, as bordas das regiões resultantes não são regulares



Algoritmo SLIC - Superpixels

A faixa de variação das variáveis de cor é conhecida (resolução radiométrica)
A faixa de variação das coordenadas linha/coluna depende do tamanho da imagem, Por isso, as variáveis devem ser normalizadas para poder calcular uma medida de similaridade como a *Distância Euclidiana*.

A similaridade é calculada como uma composição da distancia Euclidiana espectral e a distancia Euclidiana espacial da seguinte forma:

$$D(A, B) = Dc + \frac{m}{s} Dxy$$

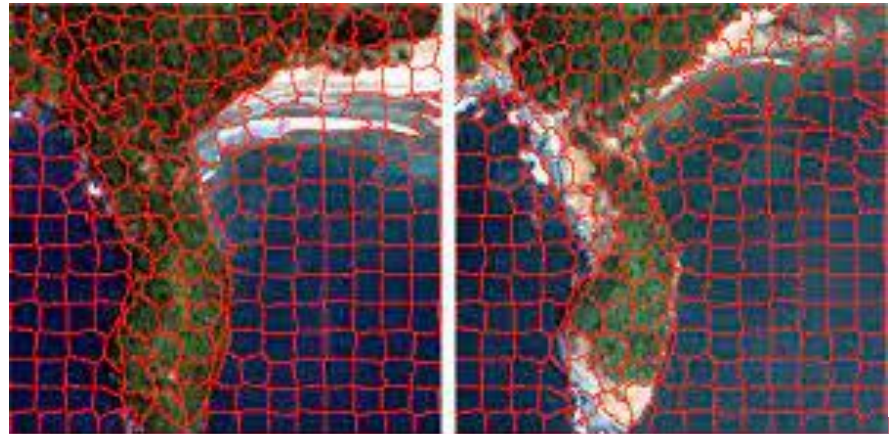
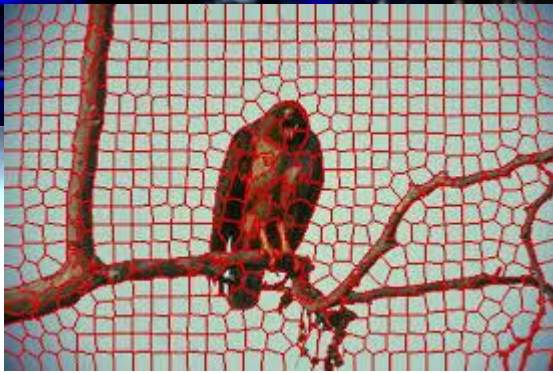
$$Dc = \sqrt{(l_A - l_B)^2 + (a_A - a_B)^2 + (b_A - b_B)^2}$$

$$Dxy = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

s : distância que separa os centros inicialmente determinados

m : parâmetro de segmentação que controla o crescimento das regiões

exemplos



(a)

(b)

Em python

```
# Converter para o espaço de cores Lab, pois SLIC funciona melhor em Lab
LAB = cv2.cvtColor(I, cv2.COLOR_BGR2LAB)
# 3. Parâmetros SLIC
# region_size: tamanho médio do superpixel
# ruler: fator de suavidade
slic = cv2.ximgproc.createSuperpixelSLIC( LAB, algorithm=cv2.ximgproc.SLIC,
region_size=20, ruler=100.0)
# Executar o algoritmo (pode definir o número de iterações)
slic.iterate(10)
# Obter os resultados
labels = slic.getLabels()           # Mapa de labels para cada pixel
mask = slic.getLabelContourMask()   # contornos para visualização
# Mascara a imagem original com as bordas
I[mask > 0] = [0, 255, 0] # Pintar bordas de verde
cv2_imshow(I)
```

Altere o fator de suavização, o tamanho do superpixel...