Image Processing using Graphs (lecture 3 - image segmentation)

Alexandre Xavier Falcão

Visual Informatics Laboratory - Institute of Computing - University of Campinas

afalcao@ic.unicamp.br

www.ic.unicamp.br/~afalcao/talks.html

Introduction

Segmentation is a challenging problem which consists of two tightly coupled tasks: recognition and delineation.

Introduction

Segmentation is a challenging problem which consists of two tightly coupled tasks: recognition and delineation.

- Recognition consists of indicating the approximate object's location (humans $\gg$ computers).

Introduction

Segmentation is a challenging problem which consists of two tightly coupled tasks: recognition and delineation.

- Recognition consists of indicating the approximate object's location (humans $\gg$ computers).
- Delineation consists of defining the precise spatial extent of the object in the image (computers $\gg$ humans).

Introduction

Segmentation is a challenging problem which consists of two tightly coupled tasks: recognition and delineation.

- Recognition consists of indicating the approximate object's location (humans $\gg$ computers).
- Delineation consists of defining the precise spatial extent of the object in the image (computers $\gg$ humans).

Introduction

Segmentation is a challenging problem which consists of two tightly coupled tasks: recognition and delineation.

- Recognition consists of indicating the approximate object's location (humans $\gg$ computers).
- Delineation consists of defining the precise spatial extent of the object in the image (computers $\gg$ humans).

- Delineation was solved by spel classification, so connectivity was not needed.

When do we need connectivity?

Simple connectivity is needed for delineation when object and other disconnected parts of the background have similar properties.

When do we need connectivity?

Simple connectivity is needed for delineation when object and other disconnected parts of the background have similar properties.

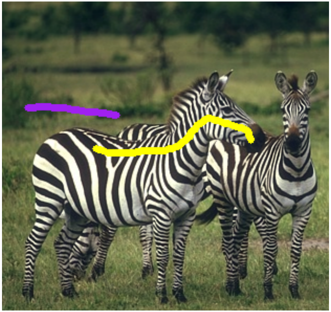
When do we need connectivity?

Simple connectivity is needed for delineation when object and other disconnected parts of the background have similar properties.

However, when do we need optimum connectivity?

When do we need optimum connectivity?

Optimum connectivity is needed for delineation when object and parts of the background with similar properties are connected to each other.



When do we need optimum connectivity?

Optimum connectivity is needed for delineation when object and parts of the background with similar properties are connected to each other.

When do we need optimum connectivity?

Optimum connectivity is needed for delineation when object and parts of the background with similar properties are connected to each other.

In this case, however, the markers needed to disconnect the object are not suitable for classification.

Our methodology

We have exploited

Our methodology

We have exploited

- fuzzy spel classification for arc-weight estimation in IFT-based segmentation methods,

Our methodology

We have exploited

- fuzzy spel classification for arc-weight estimation in IFT-based segmentation methods,
- a synergism between recognition by a human operator and IFT-based delineation in interactive segmentation, and

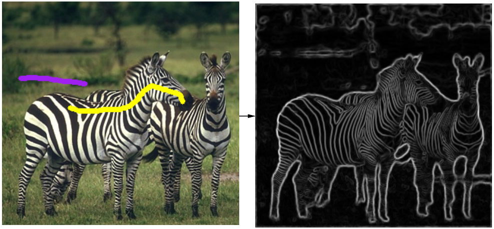
Our methodology

We have exploited

- fuzzy spel classification for arc-weight estimation in IFT-based segmentation methods,
- a synergism between recognition by a human operator and IFT-based delineation in interactive segmentation, and
- a synergism between recognition by object models and IFT-based delineation in automatic segmentation.

Organization of this lecture

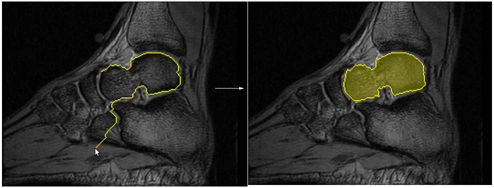
This lecture will cover interactive and automatic segmentation methods.



- Arc-weight estimation [1, 2].

Organization of this lecture

This lecture will cover interactive and automatic segmentation methods.



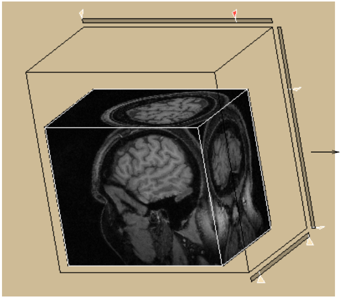
- Arc-weight estimation [1, 2].
- Boundary-based delineation [3].

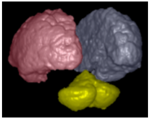
This lecture will cover interactive and automatic segmentation methods.

- Arc-weight estimation [1, 2].
- Boundary-based delineation [3].
- Region-based delineation [4, 5].

Organization of this lecture

This lecture will cover interactive and automatic segmentation methods.





- Arc-weight estimation [1, 2].
- Boundary-based delineation [3].
- Region-based delineation [4, 5].
- Cloud system model (CSM) [6].

Arc-weight estimation

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).

Arc-weight estimation

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, which affects path function $f(\pi_t)$.

Arc-weight estimation

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, which affects path function $f(\pi_t)$.
- The weight $0 \leq w(s,t) \leq K$ of an arc $(s,t) \in \mathcal{A}$ can be a linear combination

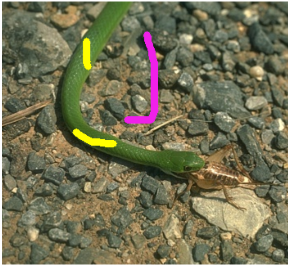
$$w(s,t) = \alpha w_o(s,t) + (1-\alpha) w_i(s,t),$$

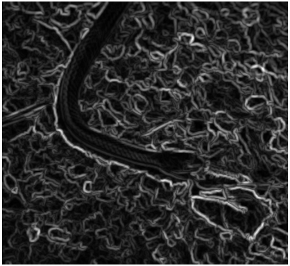
  where
  - $w_o(s,t)$ takes into account object information (e.g., its features, markers, model), and
  - $w_i(s,t)$ takes into account local image features.
  - $0 \leq \alpha \leq 1$ gives the importance of each component.

Arc-weight estimation

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

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

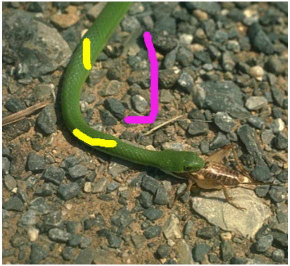


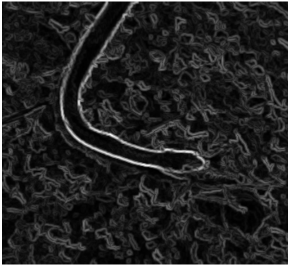


using $w_i(s,t)$ only, for $\alpha = 0.0$.

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 $\alpha = 0.8$.

Arc-weight estimation

For interactive segmentation of natural images:

- Object-based weights $w_o(s,t)$ can be derived from
  - color image features,
  - user-drawn markers, and
  - fuzzy spel classification.

Arc-weight estimation

For interactive segmentation of natural images:

- Object-based weights $w_o(s,t)$ can be derived from
  - color image features,
  - user-drawn markers, and
  - fuzzy spel classification.
- Image-based weights $w_i(s,t)$ use only the local values of the same image features.

Arc-weight estimation

For interactive segmentation of natural images:

- Object-based weights $w_o(s,t)$ can be derived from
  - color image features,
  - user-drawn markers, and
  - fuzzy spel classification.
- Image-based weights $w_i(s,t)$ use only the local values of the same image features.
- Markers used for IFT delineation are never used for arc-weight estimation.

Arc-weight estimation

We may

- use connected filters to reduce noise,

Arc-weight estimation

We may

- use connected filters to reduce noise,
- convert image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ from the RGB to the Lab color space, creating a feature image $\mathbf{F} = (\mathcal{D}_I, \vec{F})$,

Arc-weight estimation

We may

- use connected filters to reduce noise,
- convert image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ from the RGB to the Lab color space, creating a feature image $\mathbf{F} = (\mathcal{D}_I, \vec{F})$,
- create an object map $\mathbf{O} = (\mathcal{D}_I, O)$ by fuzzy classification, and

Arc-weight estimation

We may

- use connected filters to reduce noise,
- convert image $\mathbf{I} = (\mathcal{D}_I, \vec{I})$ from the RGB to the Lab color space, creating a feature image $\mathbf{F} = (\mathcal{D}_I, \vec{F})$,
- create an object map $\mathbf{O} = (\mathcal{D}_I, O)$ by fuzzy classification, and
- define image-based and object-based weights by

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

where $O(s)$ is higher when s is an object spel.

Fuzzy spel classification

- For a given image.

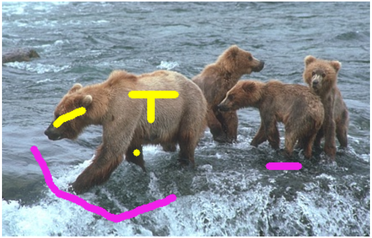
Fuzzy spel classification

- For a given image.
- Training markers are drawn in distinct parts of object and background.

Fuzzy spel classification

- For a given image.
- Training markers are drawn in distinct parts of object and background.
- Classification gives a visual feedback to guide new marker selection (synergism).

Fuzzy spel classification



- For a given image.
- Training markers are drawn in distinct parts of object and background.
- Classification gives a visual feedback to guide new marker selection (synergism).
- As markers are added, classification improves the object map.

The arc weights $w(s,t)$ or their complement $\bar{w}(s,t)$ may be used in several graph-based segmentation methods.
We will discuss

The arc weights $w(s,t)$ or their complement $\bar{w}(s,t)$ may be used in several graph-based segmentation methods.
We will discuss

- boundary-based delineation by live-wire-on-the-fly [3],

The arc weights $w(s,t)$ or their complement $\bar{w}(s,t)$ may be used in several graph-based segmentation methods.
We will discuss

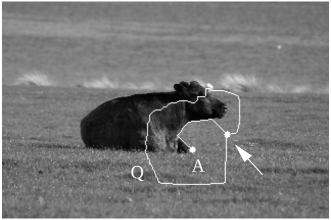
- boundary-based delineation by live-wire-on-the-fly [3],
- region-based delineation using the differential IFT algorithm with seed competition (IFTSC) [4], and

The arc weights $w(s,t)$ or their complement $\bar{w}(s,t)$ may be used in several graph-based segmentation methods.
We will discuss

- boundary-based delineation by live-wire-on-the-fly [3],
- region-based delineation using the differential IFT algorithm with seed competition (IFTSC) [4], and
- a comparative analysis between IFTSC and segmentation based on the min-cut/max-flow algorithm [5].

Live-wire-on-the-fly (LWOF)

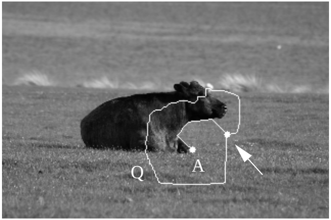
Optimum paths are incrementally computed from the wavefront Q.



- The user selects a point A on the object's boundary, and

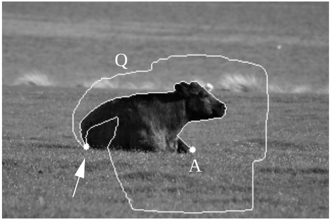
Live-wire-on-the-fly (LWOF)

Optimum paths are incrementally computed from the wavefront Q.



- The user selects a point A on the object's boundary, and
- for any subsequent position of the cursor, an optimum path from A to that position is displayed in real time.

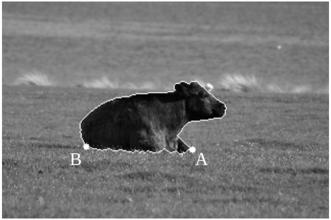
Optimum paths are incrementally computed from the wavefront Q.



- The user selects a point A on the object's boundary, and
- for any subsequent position of the cursor, an optimum path from A to that position is displayed in real time.
- When the cursor is close to the boundary, the path snaps on to it.

Live-wire-on-the-fly (LWOF)

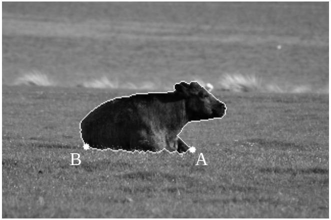
Optimum paths are incrementally computed from the wavefront Q.



- The user selects a point A on the object's boundary, and
- for any subsequent position of the cursor, an optimum path from A to that position is displayed in real time.
- When the cursor is close to the boundary, the path snaps on to it.
- The user may accept it as a boundary segment, and

Live-wire-on-the-fly (LWOF)

Optimum paths are incrementally computed from the wavefront Q.



- The user selects a point A on the object's boundary, and
- for any subsequent position of the cursor, an optimum path from A to that position is displayed in real time.
- When the cursor is close to the boundary, the path snaps on to it.
- The user may accept it as a boundary segment, and
- the process is repeated from its terminus B until the user decides to close the contour.

Live-wire-on-the-fly (LWOF)

The IFT algorithm with early termination and function f_{sum} finds optimum paths from a starting point s^* on anti-clockwise oriented boundaries.

$$
\begin{aligned}
f_{sum}(\langle t\rangle) &= \begin{cases} 0 & \text{if } t = s^* \\ +\infty & \text{otherwise} \end{cases} \\
f_{sum}(\pi_s \cdot \langle s, t\rangle) &= \begin{cases} f_{sum}(\pi_s) + \bar{w}^{\beta}(s,t) & \text{if } O(l) \geq O(r) \\ f_{sum}(\pi_s) + K^{\beta} & \text{otherwise,} \end{cases}
\end{aligned}
$$

where l and r are the spels at the left and right sides of arc $\langle s, t\rangle$. The weights $\bar{w}(s,t)$ are lower on the boundary than inside and outside it and $\beta \geq 1$ favors longer segments. (Show software)

Path computation from s^* to t in LWOF

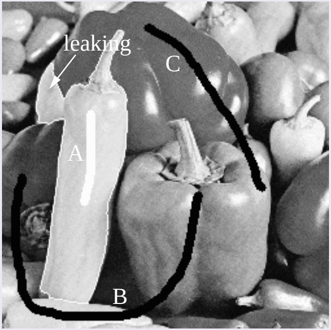
Algorithm

– PATH COMPUTATION FROM s^* TO t IN LWOF

If $V(t) = +\infty$ *or* $t \in Q$, **then**
    While Q *is not empty, do*
        Remove from Q *a spel* s *such that* $V(s)$ *is minimum.*
        If $s = t$ *then return.*
        For each $t \in \mathcal{A}(s)$ *such that* $V(t) > V(s)$, *do*
            If $O(l) \geq O(r)$,
                then *set* $tmp \leftarrow V(s) + \bar{w}^{\beta}(s,t)$
                Else *set* $tmp \leftarrow V(s) + K^{\beta}$.
            If $tmp < V(t)$, **then**
                If $V(t) \neq +\infty$, *remove* t *from* Q.
                Set $P(t) \leftarrow s$ *and* $V(t) \leftarrow tmp$.
                Insert t *in* Q.

Differential IFT with seed competition (IFTSC)

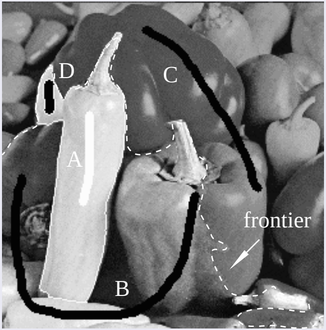
According to lecture 1, the object can also be defined by using the differential IFT algorithm with seed competition [4].



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

Differential IFT with seed competition (IFTSC)

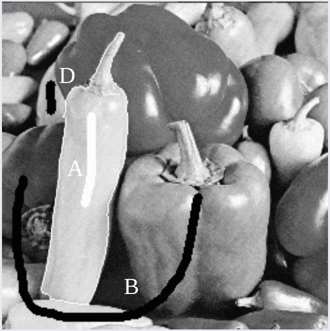
According to lecture 1, the object can also be defined by using the differential IFT algorithm with seed competition [4].



- 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.

Differential IFT with seed competition (IFTSC)

According to lecture 1, the object can also be defined by using the differential IFT algorithm with seed competition [4].



- 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.

Differential IFT with seed competition (IFTSC)

In this case, the object is an optimum-path forest for $f_{\max}$ rooted at internal seeds.

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

where $\mathcal{S}_i$ and $\mathcal{S}_e$ are internal and external seed sets.

Differential IFT with $f_{\max}$ and seed competition (IFTSC)

Algorithm

– ALGORITHM IFTSC

$(V, P, L, \mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{S}) \leftarrow$DIFT-FORESTREMOVAL$(V, P, L, \mathcal{A}, \mathcal{R}_{\mathcal{M}})$.
While $\mathcal{S} \neq \emptyset$, *remove* t *from* $\mathcal{S}$, *set* $V(t) \leftarrow 0$,
    set $L(t) \leftarrow \lambda(t)$, $P(t) \leftarrow nil$, *and* $\mathcal{F} \leftarrow \mathcal{F} \cup \{t\}$.
While $\mathcal{F} \neq \emptyset$, *remove* t *from* $\mathcal{F}$ *and insert* t *in* Q.
While Q *is not empty* **do**
    Remove s *from* Q *such that* $V(s)$ *is minimum.*
    For each $t \in \mathcal{A}(s)$, **do**
        Compute $tmp \leftarrow \max\{V(s), w(s,t)\}$.
        If $tmp < V(t)$ *or* $P(t) = s$, **then**
            If $t \in Q$, **then** *remove* t *from* Q.
            Set $P(t) \leftarrow s$, $V(t) \leftarrow tmp$, $R(t) \leftarrow R(s)$.
            Insert t *in* Q.

Differential IFT with seed competition (IFTSC)

As advantages, this approach

- can handle multiple objects in sublinear time,

Differential IFT with seed competition (IFTSC)

As advantages, this approach

- can handle multiple objects in sublinear time,
- is easily extended to 3D images, and

Differential IFT with seed competition (IFTSC)

As advantages, this approach

- can handle multiple objects in sublinear time,
- is easily extended to 3D images, and
- is much faster and more robust than segmentation methods based on the min-cut/max-flow algorithm.

IFTSC as a graph cut segmentation

From lecture 1, we know that an optimum-path forest for $f_{\max}$ provides the graph cut whose minimum arc weight

$$\min_{\forall (s,t) \in \mathcal{A}, L(s)=1, L(t)=0} w(s,t)$$

is maximum, considering all possible cuts between internal and external seeds [5].

IFTSC as a graph cut segmentation

From lecture 1, we know that an optimum-path forest for $f_{\max}$ provides the graph cut whose minimum arc weight

$$\min_{\forall (s,t)\in\mathcal{A}, L(s)=1, L(t)=0} w(s,t)$$

is maximum, considering all possible cuts between internal and external seeds [5].

- In fact, the boundaries obtained by IFTSC are also piecewise optimum.

IFTSC as a graph cut segmentation

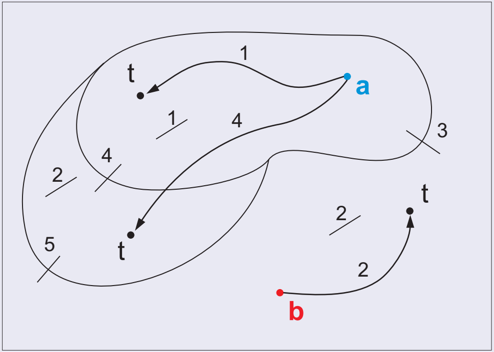
From lecture 1, we know that an optimum-path forest for $f_{\max}$ provides the graph cut whose minimum arc weight

$$\min_{\forall (s,t)\in\mathcal{A}, L(s)=1, L(t)=0} w(s,t)$$

is maximum, considering all possible cuts between internal and external seeds [5].

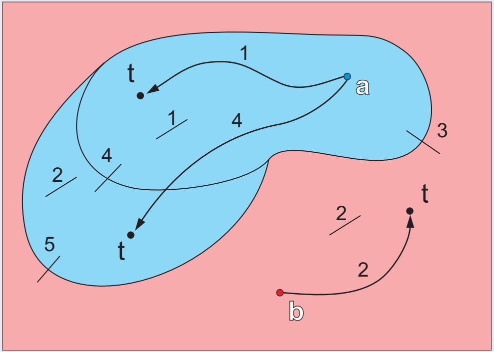
- In fact, the boundaries obtained by IFTSC are also piecewise optimum.
- That is, the above optimization holds to any part of the segmented boundary.

IFTSC as a graph cut segmentation



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 value 3.

IFTSC as a graph cut segmentation



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 value 3.

Segmentation by the min-cut/max-flow algorithm

- Methods based on the min-cut/max-flow algorithm usually aim to minimize the sum of arc weights $\bar{w}^{\beta}(s,t)$ in the cut boundary.

$$\sum_{\forall (s,t)\in\mathcal{A} \mid \; L(s)=1, L(t)=0} \bar{w}^{\beta}(s,t),$$

for $\beta \geq 1$, where lower values of β favor undesirable small cuts.

Segmentation by the min-cut/max-flow algorithm

- Methods based on the min-cut/max-flow algorithm usually aim to minimize the sum of arc weights $\bar{w}^{\beta}(s,t)$ in the cut boundary.

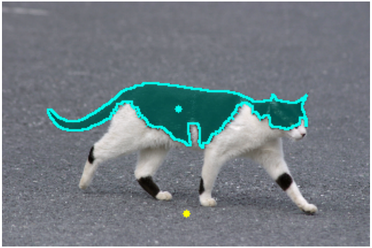
$$\sum_{\forall (s,t)\in\mathcal{A} \mid \; L(s)=1, L(t)=0} \bar{w}^{\beta}(s,t),$$

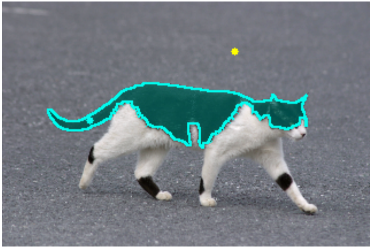
  for $\beta \geq 1$, where lower values of β favor undesirable small cuts.
- For higher values of β, the above optimization tends to minimize the maximum arc weight $\bar{w}(s,t)$ in the cut boundary,

$$\max_{\forall (s,t)\in\mathcal{A} \mid \; L(s)=1, L(t)=0} \bar{w}(s,t),$$

  which is essentially the same of maximizing the minimum arc weight $w(s,t)$ in the cut boundary, as done by IFTSC.

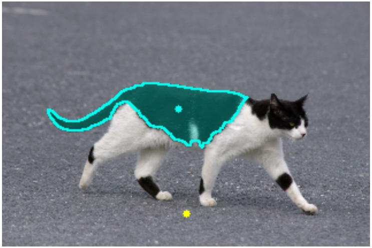
A same connected component is always obtained with IFTSC, independently of seed location.

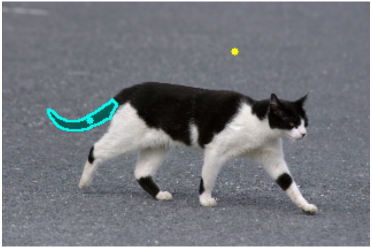




Robustness of IFTSC

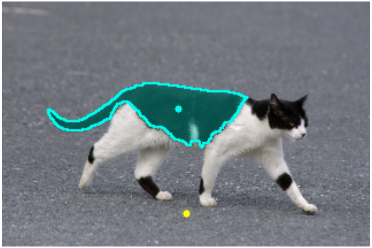
The same does not happen in the segmentation by the min-cut/max-flow algorithm, when β is not high enough.

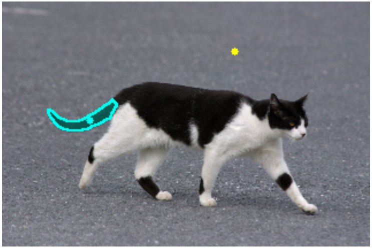




Robustness of IFTSC

The same does not happen in the segmentation by the min-cut/max-flow algorithm, when β is not high enough.





Therefore, the best result of the min-cut/max-flow algorithm is the one obtained by the IFTSC algorithm. Show software.

The cloud system model

- Automatic segmentation is feasible when the user can be substituted by some prior information, such as an object model.

The cloud system model

- Automatic segmentation is feasible when the user can be substituted by some prior information, such as an object model.
- We will present the cloud system model (CSM), which

The cloud system model

- Automatic segmentation is feasible when the user can be substituted by some prior information, such as an object model.
- We will present the cloud system model (CSM), which
  - represents shape variations of multiple objects with respect to a common reference point,

The cloud system model

- Automatic segmentation is feasible when the user can be substituted by some prior information, such as an object model.
- We will present the cloud system model (CSM), which
  - represents shape variations of multiple objects with respect to a common reference point,
  - does not require landmarks, point correspondences or deformable registration, only

The cloud system model

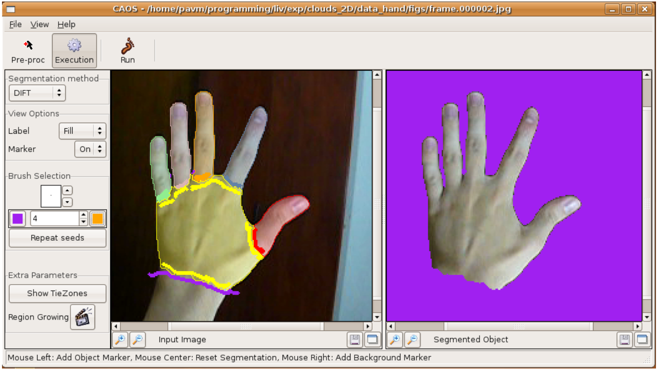
- Automatic segmentation is feasible when the user can be substituted by some prior information, such as an object model.
- We will present the cloud system model (CSM), which
  - represents shape variations of multiple objects with respect to a common reference point,
  - does not require landmarks, point correspondences or deformable registration, only
  - requires translation, IFTSC delineation and evaluation of delineation by a criterion function, and

The cloud system model

- Automatic segmentation is feasible when the user can be substituted by some prior information, such as an object model.
- We will present the cloud system model (CSM), which
  - represents shape variations of multiple objects with respect to a common reference point,
  - does not require landmarks, point correspondences or deformable registration, only
  - requires translation, IFTSC delineation and evaluation of delineation by a criterion function, and
  - exploits model orientation in arc-weight estimation for f_{max}.

The cloud system model

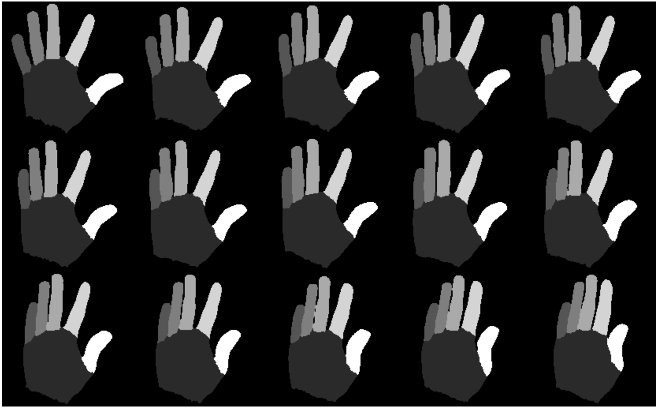
A set of training objects is first provided by interactive IFTSC segmentation.



Each image with multiple objects forms an object system with a common reference point (e.g., the geometric center of the objects).

The cloud system model

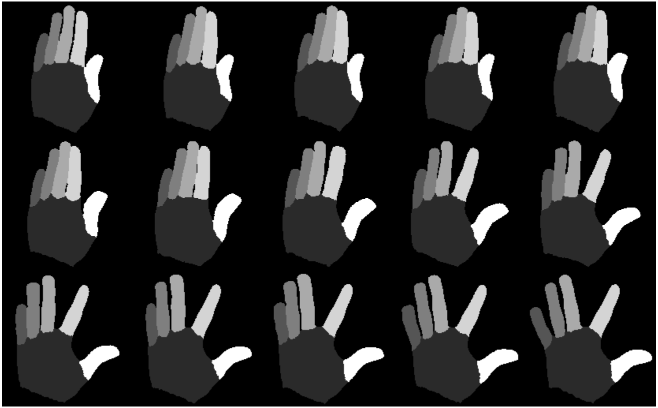
A set of training objects is first provided by interactive IFTSC segmentation.



Each image with multiple objects forms an object system with a common reference point (e.g., the geometric center of the objects).

The cloud system model

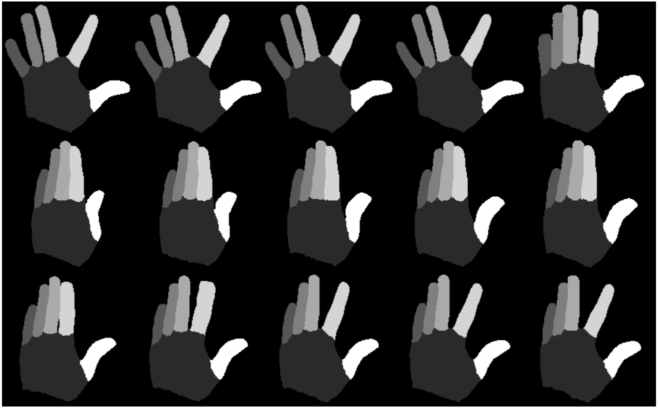
A set of training objects is first provided by interactive IFTSC segmentation.



Each image with multiple objects forms an object system with a common reference point (e.g., the geometric center of the objects).

The cloud system model

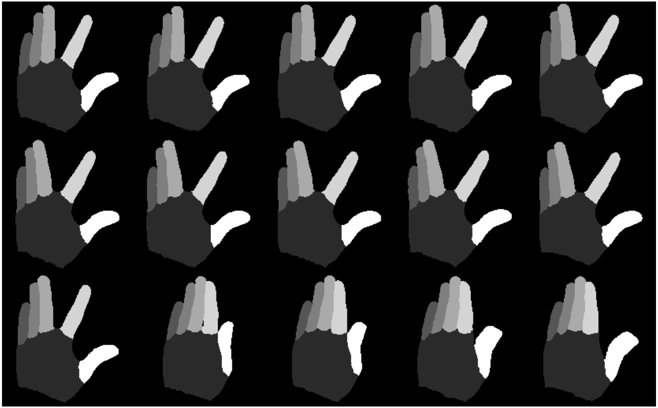
A set of training objects is first provided by interactive IFTSC segmentation.



Each image with multiple objects forms an object system with a common reference point (e.g., the geometric center of the objects).

The cloud system model

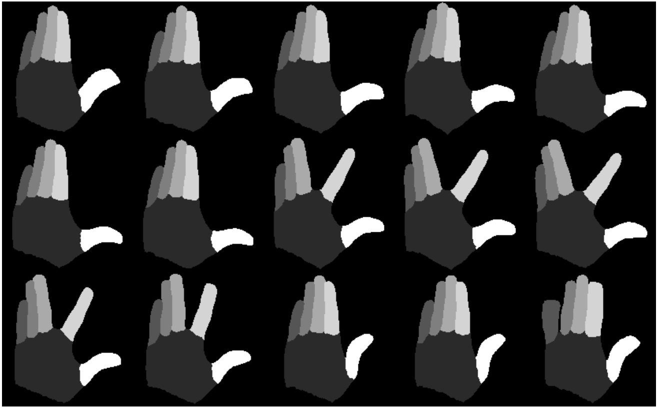
A set of training objects is first provided by interactive IFTSC segmentation.



Each image with multiple objects forms an object system with a common reference point (e.g., the geometric center of the objects).

The cloud system model

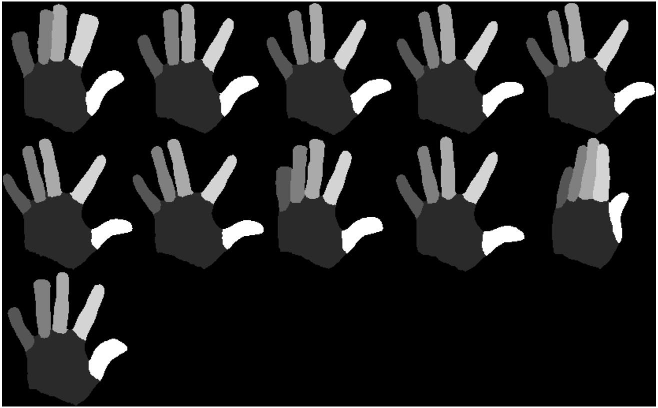
A set of training objects is first provided by interactive IFTSC segmentation.



Each image with multiple objects forms an object system with a common reference point (e.g., the geometric center of the objects).

The cloud system model

A set of training objects is first provided by interactive IFTSC segmentation.



Each image with multiple objects forms an object system with a common reference point (e.g., the geometric center of the objects).

The cloud system model

Groups of object systems in which the corresponding objects have similar shapes, sizes and positions form different cloud system models, as follows.

The cloud system model

Groups of object systems in which the corresponding objects have similar shapes, sizes and positions form different cloud system models, as follows.

- Each object system becomes a node of a complete graph, where the weight of each arc derives from the similarities between the corresponding objects in shape, size and position.

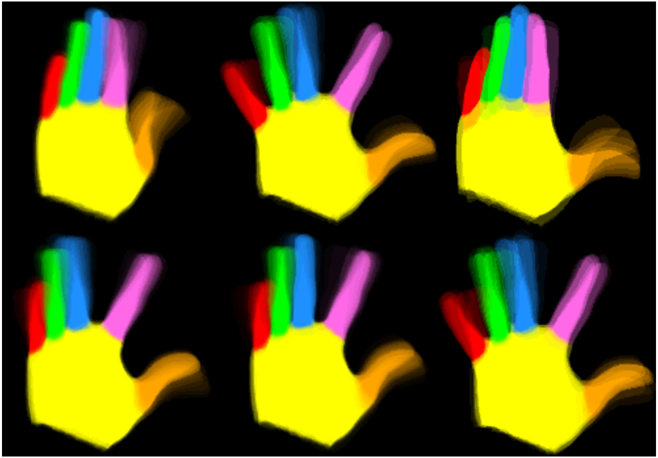
The cloud system model

Groups of object systems in which the corresponding objects have similar shapes, sizes and positions form different cloud system models, as follows.

- Each object system becomes a node of a complete graph, where the weight of each arc derives from the similarities between the corresponding objects in shape, size and position.
- The groups are found as maximal cliques in which all arc weights are higher than a threshold.

The cloud system model

The object systems in each group are finally translated to a same reference point and the corresponding object masks are averaged, forming a set of cloud systems.



A cloud system model (CSM) then consists of three elements:

A cloud system model (CSM) then consists of three elements:

- A probabilistic map (object clouds), which indicates an object uncertainty region with values strictly lower than 1 and higher than 0.

The cloud system model

A cloud system model (CSM) then consists of three elements:

- A probabilistic map (object clouds), which indicates an object uncertainty region with values strictly lower than 1 and higher than 0.
- A delineation algorithm (IFTSC), whose execution is constrained in the uncertainty region.

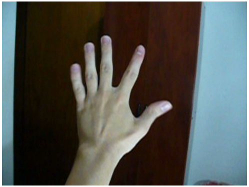
The cloud system model

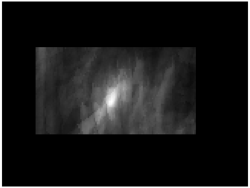
A cloud system model (CSM) then consists of three elements:

- A probabilistic map (object clouds), which indicates an object uncertainty region with values strictly lower than 1 and higher than 0.
- A delineation algorithm (IFTSC), whose execution is constrained in the uncertainty region.
- A criterion function, which assigns a score to any set of delineated objects.

The cloud system model

Segmentation using CSM consists of a search for the translation to the image location which produces the highest score, when the reference point of the most suitable cloud system is at that position.

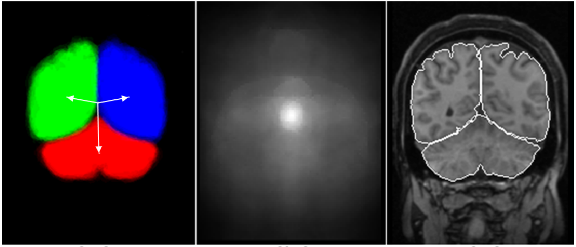




show video handsearch.mpg

Brain structure segmentation

In MR-brain images, 3D objects are cerebellum and each cerebral hemisphere without brain stem.



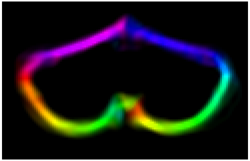
(a) (b) (c)

We are using four 3D cloud systems created from MR-images of 40 normal subjects and a three-scale search for each, which is exaustive in the lowest resolution and local in the higher ones.

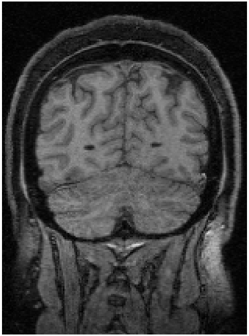
The criterion function

The criterion function assigns a score proportional to the mean arc weight in the graph cut, which separates all objects from the background.

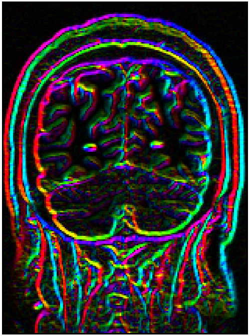
Brain structure segmentation



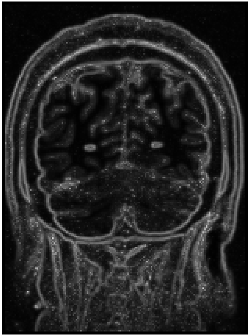
Each object cloud can also give us hints about the expected surface normal, which may be used to penalize arc weights $w(s,t)$ with wrong gradient orientation across them.



scene

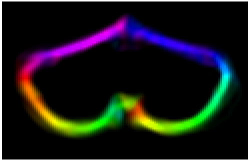


gradient

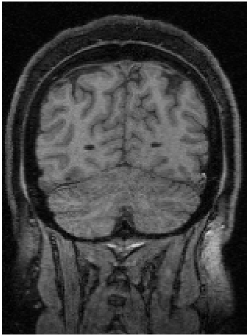


weight (cerebellum)

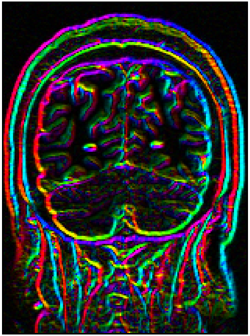
Brain structure segmentation



Each object cloud can also give us hints about the expected surface normal, which may be used to penalize arc weights $w(s, t)$ with wrong gradient orientation across them.



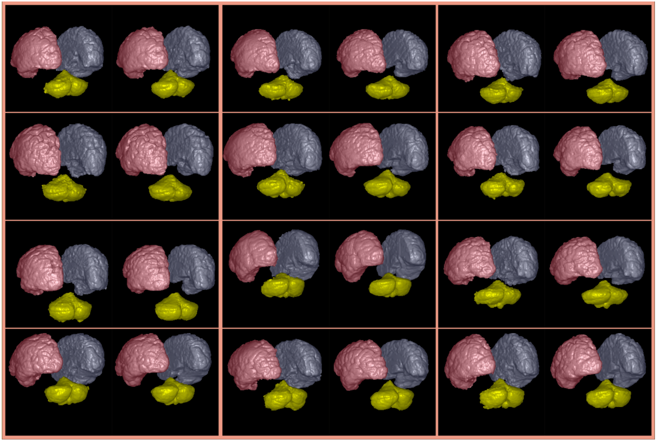
scene



gradient

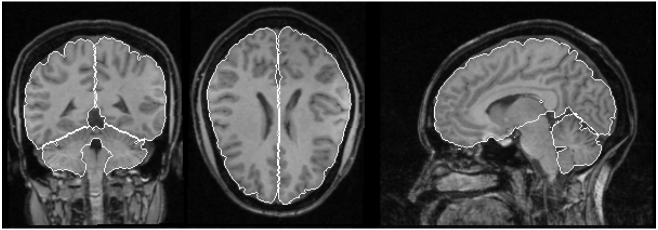
weight (cerebellum)

Results



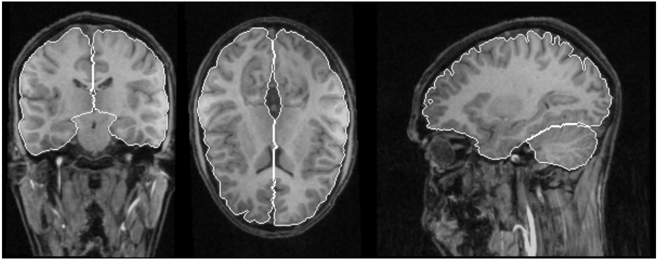
CSM has been evaluated with MR-images from normal subjects and patients. The results show Dice measure around 97% for the brain hemispheres and 95% for the cerebellum.

Results



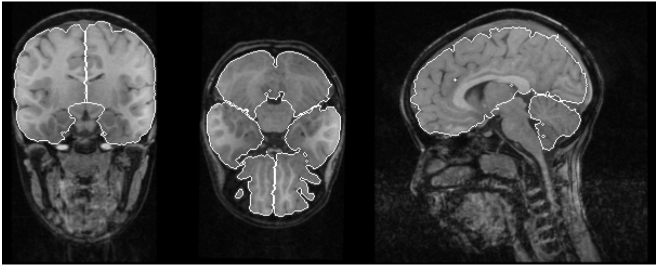
CSM has been evaluated with MR-images from normal subjects and patients. The results show Dice measure around 97% for the brain hemispheres and 95% for the cerebellum.

Results



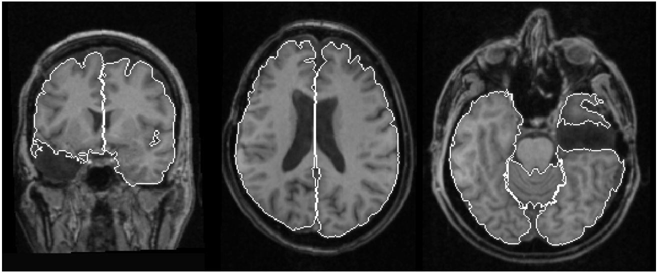
CSM has been evaluated with MR-images from normal subjects and patients. The results show Dice measure around 97% for the brain hemispheres and 95% for the cerebellum.

Results



CSM has been evaluated with MR-images from normal subjects and patients. The results show Dice measure around 97% for the brain hemispheres and 95% for the cerebellum.

Results



CSM has been evaluated with MR-images from normal subjects and patients. The results show Dice measure around 97% for the brain hemispheres and 95% for the cerebellum.

Conclusion

- Arc-weight estimation can exploit model orientation and be used in other segmentation methods based on similar concepts: similarity, speed function, affinity, cost, distance, etc.

Conclusion

- Arc-weight estimation can exploit model orientation and be used in other segmentation methods based on similar concepts: similarity, speed function, affinity, cost, distance, etc.
- The IFTSC approach is certainly among the best delineation methods today.

Conclusion

- Arc-weight estimation can exploit model orientation and be used in other segmentation methods based on similar concepts: similarity, speed function, affinity, cost, distance, etc.
- The IFTSC approach is certainly among the best delineation methods today.
- IFTSC with automatic seed propagation along video frames has been sucessfully used for object tracking [7].

- Arc-weight estimation can exploit model orientation and be used in other segmentation methods based on similar concepts: similarity, speed function, affinity, cost, distance, etc.
- The IFTSC approach is certainly among the best delineation methods today.
- IFTSC with automatic seed propagation along video frames has been sucessfully used for object tracking [7].
- The cloud system model can be exploited in other applications rather than brain structure segmentation.

Next lecture

- The IFT framework.
- Connected filters.
- Interactive and automatic segmentation methods.
- Shape representation and description.
- Clustering and classification.

Thanks for your attention

[1] 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.

[2] T.V. Spina, J.A. Montoya-Zegarra, A.X. Falcão, and P.A.V. Miranda.
Fast interactive segmentation of natural images using the image foresting transform.
In *Proc. of the 16th Intl. Conf. on Digital Signal Processing*, Santorini, Greece, 2009. IEEE.

[3] 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.

[4] 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.

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

Cloud models: Their construction and employment in automatic MRI segmentation of the brain.

Technical Report IC-10-08, Institute of Computing, University of Campinas, Mar 2010.

[7] R. Minetto, J.P. Papa, T.V. Spina, A.X. Falcão, N.J. Leite, and J. Stolfi.

Fast and robust object tracking using image foresting transform.

In *Proc. of the 16th Intl. Conf. on Systems, Signals, and Image Processing*, Chalkida, Greece, 2009. IEEE.