

Fundamentos de Processamento de Imagens

Aula 24

Segmentação de Imagens

Segmentação

- Particiona uma imagem em regiões distintas, correlacionadas com objetos ou elementos presentes na imagem
- Também pode ser entendido como o processo de agrupar pixels que apresentem atributos similares
- Geralmente, a primeira etapa no processo de tentar interpretar ou analisar imagens automaticamente
- A obtenção de segmentação correta e confiável é, em geral, muito difícil de obter de forma totalmente automatizada

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Tipos de Segmentação

- Não Contextual
 - Pixels são agrupados com base apenas em algum atributo global como, por exemplo, o nível de tons de cinza ou cor
 - Exemplo: limiarização (*thresholding*)
- Contextual
 - Pixels são agrupados com base em algum atributo global e na proximidade espacial
 - Exemplos: conectividade de pixels, similaridade de regiões, crescimento de regiões, algoritmo *split and merge*

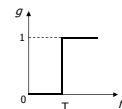
Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Limiarização

- Transforma uma imagem com valores em uma determinada faixa em uma nova imagem contendo apenas dois valores (imagem binária)

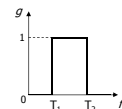
- Forma mais comum

$$g(x, y) = \begin{cases} 0, & f(x, y) < T \\ 1 \text{ ou } 255, & f(x, y) \geq T \end{cases}$$



- No caso de dois limiares

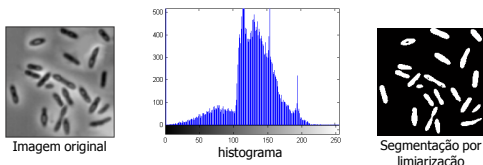
$$g(x, y) = \begin{cases} 0, & f(x, y) < T_1 \\ 1 \text{ ou } 255, & T_1 \leq f(x, y) \leq T_2 \\ 0, & f(x, y) > T_2 \end{cases}$$



Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Exemplo de Limiarização (1)

- Segmentando bactérias

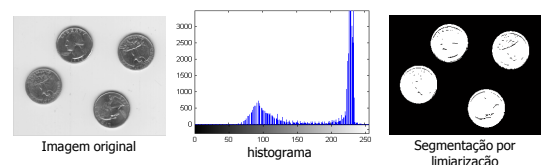


$$g(x, y) = \begin{cases} 255, & f(x, y) < 100 \\ 0, & f(x, y) \geq 100 \end{cases}$$

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS


Exemplo de Limiarização (2)

- Segmentando moedas



$$g(x, y) = \begin{cases} 255, & f(x, y) < 175 \\ 0, & f(x, y) \geq 175 \end{cases}$$


Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS



Limiarização

- O sucesso do processo de limiarização depende da escolha do(s) limiar(es)
- Os valores de limiar adequados variam entre imagens
 - Diferenças nas condições de iluminação e de contraste
 - Desejável procedimento automático para cálculo de limiar
- Abordagem mais geral baseia-se na análise do histograma
 - Picos no histograma correspondem aos elementos de interesse
 - Limiar: vale entre dois picos adjacentes
 - Em geral, picos adjacentes se sobrepõem, fazendo com que alguns pixels sejam detectados ou rejeitados erroneamente
 - Limiar ótimo minimiza o número de falsos positivos e negativos

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS




Determinação do Limiar

```

Determinação_iterativa_do_limiar (imagem f, float *limiar)
{
    // f: imagem original;
    // limiar: valor do limiar calculado iterativamente
    float m1 = média_dos_tons_de_cinza_dos_pixels_dos_cantos(f); // assume como fundo
    float m2 = média_dos_tons_de_cinza_dos_demais_pixels(f);
    float limiar_ANT = 0.0;
    float limiar_ATUAL = (m1 + m2)/2;
    //
    enquanto (limiar_ATUAL != limiar_ANT) faça
        m1 = média_dos_tons_de_cinza_dos_pixels_com_f(x,y) < limiar_ATUAL;
        m2 = média_dos_tons_de_cinza_dos_pixels_com_f(x,y) ≥ limiar_ATUAL;
        limiar_ANT = limiar_ATUAL;
        limiar_ATUAL = (m1 + m2)/2;
    fim enquanto
    *limiar = limiar_ATUAL;
}
            
```

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS



Exemplo de Determinação Automática do Limiar

- Resultados da execução do algoritmo de determinação do limiar sobre a imagem das bactérias

m ₁	m ₂	Limiar_atual
0.00	125.81	62.90
29.88	132.78	81.33
46.42	135.70	91.06
53.00	136.94	94.97
54.82	137.26	96.04
51.94	137.45	96.69

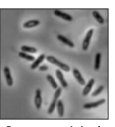
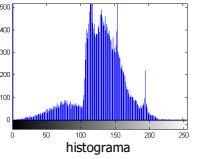






Imagem original

histograma

Segmentação por limiarização

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS




Limiarização em Imagens RGB

- Definem-se limiares independentemente para cada canal
 - Uma cor corresponde a um ponto em um espaço 3D de cores
 - A limiarização corresponde a um particionamento deste espaço
- Alternativamente, pode-se definir um limiar para a distância medida com relação a uma cor de referência (R₀, G₀, B₀)

$$g(x, y) = \begin{cases} 0, & d(x, y) < d_{\max} \\ 1 & \text{ou } 255, & d(x, y) \geq d_{\max} \end{cases}$$

onde $d(x, y)$ é a distância Euclidiana da cor associada ao pixel $f(x, y)$ e a cor de referência


Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS



Segmentação Contextual



- Principais Técnicas
 - Conectividade de Pixels (*Pixel Connectivity*)
 - Similaridade de Regiões
 - Crescimento de Regiões
 - Algoritmo *split and merge*

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS



Vizinhança entre Pixels

- Componente importante nas técnicas de segmentação baseadas em regiões
- Vizinhança-4 (4-neighbourhood) x vizinhança-8 (8-neighbourhood)

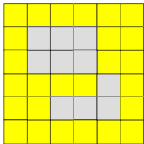



- Um caminho k-conexo entre pixels p_i e p_n é uma sequência $\{p_1, p_2, \dots, p_n\}$, onde p_{i+1} é um k-vizinho de $p_i, \forall i, i=1, \dots, n$

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Vizinhança entre Pixels

- O conjunto de pixels em cinza define
 - 1 região 8-conectada
 - 2 regiões 4-conectadas
- O conjunto de pixels em amarelo define um única região



Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Rotulação de Regiões Conexas

- Visita cada pixel da imagem, com um procedimento recursivo que rotula os pixels ainda não visitados
- Cada região conexa recebe um identificador único

```

Rotulação_de_regiões_conexas (imagem *in, imagem *out, float limiar)
// in: imagem de entrada; out: imagem de saída;
{
  int região = 1; float *média_região;
  para x = 1 até largura da imagem f faça
    para y = 1 até altura da imagem f faça
      // pixel não visitado
      se (in(x,y) > -1) então // pixel não visitado
        // rotula toda a região recursivamente
        Rotula(in, out, x, y, região, limiar, *média_região=in(x,y), *nro_pixels=1);
        região = região + 1;
}
  
```

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Rotulação de Regiões Conexas

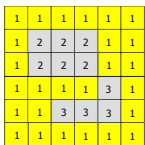
```

Rotula (imagem *in, imagem *out, int x, int y, int região, float limiar,
float *média_região, int *nro_pixels)
{ // in: imagem de entrada; out: imagem de saída
  int i, j;
  // se pixel satisfaz critério de uniformidade marca-o como visitado e rotula-o na saída
  se (abs(in(x,y) - média_região) ≤ limiar) então
    out(x,y) = região; *nro_pixels += 1;
    *média_região = ((*média_região)*(*nro_pixels - 1) + in(x,y))/(*nro_pixels);
    in(x,y) = -1;
    //
    para conect = 1 até conectividade faça // vizinhança-4 ou -8
      i = x + delta[conect].x; // soma deslocamentos
      j = y + delta[conect].y;
      se (Limites(in, i, j) && in(i,j) > -1) então
        Rotula(in, out, i, j, região, limiar, média_região, nro_pixels); // rotula
}
Limites (imagem *in, int i, int j)
{ retorna (i > 0 && i ≤ largura(in) && j > 0 && j ≤ altura(in)); }
  
```

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Rotulação de Regiões Conexas Exemplo

- Assumindo regiões 4-conectadas



Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Similaridade de Regiões

- Baseia-se na uniformidade das regiões conexas
- Predicado de Uniformidade em uma região R, P(R)
 - Condição que expressa similaridade entre tons de cinza ou cores

$$P(R) = \begin{cases} TRUE, & se \ |f(i, j) - \alpha| \leq \Delta \\ FALSE, & caso contrário \end{cases}$$

- onde α pode representar
 - o tom/cor de um pixel vizinho ($f(m, n)$)
 - a média dos tons/cores na região R (μ_R), excluindo-se o pixel em (i, j)
- Δ representa a máxima diferença definida para o critério de similaridade

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Crescimento de Regiões

- Cresce regiões similares a partir de um conjunto de pixels "sementes"
- Um pixel é incorporado a uma dada região se e somente se:
 - Ele ainda não pertence a nenhuma região
 - Ele encontra-se na vizinhança (fronteira) daquela região
 - A região permanece uniforme após a inclusão do pixel

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Crescimento de Regiões Exemplo

- Regiões 8-conectadas e $\Delta = 2$

6	7	8	6	7	8
6	1	1	1	1	6
1	0	1	0	0	7
6	7	8	6	7	8
6	1	1	6	1	2
1	0	2	1	2	1

Pixels "sementes"

6	7	8	6	7	8
6	1	1	1	1	6
1	0	1	0	0	7
6	7	8	6	7	8
6	1	1	6	1	2
1	0	2	1	0	1

Após 1a iteração

6	7	8	6	7	8
6	1	1	1	1	6
1	0	1	0	0	7
6	7	8	6	7	8
6	1	1	6	1	2
1	0	2	1	0	1

Resultado final

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Crescimento de Regiões

```

Crescimento_de_regiões (imagem *in, imagem *out, float Δ)
{
    // in: imagem de entrada; out: imagem de saída; Δ: dist. da média da região
    //
    Inicializa a imagem de saída com zeros
    Defina regiões, R1, R2, ..., Rn, na imagem saída, cada uma com um pixel semente
    Inicializa mregião = média dos tons/cores dos pixels na região Rregião
    //
    repita
        para região = 1 até n faça // para cada região
            para cada pixel p na borda da região Rregião faça
                para todos os vizinhos de p faça // considerar tipo de vizinhança: 4/8
                    (x,y) ← coordenadas do pixel vizinho de p;
                    se (out(x,y) == 0 && |in(x,y) - mregião| ≤ Δ) então
                        out(x,y) = Rregião
                        atualiza mregião;
            até que nenhum pixel possa mais ser incorporado a uma região
    }
    
```

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Crescimento de Regiões Limitações

- Operação não é estável
 - O resultado pode variar com o tipo de vizinhança (4 ou 8)
- Resultado depende da escolha do predicado de uniformidade
- Não garante uma segmentação completa da imagem

6	7	8	6	7	8
6	1	1	1	1	6
1	0	1	0	0	7
6	7	8	6	7	8
6	1	1	6	1	2
1	0	2	1	2	1

Regiões 8-conectadas e $\Delta = 2$

Pixels "sementes"

6	7	8	6	7	8
6	1	1	1	1	6
1	0	1	0	0	7
6	7	8	6	7	8
6	1	1	6	1	2
1	0	2	1	0	1

Resultado final

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Segmentação Completa

- Critérios para segmentação completa
 - Todos os pixels devem ser associados a alguma região
 - Cada pixel deve pertencer a exatamente uma região
 - Cada região define um conjunto conexo de pixels
 - Cada região deve ser uniforme
 - Regiões adjacentes devem ser não uniformes
- Crescimento de Regiões satisfaz apenas o 3º e 4º critérios
 - O número de sementes pode não ser suficiente para satisfazer o 1º e 2º critérios
 - Duas regiões adjacentes podem ser similares
 - Basta inicializar duas sementes numa área uniforme

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

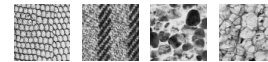
Algoritmo "Split and Merge"

- Produce segmentação completa
- Baseia-se em uma estratégia top-down
- Subdivide a imagem recursivamente em quadrantes até que P(R) seja verdadeiro
- Agrupar regiões adjacentes similares

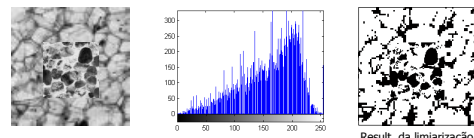
Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS

Segmentação Usando Outras Propriedades de Imagens

- Técnicas de segmentação baseadas nos valores de tons de cinza ou cores, em geral, não se aplicam a imagens contendo texturas (complexas)



- Exemplo: Sobreposição de 2 texturas (e limiarização em 128)



Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS



Segmentação de Texturas

- Na presença de texturas pode-se utilizar:
 - Técnicas estatísticas (*e.g.*, análise de variância de tons)
 - Técnicas de análise espectral

Copyright Manuel Menezes de Oliveira Neto, Informática UFRGS