Advanced Image Processing and Image Segmentation Techniques – Segmentation

Joanna Czajkowska, PhD
Media Systems Group
Institute for Vision and Graphics, University of Siegen

Fuzzy Connectedness Analysis

I. Introduction:

1. Fuzzy relation
2. Fuzzy digital space and paths
3. Fuzzy adjacency and affinity
4. Fuzzy connectedness

II. Fuzzy connectedness-based image segmentation:

1. Fuzzy objects
2. Starting points
3. Fuzzy connectedness scene
4. Multi-objects fuzzy connectedness

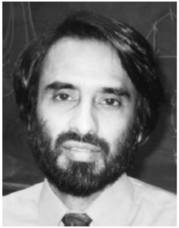
III. Implementation:

1. Graph searching using dynamic programming
2. Graph searching using Dijkstra algorithm
3. Matrix transformation-based graph analysis

Fuzzy connectedness

- The fuzzy connectedness (FC) theory is a branch of science using fuzzy logic, sets and relations.
- Authors: Jayaram Udupa and Supun Samarasekeraw (1996), Azriel Rosenfeld (1979) - Digital topology

Azriel Rosenfeld



Jayaram Udupa

Articles:

1. A. Rosenfeld, Digital Topology, American Mathematical Monthly, (86):621–630, 1979.
2. A. Rosenfeld, Fuzzy Digital Topology, Information and Control, 40(1): 76–87, 1979.
3. J. Udupa, S. Samarasekera, Fuzzy Connectedness and Object Definition: Theory, Algorithms, and Applications in Image Segmentation, Graphical Models and Image Processing, 58(3): 246–261, 1996.
4. P. Saha, J. Udupa, D. Odhner, Scale-Based Fuzzy Connected Image Segmentation: Theory, Algorithms, and Validation, Computer Vision and Image Understanding, 77(9): 145–174, 2000.
5. J. Udupa, P. Saha, Fuzzy Connectedness and Image Segmentation, Proceedings of the IEEE, 91: 1649–1669, 2003.

Fuzzy connectedness

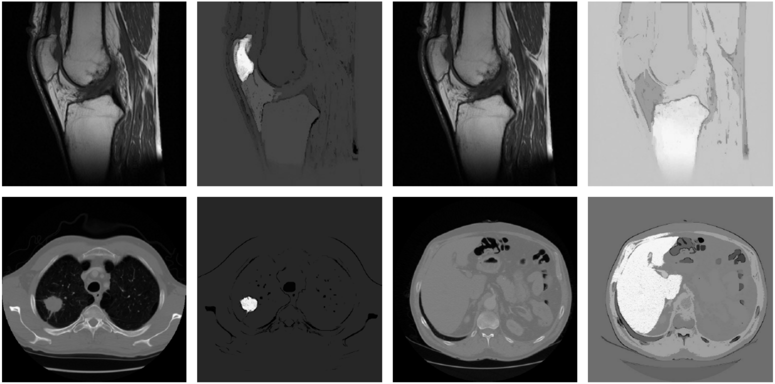
Motivation

- heterogeneity of intensity analysis
- heterogeneous object segmentation
- the analysis of ordered and connected data
- natural grouping of voxels

Applications

- image processing - image segmentation

Motivation - to segment the objects / create the fuzzy connectedness scene



Fuzzy connectedness

Methodology

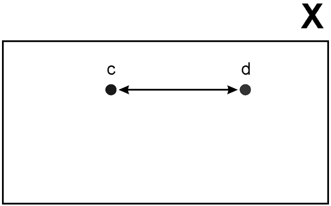
- operating on multidimensional and multifeature sets of ordered data
- points classified into single objects are strongly connected to each other by some more or less abstract relations
- their relations to points constituting other objects have relatively lower values

Fuzzy connectedness

Fuzzy connectedness = some fuzzy relation

Fuzzy relation ρ binds a pair of spels (spacial elements) c, d of some set $\mathbb{X}$ using the function $\mu_\rho(c, d) \in \langle 0, 1 \rangle$:

$$\rho = \{(c, d), \mu_\rho(c, d), \quad c, d \in \mathbb{X} \times \mathbb{X}\}$$

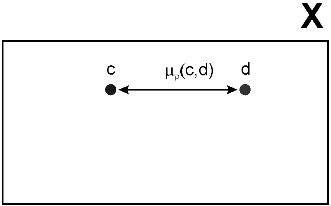


Fuzzy connectedness

Fuzzy connectedness = some fuzzy relation

Fuzzy relation ρ binds a pair of spels (spacial elements) c, d of some set $\mathbb{X}$ using the function $\mu_\rho(c,d) \in \langle 0, 1 \rangle$:

$$\rho = \{(c,d), \mu_\rho(c,d), \quad c,d \in \mathbb{X} \times \mathbb{X}\}$$

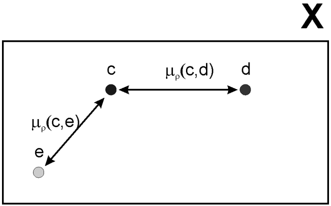


Fuzzy connectedness

Fuzzy connectedness = some fuzzy relation

Fuzzy relation ρ binds a pair of spels (spacial elements) c, d of some set $\mathbb{X}$ using the function $\mu_\rho(c, d) \in \langle 0, 1\rangle$:

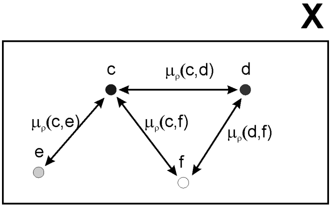
$$\rho = \{(c, d), \mu_\rho(c, d), \quad c, d \in \mathbb{X} \times \mathbb{X}\}$$



Fuzzy connectedness = some fuzzy relation

Fuzzy relation ρ binds a pair of spels (spacial elements) c, d of some set $\mathbb{X}$ using the function $\mu_\rho(c, d) \in \langle 0, 1 \rangle$:

$$\rho = \{(c, d), \mu_\rho(c, d), \quad c, d \in \mathbb{X} \times \mathbb{X}\}$$



Fuzzy connectedness

If $\mathbb{X}$ is *n*-dimensional image of field $\mathbb{X}^n$, then $\mathbf{c}, \mathbf{d}$ are points (pixels, voxels), being vectors of *n* coordinates:

$$\rho = \{(\mathbf{c}, \mathbf{d}), \mu_\rho(\mathbf{c}, \mathbf{d}), \quad \mathbf{c}, \mathbf{d} \in \mathbb{X}^n \times \mathbb{X}^n\}$$

Fuzzy connectedness

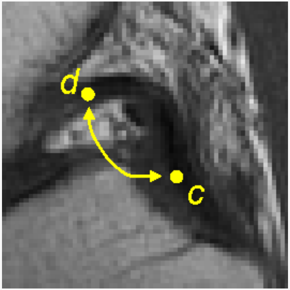
If $\mathbb{X}$ is n-dimensional image of field $\mathbb{X}^n$, then $\mathbf{c}, \mathbf{d}$ are points (pixels, voxels), being vectors of n coordinates:

$$\rho = \{(\mathbf{c}, \mathbf{d}), \mu_\rho(\mathbf{c}, \mathbf{d}), \quad \mathbf{c}, \mathbf{d} \in \mathbb{X}^n \times \mathbb{X}^n\}$$

Fuzzy connectedness

If $\mathbb{X}$ is *n*-dimensional image of field $\mathbb{X}^n$, then $\mathbf{c}, \mathbf{d}$ are points (pixels, voxels), being vectors of *n* coordinates:

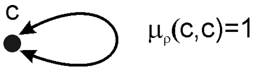
$$\rho = \{(\mathbf{c}, \mathbf{d}), \mu_\rho(\mathbf{c}, \mathbf{d}), \quad \mathbf{c}, \mathbf{d} \in \mathbb{X}^n \times \mathbb{X}^n\}$$



Properties of fuzzy relations (this corresponds to the equivalence relation in hard sets) – similitude relation

- Reflexivity:

$$\forall (c, c) \in \mathbb{X} \times \mathbb{X} : \qquad \mu_\rho(c, c) = 1.$$



Note:

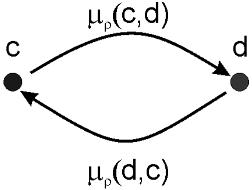
Each element is connected with itself with maximal value of μ_ρ

Fuzzy connectedness

Properties of fuzzy relations

- Symmetry:

$$\forall (c, d) \in \mathbb{X} \times \mathbb{X} : \qquad \mu_\rho(c, d) = \mu_\rho(d, c).$$



Note:

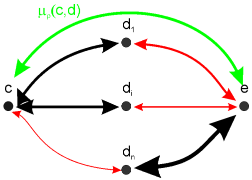
Each two elements are connected with the same relation in both sides

Fuzzy connectedness

Properties of fuzzy relations

- Transitivity:

$$\forall (c,d),(d,e),(c,e) \in \mathbb{X} \times \mathbb{X} :$$
$$\mu_\rho(c,e) = \max_d \left\{\min\left\{\mu_\rho(c,d), \mu_\rho(d,e)\right\}\right\}.$$



Note:

The relation of two elements may consist of their relations with third element, according to above rule

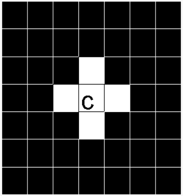
Fuzzy connectedness

Crisp adjacency

An image - a **topological ordered set**: the spacial relations between its elements (points) are defined.
Crisp adjacency relations α are e.g. 4-, 8-adjacency for 2D images or 6-, 18- and 26-adjacency for 3D image series. Exemplary 4-adjacency:

$$\forall \mathbf{c} = (c_x, c_y), \mathbf{d} = (d_x, d_y) \in \mathbb{X}^2 \times \mathbb{X}^2 :$$

$$\mu_\alpha(\mathbf{c}, \mathbf{d}) = \begin{cases} 1 & \Leftrightarrow \quad \sum_{i=x,y} (|c_i - d_i|) \leq 1, \\ 0 & \quad\;\; \text{otherwise.} \end{cases}$$



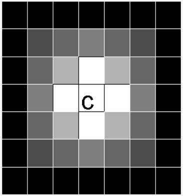
Fuzzy connectedness

Fuzzy adjacency relation

Fuzzy spel adjacency is a reflexive and symmetric fuzzy relation $\mu_\alpha \in \langle 0, 1\rangle$ in $\mathbb{X}^2$ and assigns a value to a pair of spels (c, d) based on how close they are spatially, eg.:

$$\forall \mathbf{c} = (c_x, c_y), \mathbf{d} = (d_x, d_y) \in \mathbb{X}^2 \times \mathbb{X}^2 :$$

$$\mu_\alpha(\mathbf{c}, \mathbf{d}) = \begin{cases} \frac{1}{||c-d||} & \Leftrightarrow \quad ||c-d|| < 3, \\ 0 & \quad\quad \text{otherwise.} \end{cases}$$



Fuzzy connectedness

Fuzzy affinity κ - the primary fuzzy relation in FC theory

Fuzzy affinity membership function $\mu_\kappa \in [0, 1]$ takes non-zero values for adjacent (fuzzy adjacent) elements only.

It assigns a value to a pair of spels (c, d) based on how close they are spatially and intensity-based-property-wise (local hanging-togetherness).

The value of μ_κ is based on the c and d features, like **coordinate adjacency**, **image intensities** $I(c), I(d)$ or **local intensity gradients**.

Fuzzy spel affinity

The most popular formula using in image processing:

$$\mu_\kappa(\mathbf{c}, \mathbf{d}) = \mu_\alpha(\mathbf{c}, \mathbf{d}) \cdot g\left(\mu_\phi(\mathbf{c}, \mathbf{d}), \mu_\psi(\mathbf{c}, \mathbf{d})\right),$$

where:

- μ_ϕ – intensity component, describing expected mean intensity of connected points,
- μ_ψ – gradient component, describing expected difference of intensities of connected points.

Fuzzy connectedness

Fuzzy spel affinity

The most popular formula using in image processing:

$$\mu_\kappa(\mathbf{c},\mathbf{d}) = \mu_\alpha(\mathbf{c},\mathbf{d}) \cdot g\left(\mu_\phi(\mathbf{c},\mathbf{d}), \mu_\psi(\mathbf{c},\mathbf{d})\right),$$

Expected properties of g:

- range within [0,1]
- monotonically non-decreasing in both arguments

Examples:

$$\mu_\kappa(\mathbf{c},\mathbf{d}) = \frac{1}{2}\mu_\alpha(\mathbf{c},\mathbf{d})\left(\mu_\phi(\mathbf{c},\mathbf{d}) + \mu_\psi(\mathbf{c},\mathbf{d})\right)$$

$$\mu_\kappa(\mathbf{c},\mathbf{d}) = \mu_\alpha(\mathbf{c},\mathbf{d})\sqrt{\left(\mu_\phi(\mathbf{c},\mathbf{d}) + \mu_\psi(\mathbf{c},\mathbf{d})\right)}$$

Fuzzy connectedness

Fuzzy spel affinity:

$$\kappa = \{((\underline{e}, \underline{d}), \mu_\kappa(\underline{e}, \underline{d})) : (\underline{e}, \underline{d}) \in C \times C\},$$

$$\mu_\kappa(\underline{e}, \underline{d}) = \mu_\alpha \cdot (w_1 H_1(\underline{e}, \underline{d}) + w_2 H_2(\underline{e}, \underline{d})),$$

with parameters w_1 and w_2 denoting such positive constants that

$$w_1 + w_2 = 1.$$

Components H_1 and H_2 are defined as:

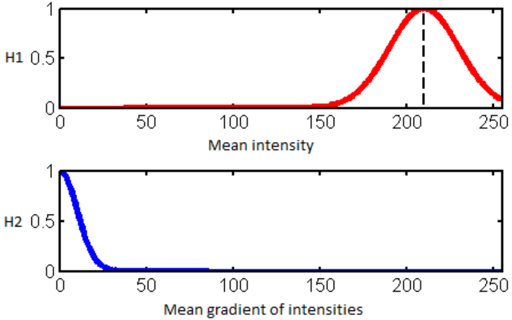
$$H_1(\underline{e}, \underline{d}) = \exp\left(-\frac{1}{2\sigma_1^2}\left(\frac{I(\underline{e})+I(\underline{d})}{2} - \lambda_1\right)^2\right),$$

$$H_2(\underline{e}, \underline{d}) = \exp\left(-\frac{1}{2\sigma_2^2}\left(|I(\underline{e}) - I(\underline{d})| - \lambda_2\right)^2\right).$$

Fuzzy connectedness

Fuzzy spel affinity

$$\mu_\kappa(\mathbf{c},\mathbf{d}) = \mu_\alpha(\mathbf{c},\mathbf{d}) \left(w_1 e^{\frac{\left(\frac{I(\mathbf{c})+I(\mathbf{d})}{2} - \lambda_1\right)^2}{2\sigma_1^2}} + w_2 e^{\frac{\left(|I(\mathbf{c})-I(\mathbf{d})| - \lambda_2\right)^2}{2\sigma_2^2}} \right) :$$



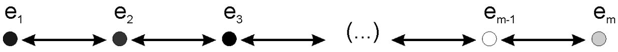
Fuzzy connectedness

Path

A single pair of points $\langle \mathbf{e}_i, \mathbf{e}_j \rangle$ with non-zero affinity is called **link** and the value of $\mu_\kappa(\mathbf{e}_i, \mathbf{e}_j)$ – its strength.

Path

A path is any sequence of $m \geq 2$ spels $\langle \mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_m \rangle$ such that for any $i \in [1, m-1]$ a pair $\langle \mathbf{e}_i, \mathbf{e}_{i+1} \rangle$ is a link. It is noted p_{cd} if $\mathbf{c} = \mathbf{e}_1$ and $\mathbf{d} = \mathbf{e}_m$.



Let P_{cd} denotes the set of all possible paths p_{cd} from $\mathbf{c}$ to $\mathbf{d}$.

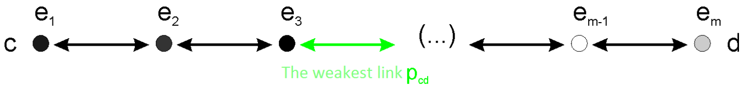
Then the set of all possible paths in $\mathbb{X}^2$ is

$$P = \bigcup_{\mathbf{c},\mathbf{d} \in \mathbb{X}^2} P_{cd}$$

Fuzzy connectedness

Strength of path

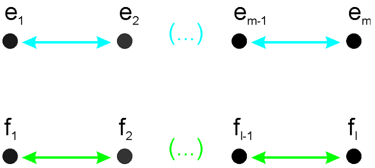
Fuzzy membership function $\mu_{\mathcal{N}}(p_{cd})$ describing any path $p_{cd} \in P_{cd}$ is said to be its **strength** and is the smallest spel affinity along p_{cd}



$$\mu_{\mathcal{N}}(p_{cd}) = \begin{cases} \min\limits_{i=1..m-1} \{\mu_\kappa(\mathbf{e}_i, \mathbf{e}_{i+1})\} & \Leftrightarrow \quad p_{cd} = \langle \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m \rangle \\ 0 & \Leftrightarrow \quad p_{cd} = \langle \rangle. \end{cases}$$

Path – join to path operation

A binary *join to* operation denoted "+" is defined as follows



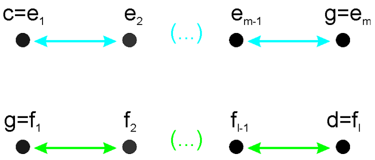
$$\langle \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m \rangle,$$

$$\langle \mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l \rangle.$$

Fuzzy connectedness

Path – join to path operation

A binary *join to* operation denoted "+" is defined as follows



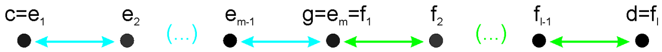
$$p_{cg} = \langle \mathbf{c}, \mathbf{e}_2, \ldots, \mathbf{g} \rangle,$$

$$p_{gd} = \langle \mathbf{g}, \mathbf{f}_2, \ldots, \mathbf{d} \rangle.$$

Fuzzy connectedness

Path – join to path operation

A binary *join to* operation denoted "$+$" is defined as follows



$$p_{cg} + p_{gd} = \langle \mathbf{c}, \mathbf{e}_2, \ldots, \mathbf{g}, \mathbf{f}_2, \ldots, \mathbf{d} \rangle.$$

Path – join to path operation

For empty paths:

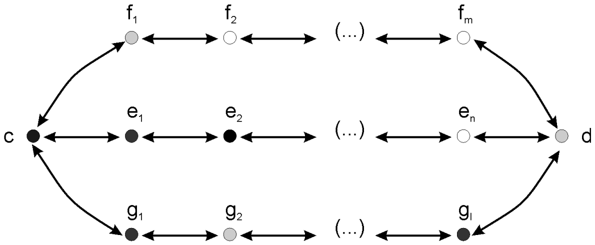
$$p_{cd} + \langle\rangle = p_{cd},$$

$$\langle\rangle + p_{cd} = p_{cd},$$

$$\langle\rangle + \langle\rangle = \langle\rangle.$$

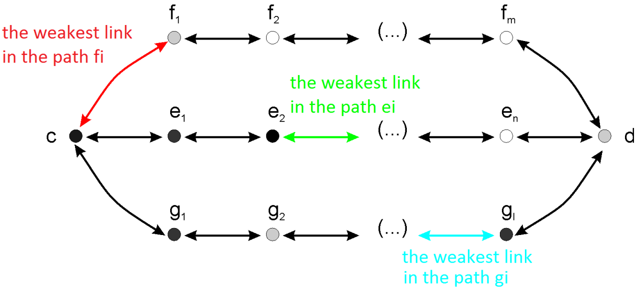
Fuzzy connectedness

There exist many different paths p_{cd} constituting a set P_{cd}.



Fuzzy connectedness

Each path p_{cd} has a defined strength $\mu_{\mathcal{N}}$.

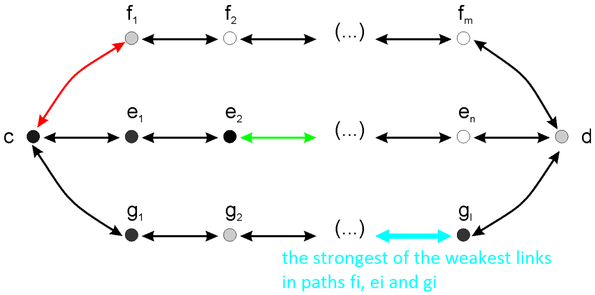


$$\mu_{\mathcal{N}}(p_{cd}) = \begin{cases} \min\limits_{i=1..m-1} \{\mu_{\kappa}(\mathbf{e}_i, \mathbf{e}_{i+1})\} & \Leftrightarrow \quad p_{cd} = \langle \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m \rangle \\ 0 & \Leftrightarrow \quad p_{cd} = \langle \rangle. \end{cases}$$

Fuzzy connectedness

Any image spels **c** and **d** are fuzzy connected according to relation K.

The membership function of fuzzy connectedness $\mu_K(\mathbf{c}, \mathbf{d})$ is the strength of the strongest path p_{cd} of all the paths between **c** and **d**, forming a set P_{cd} (global hanging-togetherness):

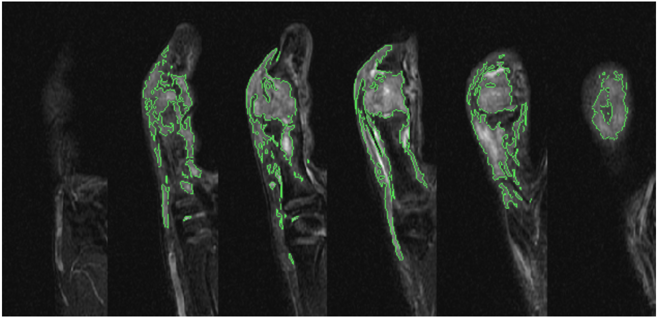


Fuzzy connectedness

Fuzzy connectedness vs image segmentation

If the set $\mathbb{X}$ is an image, then the information concerning fuzzy relation between its points (pixels) makes it possible to group them into semantically interpretable regions - **to segment them**.

The theorems were given in (Udupa&Samarasekera, 1996).



Fuzzy connectedness

Fuzzy connected object

Fuzzy $\kappa\theta_x$ object – $\mathcal{O}_{\theta_x}(\mathbf{o})$ containing a seed spel $\mathbf{o}$ is a fuzzy subset of $\mathbb{X}$ whose membership function is:

$$\mu_{\mathcal{O}_{\theta_x}(\mathbf{o})}(\mathbf{c}) = \begin{cases} \eta(\mathbf{c}) & \Leftrightarrow \quad c \in \mathcal{O}_{\theta_x}(\mathbf{o}) \\ 0 & \quad\quad\; \textit{otherwise} \end{cases}$$

where η assigns an objectness value to each spel perhaps based on $f(\mathbf{c})$ and $\mu_\kappa(\mathbf{o}, \mathbf{c})$.
Fuzzy digital space

$$(\mathbb{X}^n, \alpha)$$

Scene (over a fuzzy digital space)

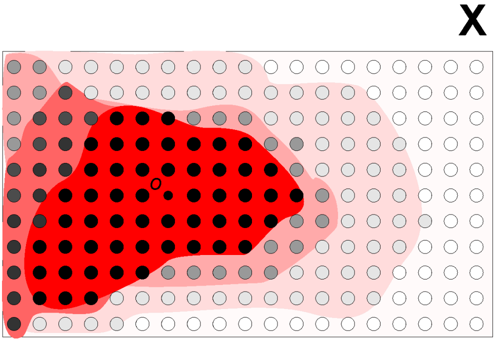
$$C_o = (C, f), \quad C \subset \mathbb{X}^n, f : C \to [0, 1]$$

Fuzzy connected object

Fuzzy $\kappa\theta_x$ object $\mathcal{O}_{\theta_x(\mathbf{o})}$ consists then of all the elements, including $\mathbf{o}$, connected with each other with the value greater x, where $x \in [0, 1]$, towards the defined fuzzy affinity relation κ

$$\mathcal{O}_{\theta_x(\mathbf{o})}(\mathbf{c}) = \begin{cases} 1 & \Leftrightarrow \quad c \in C_o \geq x \\ 0 & \quad \textit{otherwise} \end{cases}$$

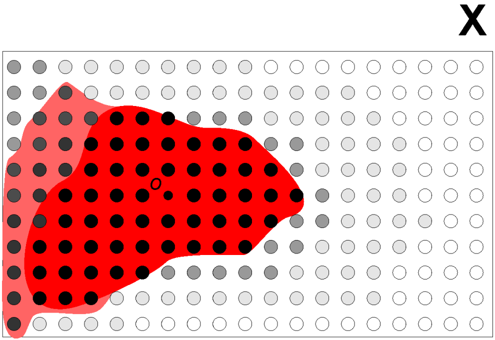
Fuzzy connected object



Fuzzy $\kappa\theta_x$ object $\mathcal{O}_{\theta_x(\mathbf{o})}$ consists then of all the elements, including $\mathbf{o}$, connected with each other with the value greater x, where $x \in [0, 1]$, towards the defined fuzzy affinity relation κ.

Fuzzy connectedness

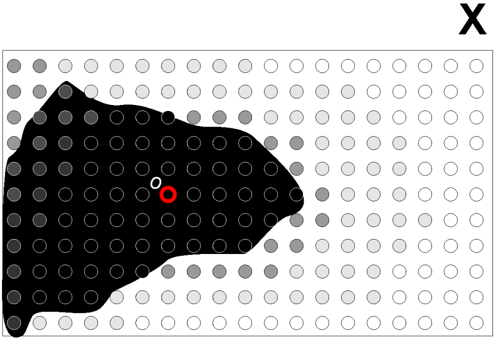
Fuzzy connected object



Fuzzy $\kappa\theta_x$ object $\mathcal{O}_{\theta_x(\mathbf{o})}$ consists then of all the elements, including $\mathbf{o}$, connected with each other with the value greater x, where $x \in [0, 1]$, towards the defined fuzzy affinity relation κ.

Fuzzy connectedness

Fuzzy connected object → binary object



Defuzzyfication of fuzzy connected object $\mathcal{O}_{\theta_x(\mathbf{o})}$ results in binary object $O_{\theta_x}(\mathbf{o})$, as a set of points in $\mathcal{O}_{\theta_x}(\mathbf{o})$.

Fuzzy connectedness

Seed points

According to the already given definition, to obtain the fuzzy connected object is necessary to calculate fuzzy connectedness of all the pair of elements in $\mathbb{X}$.

Seed points selection procedure is then needed.

In different applications a multiseeded approach is used, where set O of $M (M > 1)$ points $\mathbf{o}_i$ is required.

Fuzzy connectivity scene

In the segmentation procedure the **fuzzy connectivity scene** C_o is created by assigning a strength of connectedness to each possible path between some predefined seed point $\mathbf{o}$ and any other image element:

$$C_o(\mathbf{c}) = \mu_K(\mathbf{o}, \mathbf{c}).$$

If there exists the set O of $M (M > 1)$ points $\mathbf{o}_i$, then $C_o(\mathbf{c})$ is equal to the strongest of connectedness with points in O:

$$C_o(\mathbf{c}) = \max_i \{\mu_K(\mathbf{o}_i, \mathbf{c})\}.$$

The scene C_o for a set O is a fuzzy union:

$$C_o(\mathbf{c}) = \bigcup_{\mathbf{o}_i \in O} C_{\mathbf{o}_i} = \max_i \{\mu_K(\mathbf{o}_i, \mathbf{c})\}.$$

Fuzzy connectedness

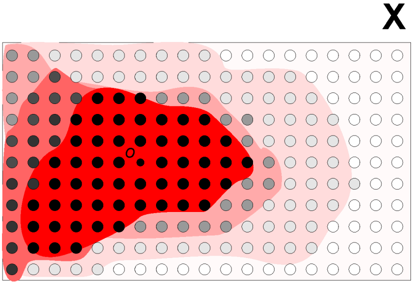
Fuzzy connectivity scene

According to the reflexivity properties, for the point **o**:

$$C_o(\mathbf{o}) = \mu_K(\mathbf{o}, \mathbf{o}) = 1.$$

The far from the starting point [a], the smaller the fuzzy connectedness value is.

[a] in the meaning of connectedness



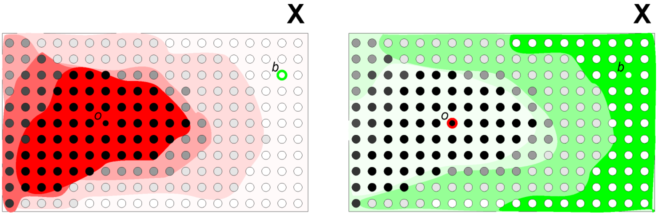
Relative fuzzy connectedness

The relative fuzzy connectedness (RFC) variant uses the second object, treated as a background region with its own seed point $\mathbf{b}$, to determine $O_{\kappa_x(\mathbf{b})}$ in a rivalry mode.

Spel $\mathbf{c} \in C$ belongs to an object according to affinity κ and connectedness K if $\mu_K(\mathbf{o}, \mathbf{c}) > \mu_K(\mathbf{b}, \mathbf{c})$. It leads to fuzzy connectivity scenes C_o and C_b and their comparison in a form of a "division of spoils".

Relative fuzzy connectedness

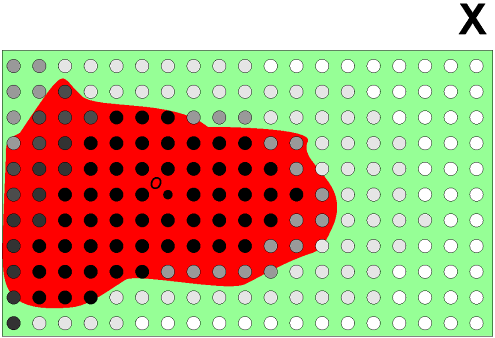
After creating scenes C_o and C_b each point $\mathbf{c}$ is assigned to the region (eg. object and background), with which starting point is connected stronger.



$$\mathbf{c} \in P_{ob_\kappa} \quad \Leftrightarrow \quad C_o(\mathbf{c}) > C_b(\mathbf{c}).$$

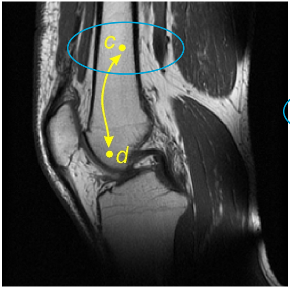
Relative fuzzy connectedness

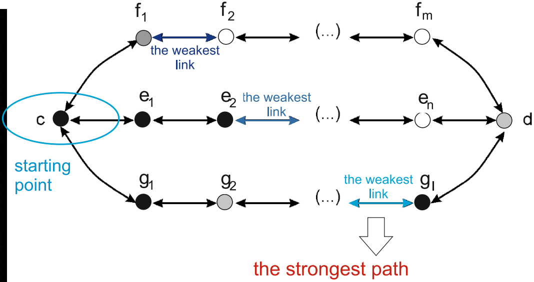
After creating scenes C_o and C_b each point $\mathbf{c}$ is assigned to the region (eg. object and background), with which starting point is connected stronger.



$$\mathbf{c} \in P_{ob_\kappa} \quad \Leftrightarrow \quad C_o(\mathbf{c}) > C_b(\mathbf{c}).$$

3-D relative fuzzy connectedness analysis:





1. Find the set of objects and background starting points respectively
2. Chose the fuzzy spel affinity κ and fuzzy adjacency ρ
3. Extract the fuzzy relative connected objects

Fuzzy connectedness

Fuzzy connectivity scene creation - implementation

Fuzzy connectivity scene C_o creation - graph searching:

- dynamic programming approach (Udupa and Samarasekera, 1996)
- Dijkstra algorithm (Carvalho et al., 1999)
- matrix transformation-based approach (Kawa, 2007)

Fuzzy connectedness

Implementation - dynamic programming

- The DP algorithm uses queue Q, into/from which successive image points are inserted or removed.
- In a single iteration t one point ($\mathbf{d}$) from the front of queue is taken. All its neighboring pixels (voxels) are tested.
- It is checked, if there exists a path p_{cd}, connecting points $\mathbf{c}$ and $\mathbf{d}$ with stronger path, then previously estimated $C_o(\mathbf{d})$.
- If yes, $C_o(\mathbf{d})$ is updated, and all the neighbors of $\mathbf{d}$ are put into queue Q.

Fuzzy connectedness

Implementation - dynamic programming

Features:

- A single iteration t takes short time.
- A lot of iterations is required.
- One point can be analyzed several times.

Implementation - dynamic programming

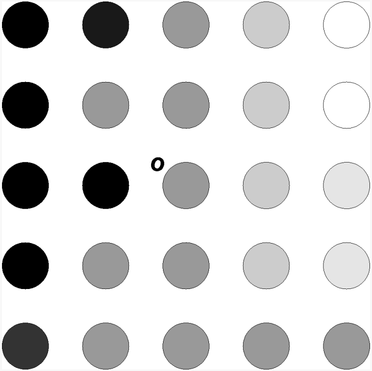
=== Algorithm 1. Dynamic Programming ===

Initialize scene C_o with zeros;
Set $C_o(\mathbf{o}_i) = 1$ for all the starting points $\mathbf{o}_i$;
Put into the queue Q all $\mathbf{c}: \ \mu_\kappa(\mathbf{o}_i, \mathbf{c}) > 0$;
Until Q is not empty
Take and remove $\mathbf{c}$ from the front of queue Q;
    $f_{max} = \max_{\mathbf{d}} \left[\min\left\{C_o(\mathbf{d}), \mu_\kappa(\mathbf{c}, \mathbf{d})\right\}\right]$;
    If $f_{max} > C_o(\mathbf{c})$
    $C_o(\mathbf{c}) = f_{max}$;
    Then put to Q all $\mathbf{e}: \ \mu_\kappa(\mathbf{c}, \mathbf{e}) > 0$;
    end
end

==

Fuzzy connectedness

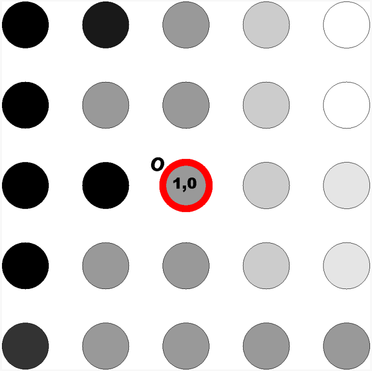
Dynamic programming



1: Initialize scene C_o with zeros;

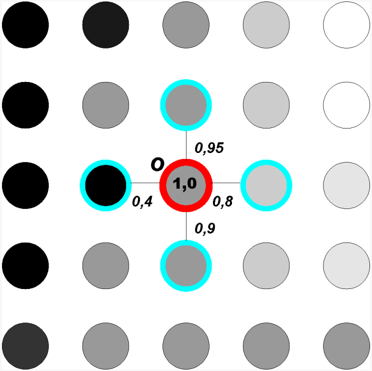
Fuzzy connectedness

Dynamic programming



2: Set $C_o(\mathbf{o}_i) = 1$ for all starting points $\mathbf{o}_i$;

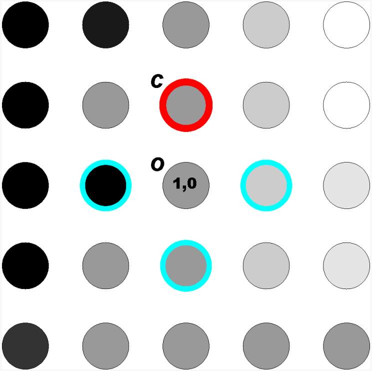
Dynamic programming



3: `Put to the queue` Q `all` $\mathbf{c}$: $\mu_\kappa(\mathbf{o}_i, \mathbf{c}) > 0$;

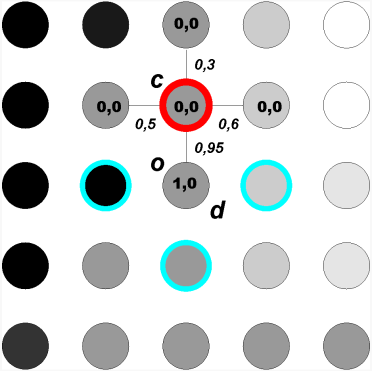
Fuzzy connectedness

Dynamic programming



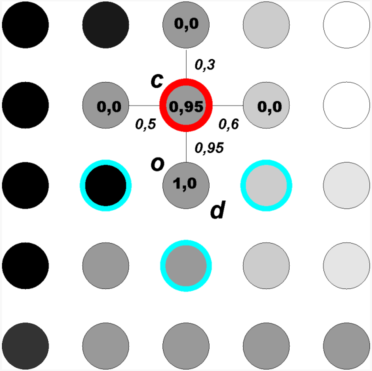
5: Take and remove **c** from the front of queue Q;

Dynamic programming



6 : $f_{max} = \max_{\mathbf{d}} \left[\min\left\{C_o(\mathbf{d}), \mu_\kappa(\mathbf{c}, \mathbf{d})\right\}\right];$

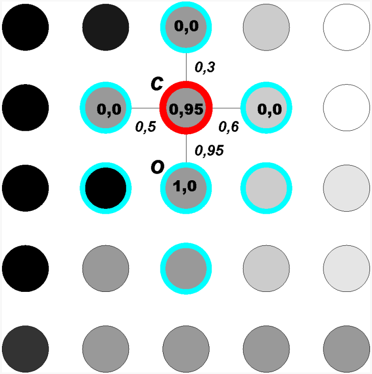
Dynamic programming



7: `If` $f_{max} > C_o(\mathbf{c})$

8: $C_o(\mathbf{c}) = f_{max}$;

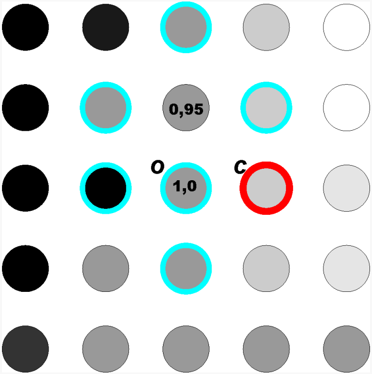
Dynamic programming



9: Put to Q all $\mathbf{e} : \mu_\kappa(\mathbf{c}, \mathbf{e}) > 0$;

Fuzzy connectedness

Dynamic programming

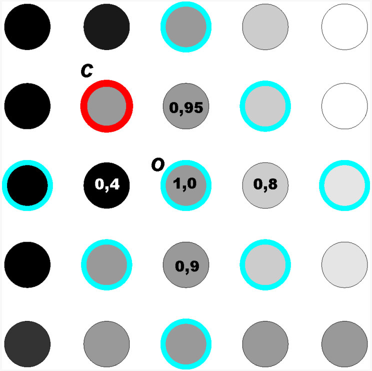


5: Take and remove **c** from the front of queue Q;

Dynamic programming

After a while...

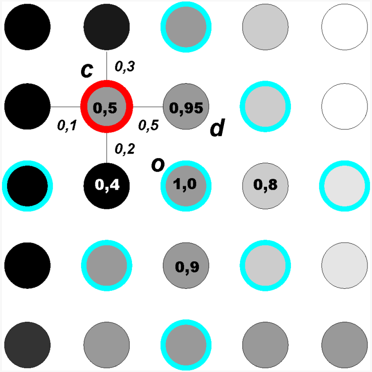
Dynamic programming



5: Take and remove **c** from the front of queue Q;

Fuzzy connectedness

Dynamic programming



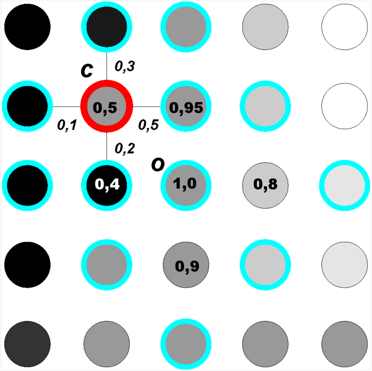
6: $f_{max} = \max_{\mathbf{d}} \left[\min\left\{C_o(\mathbf{d}), \mu_\kappa(\mathbf{c}, \mathbf{d})\right\}\right];$

7: If $f_{max} > C_o(\mathbf{c})$

8: $C_o(\mathbf{c}) = f_{max};$

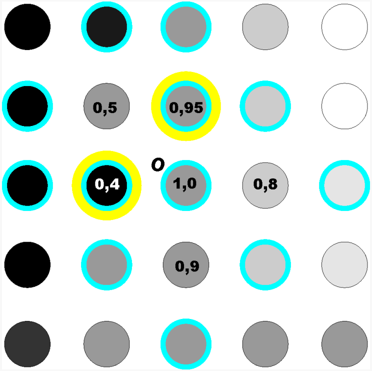
Fuzzy connectedness

Dynamic programming



9: `Put to` Q `all` $\mathbf{e}: \mu_\kappa(\mathbf{c},\mathbf{e}) > 0$;

Dynamic programming



The selected points are put into the queue once again!

Fuzzy connectedness

Implementation - Dijkstra algorithm

- I a single iteration t the point **is not** from the front of queue Q, but the point with $\mathbf{c}$ with the maximal value $C_o(\mathbf{c})$.
- It is treated as the second last point of the path leading to each of its neighbors $\mathbf{d}$.
- If it causes the improvement of $C_o(\mathbf{d})$, then the value $C_o(\mathbf{d})$ is updated and $\mathbf{d}$ is put into the queue.

Fuzzy connectedness

Implementation - Dijkstra algorithm

Features:

- A point **c** taken from the queue **is not** put there again - the number of iterations T does not exceed the number of image points.
- It is necessary to perform time consuming maximum operation.

Fuzzy connectedness

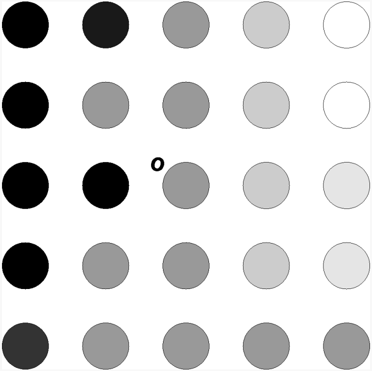
Implementation - Dijkstra algorithm

=== Algorythm 2. Dijkstra algorithm ==========

Initialize scene C_o with zeros;
Set $C_o(\mathbf{o}_i) = 1$ for all the starting points $\mathbf{o}_i$;
Put into the queue Q all $\mathbf{o}_i$;
Until Q is not empty
  Take and remove from the queue Q the point $\mathbf{c}$ with maximum $C_o(\mathbf{c})$;
  For all $\mathbf{d}: \ \mu_\kappa(\mathbf{c}, \mathbf{d}) > 0$
    $\nu = \min\{C_o(\mathbf{c}), \mu_\kappa(\mathbf{c}, \mathbf{d})\}$;
    If $\nu > C_o(\mathbf{d})$
      $C_o(\mathbf{d}) = \nu$;
      Put $\mathbf{d}$ into Q;
    end
  end
end

==

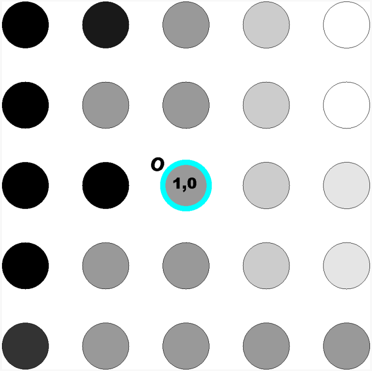
Implementation - Dijkstra algorithm



1: Initialize scene C_o with zeros;

Fuzzy connectedness

Implementation - Dijkstra algorithm

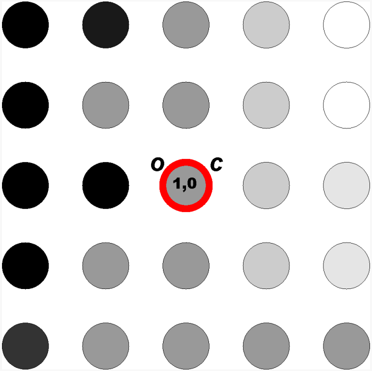


2: Set $C_o(\mathbf{o}_i) = 1$ for all starting points $\mathbf{o}_i$;

3: Put into the queue Q all $\mathbf{o}_i$;

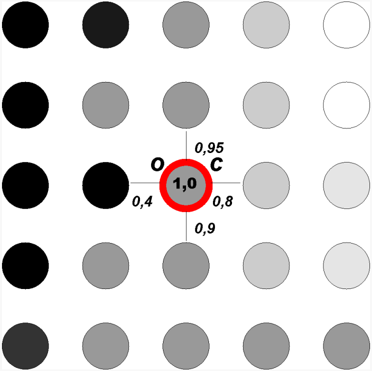
Fuzzy connectedness

Implementation - Dijkstra algorithm



5: Take and remove from the queue Q the point **c**

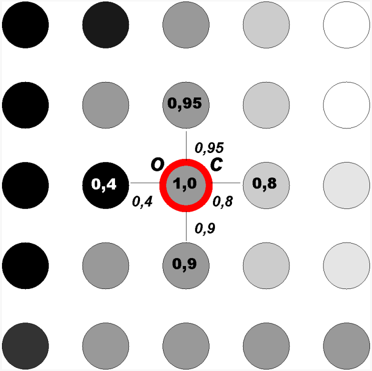
Implementation - Dijkstra algorithm



6: For all $\mathbf{d}: \ \mu_\kappa(\mathbf{c},\mathbf{d}) > 0$

7: $\nu = \min\{C_o(\mathbf{c}), \mu_\kappa(\mathbf{c},\mathbf{d})\}$;

Implementation - Dijkstra algorithm

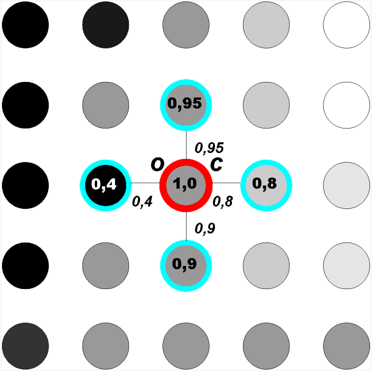


8: If $\nu > C_o(\mathbf{d})$

9: $C_o(\mathbf{d}) = \nu$;

Fuzzy connectedness

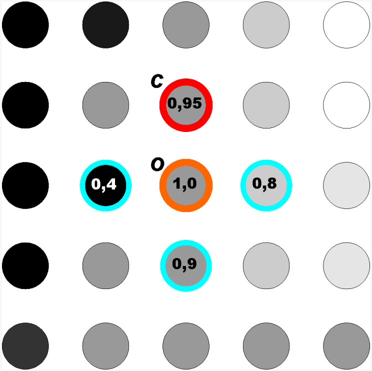
Implementation - Dijkstra algorithm



10: Put **d** into Q;

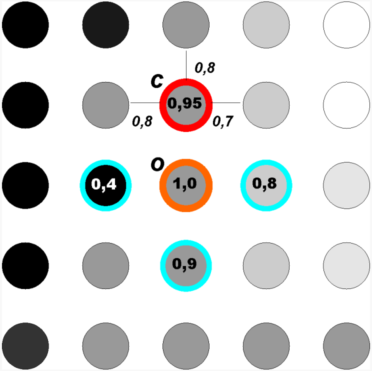
Fuzzy connectedness

Implementation - Dijkstra algorithm



5: Take and remove from the queue Q the point **c**

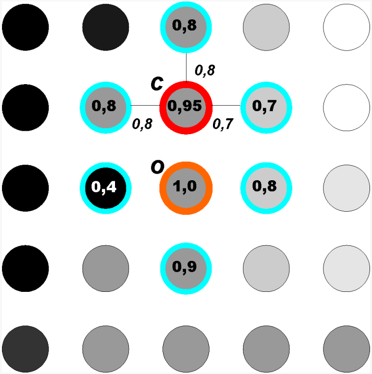
Implementation - Dijkstra algorithm



6: For all $\mathbf{d}$: $\mu_\kappa(\mathbf{c}, \mathbf{d}) > 0$

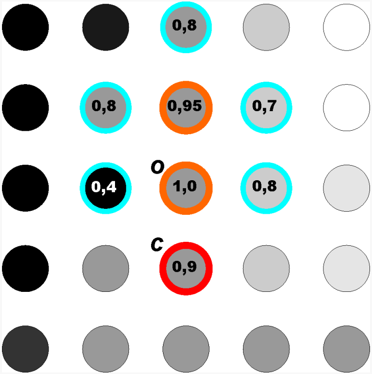
7: $\nu = \min\{C_o(\mathbf{c}), \mu_\kappa(\mathbf{c}, \mathbf{d})\}$;

Implementation - Dijkstra algorithm



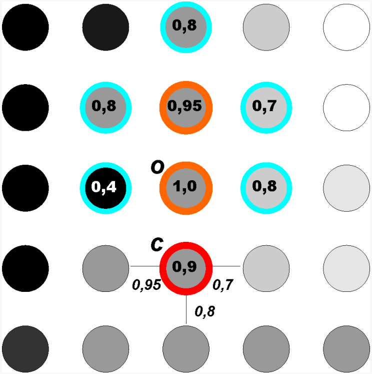
If $\nu > C_o(\mathbf{d})$
    $C_o(\mathbf{d}) = \nu$;
    Put $\mathbf{d}$ into Q;

Implementation - Dijkstra algorithm



5: Take and remove from the queue Q the point **c**

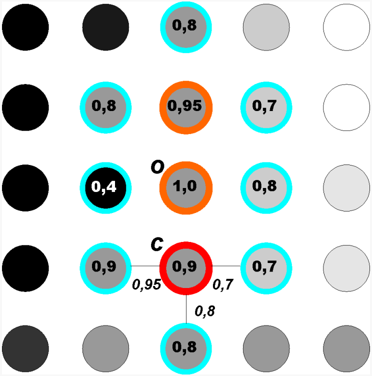
Implementation - Dijkstra algorithm



6: For all $\mathbf{d}$: $\mu_\kappa(\mathbf{c},\mathbf{d}) > 0$

7: $\nu = \min\{C_o(\mathbf{c}), \mu_\kappa(\mathbf{c},\mathbf{d})\}$;

Implementation - Dijkstra algorithm

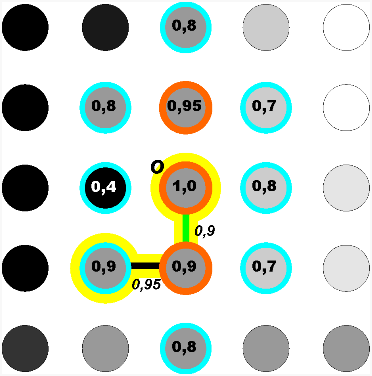


If $\nu > C_o(\mathbf{d})$

$C_o(\mathbf{d}) = \nu$;

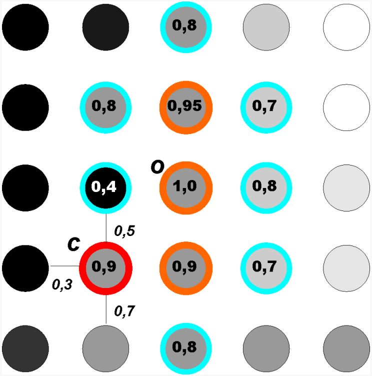
Put $\mathbf{d}$ into Q;

Implementation - Dijkstra algorithm



The path to the selected point and its weakest link.

Implementation - Dijkstra algorithm



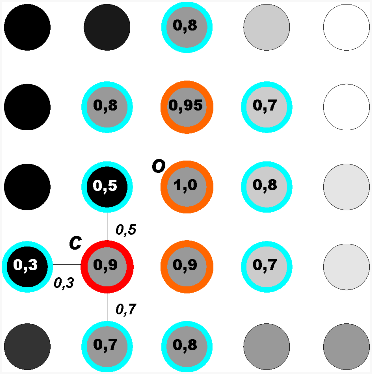
5: Take and remove from the queue Q the point $\mathbf{c}$

6: For all $\mathbf{d}: \ \mu_\kappa(\mathbf{c}, \mathbf{d}) > 0$

7: $\nu = \min\{C_o(\mathbf{c}), \mu_\kappa(\mathbf{c}, \mathbf{d})\};$

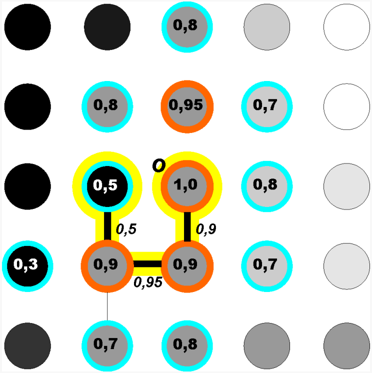
Fuzzy connectedness

Implementation - Dijkstra algorithm




```
If $\nu > C_o(\mathbf{d})$
    $C_o(\mathbf{d}) = \nu$;
    Put $\mathbf{d}$ into $Q$;
```

Implementation - Dijkstra algorithm



A stronger path is found!

Fuzzy connectedness

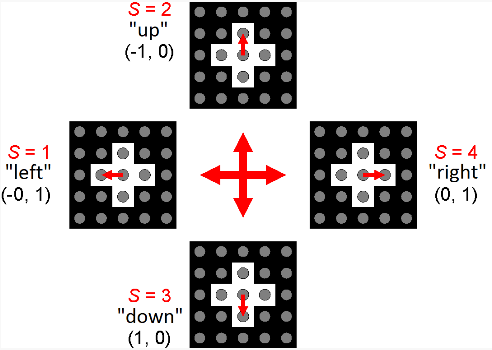
Implementation - Matrix transformation-based algorithm

- The approach uses analogy to shifts resulting from the shape of structural element in mathematical morphology.
- The previously obtained matrix of shifts makes it possible to calculate the "min" and "max" values in more efficient way, to dynamically change the fuzzy connectivity scene.

Implementation - Matrix transformation-based algorithm

Matrices adjacency:

- the exemplary crisp 4-adjacency relation $\alpha_4 = 4$ shifts s_i from the position of central pixel:



Implementation - Matrix transformation-based algorithm

```
= Algorythm 3. Matrix transformation-based algorithm =

1:   Chose the hard adjacency relation α;
2:   Decompose α to the series of shifts s
 denoting neighboring elements;
3:   Preprocess the original image by padding rows,
 columns and layers as not to permit for wrapping
 the original values in the later steps;
4:   Precompute the fuzzy affinity values for each
 neighborhood member (one table T_s  containing
 the affinity values for each  α-connected member;
 e.g. 4 tables if 4-connectivity has been chosen;
 e.g. T_1 for the one to the left);
```

Implementation - Matrix transformation-based algorithm

5: By using circular shift of the precomputed fuzzy affinity tables ensure, that the element of a new $T_s(\mathbf{c})$ denotes the affinity of original spel $\mathbf{c}$ with respect to the s-th neighbor; e.g. for a pixel $\mathbf{c} = (c_x, c_y)$, $T_1(c_x, c_y)$ contains $\mu_k(\mathbf{c}, \mathbf{d})$ with respect to the left pixel $\mathbf{d} = (c_{x-1}, c_y)$;

6: Initialize scene C_o with values zeros;

7: Set $C_o(\mathbf{o}_i) = 1$ for all the starting points $\mathbf{o}_i$;

7: Repeat

8: For each shift s

9: $T = \min(C_o, T_s)$; % matrix operation

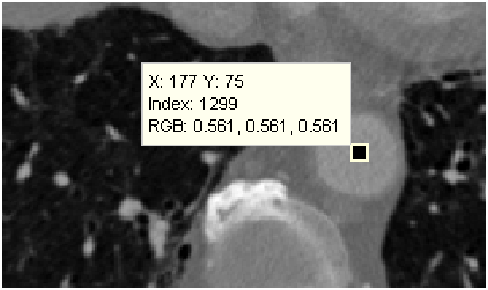
10: $C_o = \max(C_o, T)$; % matrix operation

11: end

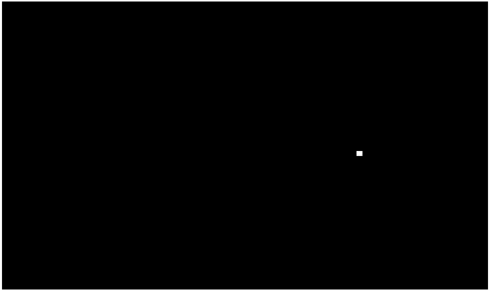
12: until C_o does not change

==

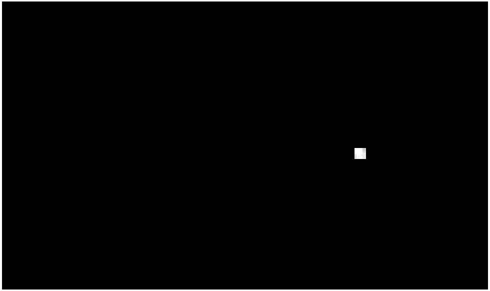
Implementation - example



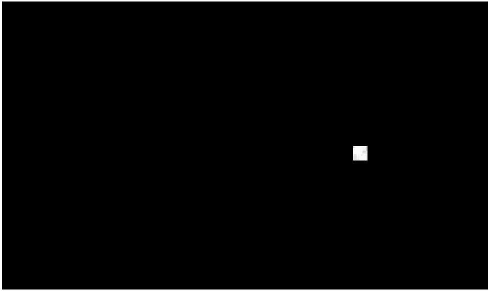
Implementation - example



Implementation - example

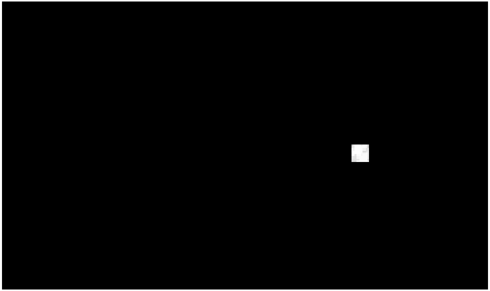


Implementation - example



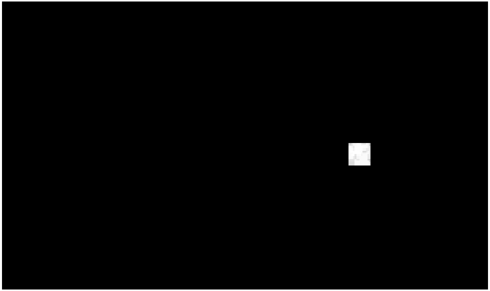
Fuzzy connectedness

Implementation - example

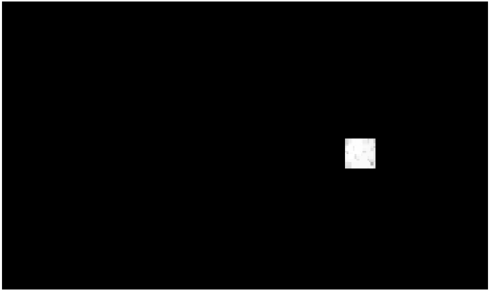


Fuzzy connectedness

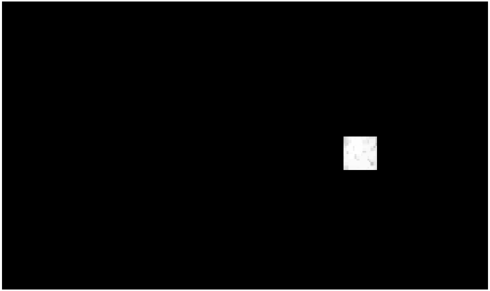
Implementation - example



Implementation - example

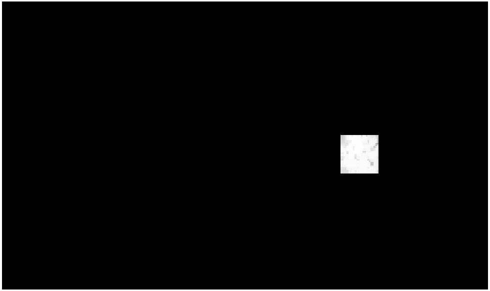


Implementation - example

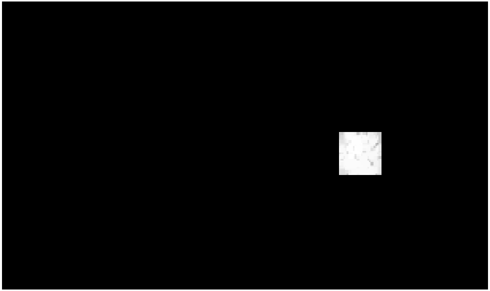


Fuzzy connectedness

Implementation - example

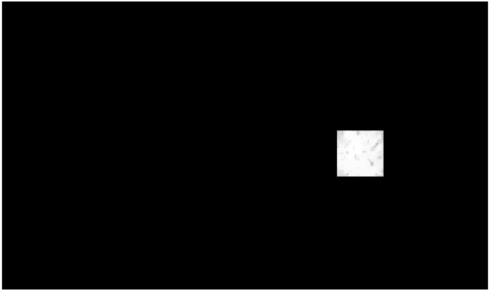


Implementation - example



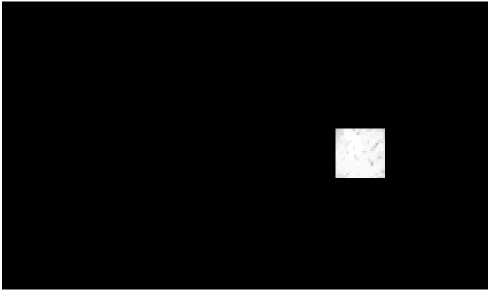
Fuzzy connectedness

Implementation - example

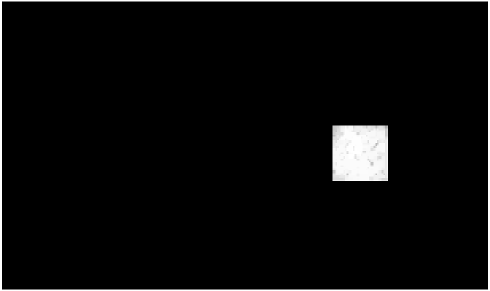


Fuzzy connectedness

Implementation - example

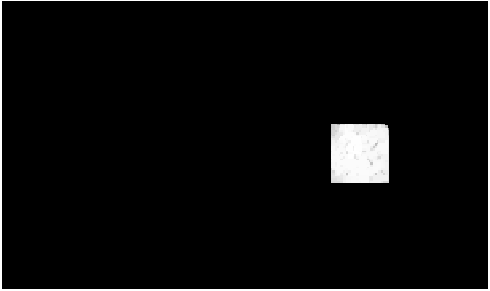


Implementation - example

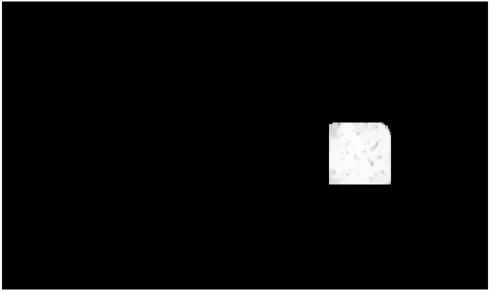


Fuzzy connectedness

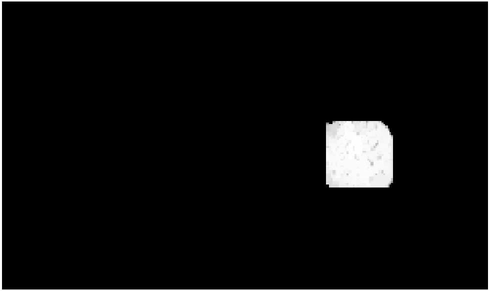
Implementation - example



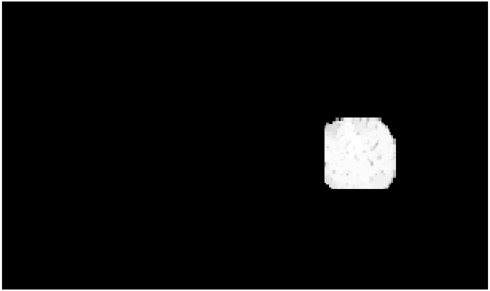
Implementation - example



Implementation - example

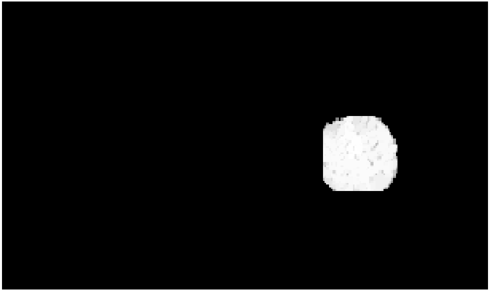


Implementation - example

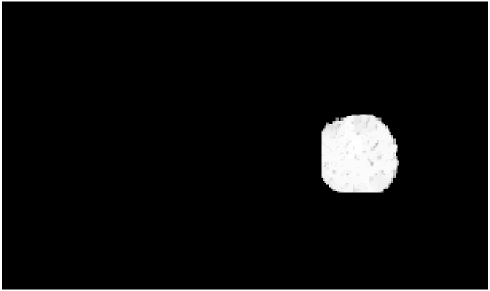


Fuzzy connectedness

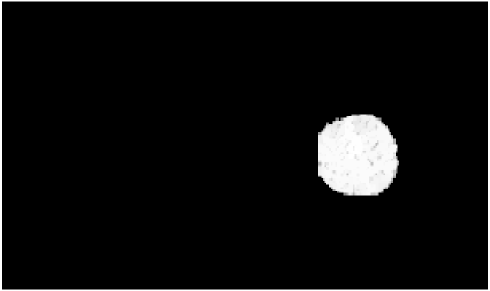
Implementation - example



Implementation - example

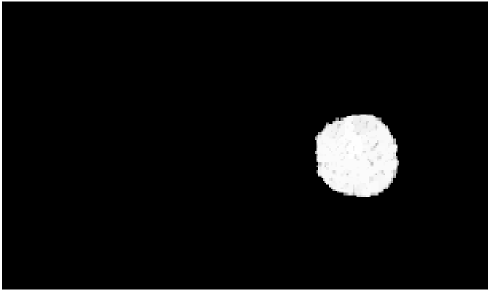


Implementation - example



Fuzzy connectedness

Implementation - example



Fuzzy connectedness

To sum up....

- Free to choose starting points
- Possibility to choose different features
- Free to select the threshold
- Simultaneous segmentation of different objects

Active Contour Model