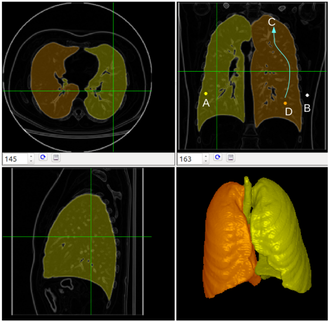
Volumetric Image Visualization

Alexandre Xavier Falcão

LIDS - Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

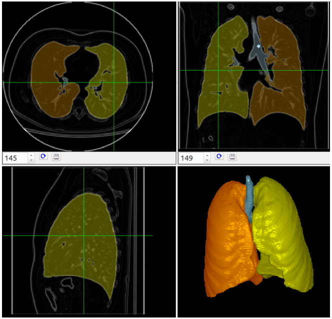
We have learned that multiple objects can be delineated by optimum seed competition.



In this lecture, we will learn the general IFT algorithm for object delineation [6, 7, 13, 14].

Object delineation by optimum seed competition

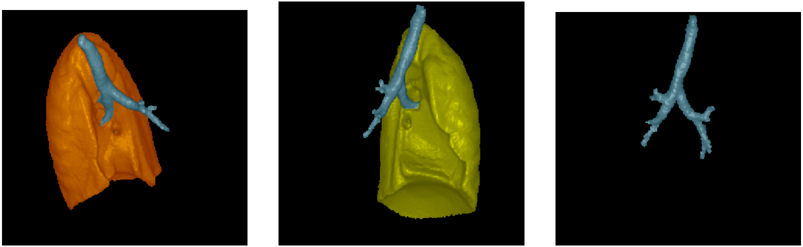
We have learned that multiple objects can be delineated by optimum seed competition.



In this lecture, we will learn the general IFT algorithm for object delineation [6, 7, 13, 14].

Object delineation by optimum seed competition

We have learned that multiple objects can be delineated by optimum seed competition.



In this lecture, we will learn the general IFT algorithm for object delineation [6, 7, 13, 14].

Object delineation by optimum seed competition

- Let $(D_I, \mathcal{A})$ be a graph derived from a 3D image $\hat{I} = (D_I, I)$, such that $\mathcal{A}$: $\{(p,q) \in D_I \times D_I \mid \|q - p\| \leq 1\}$ is a 6-neighborhood relation and $\mathcal{A}(p)$ is the set of adjacents of p.

Object delineation by optimum seed competition

- Let $(D_I, \mathcal{A})$ be a graph derived from a 3D image $\hat{I} = (D_I, I)$, such that $\mathcal{A}\colon \{(p,q) \in D_I \times D_I \mid \|q-p\| \leq 1\}$ is a 6-neighborhood relation and $\mathcal{A}(p)$ is the set of adjacents of p.

- Let $\mathcal{S} \subset D_I$ be a set of seed voxels, such that $\lambda(s) \in \{0, 1, \ldots, c\}$ indicates that seed $s \in \mathcal{S}$ belongs to one object $1 \leq i \leq c$ or the background 0.

Object delineation by optimum seed competition

- Let $(D_I, \mathcal{A})$ be a graph derived from a 3D image $\hat{I} = (D_I, I)$, such that $\mathcal{A}$: $\{(p,q) \in D_I \times D_I \mid \|q - p\| \leq 1\}$ is a 6-neighborhood relation and $\mathcal{A}(p)$ is the set of adjacents of p.

- Let $\mathcal{S} \subset D_I$ be a set of seed voxels, such that $\lambda(s) \in \{0, 1, \ldots, c\}$ indicates that seed $s \in \mathcal{S}$ belongs to one object $1 \leq i \leq c$ or the background 0.

- Let f be a path-cost function, such that

$$\begin{aligned} f(\langle q \rangle) &= H(q) \\ f(\pi_p \cdot \langle p, q \rangle) &= \max\{f(\pi_p), w(p,q)\}, \end{aligned}$$

  where $\pi_p \cdot \langle p, q \rangle$ is the extension of a path $\pi_p = \langle p_1, p_2, \ldots, p_n = p \rangle$ by an arc $(p,q) \in \mathcal{A}$ with weight $w(p,q)$, $\langle q \rangle$ is a trivial path, and $H(q)$ is a handicap function.

The general IFT algorithm

- For effective delineation, $w(p,q) \geq 0$ should be lower inside the objects and higher across their boundaries.

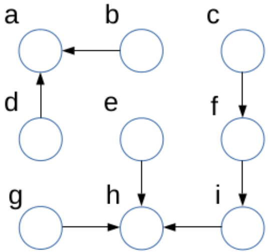
- For effective delineation, $w(p, q) \geq 0$ should be lower inside the objects and higher across their boundaries.
- $H(q)$ is usually defined as 0 for $q \in \mathcal{S}$ and $+\infty$, otherwise. Note, however, that seeds may have different priorities.

The general IFT algorithm

- For effective delineation, $w(p,q) \geq 0$ should be lower inside the objects and higher across their boundaries.
- $H(q)$ is usually defined as 0 for $q \in \mathcal{S}$ and $+\infty$, otherwise. Note, however, that seeds may have different priorities.
- While minimizing a path-cost map $C(q) = \min_{\forall \pi_q \in \Pi}\{f(\pi_q)\}$, where Π is the set of all possible paths, the IFT algorithm propagates paths from $\mathcal{S}$ in a non-decreasing order of costs and outputs an optimum-path forest rooted in $\mathcal{S}$.

The general IFT algorithm

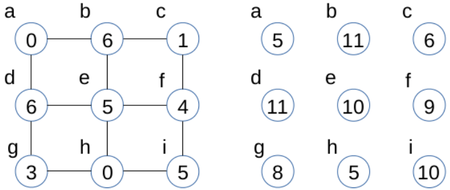
An optimum-path forest rooted in $\mathcal{S}$ is an acyclic map P of predecessors, such that $P(q) = p \in D_I \backslash \mathcal{S}$, when p is the predecessor of q in the optimum path π_q, and $P(q) = nil \notin D_I$, when $q \in \mathcal{S}$.



For instance: $\pi_c = \langle h, i, f, c \rangle$, $P(c) = f$, $\pi_a = \langle a \rangle$, and $P(a) = nil$.

The general IFT algorithm

Consider, for example, the image graph on the left, where the numbers indicate $I(q)$ and $\mathcal{A}$ is a 4-neighborhood relation.



On the right, the trivial forest for the path-cost function f with $H(q) = I(q) + 5$ and $w(p, q) = I(q)$. In this case, the roots of the forest are defined by the time p is visited for possible optimum path extension and $P(p) = nil$.

The general IFT algorithm

At each iteration of $|D_I|$ iterations,

- the algorithm selects one node p, among those of lowest cost, never selected before, for possible optimum path extension.

The general IFT algorithm

At each iteration of $|D_I|$ iterations,

- the algorithm selects one node p, among those of lowest cost, never selected before, for possible optimum path extension.

- for each adjacent $q \in \mathcal{A}(p)$, such that $C(q) > f(\pi_p \cdot \langle p, q \rangle)$, it updates the maps $C(q) \leftarrow f(\pi_p \cdot \langle p, q \rangle)$ and $P(q) \leftarrow p$.

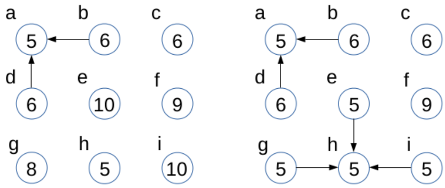
The general IFT algorithm

At each iteration of $|D_I|$ iterations,

- the algorithm selects one node p, among those of lowest cost, never selected before, for possible optimum path extension.

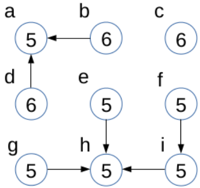
- for each adjacent $q \in \mathcal{A}(p)$, such that $C(q) > f(\pi_p \cdot \langle p, q \rangle)$, it updates the maps $C(q) \leftarrow f(\pi_p \cdot \langle p, q \rangle)$ and $P(q) \leftarrow p$.

After iterations 1 (left) and 2 (right), we will have



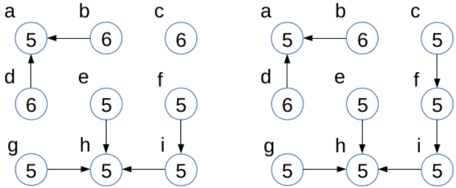
The general IFT algorithm

After iterations 3 (left) and from 4–9 (right), we will have



An optimum-path forest for the path-cost function f.

After iterations 3 (left) and from 4–9 (right), we will have



An optimum-path forest for the path-cost function f.

Its bottleneck is to determine, at each iteration, the node $p \in D_I$ of lowest cost, never selected before.

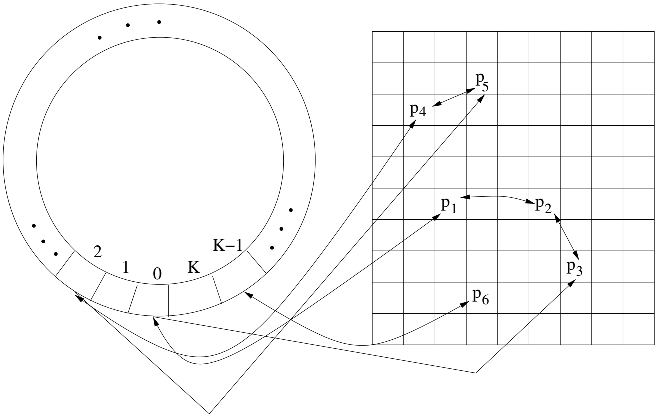
The IFT algorithm for object delineation

Input: $\hat{I} = (D_I, I)$, $\mathcal{A}$, w, and $\mathcal{S}$ labeled by λ.
Output: P, C, and a label map L, where $L(q)$ is the label of the seed that has conquered q.
Auxiliary: A priority queue Q and a variable tmp.

```
∀q ∈ D_I, do
    C(q) ← +∞ and P(q) ← nil.
    If q ∈ S then C(q) ← 0 and L(q) ← λ(q).
    insert q in Q.
While Q ≠ ∅ do
    Remove p = arg min_{q∈Q}{C(q)} from Q.
    ∀q ∈ A(p), such that q ∈ Q, do
    tmp ← max{C(p), w(p,q)}.
      If C(q) > tmp, then
        C(q) ← tmp, P(q) ← p, and L(q) ← L(p).
```

The priority queue Q

For $w(p,q) \in [0,K]$, $K \ll |D_I|$, the algorithm executes in $O(|D_I|)$, using *bucket sort*.



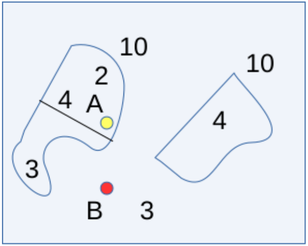
Nodes $p \in D_I$ are inserted and removed from *bucket* $C(p)$ mod $K+1$ in $O(1)$.

Arc-weight assignment

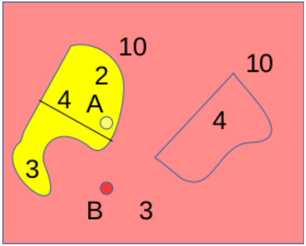
- The effectiveness of object delineation depends more on the arc-weight assignment $w(p, q)$ than on the location of the seeds inside their objects (background).

Arc-weight assignment

- The effectiveness of object delineation depends more on the arc-weight assignment $w(p, q)$ than on the location of the seeds inside their objects (background).
- It should be clear that if $w(p, q)$ is higher across the boundaries of the object than inside and outside it, any two seeds, one inside and one outside the object would be enough to complete segmentation.



(a)



(b)

Arc-weight assignment

- Pattern classifiers, such as deep neural networks, may be able to create an object map O such that the values $O(p)$ are higher inside the object than in most parts of the background.

Arc-weight assignment

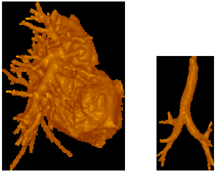
- Pattern classifiers, such as deep neural networks, may be able to create an object map O such that the values $O(p)$ are higher inside the object than in most parts of the background.

- The object map O may be used for weighted arc orientation [13], when $\lambda(p) \in \{0, 1\}$.

$$w(p,q) = \begin{cases} G^{\alpha}(q) & \text{if } O(p) > O(q) \text{ and } L(p) = 1, \\ G^{\alpha}(q) & \text{if } O(p) < O(q) \text{ and } L(p) = 0, \\ G^{\beta}(q) & \text{if } O(p) = O(q), \\ G(q) & \text{otherwise,} \end{cases}$$

where $\alpha > 1$, $0 < \beta < 1$, $G(q)$ the magnitude of a gradient vector $\vec{G}(p) = \frac{1}{|\mathcal{A}(p)|} \sum_{\forall q \in \mathcal{A}(p)} [I(q) - I(p)] \vec{v}_{pq}$, $\vec{v}_{pq} = \frac{q-p}{\|q-p\|}$.

The method in this case may be called an oriented watershed transform.



(a) (b)

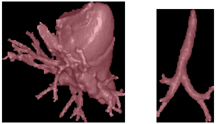
(a) Mediastinum and (b) traquea-bronchi extracted from a CT image of the thorax.

Arc-weight assignment

A watershed transform for $\lambda(p) \in \{0, 1\}$ could use the combination of image-based and object-based gradients.

$$w(p, q) \quad = \quad \alpha G_i(q) + (1 - \alpha) G_o(q),$$

where $0 \leq \alpha \leq 1$, G_i and G_o are the magnitude of the gradient vector $\vec{G}$ computed in I and O, respectively.



(a) (b)

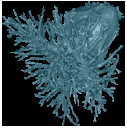
(a) Mediastinum and (b) traquea-bronchi extracted from a CT image of the thorax.

Arc-weight assignment

Another possibility for $\lambda(p) \in \{0, 1\}$ is the dynamic arc-weight estimation from the growing trees, a method called dynamic trees [2, 3].

$$w(p, q) = \begin{cases} |\mu_p - I(q)|^\alpha & \text{if } O(p) > O(q) \text{ and } L(p) = 1, \\ |\mu_p - I(q)|^\alpha & \text{if } O(p) < O(q) \text{ and } L(p) = 0, \\ |\mu_p - I(q)|^\beta & \text{if } O(p) = O(q), \\ |\mu_p - I(q)| & \text{otherwise,} \end{cases}$$

where $\alpha > 1$, $0 < \beta < 1$, $\mu_p = \frac{1}{|\mathcal{T}_p|} \sum_{\forall q \in \mathcal{T}_p} I(q)$ and $\mathcal{T}_p$ is the growing tree that has conquered p.



(a)

(b)

(a) Mediastinum and (b) traquea-bronchi extracted from a CT image of the thorax.

- The forest is optimum whenever the conditions in [5] are satisfied and this is the case of the watershed transform. Due to the orientation, those conditions are not satisfied by the oriented watershed and dynamic trees.

Optimal path-cost mapping

- The forest is optimum whenever the conditions in [5] are satisfied and this is the case of the watershed transform. Due to the orientation, those conditions are not satisfied by the oriented watershed and dynamic trees.

- The differential correction of the forest may use the algorithm in [7] for the watershed transform.

Optimal path-cost mapping

- The forest is optimum whenever the conditions in [5] are satisfied and this is the case of the watershed transform. Due to the orientation, those conditions are not satisfied by the oriented watershed and dynamic trees.

- The differential correction of the forest may use the algorithm in [7] for the watershed transform.

- The algorithm in [14] might be applied for differential corrections in root-based path-cost functions, but this is not the case of the oriented watershed and dynamic trees.

Optimal path-cost mapping

- The forest is optimum whenever the conditions in [5] are satisfied and this is the case of the watershed transform. Due to the orientation, those conditions are not satisfied by the oriented watershed and dynamic trees.

- The differential correction of the forest may use the algorithm in [7] for the watershed transform.

- The algorithm in [14] might be applied for differential corrections in root-based path-cost functions, but this is not the case of the oriented watershed and dynamic trees.

- The IFT segmentation with multiple object maps is an interesting approach, which can explore hierarchical information among objects [15].

Multiple object maps

- One may create one object map O_i for $i = 1, 2, \ldots, c$, except for the background (the diverse class).

Multiple object maps

- One may create one object map O_i for $i = 1, 2, \ldots, c$, except for the background (the diverse class).

- An object map O_0 for the background may be the complement of the union of the object maps O_i.

$$O_0(p) \quad = \quad H - \max_{i=1,2,\ldots,c} \{O_i(p)\}$$

  where H is the maximum value in O_i, $i = 1, 2, \ldots, c$.

- One may create one object map O_i for $i = 1, 2, \ldots, c$, except for the background (the diverse class).
- An object map O_0 for the background may be the complement of the union of the object maps O_i.

$$O_0(p) \quad = \quad H - \max_{i=1,2,\ldots,c}\{O_i(p)\}$$

where H is the maximum value in O_i, $i = 1, 2, \ldots, c$.

- As long as the trasition from background to O_i be from a brighter to a darker region, O_0 can be used similarly. That is, if $O_i(p) > O_i(q)$ and $L(p) = i$, $i = 0, 1, \ldots, c$, we may penalize the arc weight $w(p, q)$.

Multiple object maps

- One may create one object map O_i for $i = 1, 2, \ldots, c$, except for the background (the diverse class).

- An object map O_0 for the background may be the complement of the union of the object maps O_i.

$$O_0(p) \quad = \quad H - \max_{i=1,2,\ldots,c} \{O_i(p)\}$$

  where H is the maximum value in O_i, $i = 1, 2, \ldots, c$.

- As long as the trasition from background to O_i be from a brighter to a darker region, O_0 can be used similarly. That is, if $O_i(p) > O_i(q)$ and $L(p) = i$, $i = 0, 1, \ldots, c$, we may penalize the arc weight $w(p, q)$.

- Other variants may consider transitions between connected objects.

[1] A.M. Sousa, S.B. Martins, F. Reis, E. Bagatin, K. Irion, and A.X. Falcão. ALTIS: A Fast and Automatic Lung and Trachea CT-Image Segmentation Method. *Medical Physics*, doi 10.1002/mp.13773, 46(11), pp. 4970–4982, Nov 2019

[2] J. Bragantini, S.B. Martins, C. Castelo-Fernández, and A.X. Falcão. Graph-based Image Segmentation using Dynamic Trees. *23rd Iberoamerican Congress on Pattern Recognition*, CIARP 2018: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, doi 10.1007/978-3-030-13469-3_55, LNCS 11401, Madrid, Spain, pp. 470–478, 2019.

[3] A.X. Falcão and J. Bragantini. The Role of Optimum Connectivity in Image Segmentation: Can the Algorithm Learn Object Information During the Process? Proc. of 21st Discrete Geometry for Computer Imagery (DGCI). Couprie M., Cousty J., Kenmochi Y., Mustafa N. (eds), doi

10.1007/978-3-030-14085-4_15, LNCS 11414, pp. 180-194, March 2019.

[4] S.B. Martins, J. Bragantini, C. Yasuda, and A.X. Falcão. An Adaptive Probabilistic Atlas for Anomalous Brain Segmentation in MR Images. *Medical Physics*, doi: 10.1002/mp.13771, 46(11), pp. 4940–4950, Nov 2019.

[5] K.C. Ciesielski, A.X. Falcão, and P.A.V. Miranda. Path-value functions for which Dijkstra's algorithm returns optimal mapping. *Journal of Mathematical Imaging and Vision*, 10.1007/s10851-018-0793-1, vol. 60, pp. 1025-1036, 2018.

[6] A. X. Falcão, J. Stolfi and R. de Alencar Lotufo. The image foresting transform: theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10.1109/TPAMI.2004.1261076, 26(1), pp. 19-29, 2004.

[7] A.X. Falcão and F.P.G. Bergo . Interactive Volume Segmentation with Differential Image Foresting Transforms.

IEEE Trans. on Medical Imaging, 10.1109/TMI.2004.829335, 23(9), pp. 1100–1108, 2004.

[8] T.V. Spina, J. Stegmaier, A.X. Falcão, E. Meyerowitz, and A. Cunha. SEGMENT3D: A Web-based Application for Collaborative Segmentation of 3D Images Used in the Shoot Apical Meristem. *IEEE Intl. Symp. on Biomedical Imaging (ISBI)*. 10.1109/ISBI.2018.8363600, pp. 391-395, 2018.

[9] A.C.M. Tavares, P.A.V. Miranda, T.V. Spina, and A.X. Falcão. A Supervoxel-based Solution to Resume Segmentation for Interactive Correction by Differential Image-Foresting Transforms. *13th International Symposium on Mathematical Morphology and its Application to Signal and Image Processing*, LNCS 10225, 10.1007/978-3-319-57240-6_9, pp. 107–118, 2017.

[10] P.A.V. Miranda, A.X. Falcão, G. Ruppert and F. Cappabianco. How to Fix any 3D Segmentation Interactively via Image Foresting Transform and its use in MRI Brain

Segmentation. *8th IEEE Intl. Symp. on Biomedical Imaging: From Nano to Macro* (ISBI), 10.1109/ISBI.2011.5872811, pp. 2031–2035, 2011.

[11] T.V. Spina, S.B. Martins, and A.X. Falcão. Interactive Medical Image Segmentation by Statistical Seed Models. *XXIX SIBGRAPI - Conference on Graphics, Patterns and Images*, doi: 10.1109/SIBGRAPI.2016.045, pp. 273–280, 2016.

[12] P. Rauber, A.X. Falcão, T.V. Spina, and P.J. de Rezende. Interactive Segmentation by Image Foresting Transform on Superpixel Graphs. *Proc. of the XXVI SIBGRAPI - Conference on Graphics, Patterns and Images*, 10.1109/SIBGRAPI.2013.27, pp. 131–138, 2013.

[13] P. A. V. Miranda and L. A. C. Mansilla. Oriented Image Foresting Transform Segmentation by Seed Competition, *IEEE Transactions on Image Processing*, 10.1109/TIP.2013.2288867, 23(1), pp. 389-398, 2014.

[14] M.A.T. Condori, F.M. Cappabianco, A.X. Falcão, and P.A.V. de Miranda. An Extension of the Differential Image Foresting Transform and its Application to Superpixel Generation. *Journal of Visual Communication and Image Representation*, doi 10.1016/j.jvcir.2019.102748, 2020, to appear.

[15] L.M.C. Leon and P.A.V. de Miranda. Multi-object Segmentation by Hierarchical Layered Oriented Image Foresting Transform. *30th Conf. on Graphics, Patterns, and Images* (SIBGRAPI), IEEE, doi: 10.1109/SIBGRAPI.2017.17, Niterói, RJ, pp. 79–86, 2017.