Fundamentals of Image Processing (part III)

Alexandre Xavier Falcão

Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

Connectivity-based image transformations

- In connectivity-based image transformations, the output value of a pixel p depends on the input values of a sequence of adjacent pixels with the terminus at p, named path.

Connectivity-based image transformations

- In connectivity-based image transformations, the output value of a pixel p depends on the input values of a sequence of adjacent pixels with the terminus at p, named path.

- A path connects p to the first pixel in the sequence, called root.

Connectivity-based image transformations

- In connectivity-based image transformations, the output value of a pixel p depends on the input values of a sequence of adjacent pixels with the terminus at p, named path.
- A path connects p to the first pixel in the sequence, called root.
- The root pixel is not always important, it is important the property that is propagated from it.

Connectivity-based image transformations

- In connectivity-based image transformations, the output value of a pixel p depends on the input values of a sequence of adjacent pixels with the terminus at p, named path.
- A path connects p to the first pixel in the sequence, called root.
- The root pixel is not always important, it is important the property that is propagated from it.
- It is also convevient to interpret the input image as a graph.

Agenda

- Images as weighted graphs.

Agenda

- Images as weighted graphs.
- Paths, connectivity relation, and connected components.

Agenda

- Images as weighted graphs.
- Paths, connectivity relation, and connected components.
- Labeling of connected components.

Images as weighted graphs

Let $\hat{I} = (D_I, \mathbf{I})$ be an image and $\mathcal{A}$ be an adjacency relation.

Images as weighted graphs

Let $\hat{I} = (D_I, \mathbf{I})$ be an image and $\mathcal{A}$ be an adjacency relation.

- A set of nodes may be a subset $\mathcal{N} \subseteq D_I$ of the image domain.

Images as weighted graphs

Let $\hat{I} = (D_I, \mathbf{I})$ be an image and $\mathcal{A}$ be an adjacency relation.

- A set of nodes may be a subset $\mathcal{N} \subseteq D_I$ of the image domain.

- $(\mathcal{N}, \mathcal{A}, \mathbf{I})$ is a graph with nodes weighted by $\mathbf{I}$ and arcs defined by $\mathcal{A}$.

Images as weighted graphs

Let $\hat{I} = (D_I, \mathbf{I})$ be an image and $\mathcal{A}$ be an adjacency relation.

- A set of nodes may be a subset $\mathcal{N} \subseteq D_I$ of the image domain.
- $(\mathcal{N}, \mathcal{A}, \mathbf{I})$ is a graph with nodes weighted by $\mathbf{I}$ and arcs defined by $\mathcal{A}$.
- The arcs $(p, q) \in \mathcal{A}$ may also be weighted, for instance, by $\|\mathbf{I}(q), \mathbf{I}(p)\|_2$.

Images as weighted graphs

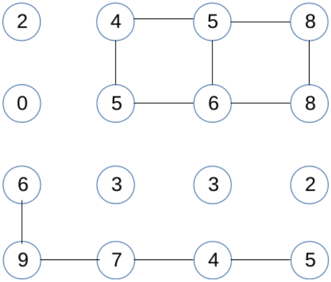
Let $\hat{I} = (D_I, \mathbf{I})$ be an image and $\mathcal{A}$ be an adjacency relation.

- A set of nodes may be a subset $\mathcal{N} \subseteq D_I$ of the image domain.
- $(\mathcal{N}, \mathcal{A}, \mathbf{I})$ is a graph with nodes weighted by $\mathbf{I}$ and arcs defined by $\mathcal{A}$.
- The arcs $(p, q) \in \mathcal{A}$ may also be weighted, for instance, by $\|\mathbf{I}(q), \mathbf{I}(p)\|_2$.

By choice of $\mathcal{A}$, there are several ways to interpret an image as a weighted graph.

Images as weighted graphs

Let the set of nodes be $\mathcal{N}$: $\{p \in D_I \mid I(p) \geq 4\}$ and $\mathcal{A}$: $\{(q,p) \in \mathcal{N} \times \mathcal{N} \mid \|q,p\|_2 \leq 1\}$ (4-neighborhood).



An image graph does not have to include all image pixels as nodes.

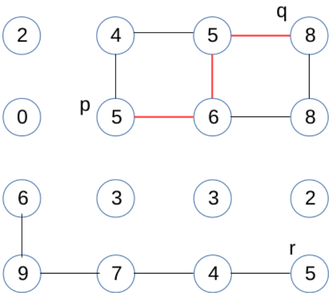
Paths and connectivity relation

- A path $\pi_p = \langle p_1, p_2, \ldots, p_K \rangle$ is a sequence of adjacent pixels $(p_k, p_{k+1}) \in \mathcal{A}$, $k \in [1, K-1]$, root p_1, and terminus $p = p_K$, being $\pi_p = \langle p \rangle$ a trivial path.

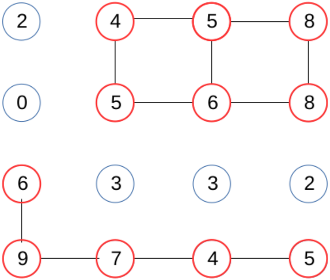
Paths and connectivity relation

- A path $\pi_p = \langle p_1, p_2, \ldots, p_K \rangle$ is a sequence of adjacent pixels $(p_k, p_{k+1}) \in \mathcal{A}$, $k \in [1, K-1]$, root p_1, and terminus $p = p_K$, being $\pi_p = \langle p \rangle$ a trivial path.

- The pixel q is said connected to a pixel p if exists a path π_q with root at p.



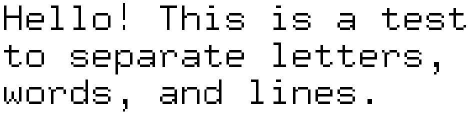
A connected component is a maximal subset $\mathcal{C} \subseteq \mathcal{N}$ in which all nodes are connected.



The graph contains two connected components.

Labeling of connected components

Let $\hat{I} = (D_I, I)$ be a binary image $I(p) \in \{0, 1\}$ of a text, in which we would like to separate letters, words, and lines.



We may define $\mathcal{N}$: $\{p \in D_I \mid I(p) = 1\}$.

Labeling of connected components

For $\mathcal{A}$ (8-neighborhood) defined as

$$\mathcal{A} \quad : \quad \{(p,q) \in \mathcal{N} \times \mathcal{N} \mid \|q,p\|_2 \leq \sqrt{2}\},$$

we can label each letter as a separated connected component.

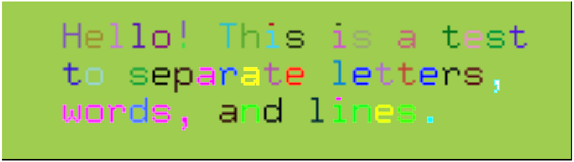
```
Hello! This is a test
to separate letters,
words, and lines.
```

The background is not labeled, but the code assigns a random color to regions with label 0.

For $\mathcal{A}$ (8-neighborhood) defined as

$$\mathcal{A} \quad : \quad \{(p,q) \in \mathcal{N} \times \mathcal{N} \mid \|q,p\|_2 \leq \sqrt{2}\},$$

we can label each letter as a separated connected component.



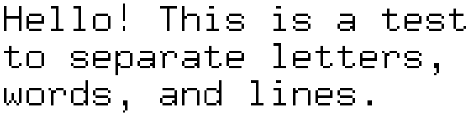
The background is not labeled, but the code assigns a random color to regions with label 0.

Labeling of connected components

For $\mathcal{A}$ defined as

$$\mathcal{A} \quad : \quad \{(p,q) \in \mathcal{N} \times \mathcal{N} \mid \|q,p\|_2 \leq 5\},$$

we can label each word as a separated connected component.

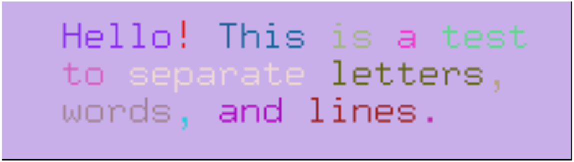


The background is not labeled, but the code assigns a random color to regions with label 0.

For $\mathcal{A}$ defined as

$$\mathcal{A} \quad : \quad \{(p,q) \in \mathcal{N} \times \mathcal{N} \mid \|q,p\|_2 \leq 5\},$$

we can label each word as a separated connected component.



The background is not labeled, but the code assigns a random color to regions with label 0.

Labeling of connected components

For $\mathcal{A}$ defined as

$$\mathcal{A} \quad : \quad \{(p,q) \in \mathcal{N} \times \mathcal{N} \mid |x_p - x_q| \leq \frac{a}{2}, \\ |y_p - y_q| \leq \frac{b}{2}\},$$

we can assign distinct labels to the lines of a text, when $a = 30$ and $b = 5$.

Hello! This is a test
to separate letters,
words, and lines.

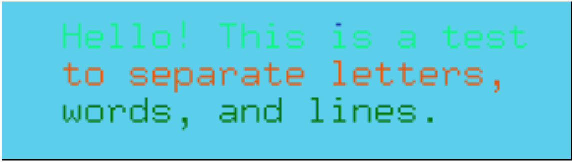
The background is not labeled, but the code assigns a random color to regions with label 0.

Labeling of connected components

For $\mathcal{A}$ defined as

$$\mathcal{A} \quad : \quad \{(p,q) \in \mathcal{N} \times \mathcal{N} \mid |x_p - x_q| \leq \frac{a}{2}, \\ |y_p - y_q| \leq \frac{b}{2}\},$$

we can assign distinct labels to the lines of a text, when $a = 30$ and $b = 5$.



The background is not labeled, but the code assigns a random color to regions with label 0.

Labeling of connected components

This algorithm creates a label map $L\colon \mathcal{N} \rightarrow \{1, 2, \ldots, c\}$ of the graph components.

1. Set $l \leftarrow 1$ and, $\forall p \in \mathcal{N}$, set $L(p) \leftarrow 0$.
2. For each $r \in \mathcal{N} \mid L(r) = 0$, do.
3. Set $L(r) \leftarrow l$ and insert r in $\mathcal{Q}$.
4. While $\mathcal{Q} \neq \emptyset$, do.
5. Remove p from $\mathcal{Q}$.
6. For each $q \in \mathcal{A}(p) \mid L(q) = 0$, do.
7. Set $L(q) \leftarrow l$ and insert q in $\mathcal{Q}$.
8. Set $l \leftarrow l + 1$.

Labeling of connected components

This algorithm creates a label map $L\colon \mathcal{N} \rightarrow \{1, 2, \ldots, c\}$ of the graph components.

1. Set $l \leftarrow 1$ and, $\forall p \in \mathcal{N}$, set $L(p) \leftarrow 0$.
2. For each $r \in \mathcal{N} \mid L(r) = 0$, do.
3. Set $L(r) \leftarrow l$ and insert r in $\mathcal{Q}$.
4. While $\mathcal{Q} \neq \emptyset$, do.
5.   Remove p from $\mathcal{Q}$.
6.   For each $q \in \mathcal{A}(p) \mid L(q) = 0$, do.
7.   Set $L(q) \leftarrow l$ and insert q in $\mathcal{Q}$.
8. Set $l \leftarrow l + 1$.

A pixel q receives label $L(r)$ of its root r when it is reached by a path π_q from a predecessor node p.

Exercise

A rooted spanning forest P is an acyclic map that assigns to very node $q \in \mathcal{N}$ its predecessor node $P(q) = p$ in the path π_q, or a marker $P(q) = nil \notin \mathcal{N}$ when q is a root in the map. P stores in backward all paths from the root set to every other node in $\mathcal{N}$.

Exercise

A rooted spanning forest P is an acyclic map that assigns to very node $q \in \mathcal{N}$ its predecessor node $P(q) = p$ in the path π_q, or a marker $P(q) = nil \notin \mathcal{N}$ when q is a root in the map. P stores in backward all paths from the root set to every other node in $\mathcal{N}$.

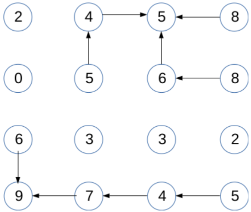
- Change the previous algorithm to create a predecessor map P.

Exercise

A rooted spanning forest P is an acyclic map that assigns to very node $q \in \mathcal{N}$ its predecessor node $P(q) = p$ in the path π_q, or a marker $P(q) = nil \notin \mathcal{N}$ when q is a root in the map. P stores in backward all paths from the root set to every other node in $\mathcal{N}$.

- Change the previous algorithm to create a predecessor map P.

- Present an algorithm to find the root r of a node p from P.



Now, if $\mathcal{N} = D_I$ and $\mathcal{A}$ is defined as

$$\mathcal{A} \quad : \quad \{(p, q) \in \mathcal{N} \times \mathcal{N} \mid \|q, p\|_2 \leq \rho \text{ and } |I(q) - I(R(p))| \leq T\},$$

where $T \geq 0$, $\rho \geq 1$, and $R(p) = r$ is the root node in π_q that reaches q, then the algorithm results

The root information requires to include $R(r) \leftarrow r$ in Line 3 and $R(q) \leftarrow R(p)$ in Line 7.

Now, if $\mathcal{N} = D_I$ and $\mathcal{A}$ is defined as

$$\mathcal{A} \quad : \quad \{(p,q) \in \mathcal{N} \times \mathcal{N} \mid \|q,p\|_2 \leq \rho \text{ and } |I(q) - I(R(p))| \leq T\},$$

where $T \geq 0$, $\rho \geq 1$, and $R(p) = r$ is the root node in π_q that reaches q, then the algorithm results

The root information requires to include $R(r) \leftarrow r$ in Line 3 and $R(q) \leftarrow R(p)$ in Line 7.

Other examples of component labeling in real images.

Other examples of component labeling in real images.

Examples

Let's see Labeling.ipynb in notebooks.tar.gz