Interactive and Automatic Image Segmentation

Alexandre Xavier Falcão

Visual Informatics Laboratory - Institute of Computing - University of Campinas

afalcao@ic.unicamp.br

Introduction

For completeness, segmentation methods should

Introduction

For completeness, segmentation methods should

- acquire and model object information;

Introduction

For completeness, segmentation methods should

- acquire and model object information;
- enhance regions wherein image properties are similar to those of the object, background, or their transition;

Introduction

For completeness, segmentation methods should

- acquire and model object information;
- enhance regions wherein image properties are similar to those of the object, background, or their transition;
- locate the object and delineate its spatial extent in the image.

Introduction

- Humans are more accurate than computers for object location.

- Humans are more accurate than computers for object location.
- Computers are more precise than humans in object delineation.

Introduction

- Humans are more accurate than computers for object location.
- Computers are more precise than humans in object delineation.

- Humans are more accurate than computers for object location.
- Computers are more precise than humans in object delineation.

In this example, delineation was solved by object enhancement followed by binarization (i.e., a spel classification), with no need for connectivity.

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.

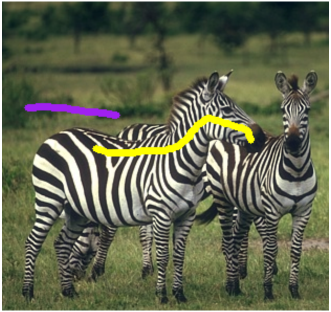
When do we need connectivity?

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.

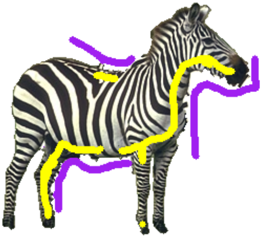


When do we need optimum connectivity?

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, some markers needed to disconnect the object are not suitable for object enhancement.

How can we effectively address segmentation?

- Markers/object models can be used for object location and enhancement.

How can we effectively address segmentation?

- Markers/object models can be used for object location and enhancement.
- Enhancement must be intelligent to extract suitable object information.

How can we effectively address segmentation?

- Markers/object models can be used for object location and enhancement.
- Enhancement must be intelligent to extract suitable object information.
- For interactive segmentation, we can exploit a synergism between object location/correction by a human operator and computer delineation.

How can we effectively address segmentation?

- Markers/object models can be used for object location and enhancement.
- Enhancement must be intelligent to extract suitable object information.
- For interactive segmentation, we can exploit a synergism between object location/correction by a human operator and computer delineation.
- For automatic segmentation, we can exploit a synergism between object location by some object model and computer delineation.

How can we effectively address segmentation?

- Markers/object models can be used for object location and enhancement.
- Enhancement must be intelligent to extract suitable object information.
- For interactive segmentation, we can exploit a synergism between object location/correction by a human operator and computer delineation.
- For automatic segmentation, we can exploit a synergism between object location by some object model and computer delineation.
- In both cases, delineation based on optimum connectivity can be used in the image domain and/or in the feature space by simple choice of the adjacency relation.

We will postpone intelligent object enhancement [1, 2, 3] for the next lecture and address now

Organization of this lecture

We will postpone intelligent object enhancement [1, 2, 3] for the next lecture and address now

- Object delineation using the image foresting transform: boundary-based [4, 5], region-based [6, 7, 8, 9], and hybrid [10] approaches.

Organization of this lecture

We will postpone intelligent object enhancement [1, 2, 3] for the next lecture and address now

- Object delineation using the image foresting transform: boundary-based [4, 5], region-based [6, 7, 8, 9], and hybrid [10] approaches.
- A comparative analysis between the IFT and the min-cut/max-flow algorithms for region-based segmentation [8].

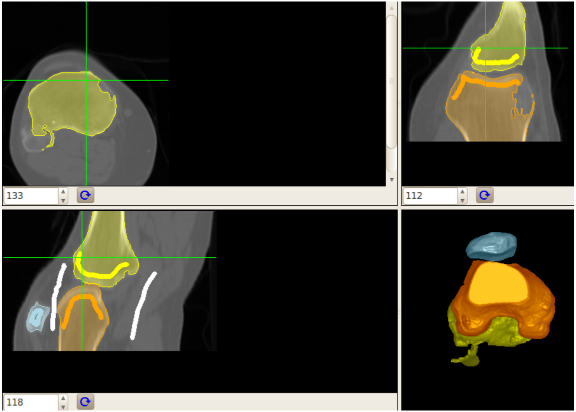
Organization of this lecture

We will postpone intelligent object enhancement [1, 2, 3] for the next lecture and address now

- Object delineation using the image foresting transform: boundary-based [4, 5], region-based [6, 7, 8, 9], and hybrid [10] approaches.
- A comparative analysis between the IFT and the min-cut/max-flow algorithms for region-based segmentation [8].
- Fuzzy object models [11, 12, 13].

Region-based object delineation

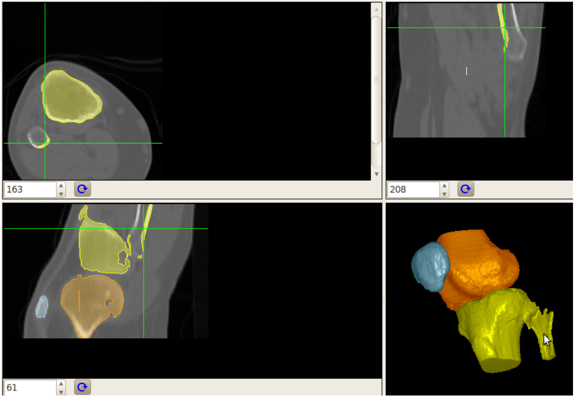
- Multiple objects can be segmented with interactive response time to the user's actions by using the differential IFT with seed competition (IFTSC)[6, 7].



- Interactive 3D visualization is crucial to help on object location and correction[14].

Region-based object delineation

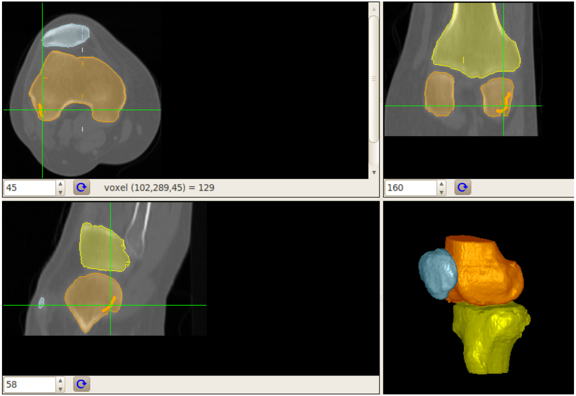
- Multiple objects can be segmented with interactive response time to the user's actions by using the differential IFT with seed competition (IFTSC)[6, 7].



- Interactive 3D visualization is crucial to help on object location and correction[14].

Region-based object delineation

- Multiple objects can be segmented with interactive response time to the user's actions by using the differential IFT with seed competition (IFTSC)[6, 7].



- Interactive 3D visualization is crucial to help on object location and correction[14].

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 seed competition (IFTSC)

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

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

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

- The dual formulation holds for fuzzy connected segmentation[9].

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

Algorithm

– Algorithm IFTSC

$(V, P, L, R, \mathcal{F}) \leftarrow$ DIFT-ForestRemoval$(V, P, L, R, \mathcal{A}, \mathcal{R}_{\mathcal{M}})$.
$\mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{S}$.
While $\mathcal{S} \neq \emptyset$, *remove* t *from* $\mathcal{S}$, *set* $V(t) \leftarrow 0$,
    set $L(t) \leftarrow \lambda(t)$, $R(t) \leftarrow t$, $P(t) \leftarrow nil$, *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 $P(t) \leftarrow s$, $V(t) \leftarrow tmp$, $R(t) \leftarrow R(s)$,
            $L(t) \leftarrow L(s)$, *and Insert* t *in* Q.

Boundary-based object delineation

An ordered sequence of optimum paths can define the object's boundary by several different ways.

Boundary-based object delineation

An ordered sequence of optimum paths can define the object's boundary by several different ways.

- Methods, such as live-wire [15, 4] and riverbed [5], present the paths as the user selects boundary points and moves the cursor on the image.

Boundary-based object delineation

An ordered sequence of optimum paths can define the object's boundary by several different ways.

- Methods, such as live-wire [15, 4] and riverbed [5], present the paths as the user selects boundary points and moves the cursor on the image.
- Iterative live-wire [16, 17] uses, as input, points nearby the boundary and executes live-wire several times, replacing those points by the mid-segment ones until convergence.

Boundary-based object delineation

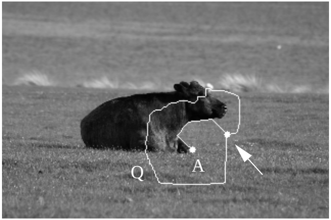
An ordered sequence of optimum paths can define the object's boundary by several different ways.

- Methods, such as live-wire [15, 4] and riverbed [5], present the paths as the user selects boundary points and moves the cursor on the image.
- Iterative live-wire [16, 17] uses, as input, points nearby the boundary and executes live-wire several times, replacing those points by the mid-segment ones until convergence.

They can be implemented by a sequence of IFTs using 4- or 8-adjacency relation and suitable connectivity function.

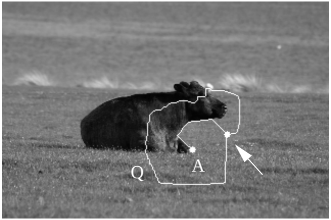
In live-wire-on-the-fly (LWOF), optimum paths are incrementally computed from the moving wavefront Q.

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



Live-wire-on-the-fly

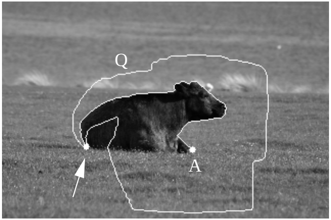
In live-wire-on-the-fly (LWOF), optimum paths are incrementally computed from the moving wavefront Q.



- 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.

Live-wire-on-the-fly

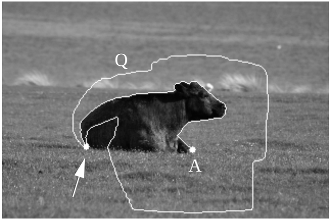
In live-wire-on-the-fly (LWOF), optimum paths are incrementally computed from the moving wavefront Q.



- 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

In live-wire-on-the-fly (LWOF), optimum paths are incrementally computed from the moving 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

In live-wire-on-the-fly (LWOF), optimum paths are incrementally computed from the moving 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 counter-clockwise oriented boundaries.

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

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.

Path computation from s^* to u in LWOF

Algorithm

– Path Computation from s^* to u in LWOF

If $V(u) = +\infty$ *or* $u \in Q$, **then**
    Set $s \leftarrow nil$.
    While $Q \neq \emptyset$ *and* $s \neq u$, *do*
        Remove from Q *a spel* s *such that* $V(s)$ *is minimum.*
        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.

Riverbed

Riverbed simulates the behavior of water flowing through a riverbed, always seeking lower ground levels, snaking through the river bends, instead of short-cutting the path as in live-wire. This leads to the following connectivity function for a starting seed point s^*:

$$f_w^{\circlearrowleft}(\langle t \rangle) = \begin{cases} 0 & \text{if } t = s^* \\ +\infty & \text{otherwise} \end{cases}$$

$$f_w^{\circlearrowleft}(\pi_s \cdot \langle s, t \rangle) = \begin{cases} \bar{w}(s,t) & \text{if } O(l) \geq O(r) \\ K & \text{otherwise.} \end{cases}$$

where l and r are the spels at the left and right sides of arc $\langle s, t \rangle$.

Riverbed

- Although $f_w^{\circlearrowleft}$ is not smooth, it selects segments such that the maximum arc weight

$$\max_{\forall (l,r)\in\mathcal{A}', L(l)=1, L(r)=0} \bar{w}(l,r)$$

is minimum, considering all possible cuts in the dual graph $(\mathcal{N}, \mathcal{A}')$.

Riverbed

- Although $f_w^{\circlearrowleft}$ is not smooth, it selects segments such that the maximum arc weight

$$\max_{\forall (l,r) \in \mathcal{A}', L(l)=1, L(r)=0} \bar{w}(l,r)$$

  is minimum, considering all possible cuts in the dual graph $(\mathcal{N}, \mathcal{A}')$.
- This implies that IFTSC and riverbed decide for the same optimum graph cut[5].

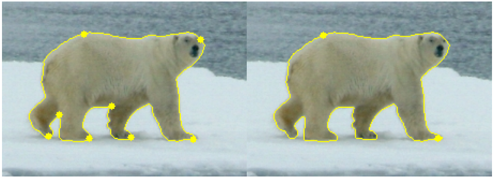
Riverbed

- Although $f_w^{\circlearrowleft}$ is not smooth, it selects segments such that the maximum arc weight

$$\max_{\forall (l,r)\in \mathcal{A}', L(l)=1, L(r)=0} \bar{w}(l,r)$$

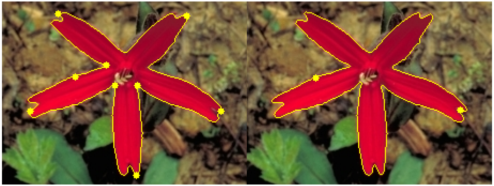
  is minimum, considering all possible cuts in the dual graph $(\mathcal{N}, \mathcal{A}')$.
- This implies that IFTSC and riverbed decide for the same optimum graph cut[5].
- Riverbed is more suitable than live-wire for more intricate shapes, but live-wire can jump weakly defined parts of the boundary.

Riverbed versus live-wire



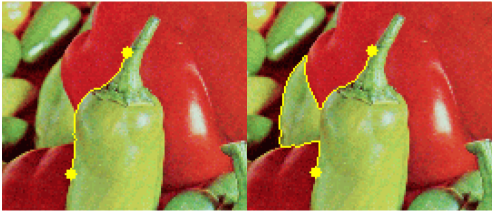
- Live-wire (left) and riverbed (right).

Riverbed versus live-wire



- Live-wire (left) and riverbed (right).
- Riverbed requires fewer anchor points for complex shapes.

Riverbed versus live-wire



- Live-wire (left) and riverbed (right).
- Riverbed requires fewer anchor points for complex shapes.
- Live-wire can jump across weakly defined segments.

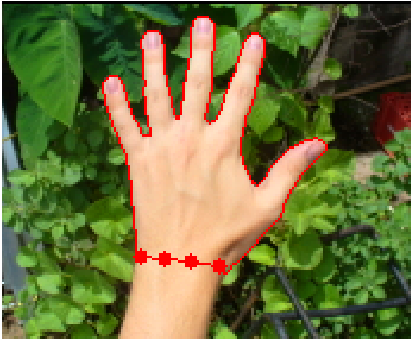
Riverbed versus live-wire

Their combination can take advantage from both approaches.

- Live-wire on complex shapes requires more anchor points.

Riverbed versus live-wire

Their combination can take advantage from both approaches.



- Live-wire on complex shapes requires more anchor points.
- Riverbed asks for more user intervention on poorly defined parts.

Riverbed versus live-wire

Their combination can take advantage from both approaches.

- Live-wire on complex shapes requires more anchor points.
- Riverbed asks for more user intervention on poorly defined parts.
- Their combination requires only two segments (live wire in cyan, riverbed in red).

Live Markers

Live Markers is another hybrid approach[10] that takes advantage from the superior ability of LWOF on weakly defined segments and from IFTSC to handle complex 2D/3D shapes of multiple objects.

Live Markers

Live Markers is another hybrid approach[10] that takes advantage from the superior ability of LWOF on weakly defined segments and from IFTSC to handle complex 2D/3D shapes of multiple objects.

The markers may be selected by the user or may come from the live-wire segments.

Live Markers

In several cases, the live-wire segments followed by IFTSC almost complete the segmentation process.

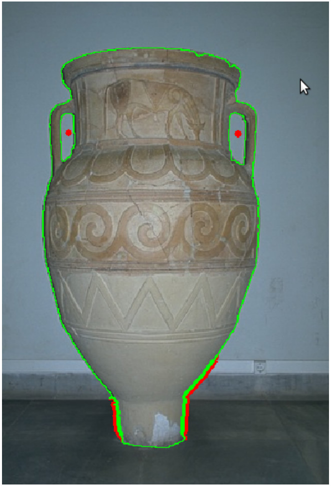
In several cases, the live-wire segments followed by IFTSC almost complete the segmentation process.

Live Markers

In several cases, the live-wire segments followed by IFTSC almost complete the segmentation process.

Live Markers

In several cases, the live-wire segments followed by IFTSC almost complete the segmentation process.



Organization of this lecture

- Object delineation using the image foresting transform.
- A comparison between IFTSC and object delineation using the min-cut/max-flow algorithm (GCMF).
- Fuzzy object models.

IFTSC and min-cut/max-flow algorithms

- From lecture 1, we know that IFTSC computes 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 [7], and this is also a piecewise optimum cut.

IFTSC and min-cut/max-flow algorithms

- From lecture 1, we know that IFTSC computes 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 [7], and this is also a piecewise optimum cut.

- Similar region-based delineation could be obtained by GCMF as a graph cut whose sum of arc weights

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

is minimum for $\beta \geq 1$, with lower values favoring smaller cuts and higher values making both equivalent[8].

IFTSC and min-cut/max-flow algorithms

- IFTSC can handle multiple object delineation in $O(|\mathcal{N}|)$.

IFTSC and min-cut/max-flow algorithms

- IFTSC can handle multiple object delineation in $O(|\mathcal{N}|)$.
- GCMF is not viable for multiple objects and takes $O(|\mathcal{N}|^{2.5})$ for single object delineation.

IFTSC and min-cut/max-flow algorithms

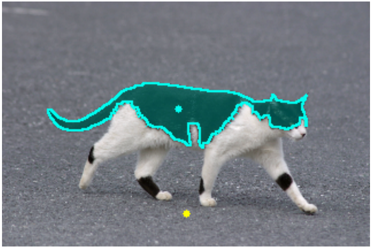
- IFTSC can handle multiple object delineation in $O(|\mathcal{N}|)$.
- GCMF is not viable for multiple objects and takes $O(|\mathcal{N}|^{2.5})$ for single object delineation.
- IFTSC is also more robust with respect to seed location than GCMF, but the latter provides smoother boundaries with less leaking than the former.

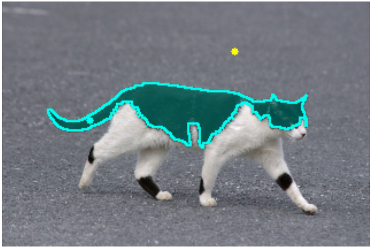
IFTSC and min-cut/max-flow algorithms

A lower β value allows GCMF (left) to obtain a smoother boundary, reducing the leaking of IFTSC (right).

IFTSC and min-cut/max-flow algorithms

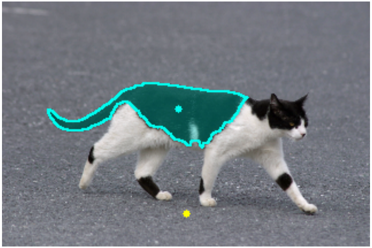
However, a same connected component is always obtained with IFTSC, independently of seed location.

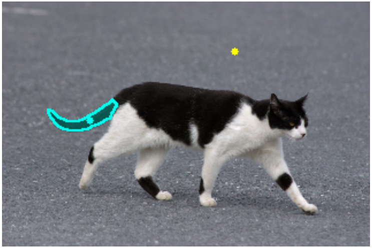




IFTSC and min-cut/max-flow algorithms

The same does not happen in GCMF, when β is not high enough.





Organization of this lecture

- Object delineation using the image foresting transform.
- A comparison between IFTSC and object delineation using the min-cut/max-flow algorithm (GCMF).
- Fuzzy object models.

Statistical object models

- Automatic segmentation is feasible when possible object deformations are captured into a statistical model (atlas).

Statistical object models

- Automatic segmentation is feasible when possible object deformations are captured into a statistical model (atlas).
- The atlas is built by registration among training images in order to estimate the probability of each spel to be inside object/background.

Statistical object models

- Automatic segmentation is feasible when possible object deformations are captured into a statistical model (atlas).
- The atlas is built by registration among training images in order to estimate the probability of each spel to be inside object/background.
- Object location in a test image is solved when it is registered with the atlas.

Statistical object models

- Automatic segmentation is feasible when possible object deformations are captured into a statistical model (atlas).
- The atlas is built by registration among training images in order to estimate the probability of each spel to be inside object/background.
- Object location in a test image is solved when it is registered with the atlas.
- Subsequent object delineation completes segmentation.

Statistical object models

- Automatic segmentation is feasible when possible object deformations are captured into a statistical model (atlas).
- The atlas is built by registration among training images in order to estimate the probability of each spel to be inside object/background.
- Object location in a test image is solved when it is registered with the atlas.
- Subsequent object delineation completes segmentation.

Registration is an expensive task that may force delineation to fit with the model irrespective to the local image information.

Fuzzy object models

We have developed fuzzy models to eliminate registration and provide more decision power to the delineation method.

Fuzzy object models

We have developed fuzzy models to eliminate registration and provide more decision power to the delineation method.

- A fuzzy model may only require a simple translation of the training objects to a common reference point for its construction [11].

Fuzzy object models

We have developed fuzzy models to eliminate registration and provide more decision power to the delineation method.

- A fuzzy model may only require a simple translation of the training objects to a common reference point for its construction [11].
- It may also require object alignment, but this only involves its own image[12, 13].

Fuzzy object models

We have developed fuzzy models to eliminate registration and provide more decision power to the delineation method.

- A fuzzy model may only require a simple translation of the training objects to a common reference point for its construction [11].
- It may also require object alignment, but this only involves its own image[12, 13].
- Segmentation is solved by translating the model and executing delineation at each location.

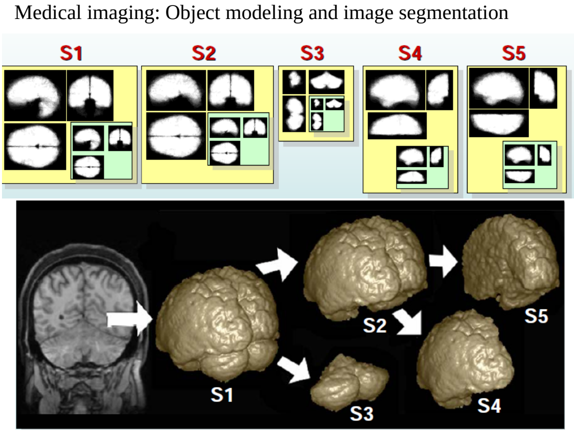
Fuzzy object models

We have developed fuzzy models to eliminate registration and provide more decision power to the delineation method.

- A fuzzy model may only require a simple translation of the training objects to a common reference point for its construction [11].
- It may also require object alignment, but this only involves its own image[12, 13].
- Segmentation is solved by translating the model and executing delineation at each location.
- It is possible to considerably speed-up this object search process by using multiple scales of the image and models.

Fuzzy object models

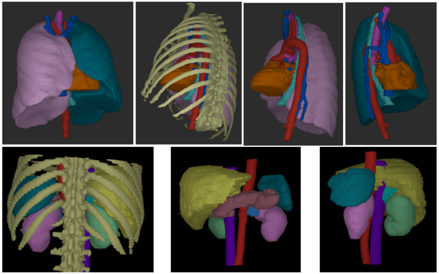
Examples of objects and their fuzzy models.



This lecture will present only the first case, named Cloud System Model (CSM) [11], using IFTSC for delineation.

Fuzzy object models

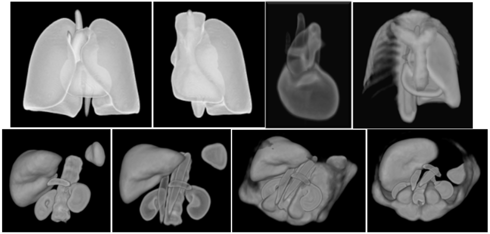
Examples of objects and their fuzzy models.



This lecture will present only the first case, named Cloud System Model (CSM) [11], using IFTSC for delineation.

Fuzzy object models

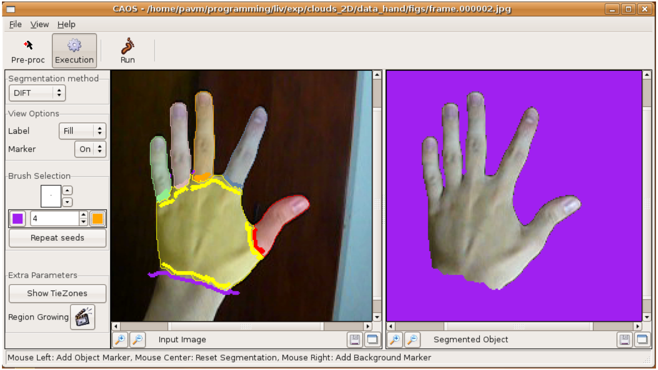
Examples of objects and their fuzzy models.



This lecture will present only the first case, named Cloud System Model (CSM) [11], using IFTSC for delineation.

The cloud system model

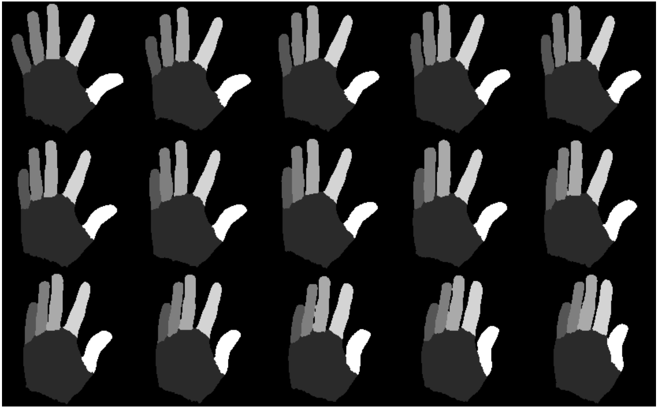
A set of training objects is first provided by interactive IFTSC segmentation.



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

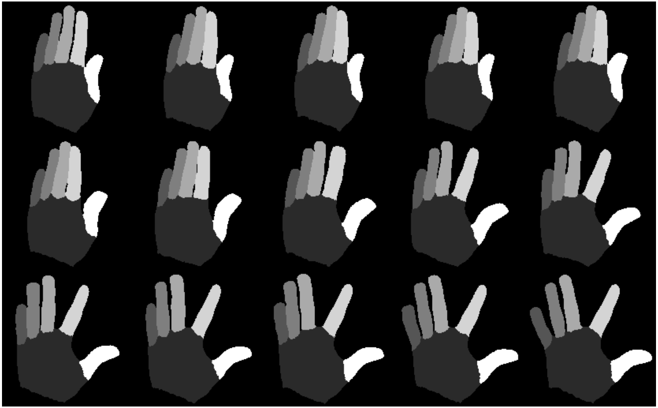
A set of training objects is first provided by interactive IFTSC segmentation.



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

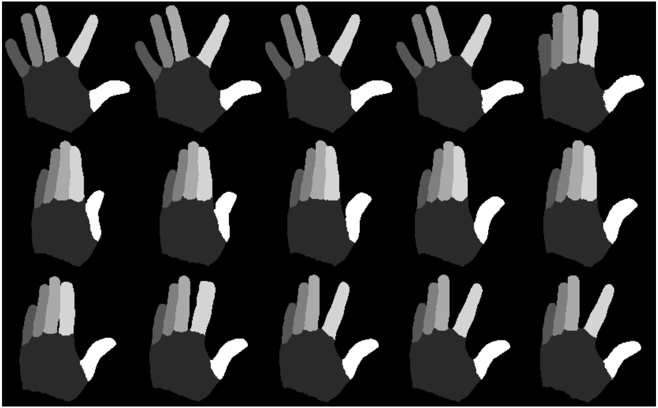
A set of training objects is first provided by interactive IFTSC segmentation.



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

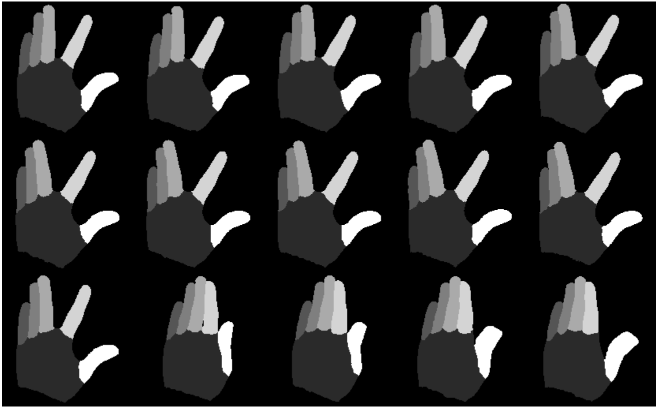
A set of training objects is first provided by interactive IFTSC segmentation.



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

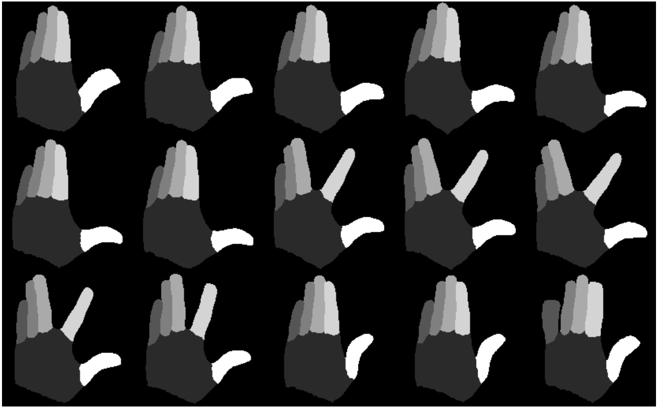
A set of training objects is first provided by interactive IFTSC segmentation.



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

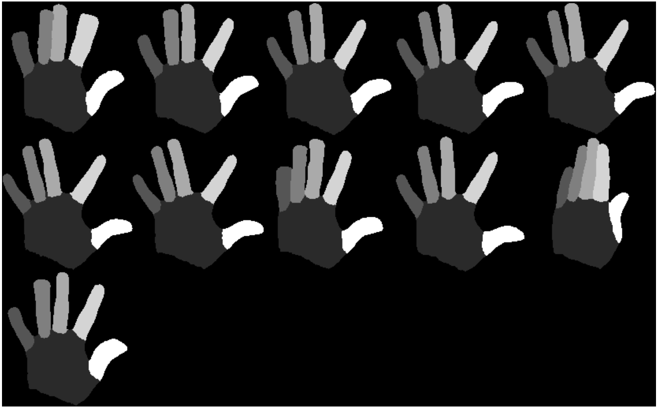
A set of training objects is first provided by interactive IFTSC segmentation.



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.

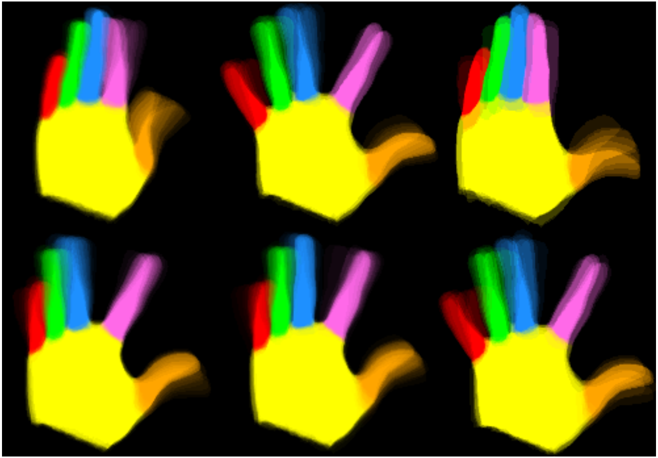
The cloud system model

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.



The cloud system model

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

The cloud system model

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

- A fuzzy membership 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 fuzzy membership 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.

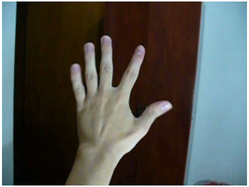
The cloud system model

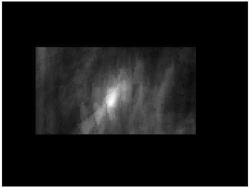
A cloud system model (CSM) then consists of three elements:

- A fuzzy membership 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

Conclusion

- It should be clear the importance of combining the strengths from distinct object delineation methods.

Conclusion

- It should be clear the importance of combining the strengths from distinct object delineation methods.
- The synergistic combination between object models and delineation methods makes automatic segmentation feasible.

Conclusion

- It should be clear the importance of combining the strengths from distinct object delineation methods.
- The synergistic combination between object models and delineation methods makes automatic segmentation feasible.
- When any automatic segmentation method fails, we have also developed solutions to reduce it into an optimum-path forest with minimum number of roots[18], so this facilitates corrections by the differential IFTSC algorithm[6, 7].

Next lecture

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

Thanks for your attention

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] P.A.V. Miranda, A.X. Falcao, and T.V. Spina.
The riverbed approach for user-steered image segmentation.
In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3133 –3136, Sep 2011.

[6] 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.

[7] 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.

[8] K.C. Ciesielski, J.K. Udupa, A.X. Falcão, and P.A.V. Miranda.
A unifying graph-cut image segmentation framework: algorithms it encompasses and equivalences among them.
In *SPIE on Medical Imaging: Image Processing*, volume 8314, page 12 pages, Feb 2012.

[9] K.C. Ciesielski, J.K. Udupa, A.X. Falcão, and P.A.V. Miranda.
Fuzzy connectedness image segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis.
Journal of Mathematical Imaging and Vision, 2012.

[10] Thiago Vallin Spina, Alexandre Xavier Falcão, and Paulo André Vechiatto Miranda.
User-steered image segmentation using live markers.
In *Proceedings of the 14th international conference on Computer analysis of images and patterns - Volume Part I*, CAIP'11, pages 211–218. Springer-Verlag, 2011.

[11] 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.

[12] J.K. Udupa, D. Odhner, A.X. Falcão, K.C. Ciesielski, P.A.V. Miranda, P.Vaideeshhwaran, S. Mishra, A. Shaheryar, G.J. Grevera, B. Saboury, and D.A. Torigian.
Fuzzy object modeling.
In *SPIE on Medical Imaging: Visualization, Image-Guided Procedures, and Modeling*, volume 7964, page 10 pages, Feb 2011.

[13] J.K. Udupa, D. Odhner, A.X. Falcão, K.C. Ciesielski, P.A.V. Miranda, S. Mishra, G. Grevera, B. Saboury, and D. Torigian.
Automatic anatomy recognition via fuzzy object models.
In *SPIE on Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 8316, page 8 pages, Feb 2012.

[14] R. Audigier, R.A. Lotufo, and A.X. Falcão.
3d visualization to assist iterative object definition from medical images.
Computerized Medical Imaging and Graphics, 30(4):217–230, Jun 2006.

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

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

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

[16] A. Souza, J.K. Udupa, G. Grevera, Y. Sun, D. Odhner, N. Suri, and M.D. Schnall.

Iterative live wire and live snake: new user-steered 3d image segmentation paradigms.

In *Proc. of SPIE on Image Processing*, volume 6144 - PART 2, 2006.

[17] J. Liu and J.K. Udupa.

Oriented active shape models.

Medical Imaging, IEEE Transactions on, 28:571–584, Apr 2009.

[18] P.A.V. Miranda, A.X. Falcao, G.C.S. Ruppert, and F.A.M. Cappabianco.

How to fix any 3d segmentation interactively via image foresting transform and its use in mri brain segmentation.

In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 2031 –2035, Apr 2011.