# Clustering and Classification

Alexandre Xavier Falcão

Visual Informatics Laboratory - Institute of Computing - University of Campinas

afalcao@ic.unicamp.br

In Sciences and Engineering, many problems require an optimum partition of a dataset $\mathcal{Z}$ into

In Sciences and Engineering, many problems require an optimum partition of a dataset $\mathcal{Z}$ into

- groups of samples based on their similarities (clustering), and/or

# Introduction

In Sciences and Engineering, many problems require an optimum partition of a dataset $\mathcal{Z}$ into

- groups of samples based on their similarities (clustering), and/or
- classes of samples from a set of possible categories (classification).

# Introduction

In Sciences and Engineering, many problems require an optimum partition of a dataset $\mathcal{Z}$ into

- groups of samples based on their similarities (clustering), and/or
- classes of samples from a set of possible categories (classification).

In both cases, we wish to design a pattern classifier (unsupervised / supervised), which can predict the cluster/class of any new sample.

# Introduction

- For classification, a labeled training set $\mathcal{T} \subset \mathcal{Z}$ with examples from each category is always required (supervised learning).

- For classification, a labeled training set $\mathcal{T} \subset \mathcal{Z}$ with examples from each category is always required (supervised learning).
- For clustering, the unlabeled dataset is the training set, but when it is too large, a subset of it can be used (unsupervised learning).

# Introduction

- For classification, a labeled training set $\mathcal{T} \subset \mathcal{Z}$ with examples from each category is always required (supervised learning).
- For clustering, the unlabeled dataset is the training set, but when it is too large, a subset of it can be used (unsupervised learning).
- In both cases, each sample $s \in \mathcal{Z}$ is represented by a point/vector $\vec{v}(s)$ in some feature space and the similarity between samples is measured by a distance function $d(s,t)$.

# Introduction

- For classification, a labeled training set $\mathcal{T} \subset \mathcal{Z}$ with examples from each category is always required (supervised learning).
- For clustering, the unlabeled dataset is the training set, but when it is too large, a subset of it can be used (unsupervised learning).
- In both cases, each sample $s \in \mathcal{Z}$ is represented by a point/vector $\vec{v}(s)$ in some feature space and the similarity between samples is measured by a distance function $d(s, t)$.
- The methods usually decide for the cluster/class of a new sample based on its position in the feature space and/or similarity with respect to the training samples/parametric model.

They may also assume that

They may also assume that

- the classes/clusters form compact clouds of points in the feature space,

# Introduction

They may also assume that

- the classes/clusters form compact clouds of points in the feature space,
- the classes/clusters do not overlap each other,

# Introduction

They may also assume that

- the classes/clusters form compact clouds of points in the feature space,
- the classes/clusters do not overlap each other,
- one cluster corresponds to one class,

# Introduction

They may also assume that

- the classes/clusters form compact clouds of points in the feature space,
- the classes/clusters do not overlap each other,
- one cluster corresponds to one class,
- the probability density function from classes / clusters present known shapes for parametric modeling.

# Introduction

- We only assume a transitive property — i.e., two samples in a same cluster / class should be connected by a chain of nearby samples in the feature space.

# Introduction

- We only assume a transitive property — i.e., two samples in a same cluster / class should be connected by a chain of nearby samples in the feature space.
- A graph $(\mathcal{T}, \mathcal{A})$ is defined by some adjacency relation $\mathcal{A} \subset \mathcal{T} \times \mathcal{T}$ between training samples in the feature space.

# Introduction

- We only assume a transitive property — i.e., two samples in a same cluster / class should be connected by a chain of nearby samples in the feature space.
- A graph $(\mathcal{T}, \mathcal{A})$ is defined by some adjacency relation $\mathcal{A} \subset \mathcal{T} \times \mathcal{T}$ between training samples in the feature space.
- A connectivity function $f(\pi_t)$ assigns a value to any path $\pi_t$ from its root $R(\pi_t)$ to its terminal node $t$.

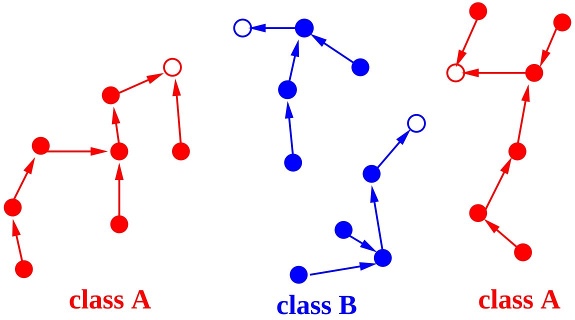
# Introduction

- We only assume a transitive property — i.e., two samples in a same cluster / class should be connected by a chain of nearby samples in the feature space.
- A graph $(\mathcal{T}, \mathcal{A})$ is defined by some adjacency relation $\mathcal{A} \subset \mathcal{T} \times \mathcal{T}$ between training samples in the feature space.
- A connectivity function $f(\pi_t)$ assigns a value to any path $\pi_t$ from its root $R(\pi_t)$ to its terminal node $t$.
- The minimization (maximization) of the connectivity map

$$V(s) = \min_{\forall t \in \Pi(\mathcal{T}, \mathcal{A}, t)} \{f(\pi_t)\}$$

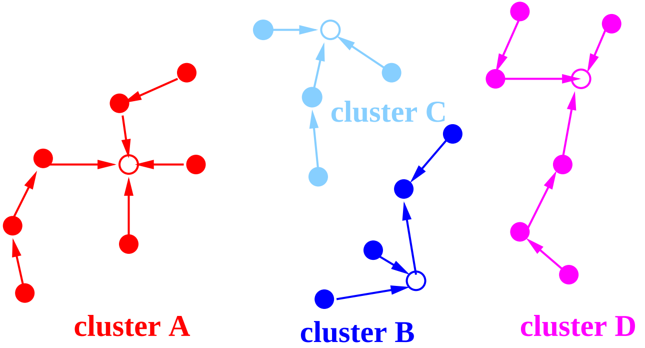
produces an optimum-path forest rooted at nodes called prototypes.

In supervised learning, each class is an optimum-path forest rooted at its prototypes, which propagate the class label to the remaining nodes of the forest.



# Introduction

In unsupervised learning, each cluster is an optimum-path tree rooted at some prototype, which propagates a cluster label to the remaining nodes of the tree.



# Introduction

- This methodology does not assume known shapes, number of groups, non-overlapping classes, or parametric models. Although, parametric models can be estimated after training.

# Introduction

- This methodology does not assume known shapes, number of groups, non-overlapping classes, or parametric models. Although, parametric models can be estimated after training.
- Both learning approaches are fast and robust for training sets of reasonable sizes.

# Introduction

- This methodology does not assume known shapes, number of groups, non-overlapping classes, or parametric models. Although, parametric models can be estimated after training.
- Both learning approaches are fast and robust for training sets of reasonable sizes.
- Label propagation to new samples is efficiently performed based on a local processing of the forest's attributes and distances with respect to the training nodes.

# Organization of this lecture

- Classification based on optimum-path forest[1, 2, 3].

# Organization of this lecture

- Classification based on optimum-path forest[1, 2, 3].
- Its application to enhance objects in natural scenes[4].

# Organization of this lecture

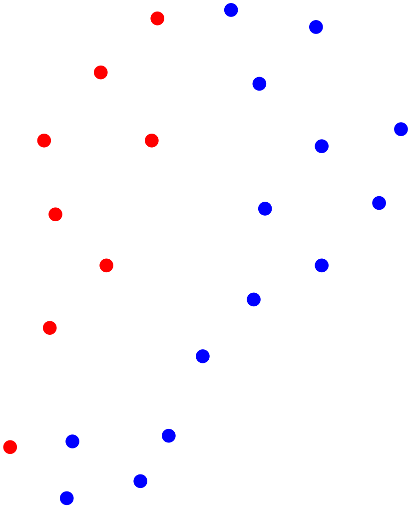
- Classification based on optimum-path forest[1, 2, 3].
- Its application to enhance objects in natural scenes[4].
- Clustering by optimum-path forest[5, 6].

# Organization of this lecture

- Classification based on optimum-path forest[1, 2, 3].
- Its application to enhance objects in natural scenes[4].
- Clustering by optimum-path forest[5, 6].
- Intelligent object enhancement based on clustering and classification[7].

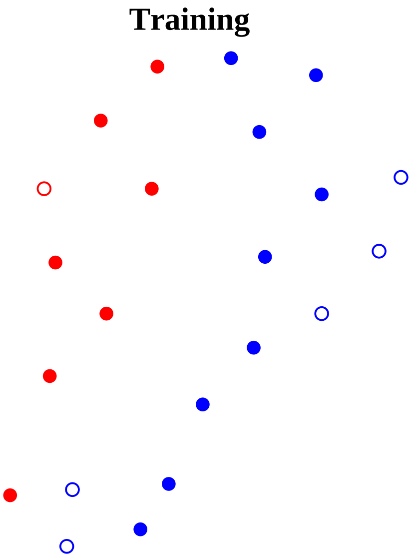
# Classification based on optimum-path forest

**Dataset**

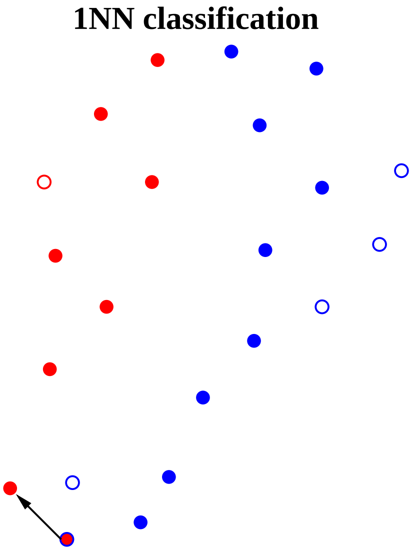


- Consider samples from two classes of a dataset.

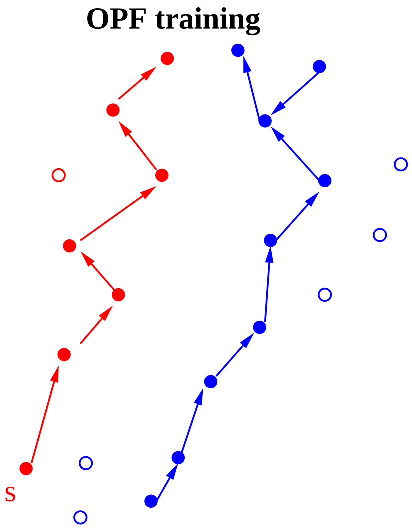
# Classification based on optimum-path forest




- Consider samples from two classes of a dataset.
- A training set (filled bullets) may not represent data distribution.

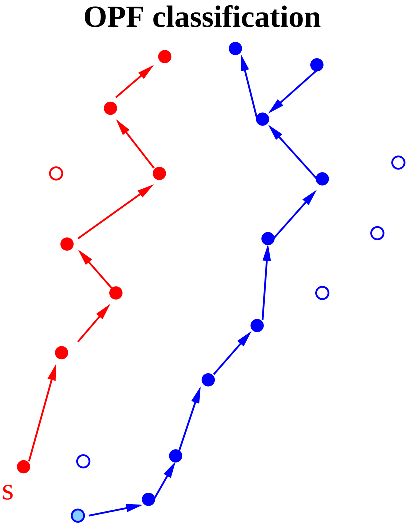



- Consider samples from two classes of a dataset.
- A training set (filled bullets) may not represent data distribution.
- Classification by nearest neighbor fails, when training samples are close to test samples (empty bullets) from other classes.




- We can create an optimum-path forest, where $V(s)$ is penalized when $s$ is not closely connected to its class.

# Classification based on optimum-path forest




- We can create an optimum-path forest, where $V(s)$ is penalized when $s$ is not closely connected to its class.
- $V(s)$ can then be used to reduce the power of $s$ to classify new samples.

# Supervised training

- We interpret $(\mathcal{T}, \mathcal{A})$ as a complete graph with undirected arcs between labeled training samples.

# Supervised training

- We interpret $(\mathcal{T}, \mathcal{A})$ as a complete graph with undirected arcs between labeled training samples.
- For a given set $\mathcal{S} \subset \mathcal{T}$ of prototypes from all classes, the connectivity map $V(t)$ is minimized for

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

where $d(s, t)$ is the distance between $s$ and $t$.

- We interpret $(\mathcal{T}, \mathcal{A})$ as a complete graph with undirected arcs between labeled training samples.
- For a given set $\mathcal{S} \subset \mathcal{T}$ of prototypes from all classes, the connectivity map $V(t)$ is minimized for

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

  where $d(s, t)$ is the distance between $s$ and $t$.
- The prototypes are the closest samples between classes.

# Supervised training algorithm

## Algorithm

– SUPERVISED TRAINING ALGORITHM

1. For each $t \in \mathcal{T} \backslash \mathcal{S}$, set $V(t) \leftarrow +\infty$.
2. For each $t \in \mathcal{S}$, set $L(t) \leftarrow \lambda(t)$, $V(t) \leftarrow 0$ and insert $t$ in $Q$.
3. While $Q$ is not empty, do
4. Remove from $Q$ a node $s$ such that $V(s)$ is minimum.
5. Insert $s$ in $\mathcal{T}'$.
6. For each $t \in \mathcal{T}$ such that $V(t) > V(s)$, do
7. Compute $tmp \leftarrow \max\{V(s), d(s,t)\}$.
8. If $tmp < V(t)$, then
9. If $V(t) \neq +\infty$, remove $t$ from $Q$.
10. Set $V(t) \leftarrow tmp$ and $L(t) \leftarrow L(s)$.
11. Insert $t$ in $Q$.

# Prototype estimation

The prototypes are the samples from distinct classes that share an arc in a minimum spanning tree (MST) computed on $(\mathcal{T}, \mathcal{A})$.

# Prototype estimation

The prototypes are the samples from distinct classes that share an arc in a minimum spanning tree (MST) computed on $(\mathcal{T}, \mathcal{A})$. The same algorithm

- with a non-smooth function

$$
\begin{aligned}
f_w(\langle t \rangle) &= \begin{cases} 0 & \text{for an arbitrary node } t \in \mathcal{T} \\ +\infty & \textit{otherwise}, \end{cases} \\
f_w(\pi_s \cdot \langle s, t \rangle) &= w(s, t),
\end{aligned}
$$

- and with $V(t) > V(s)$ in Line 6 replaced by $V(t) = +\infty$ or $t \in Q$

will output a MST.

# Label propagation

During classification of new samples:

- For any $t \in \mathcal{Z} \backslash \mathcal{T}$,

$$V(t) = \min_{\forall s \in \mathcal{T}} \{\max\{V(s), d(s,t)\}\}.$$

# Label propagation

During classification of new samples:

- For any $t \in \mathcal{Z} \backslash \mathcal{T}$,

$$V(t) = \min_{\forall s \in \mathcal{T}} \{\max\{V(s), d(s,t)\}\}.$$

- Let $s^* \in \mathcal{T}$ be the node that satisfies this equation, then the class of $t$ is assumed to be $L(s^*)$.

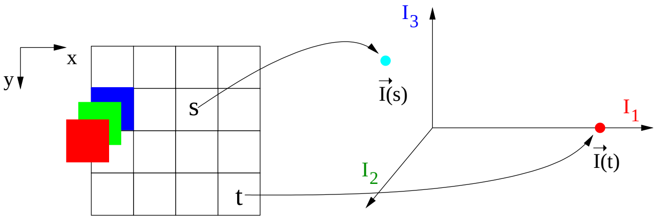
The role of the ordered set $\mathcal{T}'$ is to speed up classification [2, 3], which can halt when $\max\{V(s), d(s,t)\} < V(s')$ for a node $s'$ whose position in $\mathcal{T}'$ succeeds the position of $s$, while evaluating

$$V(t) = \min_{\forall s \in \mathcal{T}'} \{\max\{V(s), d(s,t)\}\}.$$

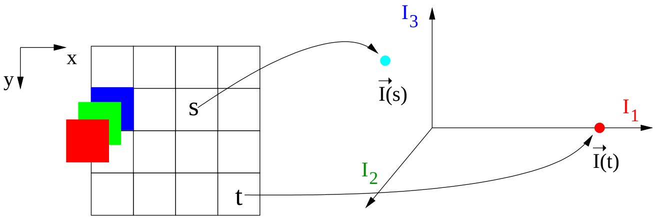
# Organization of this lecture

- Classification based on optimum-path forest.
- Its application to enhance objects in natural scenes.
- Clustering by optimum-path forest.
- Intelligent object enhancement based on clustering and classification.

Recall from lecture 1 that a spel $s \in \mathcal{D}_I$ is a point $\vec{I}(s) \in \mathcal{Z}^m$ in the RGB color space.



Recall from lecture 1 that a spel $s \in \mathcal{D}_I$ is a point $\vec{I}(s) \in \mathcal{Z}^m$ in the RGB color space.




Let $\Psi(\mathbf{I})$ be a color space transformation such that the resulting feature vector $\vec{v}(s) = (Y(s), Cb(s), Cr(s))$ corresponds to its values in the YCbCr color space (other feature spaces may be valid as well).

# Its application to object enhancement

For natural scenes, the distance function $d(s,t)$ between two spels $s$ and $t$ can be given by

$$\sqrt{\frac{1}{5}(Y(s)-Y(t))^2+(Cb(s)-Cb(t))^2+(Cr(s)-Cr(t))^2}$$

or any other function that puts more emphasis on crominance than luminance, in order to be as robust as possible to illumination variations.

# Its application to object enhancement

For a given image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ and adjacency relation $\mathcal{A}$ (e.g., 8-neighbors in 2D and 6-neighbors in 3D).

# Its application to object enhancement

For a given image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ and adjacency relation $\mathcal{A}$ (e.g., 8-neighbors in 2D and 6-neighbors in 3D).

- The success of segmentation strongly depends on the arc-weight estimation.

# Its application to object enhancement

For a given image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ and adjacency relation $\mathcal{A}$ (e.g., 8-neighbors in 2D and 6-neighbors in 3D).

- The success of segmentation strongly depends on the arc-weight estimation.
- The weight $0 \leq w(s,t) \leq K$ of an arc $(s,t) \in \mathcal{A}$ can be a linear combination

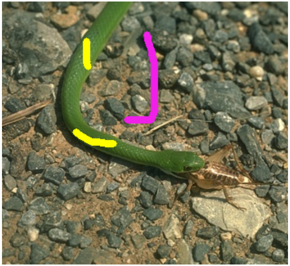
$$w(s,t) = \alpha w_o(s,t) + (1-\alpha) w_i(s,t),$$

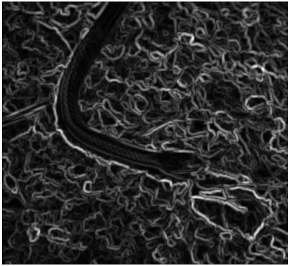
where

  - $w_o(s,t)$ takes into account object information (e.g., seed color), and
  - $w_i(s,t)$ takes into account local image features (e.g., color gradient).
  - $0 \leq \alpha \leq 1$ gives the importance of each component.

Arc weights can be visualized by a weight image $\mathbf{W} = (\mathcal{D}_I, W)$:

$$W(s) = \max_{\forall t \in \mathcal{A}(s)} \{w(s,t)\}$$



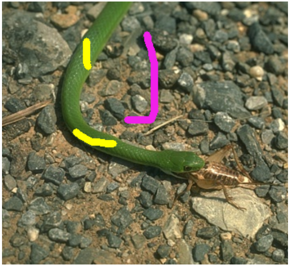


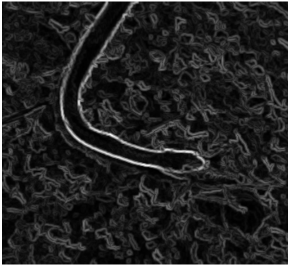
using $w_i(s,t)$ only, for $\alpha = 0.0$.

# Its application to object enhancement

Arc weights can be visualized by a weight image $\mathbf{W} = (\mathcal{D}_I, W)$:

$$W(s) = \max_{\forall t \in \mathcal{A}(s)} \{w(s,t)\}$$





using $w_o(s,t)$ and $w_i(s,t)$, for $\alpha = 0.8$.

# Its application to object enhancement

- The user selects internal and external seeds.

# Its application to object enhancement

- The user selects internal and external seeds.
- An object map $O(s)$ is created by fuzzy classification based on optimum-path forest.

# Its application to object enhancement

- The user selects internal and external seeds.
- An object map $O(s)$ is created by fuzzy classification based on optimum-path forest.
- Image-based $w_i(s,t)$ and object-based $w_o(s,t)$ arc weights are computed by

$$\begin{aligned} w_i(s,t) &\propto \|\vec{G}(s,t)\| \\ w_o(s,t) &\propto |O(t) - O(s)|. \end{aligned}$$

where $\vec{G}(s,t)$ is a color gradient at $(s,t)$.

# Its application to object enhancement

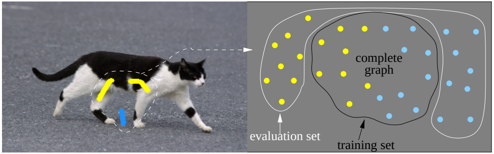
- The user selects internal and external seeds.
- An object map $O(s)$ is created by fuzzy classification based on optimum-path forest.
- Image-based $w_i(s,t)$ and object-based $w_o(s,t)$ arc weights are computed by

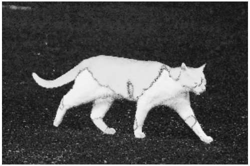
$$\begin{aligned} w_i(s,t) &\propto \|\vec{G}(s,t)\| \\ w_o(s,t) &\propto |O(t) - O(s)|. \end{aligned}$$

where $\vec{G}(s,t)$ is a color gradient at $(s,t)$.

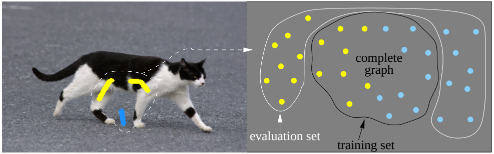
How do we compute the object map?

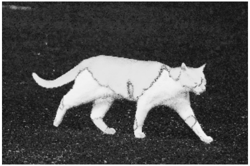
# Its application to object enhancement



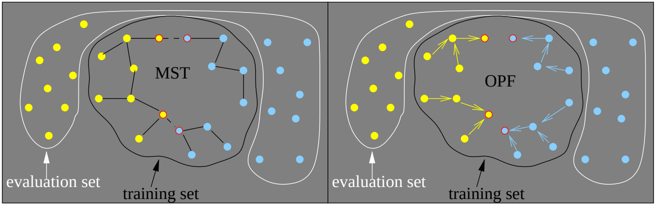
# Its application to object enhancement

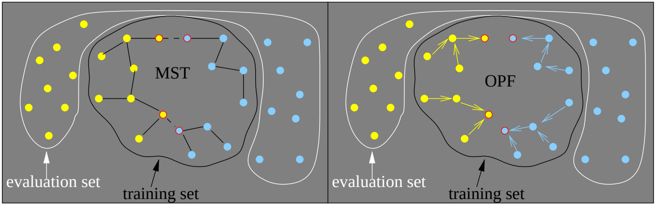


Even seed nodes may constitute large labeled sets, but they can be divided into a smaller training set $\mathcal{T}$ and a larger evaluation set $\mathcal{E}$ such that the most representative samples for $\mathcal{T}$ can be learned from $\mathcal{E}$.

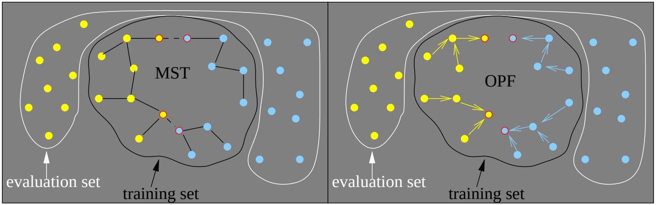
# Its application to object enhancement





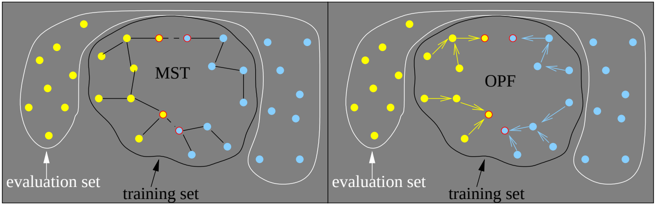

- A minimum spanning tree is computed in $(\mathcal{T}, \mathcal{A})$ and nodes that share arcs between distinct classes are taken as prototypes in $\mathcal{S}$.

# Its application to object enhancement



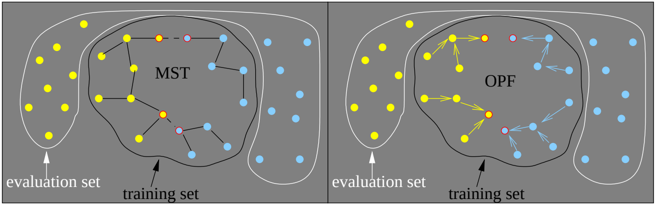

- A minimum spanning tree is computed in $(\mathcal{T}, \mathcal{A})$ and nodes that share arcs between distinct classes are taken as prototypes in $\mathcal{S}$.
- Object and background are then represented by optimum-path forests rooted in $\mathcal{S}$ (i.e., a spel classifier).

# Its application to object enhancement




- Prototypes compete among themselves and nodes in the evaluation set $\mathcal{E}$ are classified in the tree whose prototype offers an optimum path to it.

# Its application to object enhancement




- Prototypes compete among themselves and nodes in the evaluation set $\mathcal{E}$ are classified in the tree whose prototype offers an optimum path to it.
- Misclassified nodes in $\mathcal{E}$ are replaced by non-prototypes in $\mathcal{T}$ and the whole process is repeated for a few iterations in order to select the most representative nodes for $\mathcal{T}$.

# Its application to object enhancement

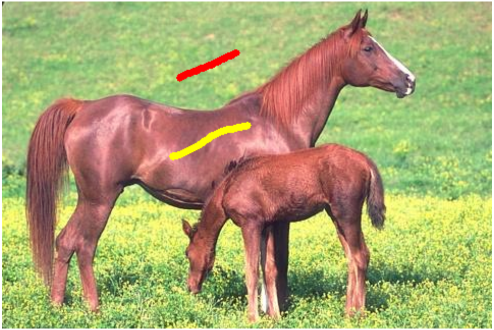
- Finally, the object map $O(t)$ can be created by fuzzy object classification.

# Its application to object enhancement

- Finally, the object map $O(t)$ can be created by fuzzy object classification.
- Let $V_o(t)$ and $V_b(t)$ be the optimum values for object and background forests, then a fuzzy object membership $\frac{V_b(t)}{V_o(t)+V_b(t)}$ can be assigned to every spel $t \in \mathcal{D}_I$.

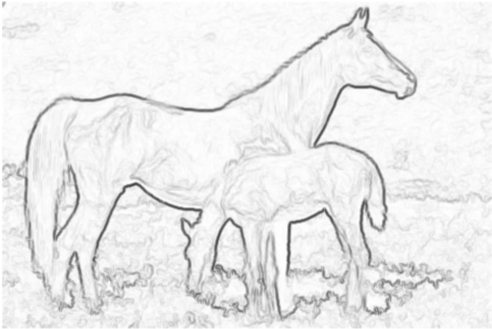
$$
\begin{aligned}
V_o(t) &= \min_{\forall s \in \mathcal{T}_o} \{\max\{V_o(s), d(s,t)\}\} \\
V_b(t) &= \min_{\forall s \in \mathcal{T}_b} \{\max\{V_b(s), d(s,t)\}\}
\end{aligned}
$$

for object $\mathcal{T}_o$ and background $\mathcal{T}_b$ seeds.



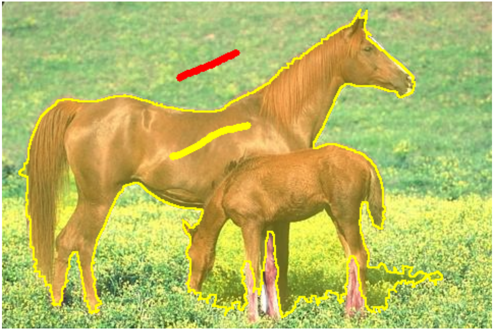
Intelligent seed selection, however, is required to reduce user involvement. The solution in [4] was to separate user interaction for enhancement and delineation. Can we avoid that?

# Need for intelligent object enhancement



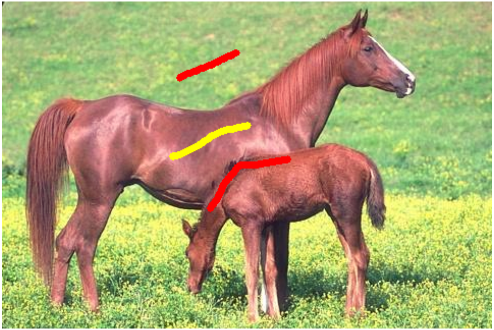
Intelligent seed selection, however, is required to reduce user involvement. The solution in [4] was to separate user interaction for enhancement and delineation. Can we avoid that?

# Need for intelligent object enhancement



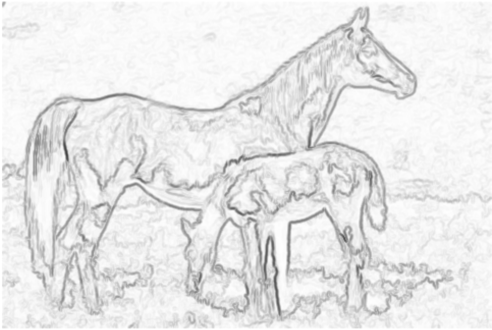
Intelligent seed selection, however, is required to reduce user involvement. The solution in [4] was to separate user interaction for enhancement and delineation. Can we avoid that?

# Need for intelligent object enhancement



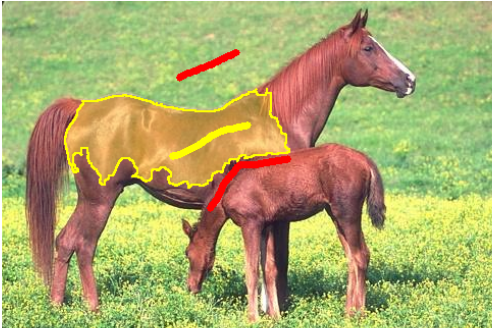
Intelligent seed selection, however, is required to reduce user involvement. The solution in [4] was to separate user interaction for enhancement and delineation. Can we avoid that?

# Need for intelligent object enhancement



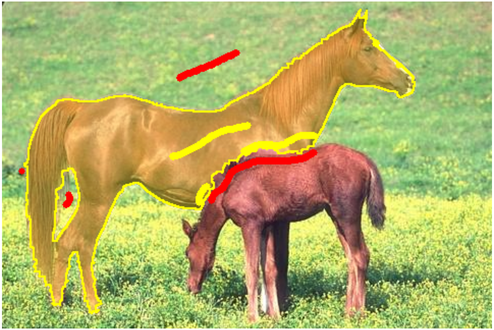
Intelligent seed selection, however, is required to reduce user involvement. The solution in [4] was to separate user interaction for enhancement and delineation. Can we avoid that?

# Need for intelligent object enhancement



Intelligent seed selection, however, is required to reduce user involvement. The solution in [4] was to separate user interaction for enhancement and delineation. Can we avoid that?

# Need for intelligent object enhancement



Intelligent seed selection, however, is required to reduce user involvement. The solution in [4] was to separate user interaction for enhancement and delineation. Can we avoid that?

# Object enhancement by PDFs

- The optimum-path forest algorithm can also be used to enhance object/background based on their probability density functions (PDFs).

# Object enhancement by PDFs

- The optimum-path forest algorithm can also be used to enhance object/background based on their probability density functions (PDFs).
- The PDFs $\rho(s \backslash \mathcal{T}_o)$ and $\rho(s \backslash \mathcal{T}_b)$ can be connectivity maps of the optimum-path forests obtained from object $\mathcal{T}_o$ and background $\mathcal{T}_b$ seeds, respectively.

# Object enhancement by PDFs

- The optimum-path forest algorithm can also be used to enhance object/background based on their probability density functions (PDFs).
- The PDFs $\rho(s\backslash\mathcal{T}_o)$ and $\rho(s\backslash\mathcal{T}_b)$ can be connectivity maps of the optimum-path forests obtained from object $\mathcal{T}_o$ and background $\mathcal{T}_b$ seeds, respectively.
- Object and background enhancement results from the density propagation (connectivity) to the remaining spels in $\mathcal{D}_I\backslash(\mathcal{T}_o \cup \mathcal{T}_b)$.

# Object enhancement by PDFs

- The optimum-path forest algorithm can also be used to enhance object/background based on their probability density functions (PDFs).
- The PDFs $\rho(s \backslash \mathcal{T}_o)$ and $\rho(s \backslash \mathcal{T}_b)$ can be connectivity maps of the optimum-path forests obtained from object $\mathcal{T}_o$ and background $\mathcal{T}_b$ seeds, respectively.
- Object and background enhancement results from the density propagation (connectivity) to the remaining spels in $\mathcal{D}_I \backslash (\mathcal{T}_o \cup \mathcal{T}_b)$.
- The PDF $\rho(s)$ can also be obtained by density propagation from an optimum-path forest computed on a random sample set $\mathcal{N} \subset \mathcal{D}_I$. In this case, it is useful for clustering and intelligent object enhancement.

# Object enhancement by PDFs

Let $\mathcal{T}_o \cup \mathcal{T}_b$ be a set of labeled seeds inside object (yellow) and background (red).

# Object enhancement by PDFs

Let $\mathcal{T}_o \cup \mathcal{T}_b$ be a set of labeled seeds inside object (yellow) and background (red).

The PDFs $\rho(s \backslash \mathcal{T}_o)$, $\rho(s \backslash \mathcal{T}_b)$, and $\rho(s)$, respectively.

# Object enhancement by PDFs

Let $\mathcal{T}_o \cup \mathcal{T}_b$ be a set of labeled seeds inside object (yellow) and background (red).

The PDFs $\rho(s \backslash \mathcal{T}_o)$, $\rho(s \backslash \mathcal{T}_b)$, and $\rho(s)$, respectively.

# Object enhancement by PDFs

Let $\mathcal{T}_o \cup \mathcal{T}_b$ be a set of labeled seeds inside object (yellow) and background (red).

The PDFs $\rho(s \backslash \mathcal{T}_o)$, $\rho(s \backslash \mathcal{T}_b)$, and $\rho(s)$, respectively.

# Object enhancement by PDFs

Let object and background be two classes $C_o$ and $C_b$, by Bayes' theorem:

$$
\begin{aligned}
P(s \in C_o \backslash \mathcal{T}) &= \frac{P(s \in C_o)\rho(s \backslash \mathcal{T}_o)}{\rho(s)}, \\
P(s \in C_b \backslash \mathcal{T}) &= \frac{P(s \in C_b)\rho(s \backslash \mathcal{T}_b)}{\rho(s)}, \\
\rho(s) &= P(s \in C_o)\rho(s \backslash \mathcal{T}_o) + P(s \in C_b)\rho(s \backslash \mathcal{T}_b).
\end{aligned}
$$

# Object enhancement by PDFs

Let object and background be two classes $C_o$ and $C_b$, by Bayes' theorem:

$$
\begin{aligned}
P(s \in C_o \backslash \mathcal{T}) &= \frac{P(s \in C_o)\rho(s \backslash \mathcal{T}_o)}{\rho(s)}, \\
P(s \in C_b \backslash \mathcal{T}) &= \frac{P(s \in C_b)\rho(s \backslash \mathcal{T}_b)}{\rho(s)}, \\
\rho(s) &= P(s \in C_o)\rho(s \backslash \mathcal{T}_o) + P(s \in C_b)\rho(s \backslash \mathcal{T}_b).
\end{aligned}
$$

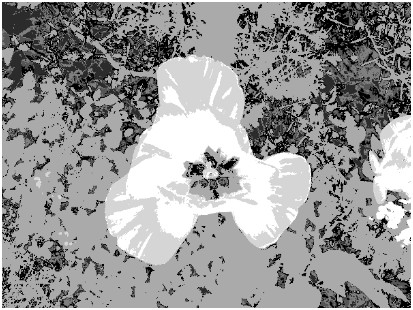
One could also try to compute object and background enhancement based on their posterior probabilities, as above, but the user-selected seeds are usually not suitable to estimate the prior probabilities $P(s \in C_o)$ and $P(s \in C_b)$.

# Organization of this lecture

- Classification based on optimum-path forest.
- Its application to enhance objects in natural scenes.
- Clustering by optimum-path forest.
- Intelligent object enhancement based on clustering and classification.

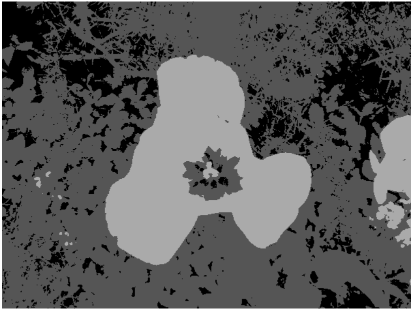
# Clustering by optimum-path-forest

- For clustering, we do not even count with seeds. We then compute one cluster for each dome of $\rho(s)$.



# Clustering by optimum-path-forest

- For clustering, we do not even count with seeds. We then compute one cluster for each dome of $\rho(s)$.



The number of groups is a scale-dependent parameter and, in this case, two groups do not separate object and background classes.

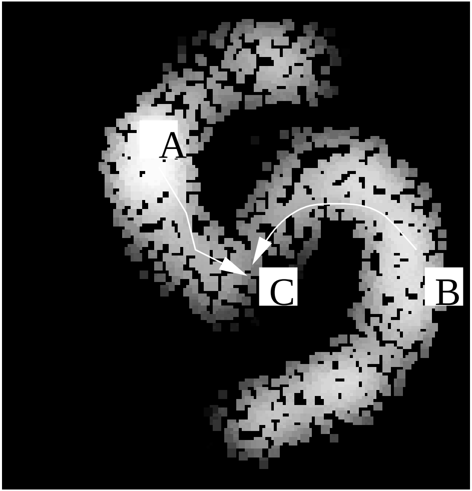
# Clustering by optimum-path-forest

- For clustering, we do not even count with seeds. We then compute one cluster for each dome of $\rho(s)$.

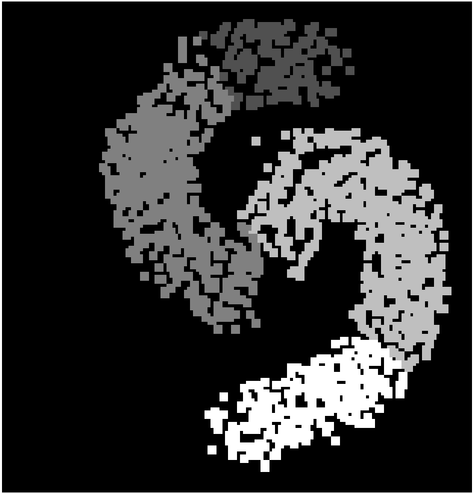
The number of groups is a scale-dependent parameter and, in this case, two groups do not separate object and background classes.

# Clustering by optimum-path-forest

The maxima of the pdf are the prototypes, but the number of maxima (clusters) reduces as the scale increases.

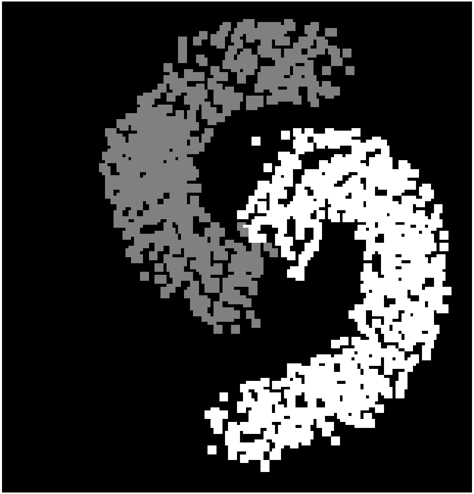


The maxima of the pdf are the prototypes, but the number of maxima (clusters) reduces as the scale increases.



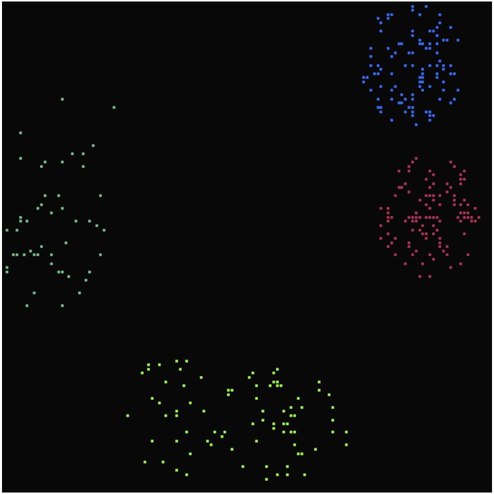
# Clustering by optimum-path-forest

The maxima of the pdf are the prototypes, but the number of maxima (clusters) reduces as the scale increases.



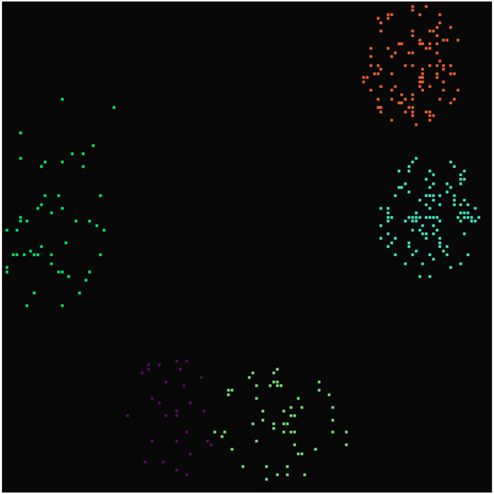
# Clustering by optimum-path-forest

The maxima of the pdf are the prototypes, but the number of maxima (clusters) reduces as the scale increases.



# Clustering by optimum-path-forest

The maxima of the pdf are the prototypes, but the number of maxima (clusters) reduces as the scale increases.



# Clustering by optimum-path-forest

- Since $\mathcal{D}_I$ is a large dataset, random samples will form an unlabeled training set $\mathcal{T}$.

# Clustering by optimum-path-forest

- Since $\mathcal{D}_I$ is a large dataset, random samples will form an unlabeled training set $\mathcal{T}$.
- We then define a knn-graph $(\mathcal{T}, \mathcal{A}_k)$ with adjacency relation

$$\mathcal{A}_k \quad : \quad (s,t) \in \mathcal{A}_k \text{ (or } t \in \mathcal{A}_k(s)) \text{ if } t \text{ is } k \text{ nearest neighbor of } s \text{ using the feature space.}$$

# Clustering by optimum-path-forest

- Since $\mathcal{D}_I$ is a large dataset, random samples will form an unlabeled training set $\mathcal{T}$.
- We then define a knn-graph $(\mathcal{T}, \mathcal{A}_k)$ with adjacency relation

$$\mathcal{A}_k \quad : \quad (s,t) \in \mathcal{A}_k \text{ (or } t \in \mathcal{A}_k(s)\text{) if } t \text{ is } k \text{ nearest neighbor of } s \text{ using the feature space.}$$

The best value of $k \in [k_{\min}, k_{\max}]$ is the one whose clustering produces a minimum normalized cut in $(\mathcal{T}, \mathcal{A}_k)$.

The graph is weighted on the arcs $(s,t) \in \mathcal{A}_k$ by $d(s,t)$ and on the nodes by the pdf $\rho(s)$.

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|\mathcal{A}_k(s)|} \sum_{\forall t \in \mathcal{A}_k(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right)$$

where $\sigma = \frac{d_f}{3}$ and $d_f = \max_{\forall(s,t)\in\mathcal{A}_k}\{d(s,t)\}$. The pdf is usually normalized within an interval $[1, K]$.

The connectivity map $V(t)$ is maximized for

$$
\begin{aligned}
f_{\min}(\langle t \rangle) &= \begin{cases} \rho(t) & \text{if } t \in \mathcal{R} \\ \rho(t) - 1 & \text{otherwise} \end{cases} \\
f_{\min}(\pi_s \cdot \langle s, t \rangle) &= \min\{f_{\min}(\pi_s), \rho(t)\}
\end{aligned}
$$

where $\mathcal{R}$ is the root set found on-the-fly and arcs are added in $\mathcal{A}_k$ to guarantee arc symmetry on the plateaus of the pdf.

# Unsupervised training algorithm

## Algorithm

– UNSUPERVISED TRAINING ALGORITHM

1. *Set* $lb \leftarrow 1$.
2. *For each* $s \in \mathcal{T}$, *set* $V(s) \leftarrow \rho(s) - 1$ *and insert* $s$ *in* $Q$.
3. *While* $Q$ *is not empty, do*
4.     *Remove from* $Q$ *a sample* $s$ *such that* $V(s)$ *is maximum*
5.     *Insert* $s$ *in* $\mathcal{T}'$.
6.     *If* $P(s) = nil$, *then*
7.         *Set* $L(s) \leftarrow lb$, $lb \leftarrow lb + 1$, *and* $V(s) \leftarrow \rho(s)$.
8.     *For each* $t \in \mathcal{A}_k(s)$ *and* $V(t) < V(s)$, *do*
9.         *Compute* $tmp \leftarrow \min\{V(s), \rho(t)\}$.
10.         *If* $tmp > V(t)$ *then*
11.             *Set* $L(t) \leftarrow L(s)$ *and* $V(t) \leftarrow tmp$.
12.             *Update position of* $t$ *in* $Q$.

# Fast label propagation

The role of the ordered set $\mathcal{T}'$ is to speed up label propagation to new nodes $t \in \mathcal{Z} \backslash \mathcal{T}$ [6], which can halt when $s^*$ is found in

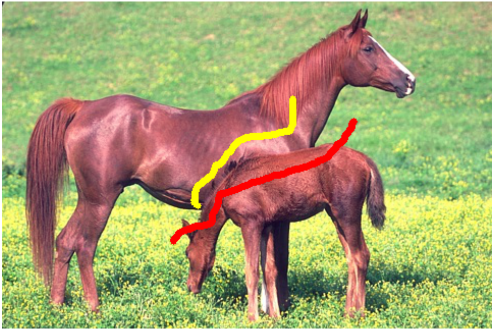
$$V(s^*) = \max_{\forall s \in \mathcal{T}' | d(s,t) \leq \omega(s)} \{V(s)\},$$

where $\omega(s)$ is the maximum distance between $s$ and its $k$-nearest neighbors in $\mathcal{T}$. The node $t$ then receives label $L(s^*)$.

# Intelligent object enhancement

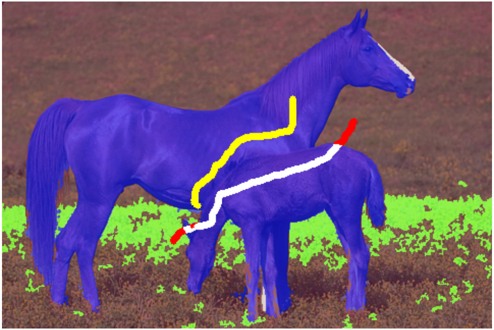
The user draws markers for delineation and the method removes the seeds from the background markers which fall in the same clusters of object marker seeds[7].

# Intelligent object enhancement



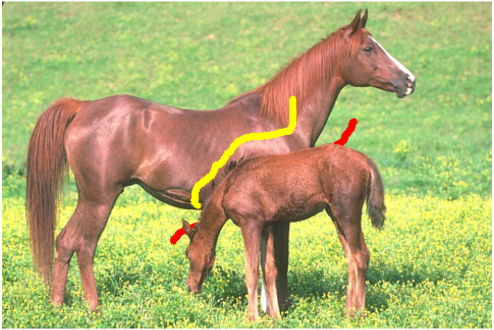
The user draws markers for delineation and the method removes the seeds from the background markers which fall in the same clusters of object marker seeds[7].

# Intelligent object enhancement



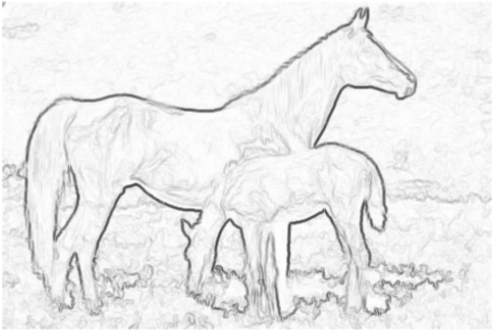
The user draws markers for delineation and the method removes the seeds from the background markers which fall in the same clusters of object marker seeds[7].

# Intelligent object enhancement



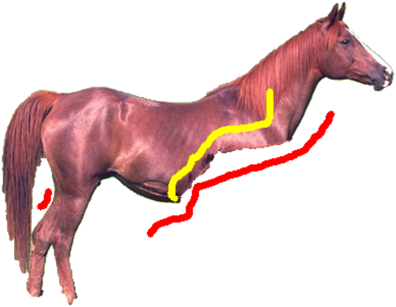
The user draws markers for delineation and the method removes the seeds from the background markers which fall in the same clusters of object marker seeds[7].

# Intelligent object enhancement



The user draws markers for delineation and the method removes the seeds from the background markers which fall in the same clusters of object marker seeds[7].

# Intelligent object enhancement



The user draws markers for delineation and the method removes the seeds from the background markers which fall in the same clusters of object marker seeds[7].

# Conclusion

- We presented the design of fast and effective clustering and classification methods based on optimum-path forest.

# Conclusion

- We presented the design of fast and effective clustering and classification methods based on optimum-path forest.
- These methods have been succeeded in several applications: CBIR[8, 9], video segmentation[10], biometry[11, 12], remote sensing[13], signal processing[14], etc.

# Conclusion

- We presented the design of fast and effective clustering and classification methods based on optimum-path forest.
- These methods have been succeeded in several applications: CBIR[8, 9], video segmentation[10], biometry[11, 12], remote sensing[13], signal processing[14], etc.
- Their C source code is available in `www.ic.unicamp.br/~afalcao/libopf`.

# Next lecture

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

Thanks for your attention

[1] J.P. Papa, A.X. Falcão, and C.T.N. Suzuki.
Supervised pattern classification based on optimum-path forest.
*Intl. Journal of Imaging Systems and Technology*, 19(2):120–131, Jun 2009.

[2] J. P. Papa, F. A. M. Cappabianco, and A. X. Falcão.
Optimizing optimum-path forest classification for huge datasets.
In *Proceedings of The 20th International Conference on Pattern Recognition*, pages 4162–4165, Istanbul, Turkey, Aug 2010.

[3] 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.

[4] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa.
Synergistic arc-weight estimation for interactive image segmentation using graphs.
*Computer Vision and Image Understanding*, 114(1):85–99, Jan 2010.

[5] 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.

[6] Fábio A.M. Cappabianco, A.X. Falcão, Clarissa L. Yasuda, and J. K. Udupa.

MR-Image Segmentation of Brain Tissues based on Bias Correction and Optimum-Path Forest Clustering.

Technical Report IC-10-07, Institute of Computing, University of Campinas, March 2010.

[7] T.V. Spina and A.X. Falcão.

Intelligent understanding of user input applied to arc-weight estimation for graph-based foreground segmentation.

In *23rd SIBGRAPI: Conference on Graphics, Patterns and Images*, pages 164–171, 2010.

[8] 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.

[9] 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.

[10] R. Minetto, T.V. Spina, A.X. Falc ao, N.J. Leite, J.P. Papa, and J. Stolfi.

Iftrace: Video segmentation of deformable objects using the image foresting transform.

*Computer Vision and Image Understanding*, 116(2):274 – 291, 2012.

[11] J.A. Montoya-Zegarra, J.P. Papa, N.J. Leite, R.S. Torres, and A.X. Falcão.

Novel approaches for exclusive and continuous fingerprint classification.

In *Proc. of the 3rd Pacific-Rim Symposium on Image and Video Technology (PSVIT 2009)*, volume LNCS 5414, pages 386–397, Tokyo, Japan, 2008. Springer.

[12] G. Chiachia, A.N. Marana, J.P. Papa, and A.X. Falcão.
Infrared face recogniton by optimum-path forest.
In *Proc. of the 16th Intl. Conf. on Systems, Signals, and Image Processing*, Chalkida, Greece, 2009. IEEE.

[13] R.J. Pisani, J.P. Papa, C.R.L. Zimback, A.X. Falcão, and A.P. Barbosa.
Land use classification using optimum-path forest.
In *XIV Brazilian Remote Sensing Symposium (SBSR)*, pages 7063–7070, Natal-RN, 2009. MCT-INPE.

[14] A.I. Iliev, M.S. Scordilis, J.P. Papa, and A.X. Falcão.
Spoken emotion recognition through optimum-path forest classification using glottal features.
*Computer Speech and Language*, 24(3):445–460, Jul 2010.