# Image Foresting Transform

Alexandre Xavier Falcão

Visual Informatics Laboratory - Institute of Computing - University of Campinas

afalcao@ic.unicamp.br

# Introduction

Image transformations are usually based on

# Introduction

Image transformations are usually based on

- pixels (e.g., thresholding).

# Introduction

Image transformations are usually based on

- pixels (e.g., thresholding).
- adjacency relations: pixels and their neighbors (e.g., linear filtering).

# Introduction

Image transformations are usually based on

- pixels (e.g., thresholding).
- adjacency relations: pixels and their neighbors (e.g., linear filtering).
- connectivity relations: sequences of adjacent pixels (e.g., component labeling).

# Introduction

- The interpretation of an image as a graph provides a more general topology to the design of image transformations.

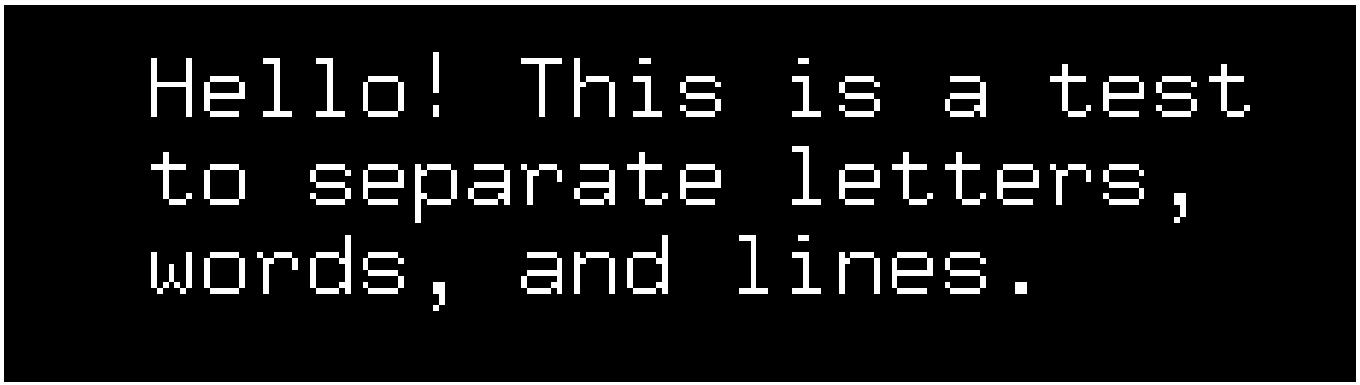
# Introduction

- The interpretation of an image as a graph provides a more general topology to the design of image transformations.
- The graph nodes may be pixels, edges, regions, and the arcs will result from a given adjacency relation.

# Introduction

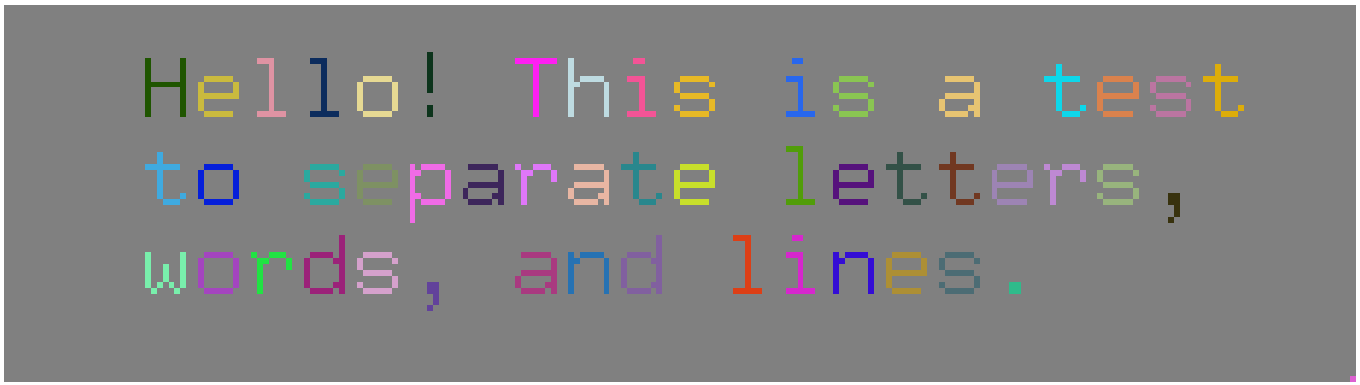
- The interpretation of an image as a graph provides a more general topology to the design of image transformations.
- The graph nodes may be pixels, edges, regions, and the arcs will result from a given adjacency relation.
- This strategy counts with several algorithms from Graph Theory and their proof of correctness.

The same algorithm with distinct adjacency relations, for example, can label letters, words and lines.

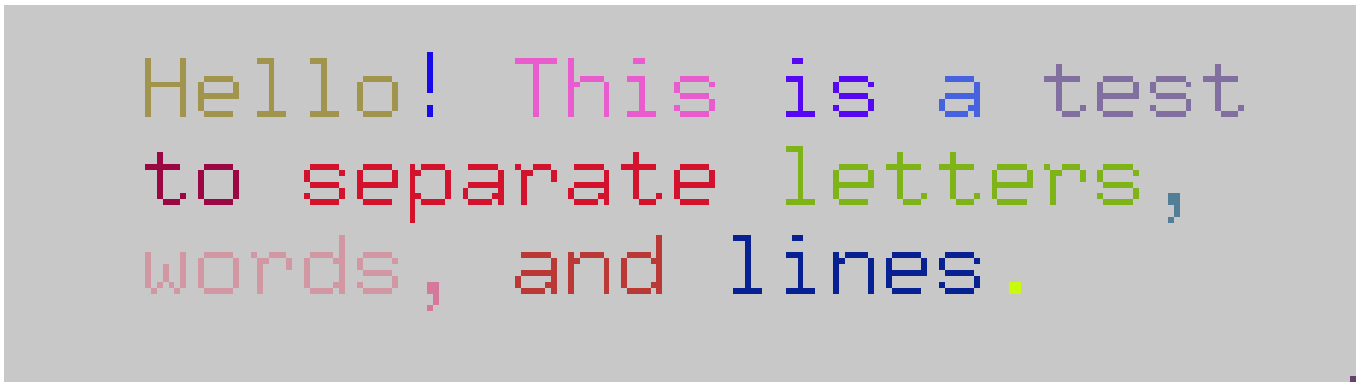


# Introduction

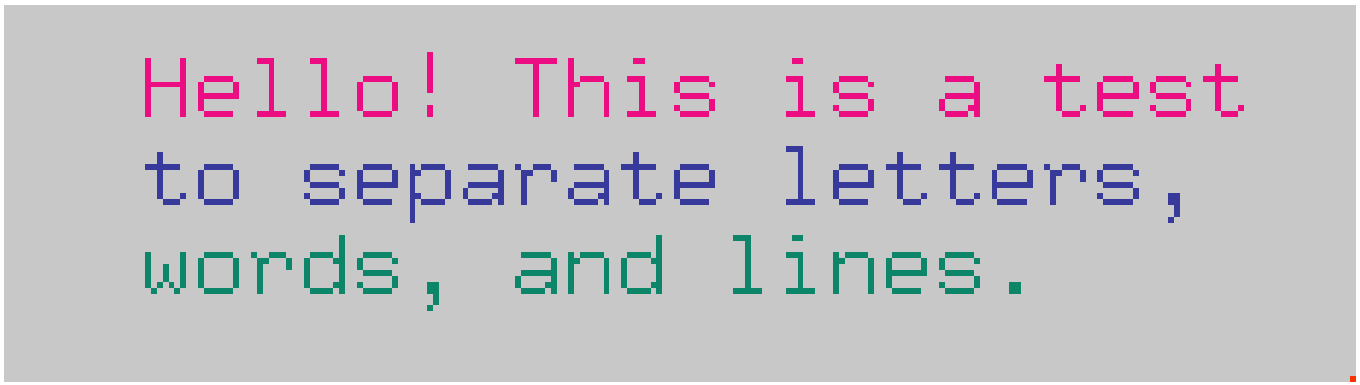
The same algorithm with distinct adjacency relations, for example, can label letters, words and lines.



The same algorithm with distinct adjacency relations, for example, can label letters, words and lines.



The same algorithm with distinct adjacency relations, for example, can label letters, words and lines.



# Introduction

This lecture presents the Image Foresting Transform (IFT) — a tool for the design of image operators based on optimum connectivity [1] — which provides

# Introduction

This lecture presents the Image Foresting Transform (IFT) — a tool for the design of image operators based on optimum connectivity [1] — which provides

- linear-time algorithms for most applications [2, 3],

# Introduction

This lecture presents the Image Foresting Transform (IFT) — a tool for the design of image operators based on optimum connectivity [1] — which provides

- linear-time algorithms for most applications [2, 3],
- unified framework to improve methods [4, 5, 6, 7, 8], including hardware-based implementations [9, 10], and

# Introduction

This lecture presents the Image Foresting Transform (IFT) — a tool for the design of image operators based on optimum connectivity [1] — which provides

- linear-time algorithms for most applications [2, 3],
- unified framework to improve methods [4, 5, 6, 7, 8], including hardware-based implementations [9, 10], and
- effective solutions for segmentation [11, 12, 13, 14, 15], filtering [4], representation [16], description [17, 18, 19], clustering [20], and classification [21, 22, 23].

# Organization of this lecture

- Basic definitions.

# Organization of this lecture

- Basic definitions.
- Images as graphs.

# Organization of this lecture

- Basic definitions.
- Images as graphs.
- Connectivity functions.

# Organization of this lecture

- Basic definitions.
- Images as graphs.
- Connectivity functions.
- Image foresting transform.

# Organization of this lecture

- Basic definitions.
- Images as graphs.
- Connectivity functions.
- Image foresting transform.
- General algorithm, variants, and implementation issues.

# General image definition

A digital image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ is a pair, where

# General image definition

A digital image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ is a pair, where

- $\mathcal{D}_I \subset \mathcal{Z}^n$ is the image domain (a set of spels — space elements), and

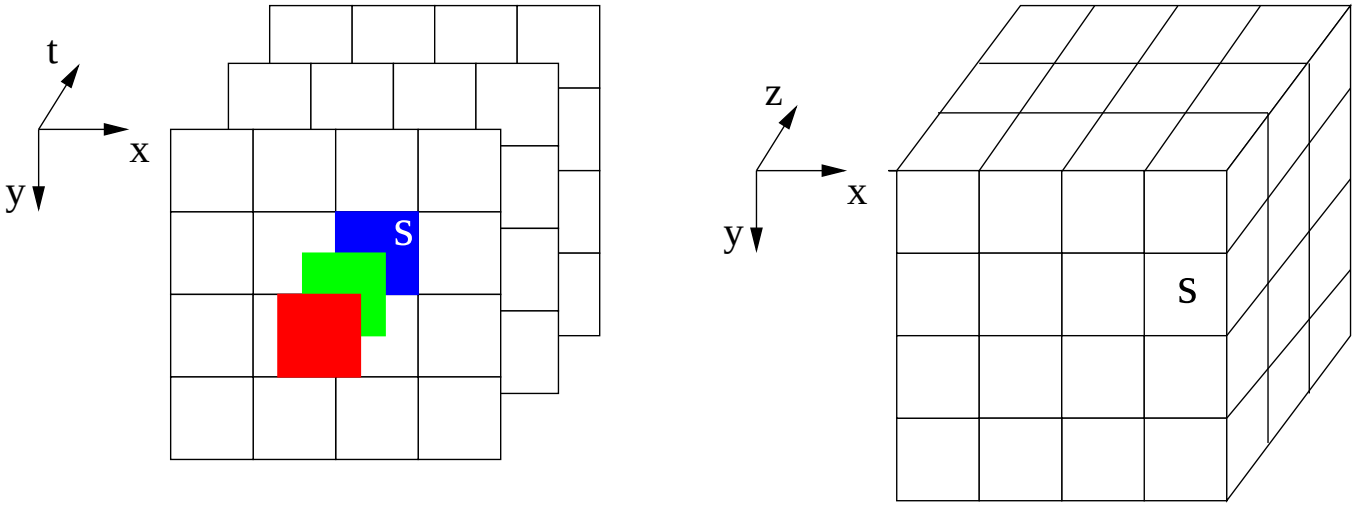
# General image definition

A digital image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ is a pair, where

- $\mathcal{D}_I \subset \mathcal{Z}^n$ is the image domain (a set of spels — space elements), and
- $\vec{I}(s) = (I_1(s), I_2(s), \ldots, I_m(s)) \in \mathcal{Z}^m$ is a vectorial mapping, which assigns a set of values to each $s \in \mathcal{D}_I$.

For $m = 1$, we use $\mathbf{I} = (\mathcal{D}_I, I)$.

# General image definition

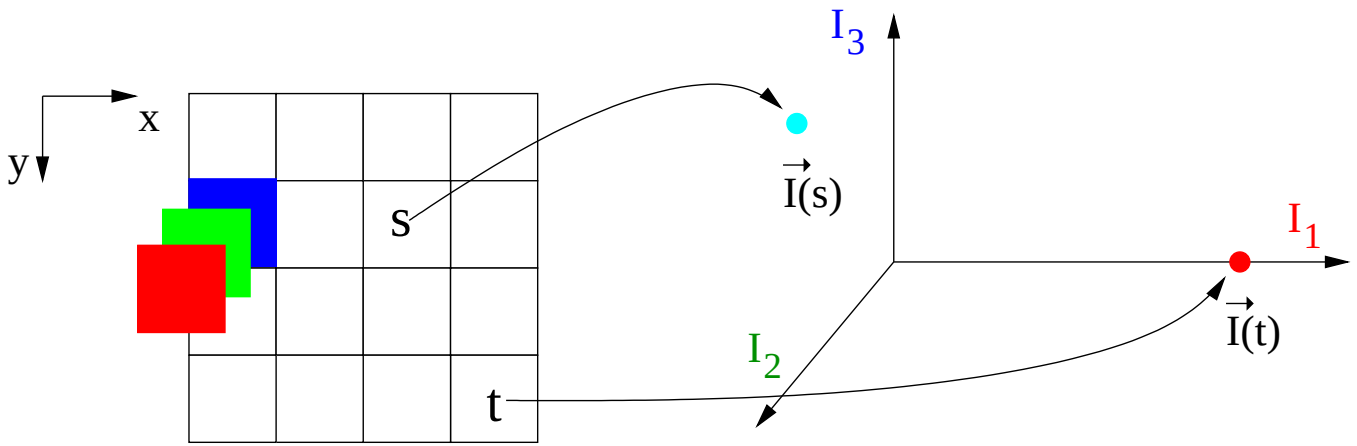



(a) (b)

(a) A RGB video $\mathbf{I}$, $n = 3$ and $m = 3$. (b) A CT image $\mathbf{I}$, $n = 3$ and $m = 1$.

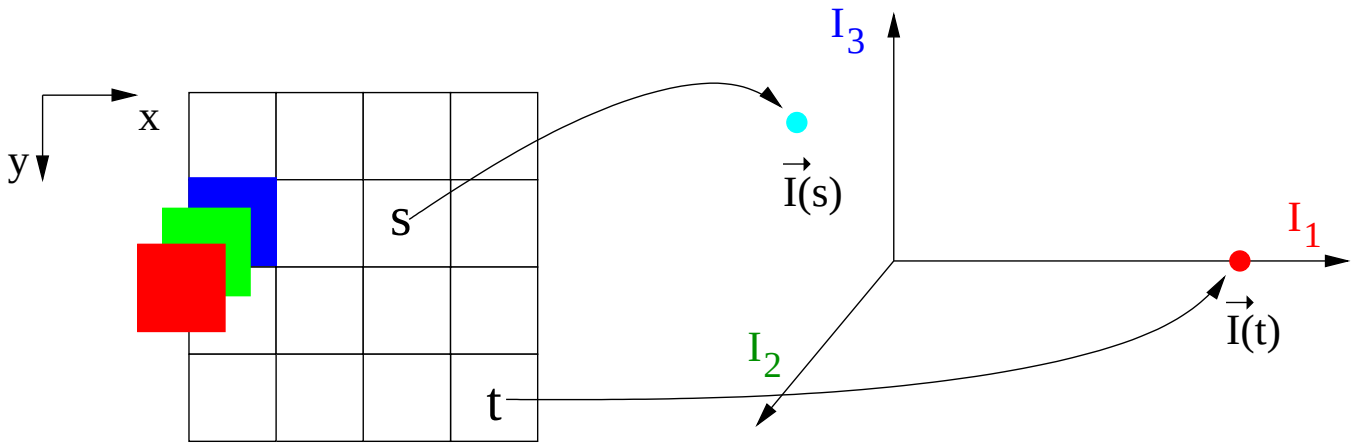
# Image domain and feature space

A spel $s \in \mathcal{D}_I$ is a point $\vec{I}(s) \in \mathcal{Z}^m$ in the feature space.



# Image domain and feature space

A spel $s \in \mathcal{D}_I$ is a point $\vec{I}(s) \in \mathcal{Z}^m$ in the feature space.




New features may result from any image transformation $\Psi(\mathbf{I})$, creating a real image $\mathbf{F} = (\mathcal{D}_F, \vec{F})$ where $\mathcal{D}_F = \mathcal{D}_I$ and $\vec{F}(s) = (F_1, F_2, \ldots, F_{m'}) \in \Re^{m'}$.

# Images as graphs

An image can be interpreted as a graph $(\mathcal{N}, \mathcal{A})$, whose

# Images as graphs

An image can be interpreted as a graph $(\mathcal{N}, \mathcal{A})$, whose

- nodes form a subset $\mathcal{N} \subseteq \mathcal{D}_I$ and

# Images as graphs

An image can be interpreted as a graph $(\mathcal{N}, \mathcal{A})$, whose

- nodes form a subset $\mathcal{N} \subseteq \mathcal{D}_I$ and
- arcs are defined by an adjacency relation $\mathcal{A} \subset \mathcal{D}_I \times \mathcal{D}_I$.

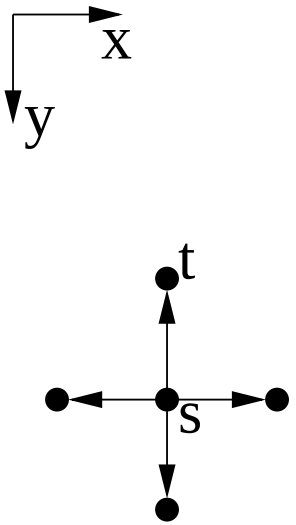
# Images as graphs

An image can be interpreted as a graph $(\mathcal{N}, \mathcal{A})$, whose

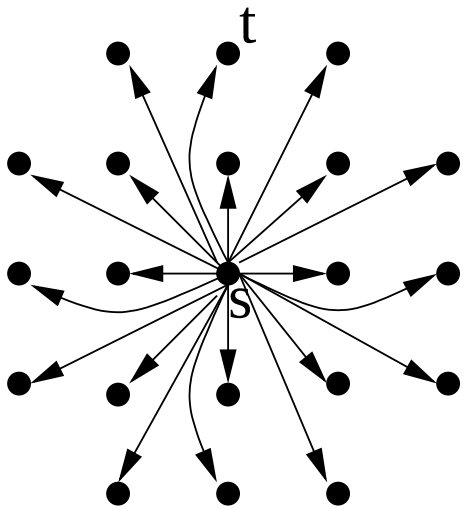
- nodes form a subset $\mathcal{N} \subseteq \mathcal{D}_I$ and
- arcs are defined by an adjacency relation $\mathcal{A} \subset \mathcal{D}_I \times \mathcal{D}_I$.

We will indicate that a spel $t$ is adjacent to spel $s$ either by $(s, t) \in \mathcal{A}$ or $t \in \mathcal{A}(s)$.

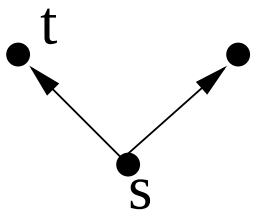
# Images as graphs




(a) Ex 1: r = 1



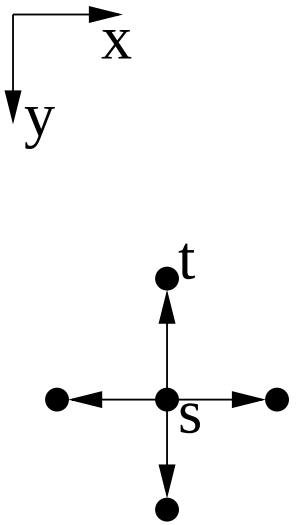

(b) Ex 1: r = $\sqrt{5}$



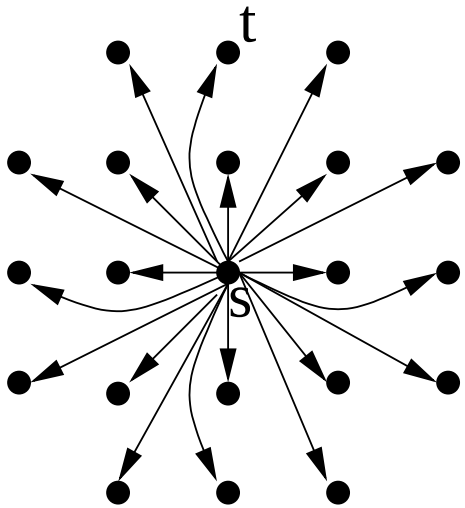

(c) Ex 2

- Example 1: $(s, t) \in \mathcal{A}$ when $\|t - s\|^2 \leq r^2$, for $r \geq 1$.

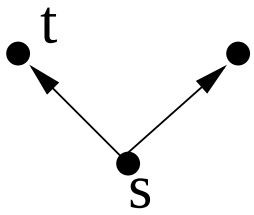
# Images as graphs




(a) Ex 1: r = 1



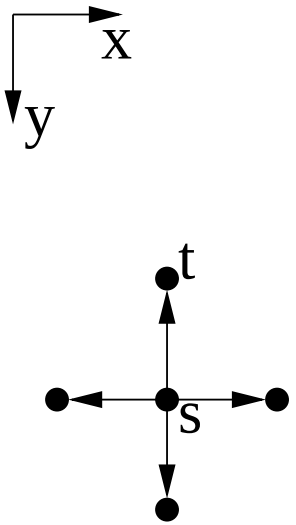

(b) Ex 1: r = $\sqrt{5}$



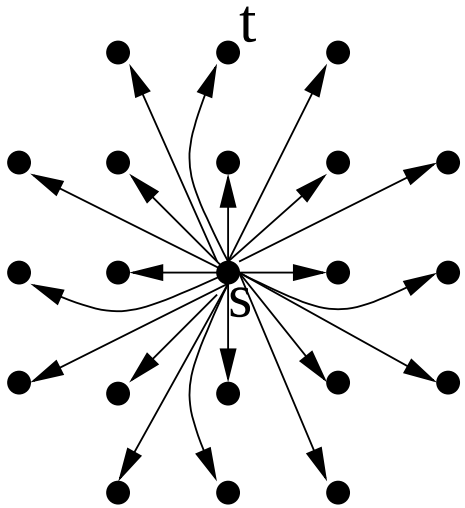

(c) Ex 2

- Example 1: $(s,t) \in \mathcal{A}$ when $\|t-s\|^2 \leq r^2$, for $r \geq 1$.
- Example 2: $(s,t) \in \mathcal{A}$ when $t-s \in \{(-1,-1),(1,-1)\}$.

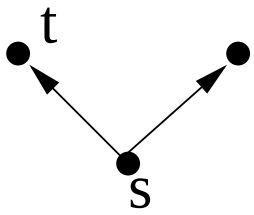
# Images as graphs




(a) Ex 1: r = 1




(b) Ex 1: r = $\sqrt{5}$




(c) Ex 2

- Example 1: $(s,t) \in \mathcal{A}$ when $\|t-s\|^2 \leq r^2$, for $r \geq 1$.
- Example 2: $(s,t) \in \mathcal{A}$ when $t-s \in \{(-1,-1),(1,-1)\}$.
- Example 3: $(s,t) \in \mathcal{A}$ when $t$ is a $k$-nearest neighbor of $s$ in the feature space, for $k \geq 1$.

# Images as graphs

Position invariant relations $\mathcal{A}$ can be represented by a vector of relative displacements

$$t - s \quad \in \quad \{(dx_1, dy_1), (dx_2, dy_2), \ldots, (dx_d, dy_d)\},$$

and fixed size $d = |\mathcal{A}(s)|$ $\forall s$, leading to an implicit graph representation.

$$(x_t, y_t) \quad = \quad (x_s, y_s) + (dx_i, dy_i), i = 1, 2, \ldots, d,$$

where $t = (x_t, y_t)$ and $s = (x_s, y_s)$.

# Connectivity functions

Connectivity-based operators use paths in $(\mathcal{N}, \mathcal{A})$.

# Connectivity functions

Connectivity-based operators use paths in $(\mathcal{N}, \mathcal{A})$.

- A path $\pi_t$ is a sequence of distinct adjacent nodes with terminus at node $t$.

# Connectivity functions

Connectivity-based operators use paths in $(\mathcal{N}, \mathcal{A})$.

- A path $\pi_t$ is a sequence of distinct adjacent nodes with terminus at node $t$.
- A spel $t$ is said connected to a spel $r$ if there exists a path $\pi_t$ from $r$ to $t$, in which case the root $R(\pi_t) = r$.

# Connectivity functions

Connectivity-based operators use paths in $(\mathcal{N}, \mathcal{A})$.

- A path $\pi_t$ is a sequence of distinct adjacent nodes with terminus at node $t$.
- A spel $t$ is said connected to a spel $r$ if there exists a path $\pi_t$ from $r$ to $t$, in which case the root $R(\pi_t) = r$.
- A connectivity function $f(\pi_t)$ assigns a value to any path.

# Connectivity functions

Connectivity-based operators use paths in $(\mathcal{N}, \mathcal{A})$.

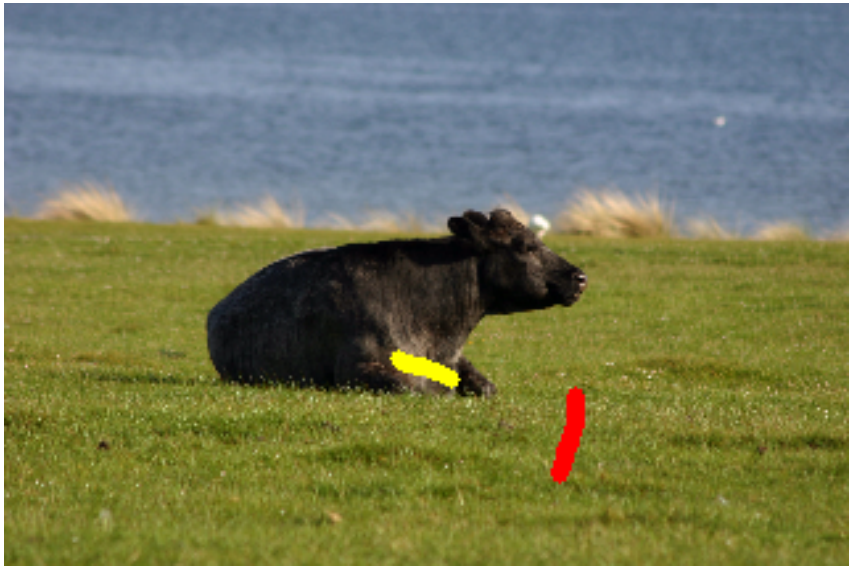
- A path $\pi_t$ is a sequence of distinct adjacent nodes with terminus at node $t$.
- A spel $t$ is said connected to a spel $r$ if there exists a path $\pi_t$ from $r$ to $t$, in which case the root $R(\pi_t) = r$.
- A connectivity function $f(\pi_t)$ assigns a value to any path.
- A path $\pi_t$ is optimum if $f(\pi_t) \leq f(\tau_t)$ for any other $\tau_t$ (minimum).

# Connectivity functions

Connectivity-based operators use paths in $(\mathcal{N}, \mathcal{A})$.

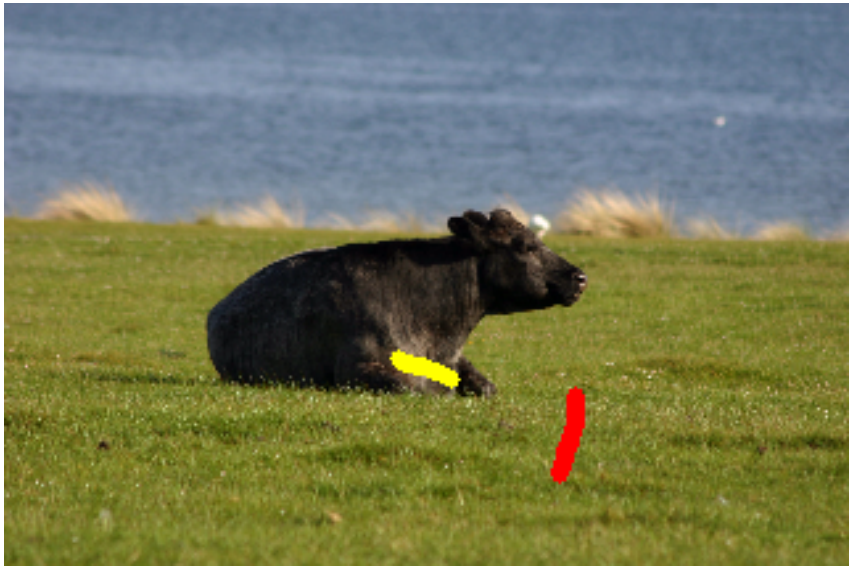
- A path $\pi_t$ is a sequence of distinct adjacent nodes with terminus at node $t$.
- A spel $t$ is said connected to a spel $r$ if there exists a path $\pi_t$ from $r$ to $t$, in which case the root $R(\pi_t) = r$.
- A connectivity function $f(\pi_t)$ assigns a value to any path.
- A path $\pi_t$ is optimum if $f(\pi_t) \leq f(\tau_t)$ for any other $\tau_t$ (minimum).
- The dual definition $f(\pi_t) \geq f(\tau_t)$ (maximum) is also valid.

# Connectivity functions



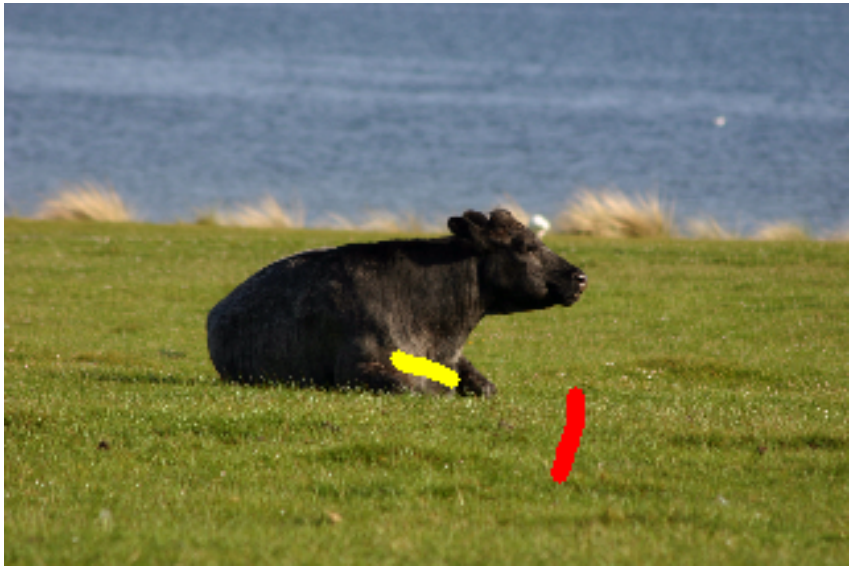
- Consider for example a segmentation problem, where internal $\mathcal{S}_i$ (yellow) and external $\mathcal{S}_e$ (red) sets of seed spels are given.

# Connectivity functions



- Consider for example a segmentation problem, where internal $\mathcal{S}_i$ (yellow) and external $\mathcal{S}_e$ (red) sets of seed spels are given.
- The seeds may compete among themselves by offering optimum paths to every spel in the image.

# Connectivity functions



- Consider for example a segmentation problem, where internal $\mathcal{S}_i$ (yellow) and external $\mathcal{S}_e$ (red) sets of seed spels are given.
- The seeds may compete among themselves by offering optimum paths to every spel in the image.
- The object can be defined by spels $t$ whose optimum paths $\pi_t$ have roots $R(\pi_t)$ in $\mathcal{S}_i$.

# Connectivity functions

We may

- interpret the image as an 8-neighborhood graph,

# Connectivity functions

We may

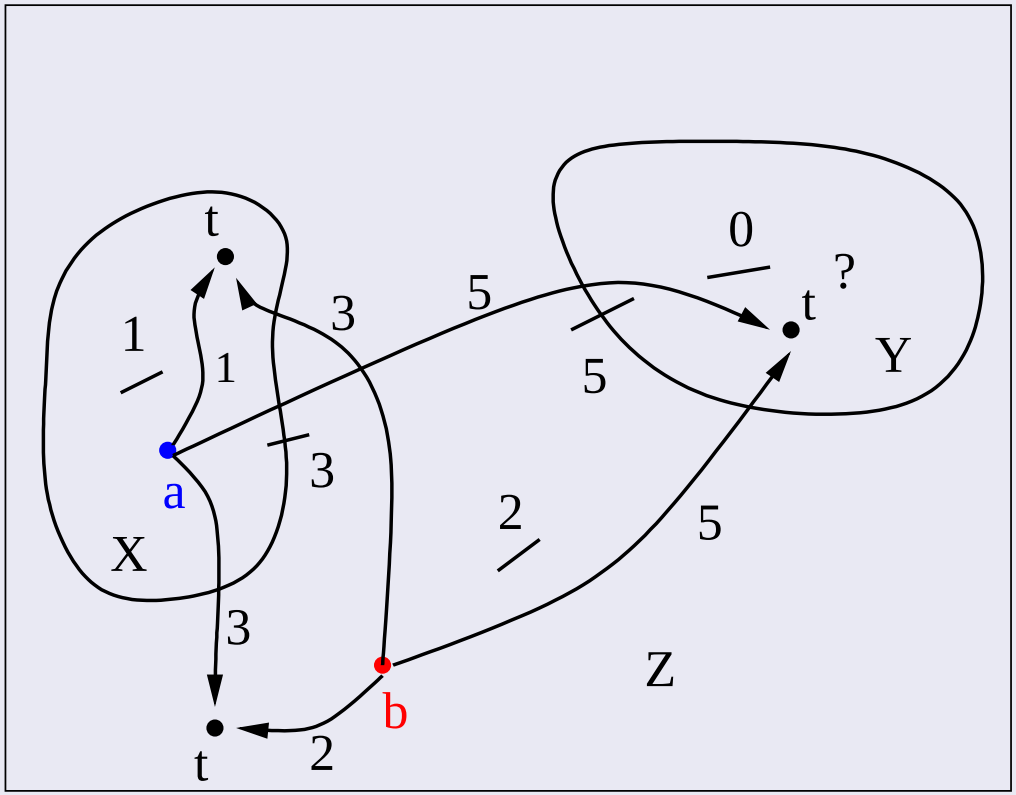
- interpret the image as an 8-neighborhood graph,
- assign higher arc weights $w(s, t)$ across the object's boundary than inside and outside it, and

# Connectivity functions

We may

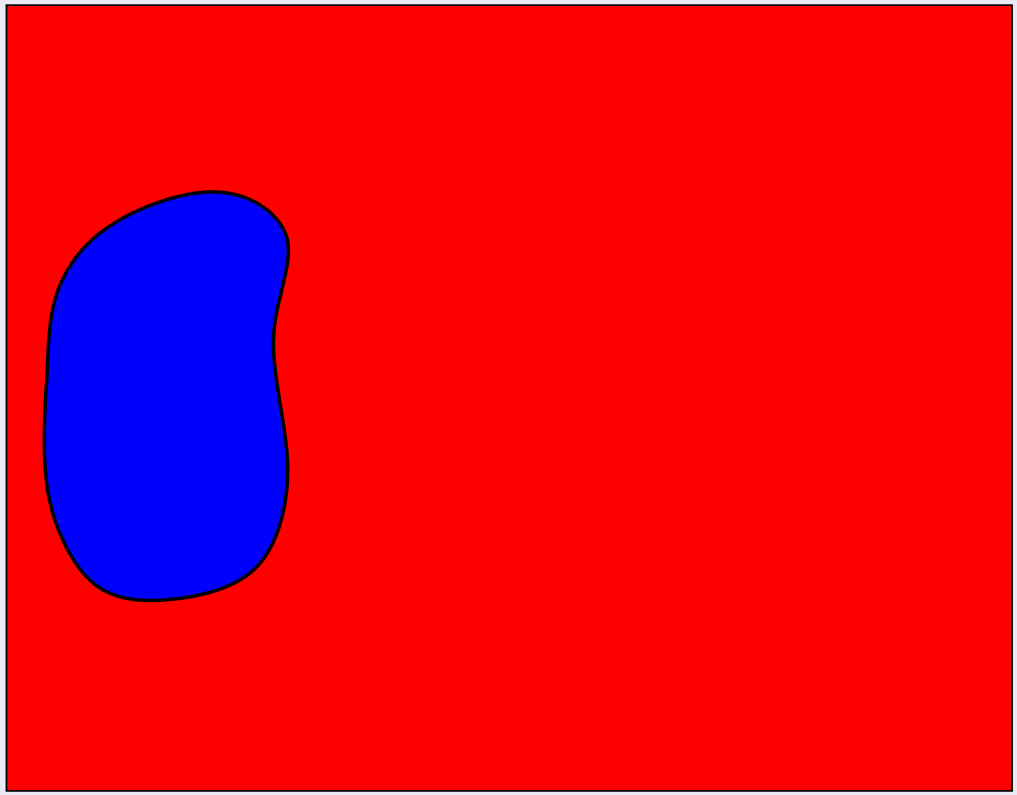
- interpret the image as an 8-neighborhood graph,
- assign higher arc weights $w(s,t)$ across the object's boundary than inside and outside it, and
- define the value of a path to be the maximum arc weight along it, such that any path that crosses the boundary will be penalized.

# The IFT computation



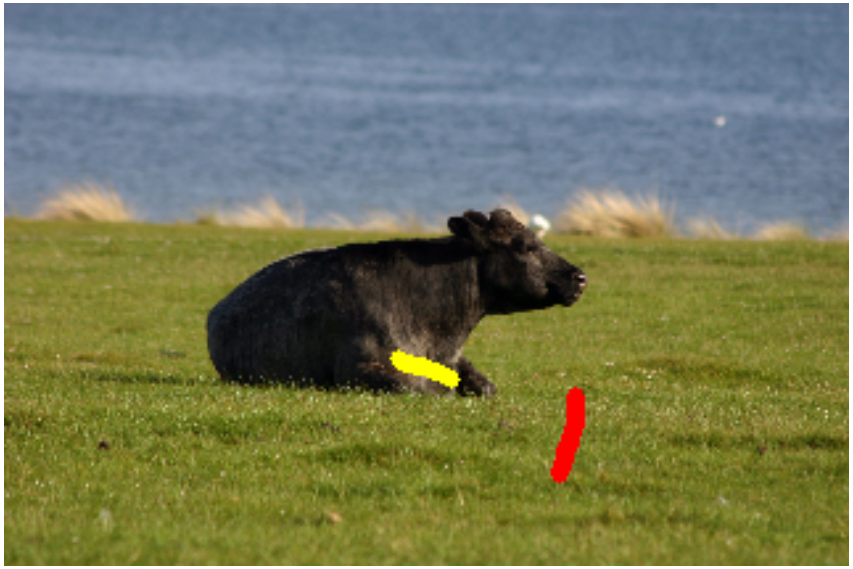

Seeds $a$ and $b$ compete between them, but $b$ conquers all spels $t$ in component $Y$ because it will be surrounded by optimum paths rooted at $b$.

# The IFT computation



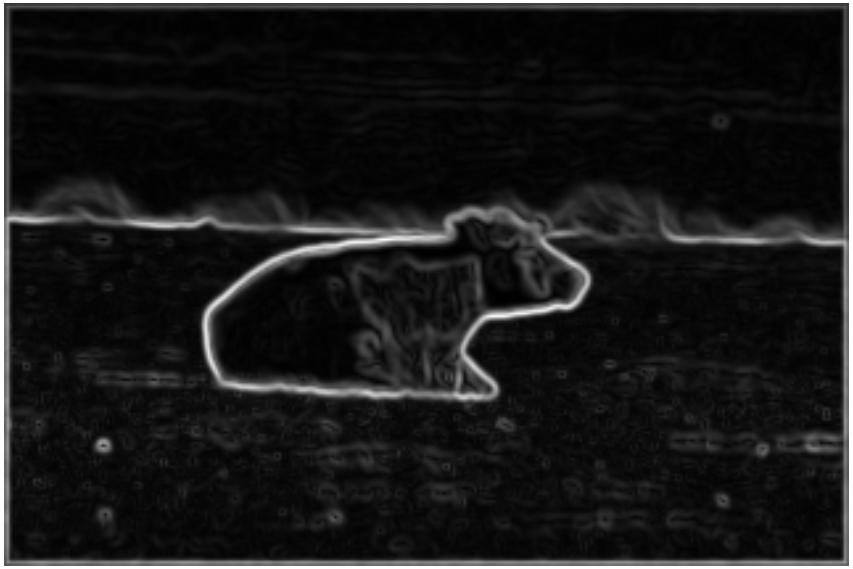
Seeds $a$ and $b$ compete between them, but $b$ conquers all spels $t$ in component $Y$ because it will be surrounded by optimum paths rooted at $b$.

# Connectivity functions



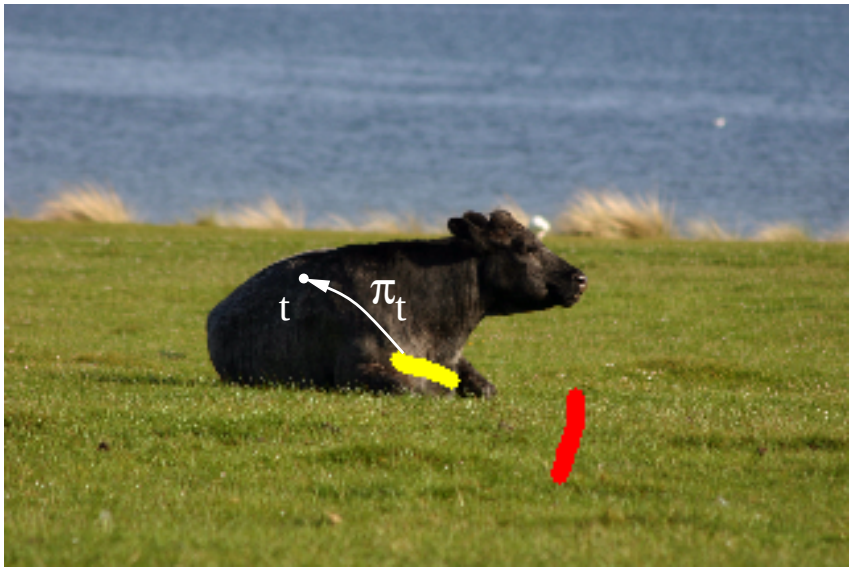
- Image with internal and external markers.

# Connectivity functions



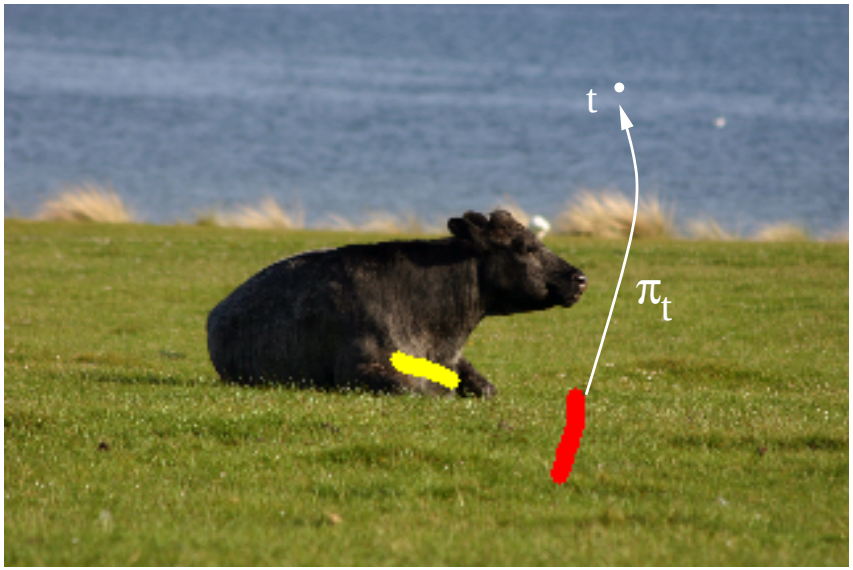
- Image with internal and external markers.
- Arc-weight image.

# Connectivity functions




- Image with internal and external markers.
- Arc-weight image.
- Optimum-paths to foreground pixels.

# Connectivity functions




- Image with internal and external markers.
- Arc-weight image.
- Optimum-paths to foreground pixels.
- Optimum-paths to background pixels.

# Connectivity functions

- Image with internal and external markers.
- Arc-weight image.
- Optimum-paths to foreground pixels.
- Optimum-paths to background pixels.
- Segmentation result.

# Connectivity functions

show video-iftsc.gif

- Image with internal and external markers.
- Arc-weight image.
- Optimum-paths to foreground pixels.
- Optimum-paths to background pixels.
- Segmentation result.

# Connectivity functions

More formally,

# Connectivity functions

More formally,

- a path $\pi_t = \pi_s \cdot \langle s, t \rangle$ is the extension of a path $\pi_s$ by an arc $(s, t)$, being $\pi_t = \langle t \rangle$ a trivial path, and

# Connectivity functions

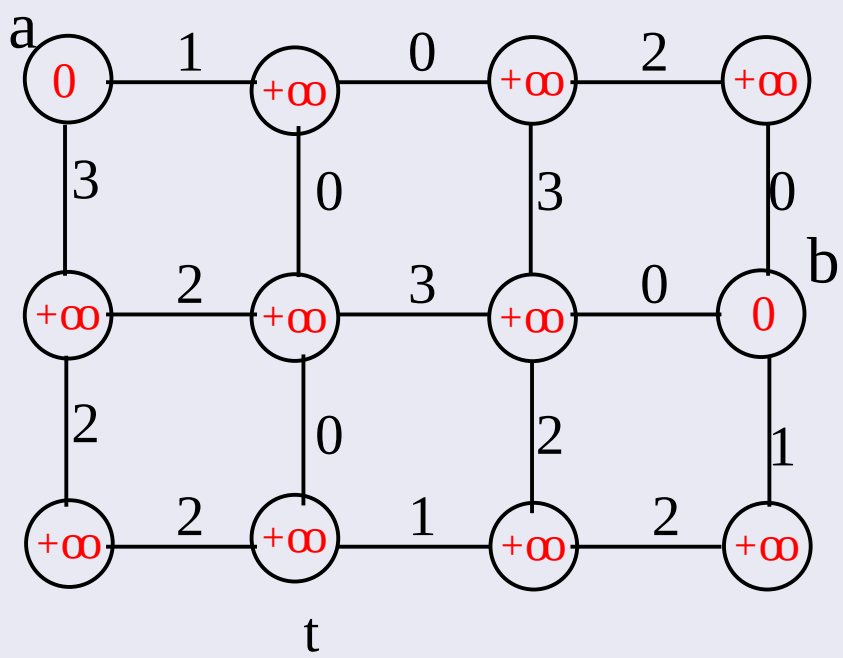
More formally,

- a path $\pi_t = \pi_s \cdot \langle s, t \rangle$ is the extension of a path $\pi_s$ by an arc $(s, t)$, being $\pi_t = \langle t \rangle$ a trivial path, and
- the max-arc path function is defined as

$$
\begin{aligned}
f_{\max}(\langle t \rangle) &= \begin{cases} 0 & \text{if } t \in \mathcal{S} = \mathcal{S}_i \cup \mathcal{S}_e \\ +\infty & \text{otherwise} \end{cases} \\
f_{\max}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{\max}(\pi_s), w(s, t)\}.
\end{aligned}
$$

# Connectivity functions

Consider, for example, a 4-neighborhood graph, path function $f_{\max}$ and two seeds $\mathcal{S}_i = \{a\}$ and $\mathcal{S}_e = \{b\}$.

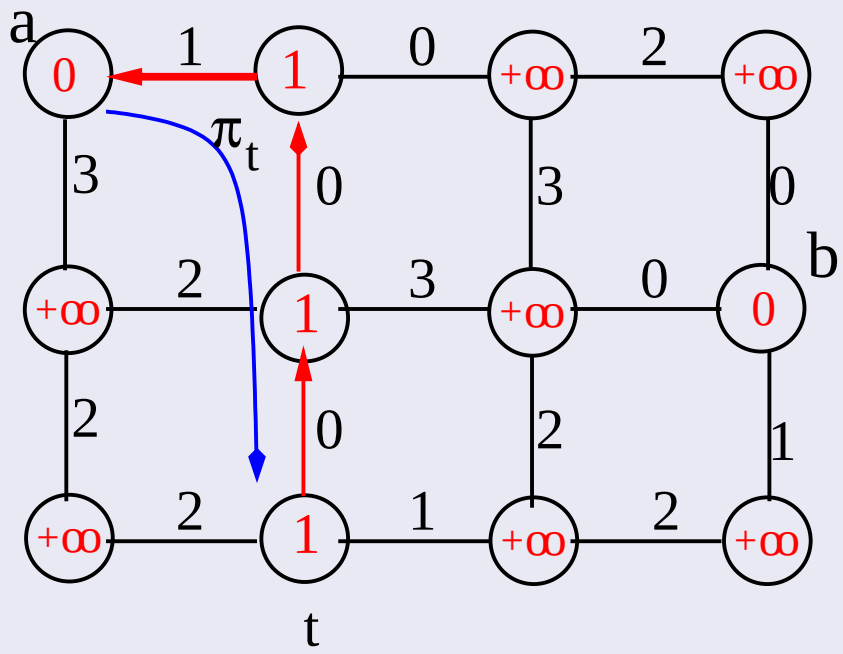



*4−neighborhood graph*

Paths are represented in backwards.

# Connectivity functions

Consider, for example, a 4-neighborhood graph, path function $f_{\max}$ and two seeds $\mathcal{S}_i = \{a\}$ and $\mathcal{S}_e = \{b\}$.

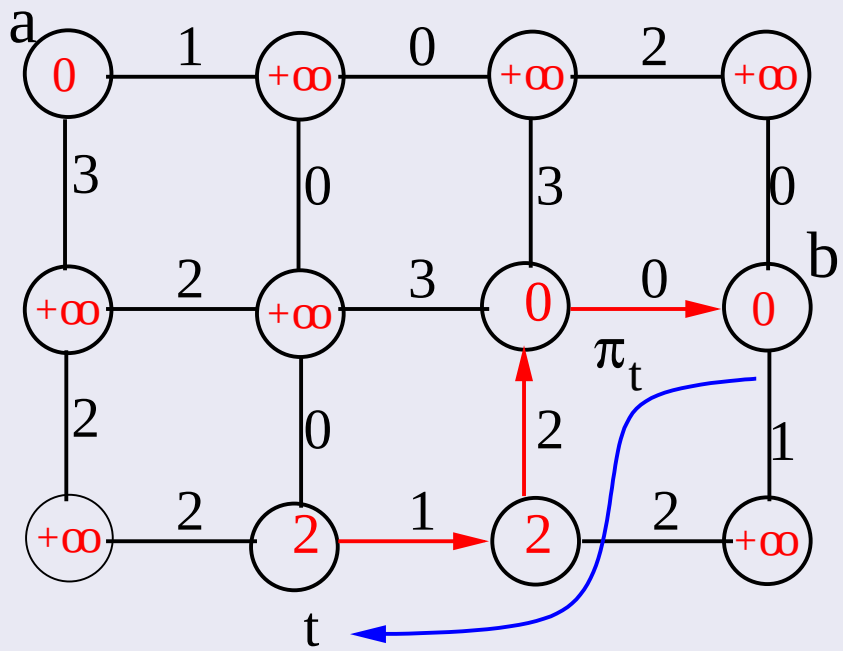



$$f_{max}(\pi_t) = 1 \quad \text{for} \quad R(\pi_t) = a$$

Paths are represented in backwards.

# Connectivity functions

Consider, for example, a 4-neighborhood graph, path function $f_{\max}$ and two seeds $\mathcal{S}_i = \{a\}$ and $\mathcal{S}_e = \{b\}$.




$$f_{max}(\pi_t) = 2 \quad \text{for} \quad R(\pi_t) = b$$

Paths are represented in backwards.

# Connectivity functions

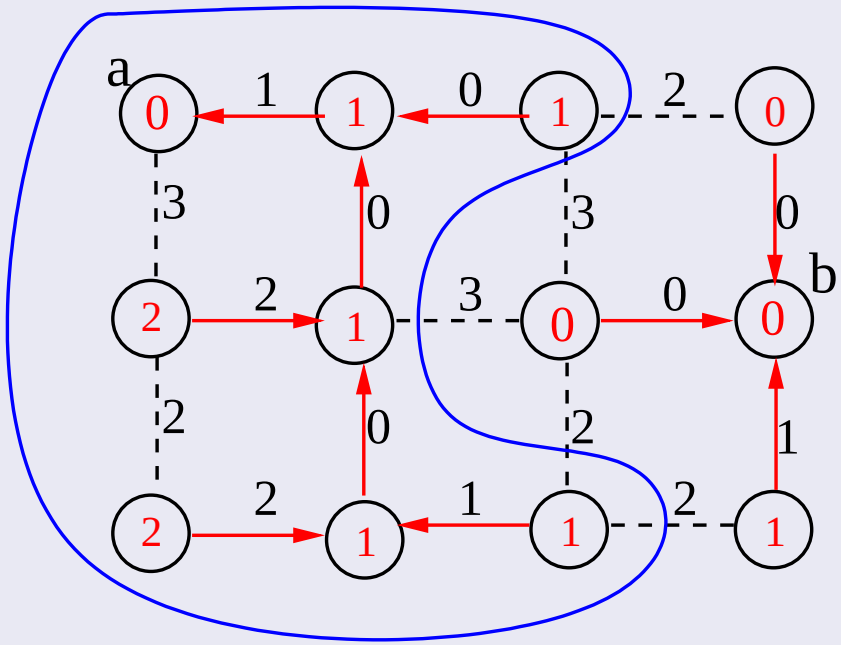
The segmentation essentially minimizes a connectivity map

$$V(t) = \min_{\forall \pi_t \in \Pi(\mathcal{N},\mathcal{A},t)} \{f_{\max}(\pi_t)\}$$

by considering the set $\Pi(\mathcal{N}, \mathcal{A}, t)$ of all paths with terminus $t$ and function $f_{\max}$.

# Connectivity function

The IFT algorithm solves this problem by computing an optimum-path forest $P$ in $(\mathcal{N}, \mathcal{A})$ — a predecessor map with no cycles, containing all optimum paths from a root set $\mathcal{R}$, which in this case is $\mathcal{S} = \mathcal{S}_i \cup \mathcal{S}_e$.



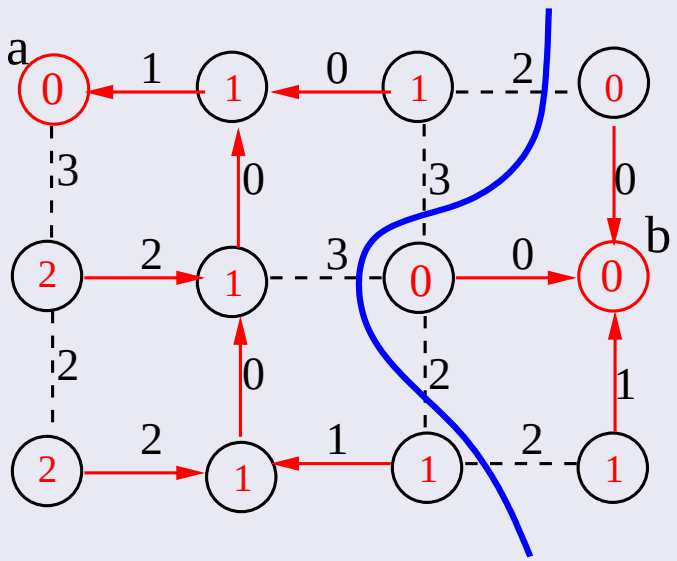

The object is defined by the optimum forest for $f_{\max}$ rooted in $\mathcal{S}_i$.

# Connectivity function

An optimum-path forest for $f_{\max}$ also provides the graph cut whose minimum arc weight

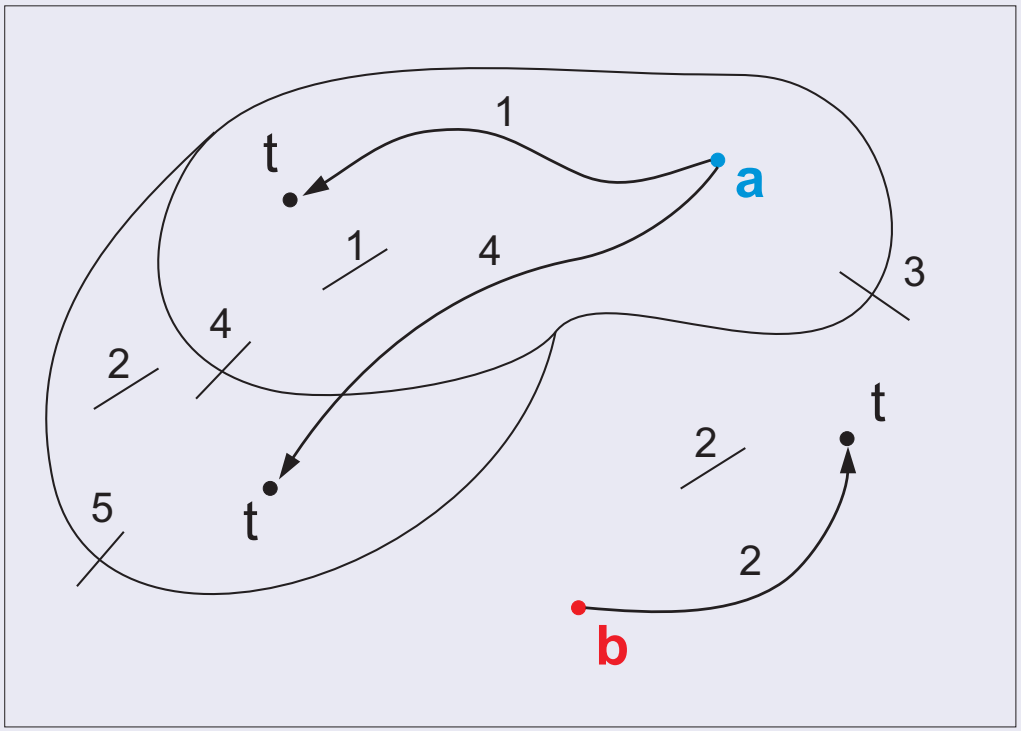
$$\min_{\forall (s,t) \in \mathcal{A}, R(\pi_s)=a, R(\pi_t)=b} w(s,t)$$

is maximum, considering all possible cuts between $a$ and $b$ [5, 7].



# Connectivity function

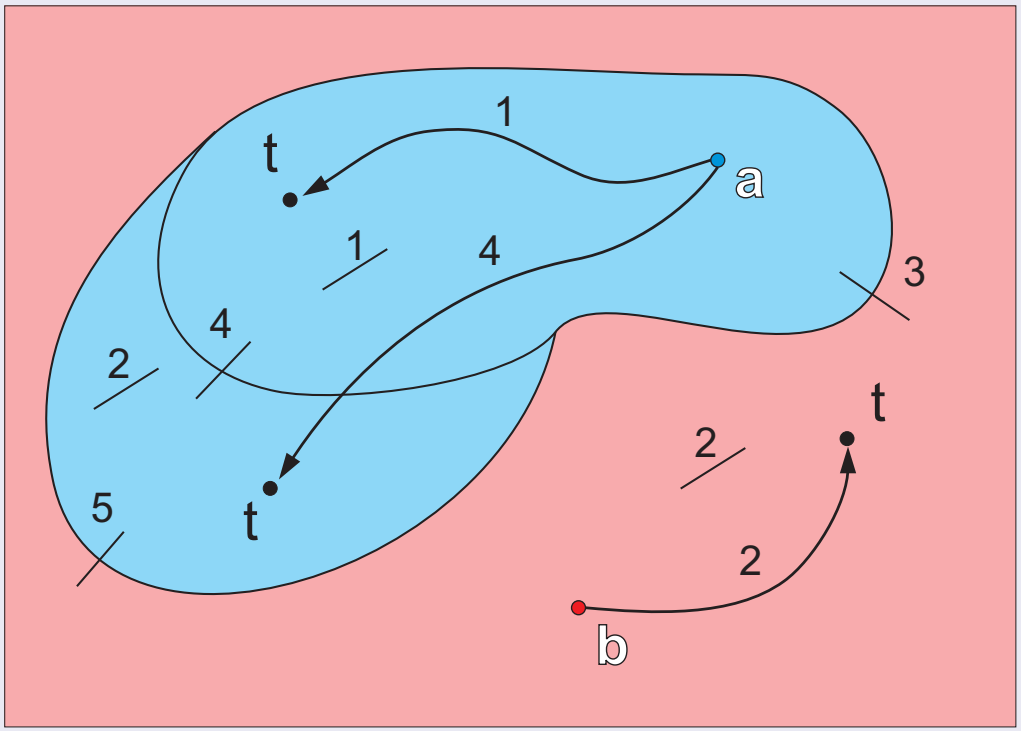
Indeed, the graph cut obtained with $f_{max}$ is piecewise optimum[5].




The boundary segment with arc weights equal to 5 has preference over the segment with weight 4, due to the piecewise optimum property. Note that both solutions lead to the same maximum cut with minimum arc weight 3.

# Connectivity function

Indeed, the graph cut obtained with $f_{max}$ is piecewise optimum[5].




The boundary segment with arc weights equal to 5 has preference over the segment with weight 4, due to the piecewise optimum property. Note that both solutions lead to the same maximum cut with minimum arc weight 3.

# Image Foresting Transform

Therefore, for suitable adjacency relation and connectivity function, the IFT essentially reduces a given image processing problem into the computation of an optimum-path forest followed by a local processing of its attributes:

# Image Foresting Transform

Therefore, for suitable adjacency relation and connectivity function, the IFT essentially reduces a given image processing problem into the computation of an optimum-path forest followed by a local processing of its attributes:

- the connectivity map $V(t)$,

# Image Foresting Transform

Therefore, for suitable adjacency relation and connectivity function, the IFT essentially reduces a given image processing problem into the computation of an optimum-path forest followed by a local processing of its attributes:

- the connectivity map $V(t)$,
- the optimum paths in $P(t)$, and

# Image Foresting Transform

Therefore, for suitable adjacency relation and connectivity function, the IFT essentially reduces a given image processing problem into the computation of an optimum-path forest followed by a local processing of its attributes:

- the connectivity map $V(t)$,
- the optimum paths in $P(t)$, and
- a root label $L(t)$.

# The IFT computation

- First, all nodes $t \in \mathcal{N}$ are trivial paths with initial connectivity values $V_0(t) = f(\langle t \rangle)$.

# The IFT computation

- First, all nodes $t \in \mathcal{N}$ are trivial paths with initial connectivity values $V_0(t) = f(\langle t \rangle)$.
- The initial roots are identified at the global minima of $V_0(t)$.

# The IFT computation

- First, all nodes $t \in \mathcal{N}$ are trivial paths with initial connectivity values $V_0(t) = f(\langle t \rangle)$.
- The initial roots are identified at the global minima of $V_0(t)$.
- They may conquer their adjacent nodes by offering them better paths.

# The IFT computation

- First, all nodes $t \in \mathcal{N}$ are trivial paths with initial connectivity values $V_0(t) = f(\langle t \rangle)$.
- The initial roots are identified at the global minima of $V_0(t)$.
- They may conquer their adjacent nodes by offering them better paths.
- The process continues from the adjacent nodes in a non-decreasing order of path values.

$$\text{if } f(\pi_s \cdot \langle s, t \rangle) < f(\pi_t) \quad \text{then } \pi_t \leftarrow \pi_s \cdot \langle s, t \rangle.$$

# The IFT computation

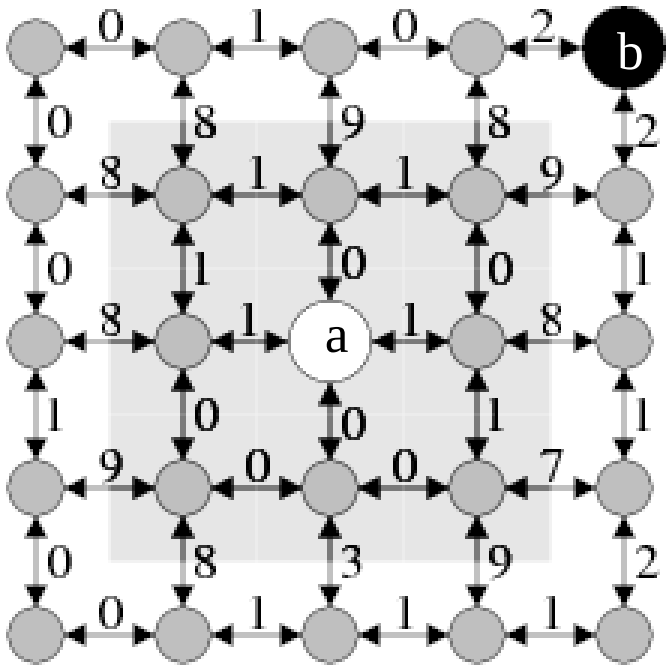
- First, all nodes $t \in \mathcal{N}$ are trivial paths with initial connectivity values $V_0(t) = f(\langle t \rangle)$.
- The initial roots are identified at the global minima of $V_0(t)$.
- They may conquer their adjacent nodes by offering them better paths.
- The process continues from the adjacent nodes in a non-decreasing order of path values.

$$\text{if } f(\pi_s \cdot \langle s,t \rangle) < f(\pi_t) \quad \text{then } \pi_t \leftarrow \pi_s \cdot \langle s,t \rangle.$$

- Essentially the minima in $V_0(t)$ compete among themselves and some of them become roots in $\mathcal{R}$, being also minima in $V(t)$. show video-iftsc-computation.gif

# Path propagation

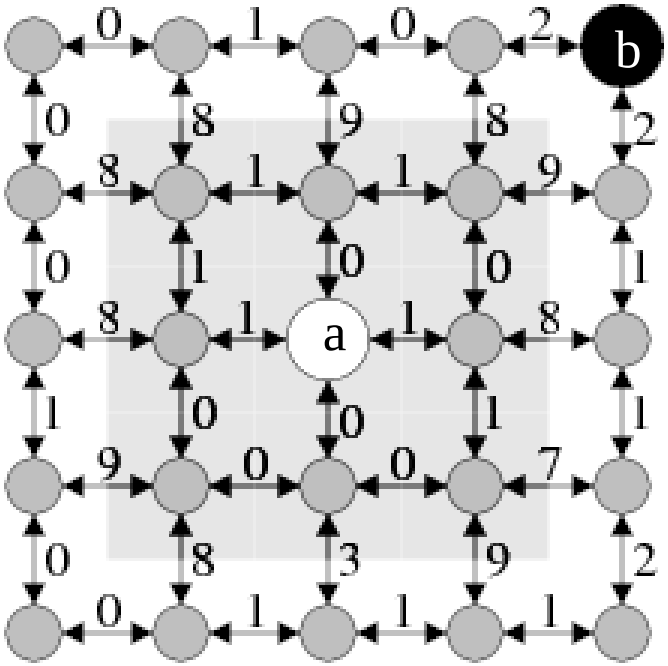
Consider the optimum path propagation for $f_{\max}$ from $\mathcal{S} = \{a, b\}$ in the 4-neighborhood graph below.

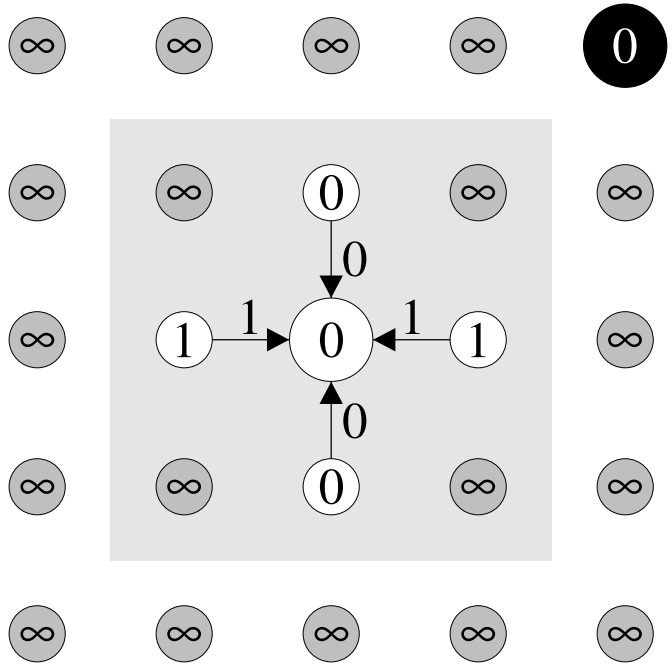



Object and background are separated by the arcs with higher weights.

# Path propagation

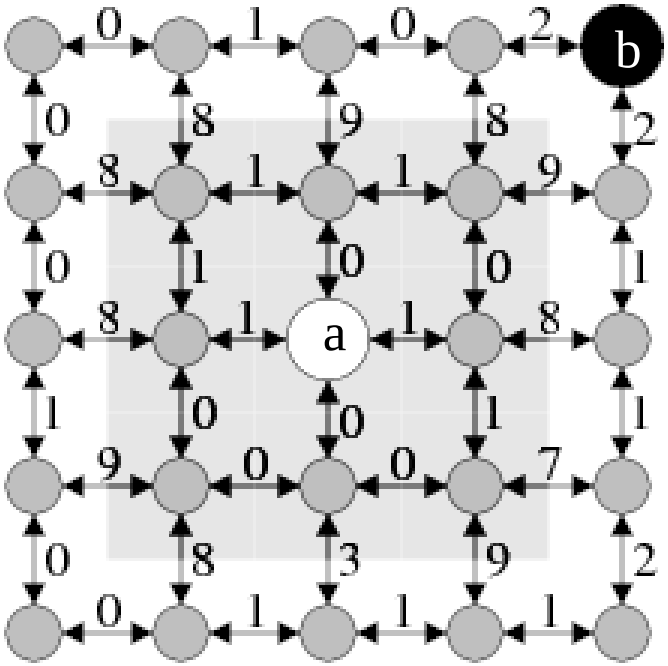
From iteration 1 to 5, iteration 12, 20, and 25.

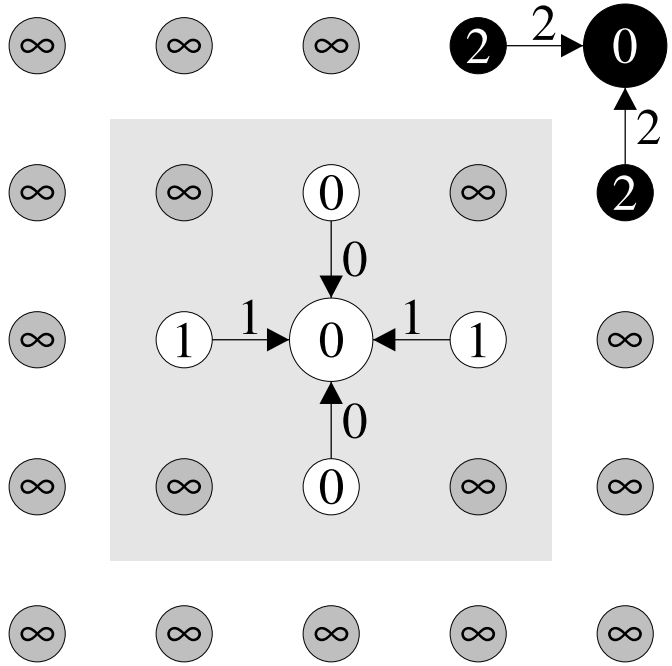


# Path propagation

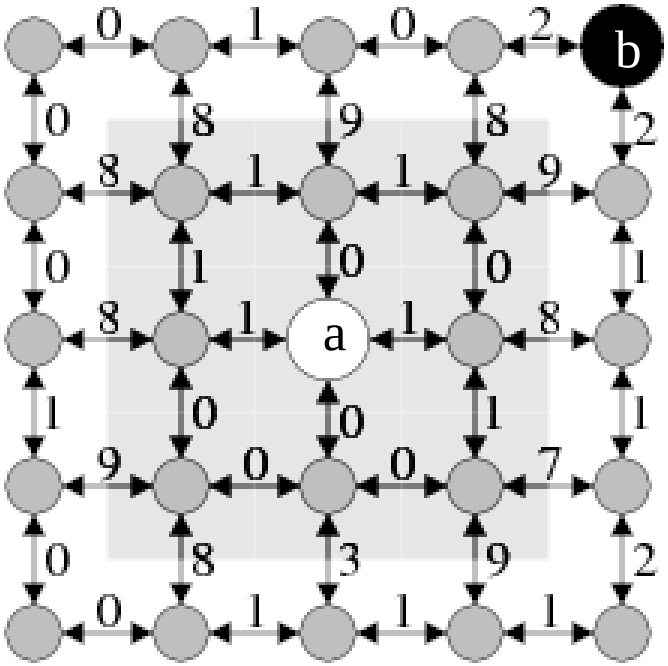
From iteration 1 to 5, iteration 12, 20, and 25.

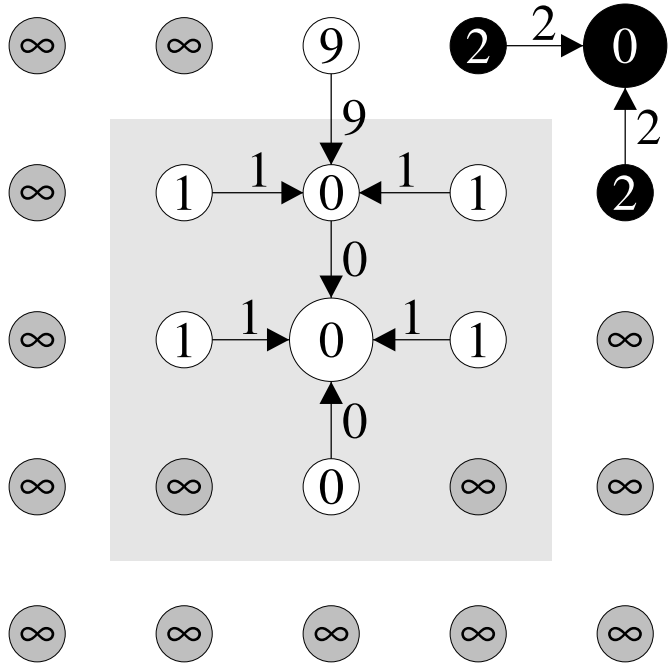


# Path propagation

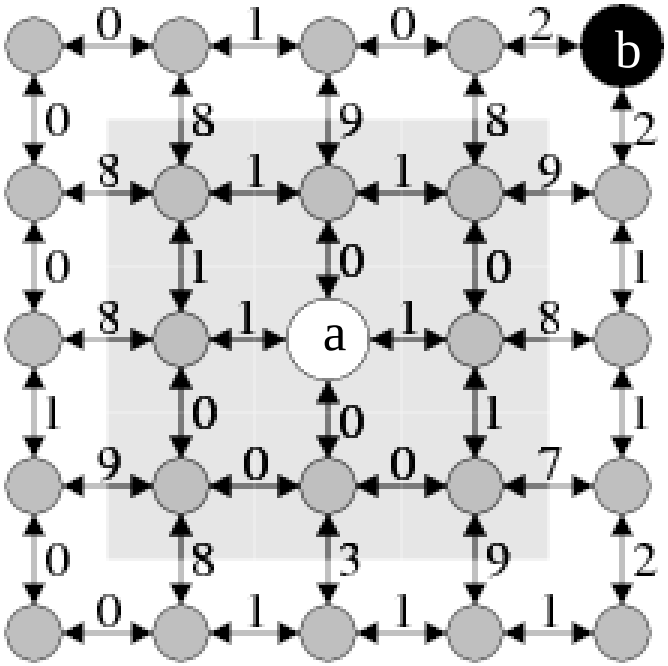
From iteration 1 to 5, iteration 12, 20, and 25.

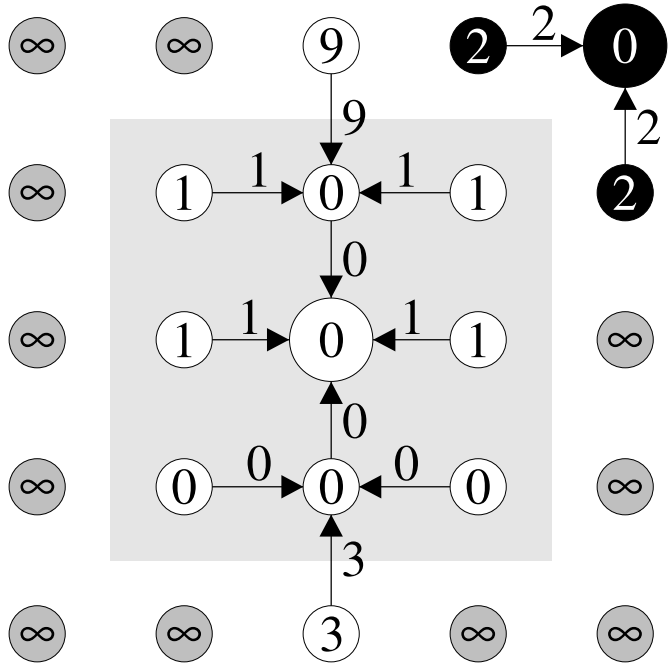


# Path propagation

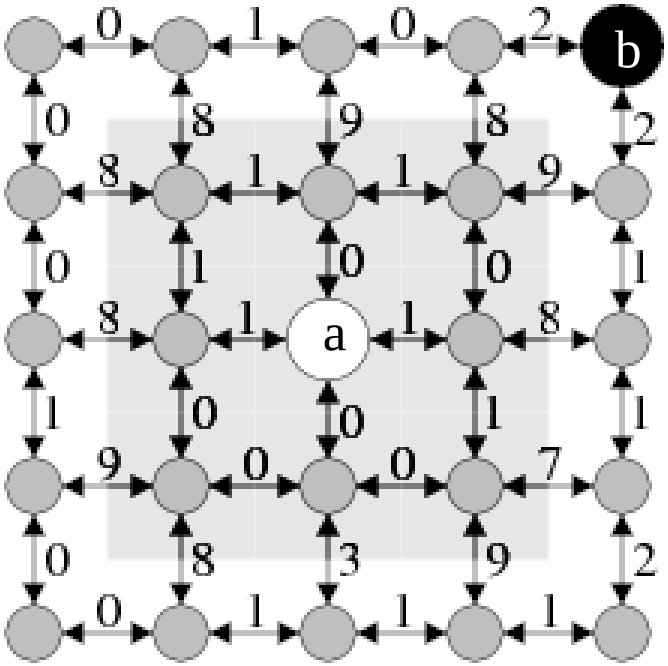
From iteration 1 to 5, iteration 12, 20, and 25.

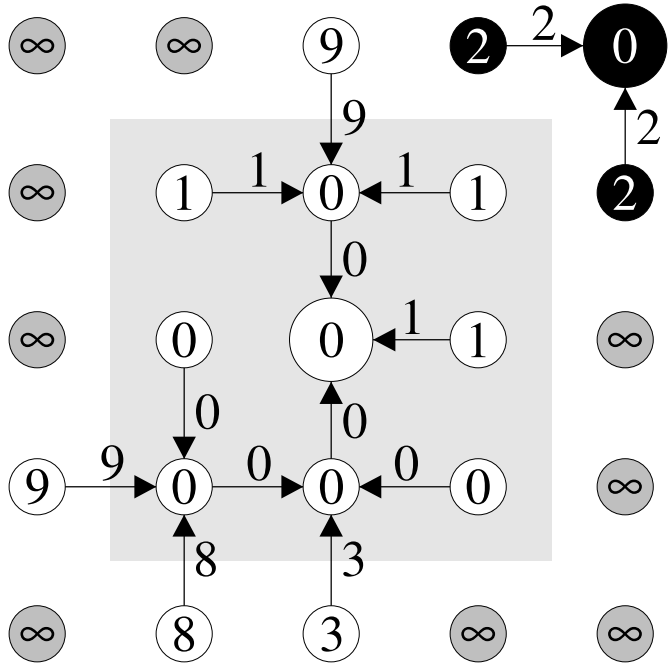


# Path propagation

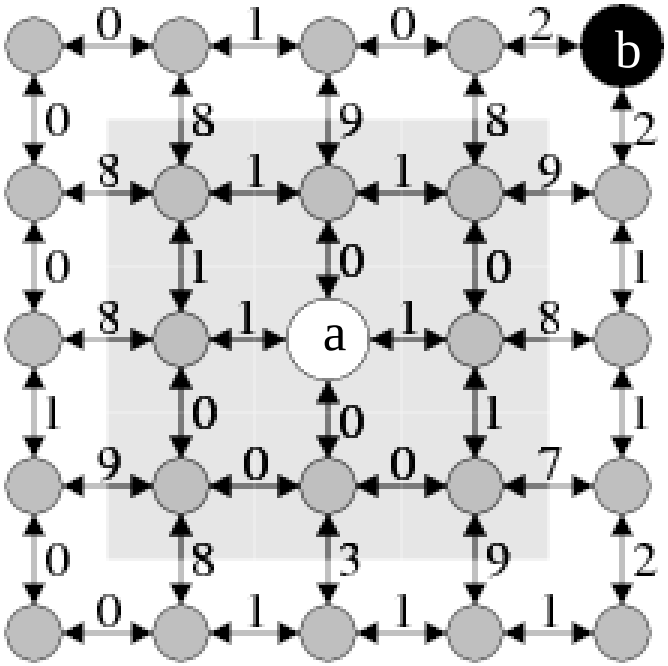
From iteration 1 to 5, iteration 12, 20, and 25.

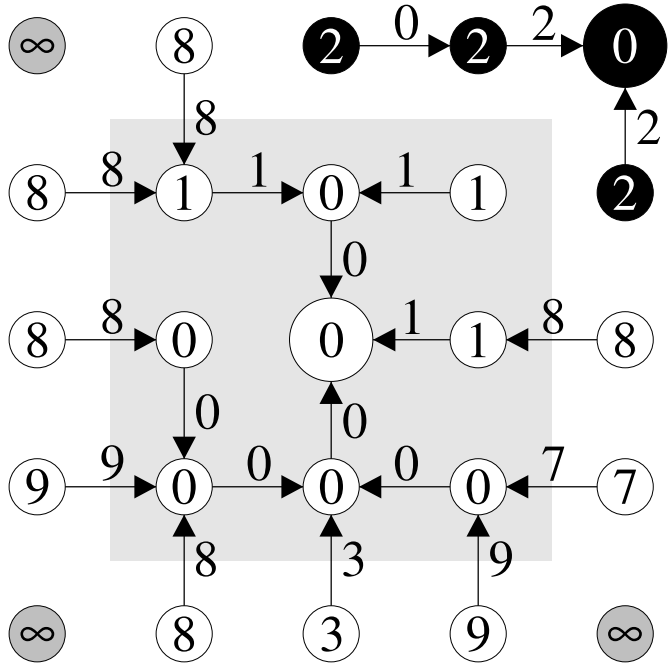


# Path propagation

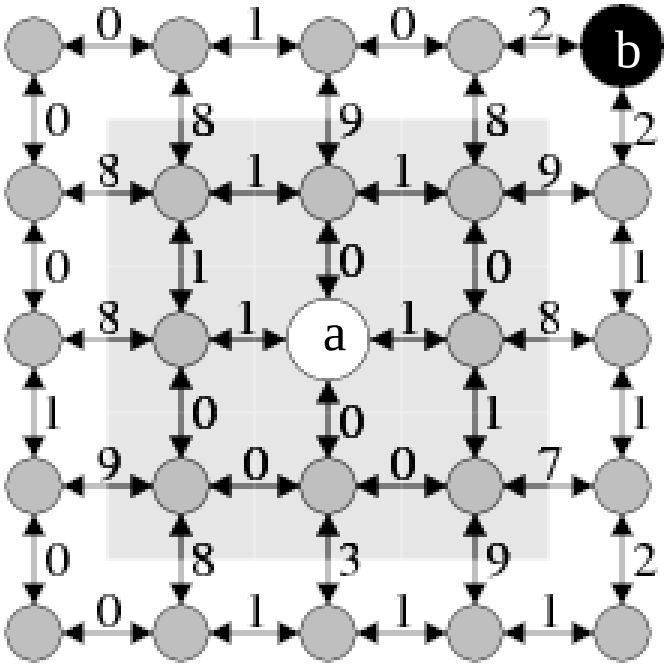
From iteration 1 to 5, iteration 12, 20, and 25.

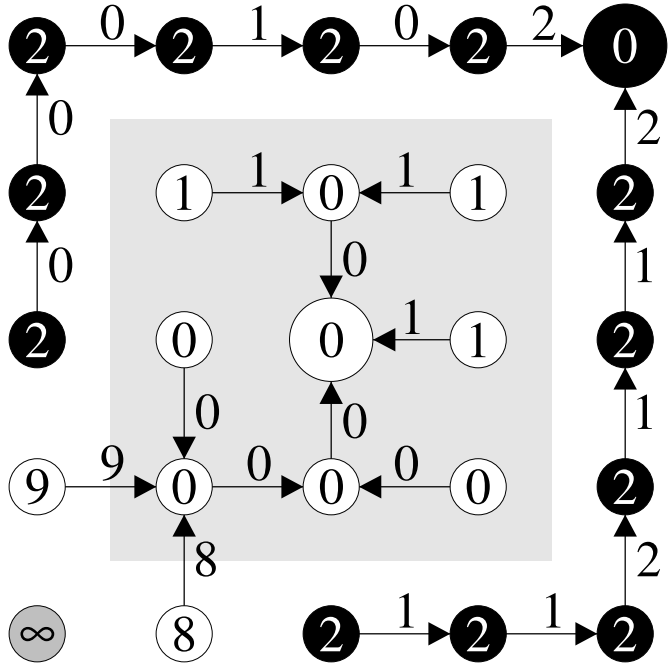


# Path propagation

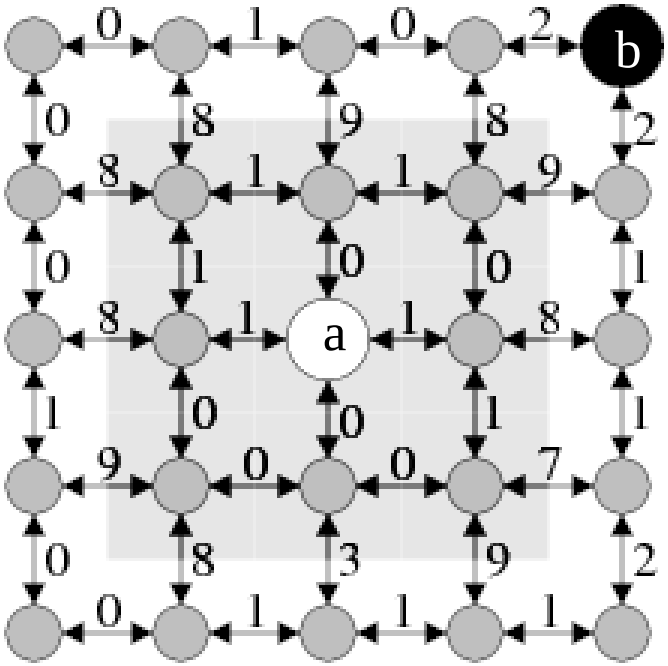
From iteration 1 to 5, iteration 12, 20, and 25.

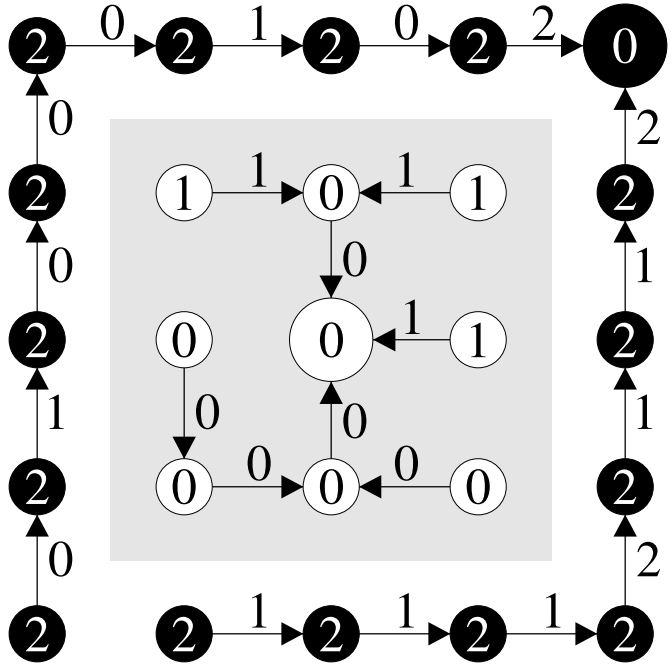


# Path propagation

From iteration 1 to 5, iteration 12, 20, and 25.



# Information propagation

- The IFT requires a priority queue $Q$ for path propagation (modified Dijkstra's algorithm).

# Information propagation

- The IFT requires a priority queue $Q$ for path propagation (modified Dijkstra's algorithm).
- It can propagate other informations to each node:

# Information propagation

- The IFT requires a priority queue $Q$ for path propagation (modified Dijkstra's algorithm).
- It can propagate other informations to each node:
  - its propagation order [24],

# Information propagation

- The IFT requires a priority queue $Q$ for path propagation (modified Dijkstra's algorithm).
- It can propagate other informations to each node:
  - its propagation order [24],
  - a graph-cut measure [25], etc.

# Information propagation

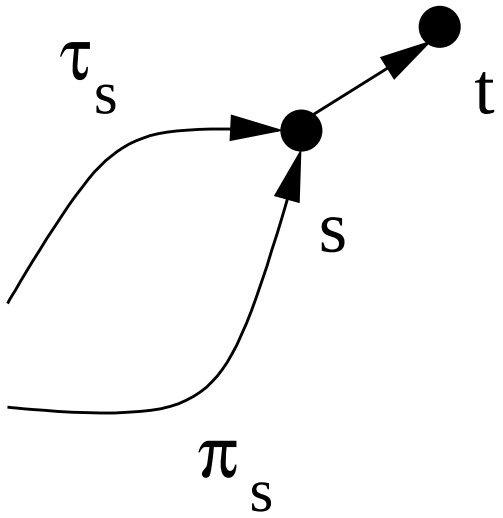
- The IFT requires a priority queue $Q$ for path propagation (modified Dijkstra's algorithm).
- It can propagate other informations to each node:
  - its propagation order [24],
  - a graph-cut measure [25], etc.
- Its correctness requires smooth connectivity functions.

# Information propagation

A path function is smooth when for every node $t$, there exists at least one optimum path $\pi_t$ which is either trivial or has the form $\pi_s \cdot \langle s, t \rangle$ where:

1. $f(\pi_s) \leq f(\pi_t)$.
2. $\pi_s$ is optimum.
3. For any optimum path $\tau_s$, $f(\tau_s \cdot \langle s, t \rangle) = f(\pi_t)$.

These conditions are applied to only optimum paths.



# The general IFT algorithm

## Algorithm

– General IFT algorithm

1. For each $t \in \mathcal{N}$, do
2. $\quad$ Set $P(t) \leftarrow nil$, $L(t) \leftarrow t$ and $V(t) \leftarrow f(\langle t \rangle)$.
3. $\quad$ If $V(t) \neq +\infty$, then insert $t$ in $Q$.
4. While $Q$ is not empty, do
5. $\quad$ Remove from $Q$ a spel $s$ such that $V(s)$ is minimum.
6. $\quad$ For each $t \in \mathcal{A}(s)$ such that $V(t) > V(s)$, do
7. $\quad\quad$ Compute $tmp \leftarrow f(\pi_s \cdot \langle s, t \rangle)$.
8. $\quad\quad$ If $tmp < V(t)$, then
9. $\quad\quad\quad$ If $V(t) \neq +\infty$, remove $t$ from $Q$.
10. $\quad\quad\quad$ Set $P(t) \leftarrow s$, $V(t) \leftarrow tmp$, $L(t) \leftarrow L(s)$.
11. $\quad\quad\quad$ Insert $t$ in $Q$.

# Important remarks and variants

- The queue $Q$ is then a wavefront whose propagation from each root follows a non-decreasing order of path values.

# Important remarks and variants

- The queue $Q$ is then a wavefront whose propagation from each root follows a non-decreasing order of path values.
- Ties are broken in $Q$, by following the FIFO policy. A variant of the algorithm with LIFO tie-breaking policy is given in [1].

# Important remarks and variants

- The queue $Q$ is then a wavefront whose propagation from each root follows a non-decreasing order of path values.
- Ties are broken in $Q$, by following the FIFO policy. A variant of the algorithm with LIFO tie-breaking policy is given in [1].
- Other root information in a function $\lambda(t)$ can be propagated by setting $L(t) \leftarrow \lambda(t)$ in Line 2.

# Important remarks and variants

- The queue $Q$ is then a wavefront whose propagation from each root follows a non-decreasing order of path values.
- Ties are broken in $Q$, by following the FIFO policy. A variant of the algorithm with LIFO tie-breaking policy is given in [1].
- Other root information in a function $\lambda(t)$ can be propagated by setting $L(t) \leftarrow \lambda(t)$ in Line 2.
- Line 7 can be simplified to $tmp \leftarrow \max\{V(s), w(s,t)\}$ in the case of $f_{\max}$.

# Important remarks and variants

- The queue $Q$ is then a wavefront whose propagation from each root follows a non-decreasing order of path values.
- Ties are broken in $Q$, by following the FIFO policy. A variant of the algorithm with LIFO tie-breaking policy is given in [1].
- Other root information in a function $\lambda(t)$ can be propagated by setting $L(t) \leftarrow \lambda(t)$ in Line 2.
- Line 7 can be simplified to $tmp \leftarrow \max\{V(s), w(s,t)\}$ in the case of $f_{\max}$.
- The dual operation $V(t) = \max_{\forall \pi_t \in \Pi(\mathcal{N},\mathcal{A},t)}\{f_{\min}(\pi_t)\}$ requires: $V(t) \neq -\infty$ in Lines 3 and 9, $V(s)$ is maximum in Line 5, $V(t) < V(s)$ in Line 6, and $tmp > V(t)$ in Line 8.

# Important remarks and variants

When $s$ is removed from $Q$ in Line 5:

# Important remarks and variants

When $s$ is removed from $Q$ in Line 5:

- Path $\pi_s$ is optimum, it can be obtained from $P$, which also contains all optimum paths $\pi_r$ with values $V(r) < V(s)$.

# Important remarks and variants

When $s$ is removed from $Q$ in Line 5:

- Path $\pi_s$ is optimum, it can be obtained from $P$, which also contains all optimum paths $\pi_r$ with values $V(r) < V(s)$.
- Besides that, if $P(s) = nil$ then $s$ is a root node of $\mathcal{R}$, found on-the-fly.

# Important remarks and variants

When $s$ is removed from $Q$ in Line 5:

- Path $\pi_s$ is optimum, it can be obtained from $P$, which also contains all optimum paths $\pi_r$ with values $V(r) < V(s)$.
- Besides that, if $P(s) = nil$ then $s$ is a root node of $\mathcal{R}$, found on-the-fly.
- Early termination is possible whenever $s$ is a destination node [2] or when $V(s)$ is greater than a given threshold [24].

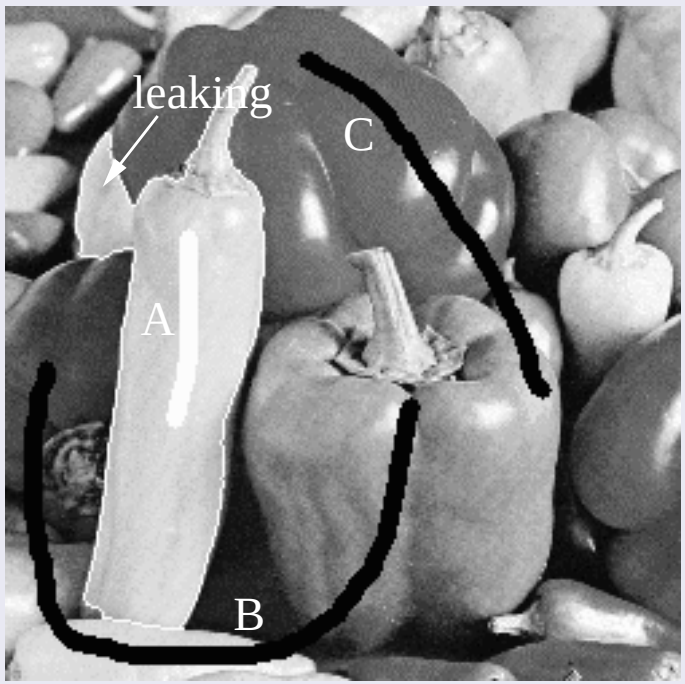
# Important remarks and variants

When $s$ is removed from $Q$ in Line 5:

- Path $\pi_s$ is optimum, it can be obtained from $P$, which also contains all optimum paths $\pi_r$ with values $V(r) < V(s)$.
- Besides that, if $P(s) = nil$ then $s$ is a root node of $\mathcal{R}$, found on-the-fly.
- Early termination is possible whenever $s$ is a destination node [2] or when $V(s)$ is greater than a given threshold [24].
- Later, whenever necessary, the remaining optimum paths $\pi_t$ with $V(t) \geq V(s)$ can be obtained incrementally from the nodes in $Q$.

# Important remarks and variants

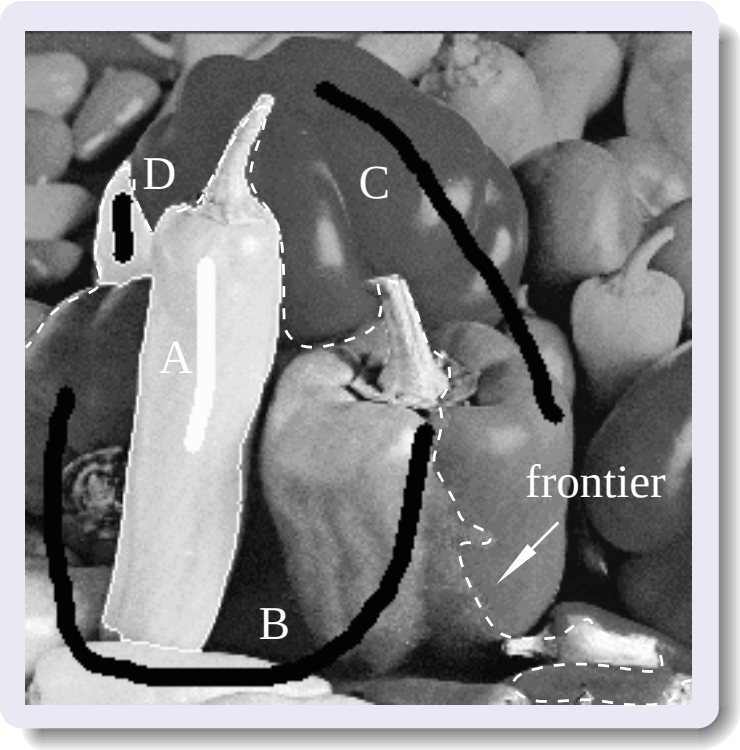
For a fixed path function $f$, the optimum forest can be updated in a differential way whenever we add/remove root nodes [3].




- Internal ($A$) and external ($B$ and $C$) markers are selected, but a "leaking" occurs.

# Important remarks and variants

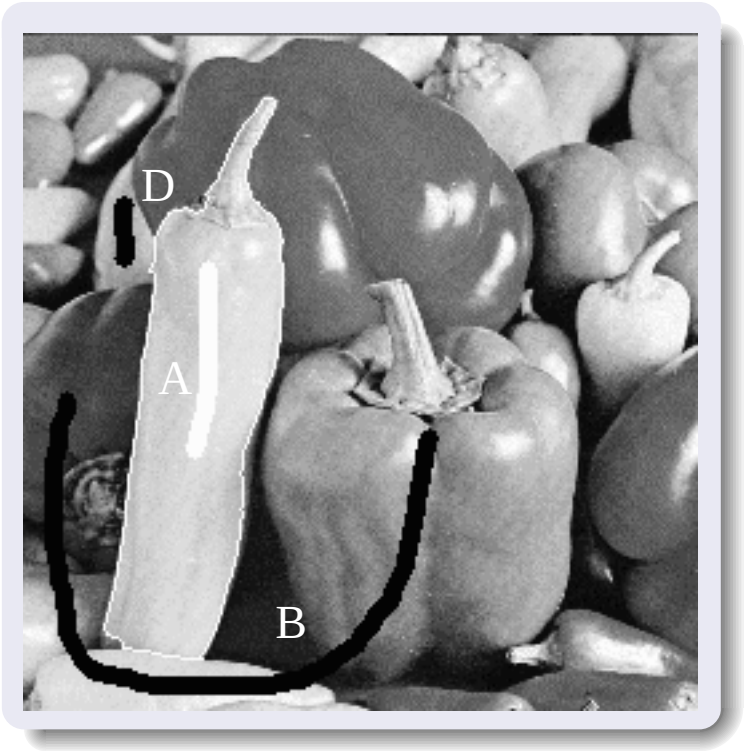
For a fixed path function $f$, the optimum forest can be updated in a differential way whenever we add/remove root nodes [3].




- Internal ($A$) and external ($B$ and $C$) markers are selected, but a "leaking" occurs.
- We add an external marker $D$ and select marker $C$ for removal. The competition involves $D$ and frontier spels (dashed line) of the forests of $A$ and $B$.

# Important remarks and variants

For a fixed path function $f$, the optimum forest can be updated in a differential way whenever we add/remove root nodes [3].




- Internal ($A$) and external ($B$ and $C$) markers are selected, but a "leaking" occurs.
- We add an external marker $D$ and select marker $C$ for removal. The competition involves $D$ and frontier spels (dashed line) of the forests of $A$ and $B$.
- Result of segmentation.

# Priority queue

- The IFT algorithm runs in $O(|\mathcal{A}| + |\mathcal{N}|^2)$.

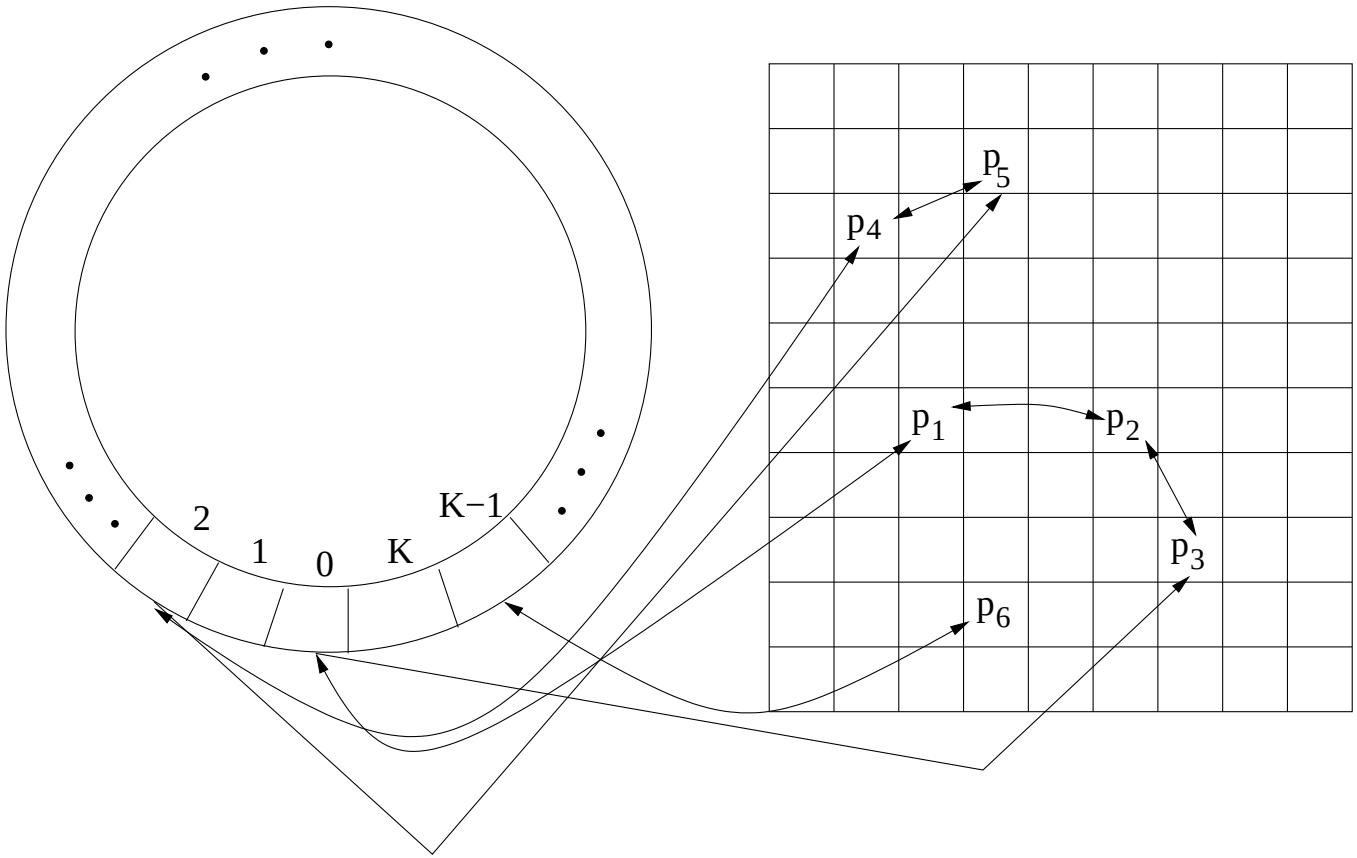
# Priority queue

- The IFT algorithm runs in $O(|\mathcal{A}| + |\mathcal{N}|^2)$.
- Its running time is $O(|\mathcal{N}| \log(|\mathcal{N}|))$, when $Q$ is a binary heap and the graph is sparse ($|\mathcal{A}(s)| \ll |\mathcal{N}|, \forall s \in \mathcal{N}$ ).

# Priority queue

- The IFT algorithm runs in $O(|\mathcal{A}| + |\mathcal{N}|^2)$.
- Its running time is $O(|\mathcal{N}| \log(|\mathcal{N}|))$, when $Q$ is a binary heap and the graph is sparse ($|\mathcal{A}(s)| \ll |\mathcal{N}|$, $\forall s \in \mathcal{N}$ ).
- Its running time is $O(|\mathcal{N}|)$, when $f(\pi_s \cdot \langle s, t \rangle) - f(\pi_s) \in [0..K]$, $K \ll |\mathcal{N}|$, are integers, the graph is sparse, and $Q$ uses bucket sort [2].

# Priority queue




Nodes $t$ are inserted in bucket $V(t)\%(K+1)$ (left), forming $K+1$ lists (right). The property $f(\pi_s \cdot \langle s,t \rangle) - f(\pi_s) \in [0..K]$ guarantees that nodes with different values are never in a same bucket.

# Conclusion

- It should be clear the advantages of interpreting images as graphs for the design of adjacency-based and connectivity-based image transformations.

# Conclusion

- It should be clear the advantages of interpreting images as graphs for the design of adjacency-based and connectivity-based image transformations.
- The IFT plays an important role in this scenario, given that it unifies image transformations based on optimum connectivity.

# Conclusion

- It should be clear the advantages of interpreting images as graphs for the design of adjacency-based and connectivity-based image transformations.
- The IFT plays an important role in this scenario, given that it unifies image transformations based on optimum connectivity.
- It requires an adjacency relation and a smooth connectivity function (path initialization and propagation rules).

# Conclusion

- It should be clear the advantages of interpreting images as graphs for the design of adjacency-based and connectivity-based image transformations.
- The IFT plays an important role in this scenario, given that it unifies image transformations based on optimum connectivity.
- It requires an adjacency relation and a smooth connectivity function (path initialization and propagation rules).
- It can be efficiently implemented in time proportional to the number of nodes for most image transformations.

# Conclusion

- It should be clear the advantages of interpreting images as graphs for the design of adjacency-based and connectivity-based image transformations.
- The IFT plays an important role in this scenario, given that it unifies image transformations based on optimum connectivity.
- It requires an adjacency relation and a smooth connectivity function (path initialization and propagation rules).
- It can be efficiently implemented in time proportional to the number of nodes for most image transformations.
- A demonstration C code is available at `http://code.google.com/p/ift-demo/`.

# Next lecture

- The Image Foresting Transform.
- Interactive and automatic segmentation methods.
- Clustering and classification.
- Connected filters.

Thanks for your attention

[1] A.X. Falcão, J. Stolfi, and R.A. Lotufo.

The image foresting transform: Theory, algorithms, and applications.

*IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.

[2] A.X. Falcão, J.K. Udupa, and F.K. Miyazawa.

An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly.

*IEEE Trans. on Medical Imaging*, 19(1):55–62, Jan 2000.

[3] A. X. Falcão and F. P. G. Bergo.

Interactive volume segmentation with differential image foresting transforms.

*IEEE Trans. on Medical Imaging*, 23(9):1100–1108, 2004.

[4] A.X. Falcão, B. S. da Cunha, and R. A. Lotufo.

Design of connected operators using the image foresting transform.

In *SPIE on Medical Imaging*, volume 4322, pages 468–479, Feb 2001.

[5] P.A.V. Miranda and A.X. Falcão.

Links between image segmentation based on optimum-path forest and minimum cut in graph.

*Journal of Mathematical Imaging and Vision*, 35(2):128–142, Oct 2009.

doi:10.1007/s10851-009-0159-9.

[6] R. Audigier and R.A. Lotufo.

Watershed by image foresting transform, tie-zone, and theoretical relationship with other watershed definitions.

In *Mathematical Morphology and its Applications to Signal and Image Processing (ISMM)*, pages 277–288, Rio de Janeiro, RJ, Oct 2007. MCT/INPE.

[7] K.C. Ciesielski, J.K. Udupa, A.X. Falcão, and P.A.V. Miranda.

A unifying graph-cut image segmentation framework: algorithms it encompasses and equivalences among them.

In *SPIE on Medical Imaging: Image Processing*, volume 8314, page 12 pages, Feb 2012.

[8] K.C. Ciesielski, J.K. Udupa, A.X. Falcão, and P.A.V. Miranda.

Fuzzy connectedness image segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis.

*Journal of Mathematical Imaging and Vision*, 2012.

[9] F. Cappabianco, G. Araujo, R. Azevedo, and A. Falcão.

A general image processing architecture for fpga.

In *V Southern Programmable Logic Conference*, pages 27–32, São Carlos, SP, 2009. Rima.

[10] Y. Zhuge, J. K. Udupa, K. C. Ciesielski, A.X. Falão, P.A. V. Miranda, and R.W. Miller.

Gpu-based iterative relative fuzzy connectedness image segmentation.

In *SPIE on Medical Imaging: Image Processing*, volume 8316, page 11 pages, Feb 2012.

[11] A.X. Falcão, J.K. Udupa, S. Samarasekera, S. Sharma, B.E. Hirsch, and R.A. Lotufo.

User-steered image segmentation paradigms: Live wire and live lane.

*Graphical Models*, 60(4):233–260, Jul 1998.

[12] R.A. Lotufo and A.X. Falcão.

The ordered queue and the optimality of the watershed approaches.

In *Mathematical Morphology and its Applications to Image and Signal Processing (ISMM)*, volume 18, pages 341–350. Kluwer, Jun 2000.

[13] F.P.G. Bergo, A.X. Falcão, P.A.V. Miranda, and L.M. Rocha.

Automatic image segmentation by tree pruning.

*Journal of Mathematical Imaging and Vision*, 29(2–3):141–162, Nov 2007.

[14] P.A.V. Miranda, A.X. Falcao, and T.V. Spina.

The riverbed approach for user-steered image segmentation.

In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3133 –3136, Sep 2011.

[15] Thiago Vallin Spina, Alexandre Xavier Falcão, and Paulo André Vechiatto Miranda.

User-steered image segmentation using live markers.

In *Proceedings of the 14th international conference on Computer analysis of images and patterns - Volume Part I*, CAIP'11, pages 211–218. Springer-Verlag, 2011.

[16] A.X. Falcão, L.F. Costa, and B.S. da Cunha.

Multiscale skeletons by image foresting transform and its applications to neuromorphometry.

*Pattern Recognition*, 35(7):1571–1582, Apr 2002.

[17] R.S. Torres, A.X. Falcão, and L.F. Costa.

A graph-based approach for multiscale shape analysis.

*Pattern Recognition*, 37(6):1163–1174, 2004.

[18] R.S. Torres and A.X. Falcão.

Contour salience descriptors for effective image retrieval and analysis.

*Image and Vision Computing*, 25(1):3–13, Jan 2007.

[19] F.A. Andaló, P.A.V. Miranda, R. da S. Torres, and A.X.Falcão.

Shape feature extraction and description based on tensor scale.

*Pattern Recognition*, 43(1):26–36, Jan 2010.

[20] L.M. Rocha, F.A.M. Cappabianco, and A.X. Falcão.

Data clustering as an optimum-path forest problem with applications in image analysis.

*Intl. Journal of Imaging Systems and Technology*, 19(2):50–68, Jun 2009.

[21] João P. Papa, Alexandre X. Falcão, Victor Hugo C. de Albuquerque, and João Manuel R. S. Tavares.

Efficient supervised optimum-path forest classification for large datasets.

*Pattern Recognition*, 45(1):512–520, January 2012.

[22] A.T. da Silva, J.A. dos Santos, A.X. Falcão, R. da S. Torres, and L.P. Magalhães.

Incorporating multiple distance spaces in optimum-path forest classification to improve feedback-based learning.

*Computer Vision and Image Understanding*, 116(4):510–523, Apr 2012.

[23] A.T. da Silva, A.X. Falcão, and L.P. Magalhães.

Active learning paradigms for cbir systems based on optimum-path forest classification.

*Pattern Recognition*, 44(12):2971 – 2978, 2011.

[24] P.A.V. Miranda, A.X. Falcão, A. Rocha, and F.P.G. Bergo.

Object delineation by $\kappa$-connected components.

*EURASIP Journal on Advances in Signal Processing*, 2008.

[25] A.X. Falcão, P.A.V. Miranda, and A. Rocha.

A linear-time approach for image segmentation using graph-cut measures.

In *8th Intl. Conf. on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, volume LNCS 4179, pages 138–149, Antwerp, Belgium, 2006. Springer.