Rotulação Contornos e Componentes Conexos

Alexandre Xavier Falcão

Instituto de Computação - UNICAMP

afalcao@ic.unicamp.br

Introdução

- Até o momento vimos operadores de imagem locais e baseados em relações de adjacência. A partir desta aula vamos abordar uma terceira categoria, a dos operadores baseados em **relações de conexidade**.

Introdução

- Até o momento vimos operadores de imagem locais e baseados em relações de adjacência. A partir desta aula vamos abordar uma terceira categoria, a dos operadores baseados em **relações de conexidade**.
- Seja $\hat{I} = (D_I, I)$ uma imagem, resultante de uma segmentação binária, tal que $I(p) \in \{0, 1\}$ para todo $p \in D_I$ (2D ou 3D), estamos interessados rotular seus **componentes conexos**.

Introdução

- Até o momento vimos operadores de imagem locais e baseados em relações de adjacência. A partir desta aula vamos abordar uma terceira categoria, a dos operadores baseados em **relações de conexidade**.
- Seja $\hat{I} = (D_I, I)$ uma imagem, resultante de uma segmentação binária, tal que $I(p) \in \{0, 1\}$ para todo $p \in D_I$ (2D ou 3D), estamos interessados rotular seus **componentes conexos**.
- Seja $\hat{L} = (D_I, L)$ uma imagem de componentes rotulados (caso 2D apenas), resultante de uma segmentação n-ária, estamos interessados em rotular **seus contornos**.

- Para uma dada relação de adjacência $\mathcal{A}$, um spel $q \in D_I$ é dito **conexo** a um spel $p \in D_I$, se existe uma sequência de spels distintos $\langle p_1, p_2, \ldots, p_n \rangle$, onde $p_1 = p$, $p_n = q$, e $(p_i, p_{i+1}) \in \mathcal{A}$.

Introdução

- Para uma dada relação de adjacência $\mathcal{A}$, um spel $q \in D_I$ é dito **conexo** a um spel $p \in D_I$, se existe uma sequência de spels distintos $\langle p_1, p_2, \ldots, p_n \rangle$, onde $p_1 = p$, $p_n = q$, e $(p_i, p_{i+1}) \in \mathcal{A}$.
- Um **componente conexo** é um conjunto $\mathcal{C} \subset D_I$ de spels onde todo par (p, q), $p \in \mathcal{C}$ e $q \in \mathcal{C}$, é conexo.

Introdução

- Para uma dada relação de adjacência $\mathcal{A}$, um spel $q \in D_I$ é dito **conexo** a um spel $p \in D_I$, se existe uma sequência de spels distintos $\langle p_1, p_2, \ldots, p_n \rangle$, onde $p_1 = p$, $p_n = q$, e $(p_i, p_{i+1}) \in \mathcal{A}$.
- Um **componente conexo** é um conjunto $\mathcal{C} \subset D_I$ de spels onde todo par (p, q), $p \in \mathcal{C}$ e $q \in \mathcal{C}$, é conexo.
- Dada a definição de componente conexo, vamos considerar apenas relações de adjacência **irreflexivas e simétricas**.

Relações de conexidade

Considere o exemplo abaixo de uma imagem 2D e responda:

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

Relações de conexidade

Considere o exemplo abaixo de uma imagem 2D e responda:

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

- Para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq 1$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?

Relações de conexidade

Considere o exemplo abaixo de uma imagem 2D e responda:

$$\begin{array}{cccccccccc} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{array}$$

- Para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq 1$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?
- Para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq \sqrt{2}$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?

Relações de conexidade

Considere o exemplo abaixo de uma imagem 2D e responda:

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

- Para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq 1$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?
- Para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq \sqrt{2}$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?
- Para $q \in \mathcal{A}(p)$ se $y_p = y_q$, $|x_q - x_p| \leq 3$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?

Relações de conexidade

Considere o exemplo abaixo de uma imagem 2D e responda:

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

- Para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq 1$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?
- Para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq \sqrt{2}$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?
- Para $q \in \mathcal{A}(p)$ se $y_p = y_q$, $|x_q - x_p| \leq 3$ e $I(p) = I(q) = 1$, quantos componentes conexos tem a imagem?
- Para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq \sqrt{2}$, quantos componentes conexos tem a imagem?

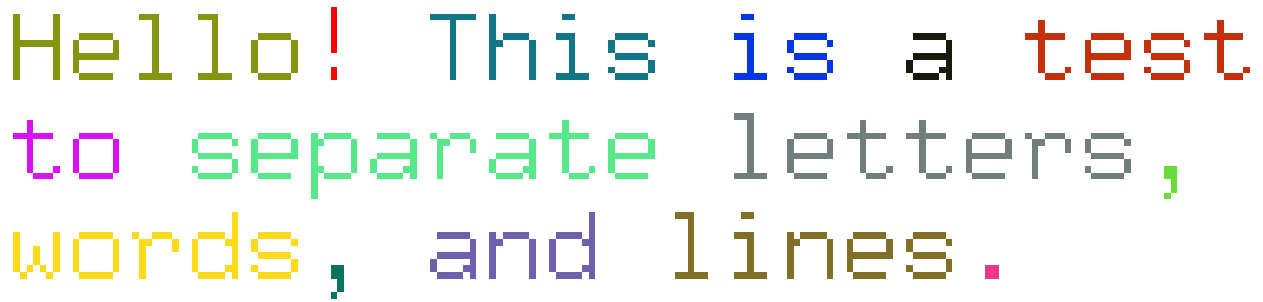
Relações de conexidade

Considere agora o problema de associar uma cor para cada componente conexo de uma imagem.

Rotulação dos pixels 1's com adjacência circular de raio $\sqrt{2}$ (letras) e raio 5 (palavras), e com adjacência retangular de tamanho 30×5 pixels (linhas).

Relações de conexidade

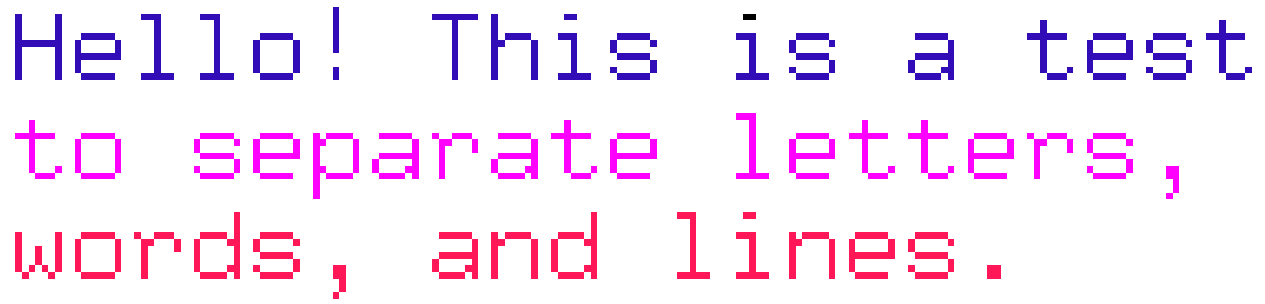
Considere agora o problema de associar uma cor para cada componente conexo de uma imagem.



Rotulação dos pixels 1's com adjacência circular de raio $\sqrt{2}$ (letras) e raio 5 (palavras), e com adjacência retangular de tamanho 30×5 pixels (linhas).

Relações de conexidade

Considere agora o problema de associar uma cor para cada componente conexo de uma imagem.

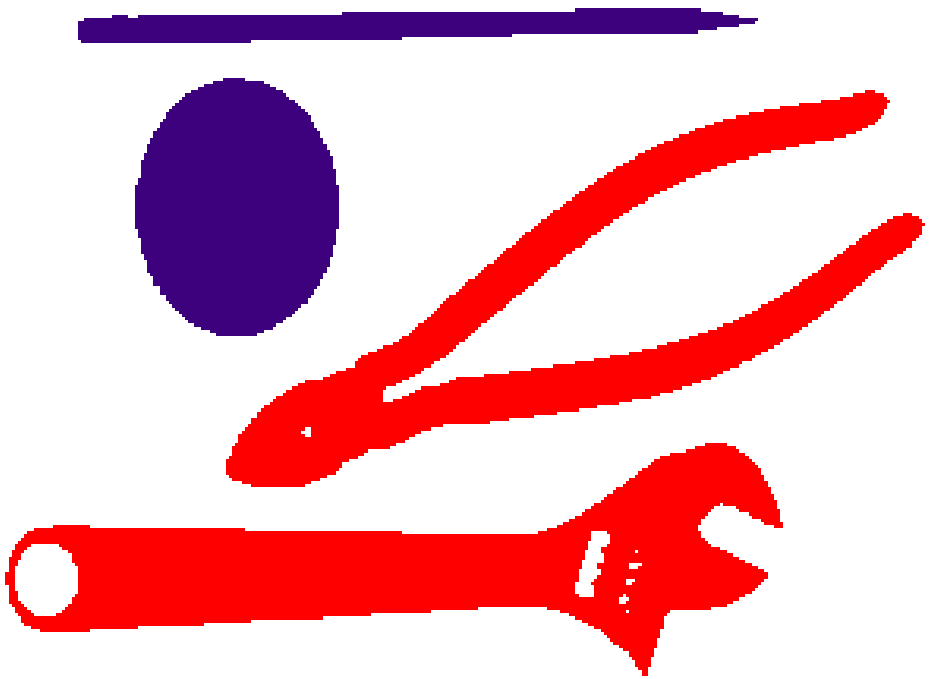


Rotulação dos pixels 1's com adjacência circular de raio $\sqrt{2}$ (letras) e raio 5 (palavras), e com adjacência retangular de tamanho 30×5 pixels (linhas).

Relações de conexidade

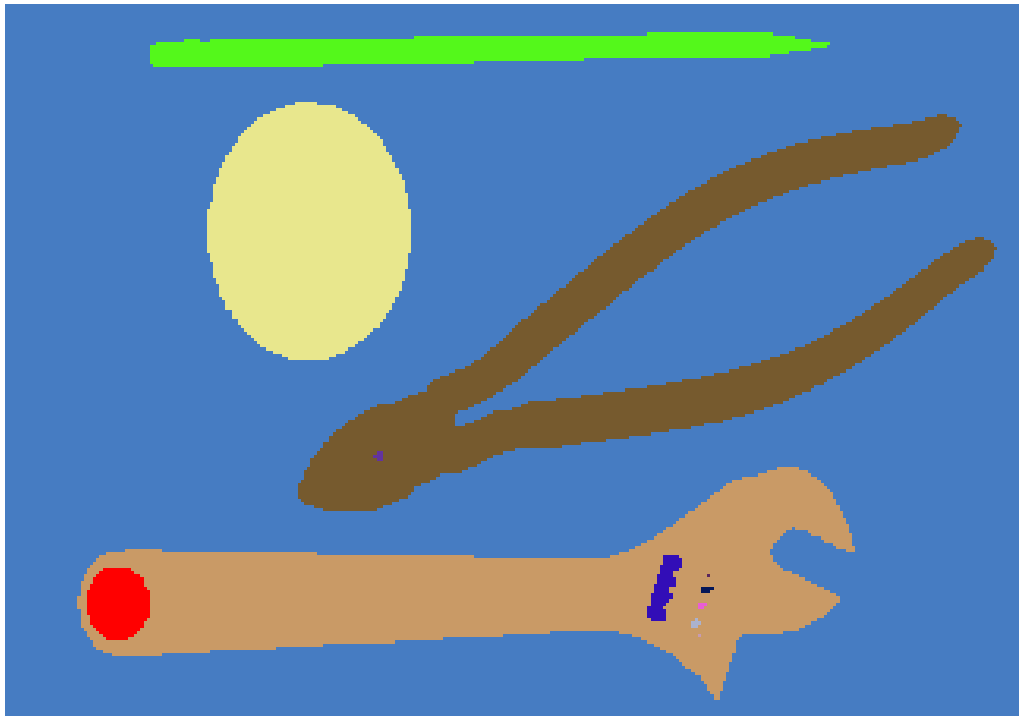
Rotulação dos pixels 1's com adjacência circular de raio $\sqrt{2}$, raio 15, e raio $\sqrt{2}$ incluindo os pixels 0's.

Relações de conexidade



Rotulação dos pixels 1's com adjacência circular de raio $\sqrt{2}$, raio 15, e raio $\sqrt{2}$ incluindo os pixels 0's.

Relações de conexidade



Rotulação dos pixels 1's com adjacência circular de raio $\sqrt{2}$, raio 15, e raio $\sqrt{2}$ incluindo os pixels 0's.

Rotulação de componentes conexos

- Considere o problema de associar um rótulo $I = 1, 2, \ldots, n_c$ a cada componente conexo **de mesmo brilho** em uma imagem binária $\hat{I} = (D_I, I)$ com n_c componentes de acordo com uma relação de adjacência $\mathcal{A}$.

Rotulação de componentes conexos

- Considere o problema de associar um rótulo $I = 1, 2, \ldots, n_c$ a cada componente conexo **de mesmo brilho** em uma imagem binária $\hat{I} = (D_I, I)$ com n_c componentes de acordo com uma relação de adjacência $\mathcal{A}$.
- Para qualquer pixel $p \in D_I$ ainda não rotulado ($L(p) = 0$), associamos um rótulo novo ($L(p) \leftarrow I$) e **propagamos** este rótulo a todos pixels q **conexos** a p usando **busca em largura** (uma fila FIFO - *First-In-First-Out*).

Rotulação de componentes conexos

- Entrada: Imagem $\hat{I} = (D_I, I)$ e relação $\mathcal{A}$.
- Saída: Imagem rotulada $\hat{L} = (D_I, L)$, onde $L(p) = 0$ inicialmente.
- Auxiliares: FIFO Q e variável inteira $l = 1$.

1. Para todo pixel $p \in D_I$, tal que $L(p) = 0$, faça
2. $\quad L(p) \leftarrow l$ e insira p em Q.
3. $\quad$ Enquanto $Q \neq \emptyset$ faça
4. $\quad\quad$ Remova p de Q.
5. $\quad\quad$ Para todo $q \in \mathcal{A}(p)$, $L(q) = 0$ e $I(p) = I(q)$, faça
6. $\quad\quad\quad L(q) \leftarrow L(p)$ e insira q em Q.
7. $\quad l \leftarrow l + 1$.

Rotulação de componentes conexos

No exemplo abaixo, o resultado para $q \in \mathcal{A}(p)$ se $\|q - p\| \leq 1$ e $I(p) = I(q)$,

0	1	0	1	1	0	0	0	1	1		1	2	3	4	4	1	1	1	5	5
0	1	1	0	1	0	0	0	1	0		1	2	2	1	4	1	1	1	5	1
0	0	0	0	0	0	1	0	0	0		1	1	1	1	1	1	6	1	1	1
0	0	0	0	0	0	1	0	0	0		1	1	1	1	1	1	6	1	1	1
1	1	0	0	1	1	1	0	0	0		7	7	1	1	6	6	6	1	1	1

Original (esquerda) e rotulada (direita).

Rotulação de contornos

A **borda** de um componente conexo $\mathcal{C} \subset D_I$ é o conjunto de spels $\mathcal{B} \subset \mathcal{C}$ tal que para todo $p \in \mathcal{B}$ existe um adjacente $q \notin \mathcal{C}$ (adjacente-4 em 2D e adjacente-6 em 3D).

1	2	2	2	2	3	3	3	3	3		a	b	b	b	b	c	c	c	c	c
1	1	2	2	2	3	2	3	3	3		a	a	b	2	b	c	b	c	3	c
1	1	2	2	2	3	2	3	3	3		a	a	b	2	b	c	b	c	3	c
1	1	2	2	2	2	2	3	3	3		a	a	b	2	b	b	b	c	3	c
1	1	2	2	2	2	2	3	3	3		a	a	b	b	b	b	b	c	c	c

Os pixels rotulados a, b, e c na imagem à direita são as bordas dos componentes $1, 2$, e 3 da imagem à esquerda.

Rotulação de contornos

Um **contorno** π (caso 2D apenas) é uma sequência de pixels distintos $\langle p_1, p_2, \ldots, p_n \rangle$ tal que $p_i \in \mathcal{B}$ e $(p_i, p_{(i\%n)+1}) \in \mathcal{A}_8$ (adjacentes-8), $i = 1, 2, \ldots, n$.

Rotulação de contornos

Um **contorno** π (caso 2D apenas) é uma sequência de pixels distintos $\langle p_1, p_2, \ldots, p_n \rangle$ tal que $p_i \in \mathcal{B}$ e $(p_i, p_{(i\%n)+1}) \in \mathcal{A}_8$ (adjacentes-8), $i = 1, 2, \ldots, n$.

- Isto é, um contorno é uma curva fechada, conexa, e orientada (curva de Jordan).

Rotulação de contornos

Um **contorno** π (caso 2D apenas) é uma sequência de pixels distintos $\langle p_1, p_2, \ldots, p_n \rangle$ tal que $p_i \in \mathcal{B}$ e $(p_i, p_{(i\%n)+1}) \in \mathcal{A}_8$ (adjacentes-8), $i = 1, 2, \ldots, n$.

- Isto é, um contorno é uma curva fechada, conexa, e orientada (curva de Jordan).
- Estamos interessados em associar um rótulo $l = 1, 2, \ldots, n$ a cada pixel de um contorno com n pixels bem como associar um rótulo $l = 1, 2, \ldots, m_c$, $m_c \geq n_c$, a cada contorno de uma imagem com n_c componentes (o algoritmo é praticamente o mesmo).

Rotulação de contornos

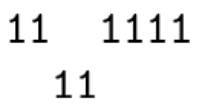
Um **contorno** π (caso 2D apenas) é uma sequência de pixels distintos $\langle p_1, p_2, \ldots, p_n \rangle$ tal que $p_i \in \mathcal{B}$ e $(p_i, p_{(i\%n)+1}) \in \mathcal{A}_8$ (adjacentes-8), $i = 1, 2, \ldots, n$.

- Isto é, um contorno é uma curva fechada, conexa, e orientada (curva de Jordan).
- Estamos interessados em associar um rótulo $l = 1, 2, \ldots, n$ a cada pixel de um contorno com n pixels bem como associar um rótulo $l = 1, 2, \ldots, m_c$, $m_c \geq n_c$, a cada contorno de uma imagem com n_c componentes (o algoritmo é praticamente o mesmo).
- Note que um componente pode produzir múltiplos contornos, caso tenha buracos.

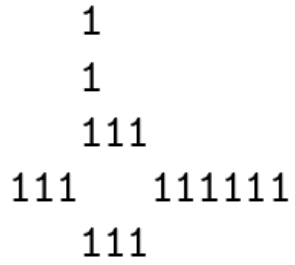
Rotulação de contornos

As bordas, porém, podem apresentar características indesejáveis para os algoritmos de rotulação de contornos.

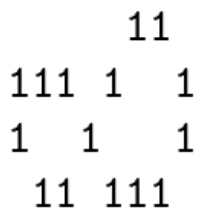
(a) Linha



(b) Contorno com ramos



(c) Contorno com pixel de estrangulamento



Rotulação de contornos

As bordas, porém, podem apresentar características indesejáveis para os algoritmos de rotulação de contornos.

(d) Dois contornos que se tocam

```
 111  11
1   11  1
1111  11
```

(e) Contorno que toca o lado oposto

```
 111  111
1   11   1
1   11   1
 111  111
```

Rotulação de contornos

- No caso (a), o componente é muito fino e todos os seus pixels são considerados borda (i.e. o componente é um **esqueleto**), sem distinção entre interior e borda.

Rotulação de contornos

- No caso (a), o componente é muito fino e todos os seus pixels são considerados borda (i.e. o componente é um **esqueleto**), sem distinção entre interior e borda.
- O caso (b) contém ramos que impossibilitam a enumeração sequencial de pixels de contorno.

Rotulação de contornos

- No caso (a), o componente é muito fino e todos os seus pixels são considerados borda (i.e. o componente é um **esqueleto**), sem distinção entre interior e borda.
- O caso (b) contém ramos que impossibilitam a enumeração sequencial de pixels de contorno.
- Os casos (c-e) geram ambiguidades se a entrada for uma imagem binária, pois não sabemos quando temos um contorno ou dois contornos que se tocam.

Rotulação de contornos

- Desejamos, portanto, um algoritmo que ignore os esqueletos (ou dilate os componentes) e os ramos, e

Rotulação de contornos

- Desejamos, portanto, um algoritmo que ignore os esqueletos (ou dilate os componentes) e os ramos, e
- resolva as ambiguidades dos casos (c-e).

Rotulação de contornos

- Desejamos, portanto, um algoritmo que ignore os esqueletos (ou dilate os componentes) e os ramos, e
- resolva as ambiguidades dos casos (c-e).
- Paratanto, devemos evitar segmentos de contorno entre pixels adjacentes-8 que passem **por dentro e por fora** do componente. Isto é, para cada $(p, q) \in \mathcal{A}_8$ temos os conceitos de pixel $e(p, q)$ à esquerda e pixel $d(p, q)$ à direita do segmento.

Rotulação de contornos

Dada uma imagem $\hat{L}_1 = (D_I, L_1)$.

C_1 Evitamos segmentos (p, q) que passam por dentro ou por fora dos componentes (i.e., evitamos $L_1(d(p,q)) = L_1(e(p,q))$ ou $L_1(d(p,q)) \neq L_1(e(p,q))$ sendo ambos diferentes de $L_1(p)$).

Rotulação de contornos

Dada uma imagem $\hat{L}_1 = (D_I, L_1)$.

- C_1 Evitamos segmentos (p, q) que passam por dentro ou por fora dos componentes (i.e., evitamos $L_1(d(p,q)) = L_1(e(p,q))$ ou $L_1(d(p,q)) \neq L_1(e(p,q))$ sendo ambos diferentes de $L_1(p)$).
- C_2 Adotamos $\mathcal{A}_8$ com pixels ordenados no sentido horário (ou anti-horário) em relação à origem.

Rotulação de contornos

Dada uma imagem $\hat{L}_1 = (D_I, L_1)$.

- C_1 Evitamos segmentos (p, q) que passam por dentro ou por fora dos componentes (i.e., evitamos $L_1(d(p,q)) = L_1(e(p,q))$ ou $L_1(d(p,q)) \neq L_1(e(p,q))$ sendo ambos diferentes de $L_1(p)$).
- C_2 Adotamos $\mathcal{A}_8$ com pixels ordenados no sentido horário (ou anti-horário) em relação à origem.
- C_3 Rotulamos cada contorno no sentido horário ($\mathcal{A}_8$ anti-horário), aceitando $d(p,q) \notin D_I$, desde que $e(p,q) \in D_I$ e $L_1(e(p,q)) = L_1(p)$.

Rotulação de contornos

- A rotulação é uma **busca em profundidade** (pilha ou fila LIFO - *Last-In-First-Out*) em cada borda $\mathcal{B}$, iniciada em um pixel p_1 com ao menos um segmento satisfazendo as condições acima.

Rotulação de contornos

- A rotulação é uma **busca em profundidade** (pilha ou fila LIFO - *Last-In-First-Out*) em cada borda $\mathcal{B}$, iniciada em um pixel p_1 com ao menos um segmento satisfazendo as condições acima.
- A busca gera um mapa de predecessores P tal que $P(p_1) = nil$ e $P(p_i) = p_{i-1}$ para $i > 1$. Os rótulos são atribuídos do final para o início de π.

Rotulação de contornos

Nos exemplos acima, temos a seguinte rotulação de pixels.

(a) Linha

```
      00   0000
        00
```

(b) Contorno com ramos

```
        0
        0
        812
007     300000
        654
```

Rotulação de contornos

Nos exemplos acima, temos a seguinte rotulação de pixels.

(c) Contorno com pixel de estrangulamento

```
    91
000 8  2
0  7   3
 00 654
```

(d) Dois contornos que se tocam

```
 912  61
8   35  2
7654  43
```

(e) Contorno que toca o lado oposto

```
  20  1  2      5  6  7
19         3  4         8
18        14 13         9
  17 16 15     12 11 10
```

Rotulação de contornos

Para o algoritmo, adotaremos:

- Entrada: Imagem de componentes rotulados $\hat{L}_1 = (D_I, L_1)$ e imagem de bordas $\hat{B} = (D_I, B)$ tal que $B(p) = L_1(p)$ se p for borda, ou $B(p) = 0$ no caso contrário.
- Saída: Imagem de pixels de contorno rotulados $\hat{L}_2 = (D_I, L_2)$.
- Auxiliares: Fila LIFO Q, variável I, mapas de cores C e predecessores P, inicializados com $C(s) = \textit{branco}$ e $L_2(s) = 0$ para todo $s \in D_I$.

Rotulação de contorno

1. Para todo $s \in D_I$, tal que $B(s) \neq 0$ e $C(s) = branco$, faça
2. Se existe $q \in A_8(s)$, $B(q) = L_1(s)$, $(s, q) \in C_1 \cap C_3$,
3. Faça $C(s) \leftarrow cinza$, $P(s) \leftarrow nil$, e insira s em Q.
4. Enquanto $Q \neq \emptyset$, faça
5. Remova p de Q e faça $C(p) \leftarrow preto$.
6. Para todo $q \in A_8(p)$, faça
7. Se $q = s$ e $P(p) \neq s$, vá para a linha 12.
8. Se $B(q) = L_1(s)$, $C(q) \neq preto$ e $(p, q) \in C_1 \cap C_3$,
9. Faça $P(q) \leftarrow p$.
10. Se $C(q) = branco$, insira q em Q e
11. $C(q) \leftarrow cinza$.
12. Faça $I \leftarrow 1$ e esvazie Q.
13. Enquanto $p \neq nil$, $L_2(p) \leftarrow I$, $p \leftarrow P(p)$ e $I \leftarrow I + 1$.