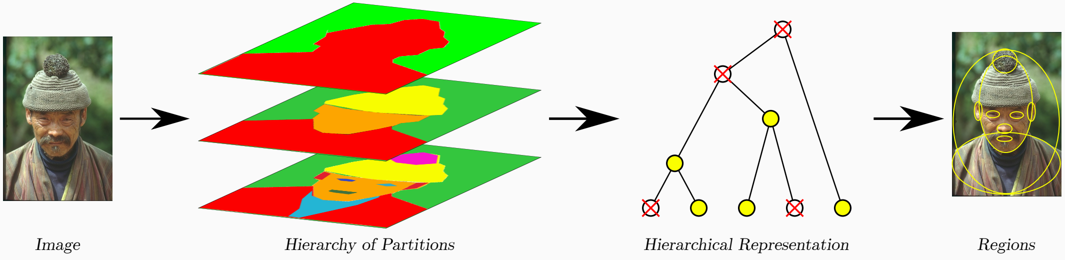
Out-of-core Attribute Algorithms for Binary Partition Hierarchies

Josselin Lefèvre, Jean Cousty, Benjamin Perret, Harold Phelippeau

Introduction

Hierarchical segmentation: introduction

- Regions/clusters of interest do not all appear at the same scale



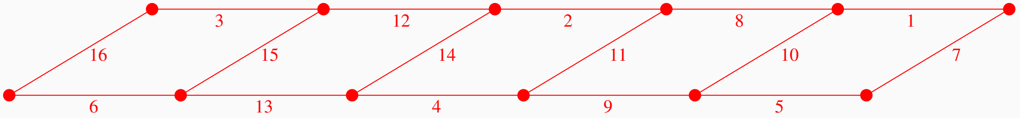
- A hierarchy is a series of nested partitions of a (image) domain:
  - a series $(\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$ of partitions of a set V such that for any i in $\{0, \ldots, \ell - 1\}$, any element of $\mathbf{P}_i$ is included in an element of $\mathbf{P}_{i+1}$

MST and BPH: introduction

- Binary Partition Hierarchies by altitude ordering (BPH) and Minimum Spanning Trees (MST) are
    - Key structures for hierarchical analysis
        - Watershed, constrained connectivity, quasi-flat zone, ultrametric opening, etc.
    - Obtained from
        - a weighted graph $G = (V, E, w)$ and
        - a total ordering $\prec$ of the edges of this graph

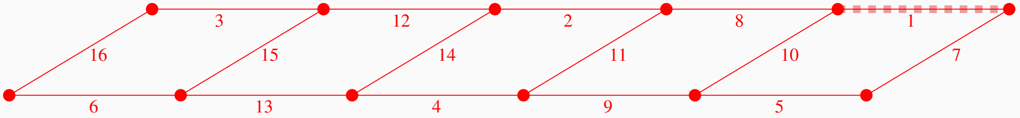
- Intuitively, the BPH can be seen as the hierarchy of partitions obtained during Kruskal's minimum-spanning-tree algorithm

MST and BPH: introduction



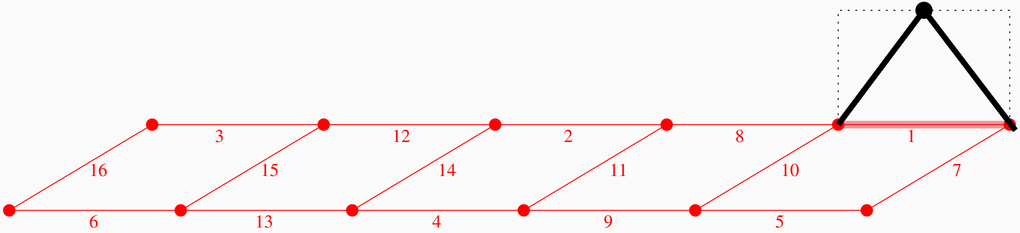
Input graph; weights indicate edge ordering

MST and BPH: introduction



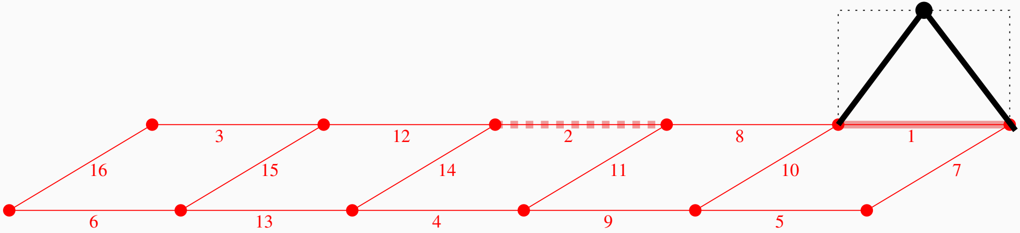
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



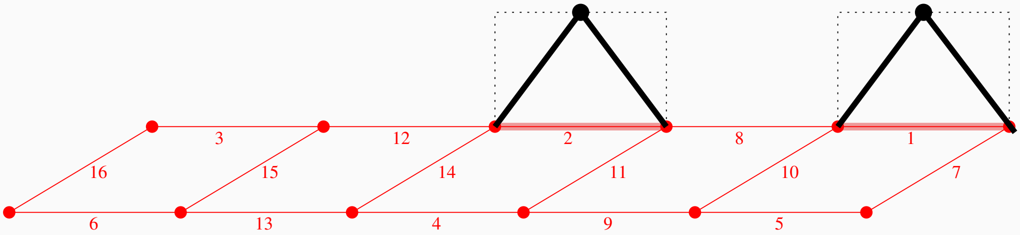
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



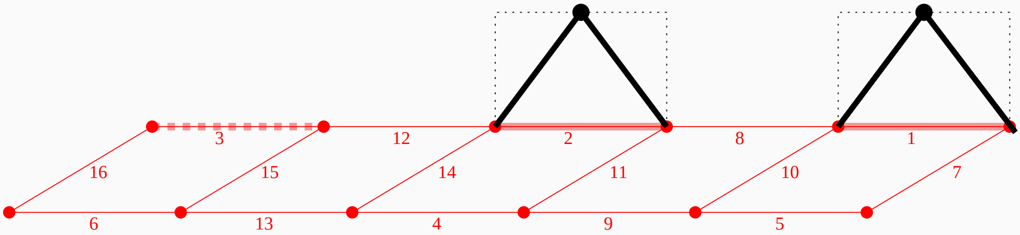
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



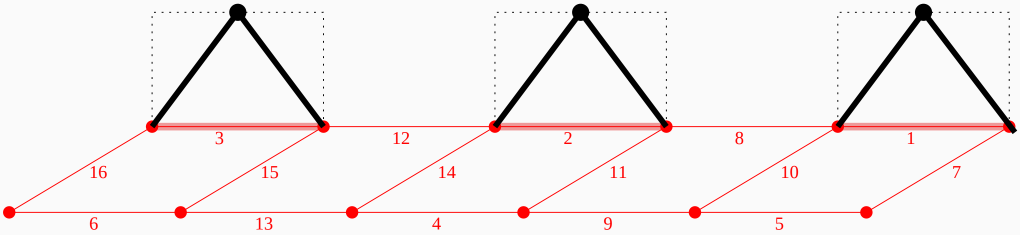
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



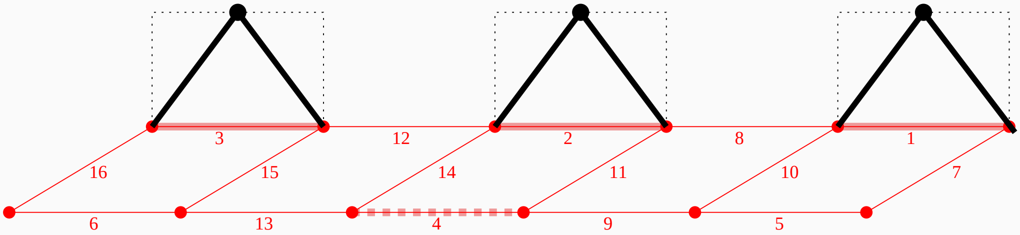
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



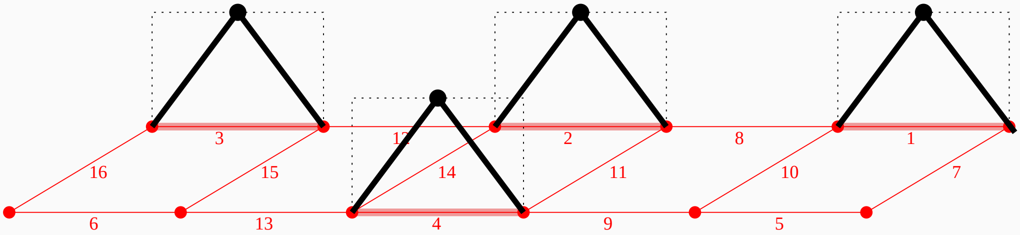
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



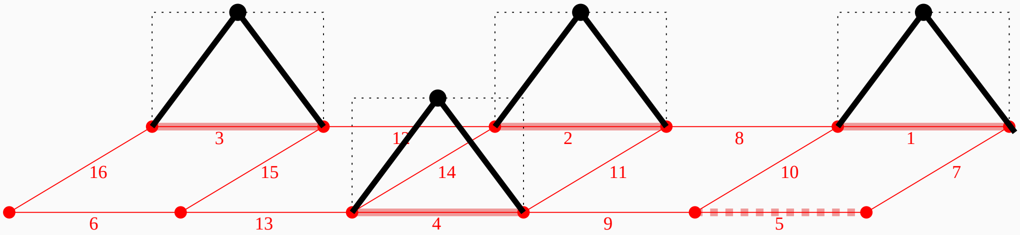
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



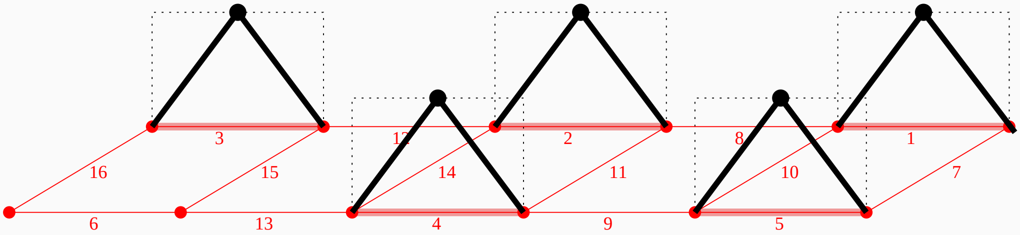
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



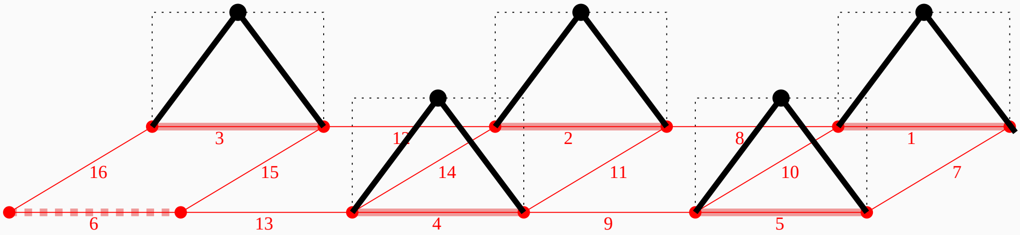
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



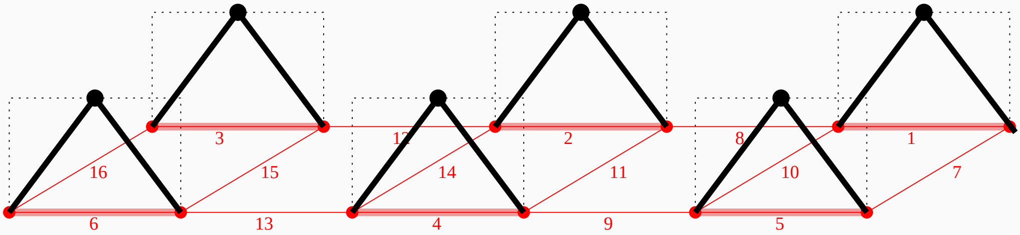
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



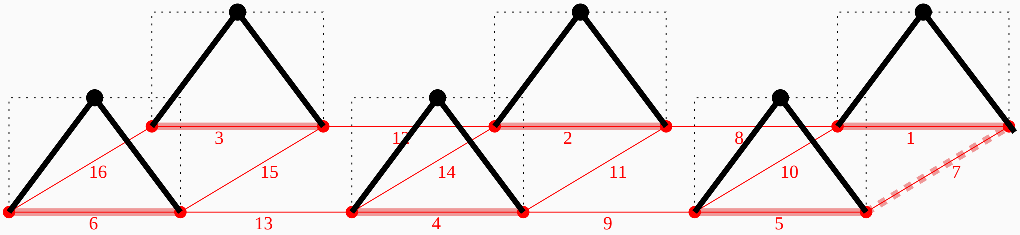
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction

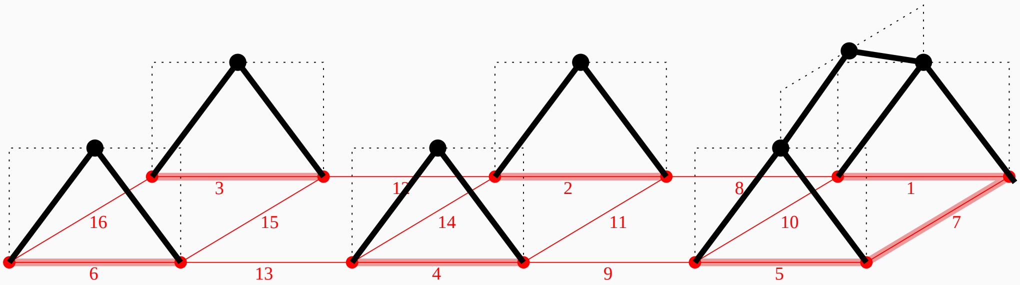


MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction

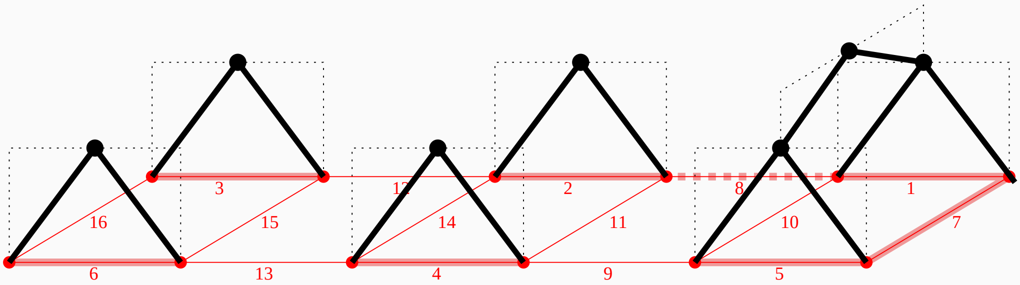


MST (**red**) and BPH (**black**) in construction



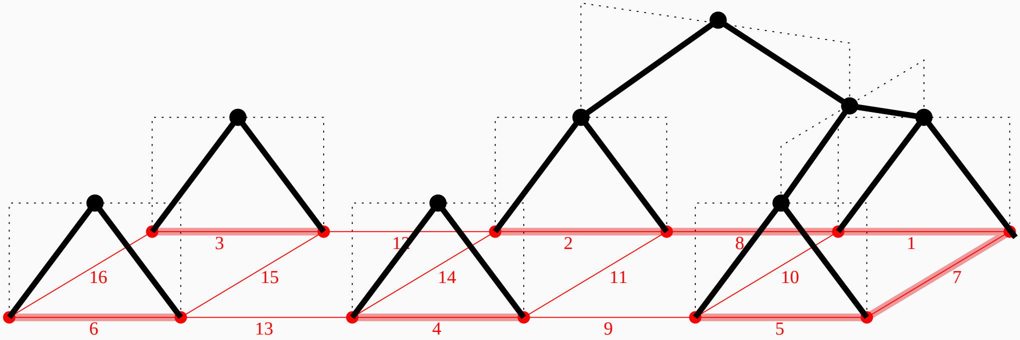
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



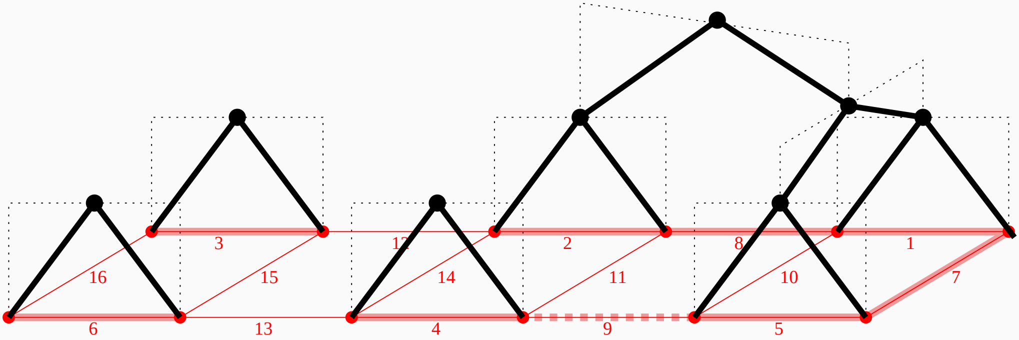
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



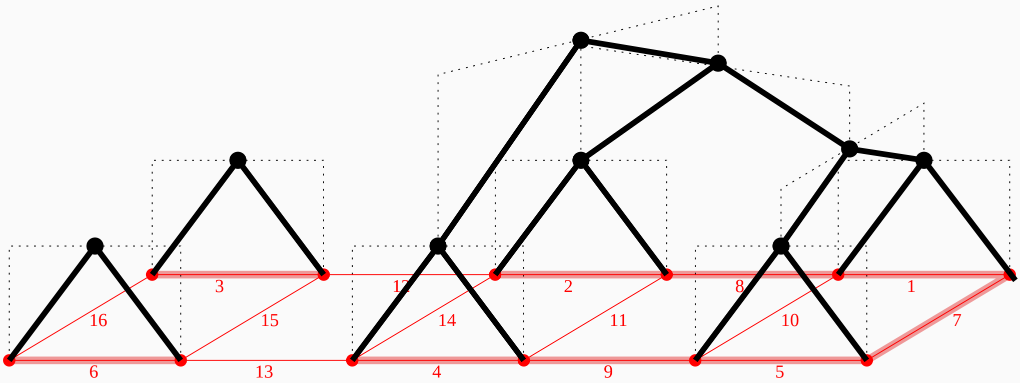
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



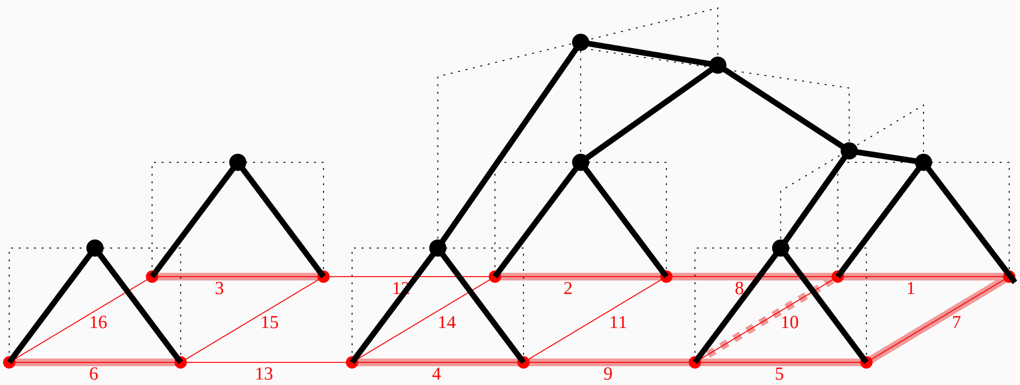
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



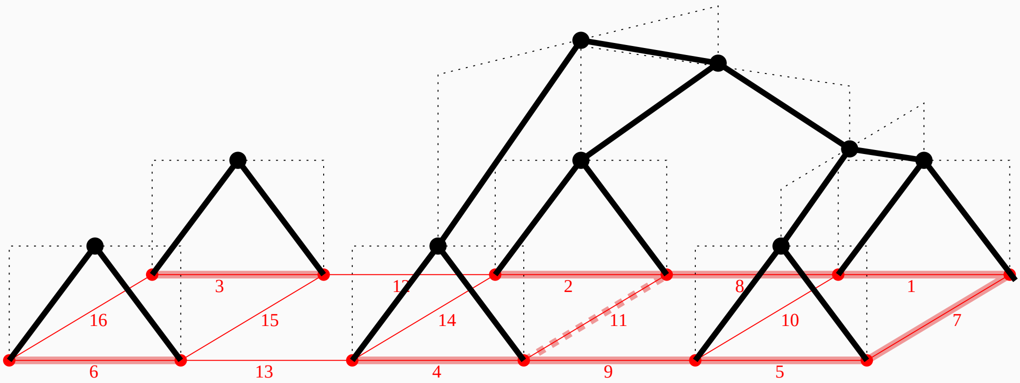
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



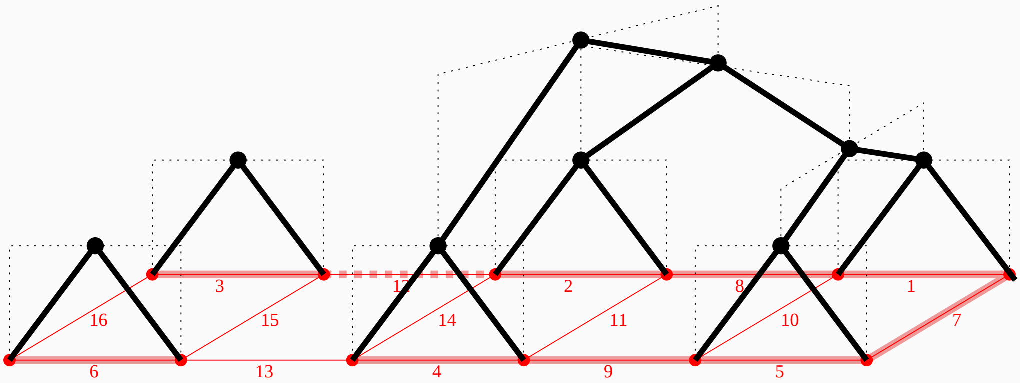
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



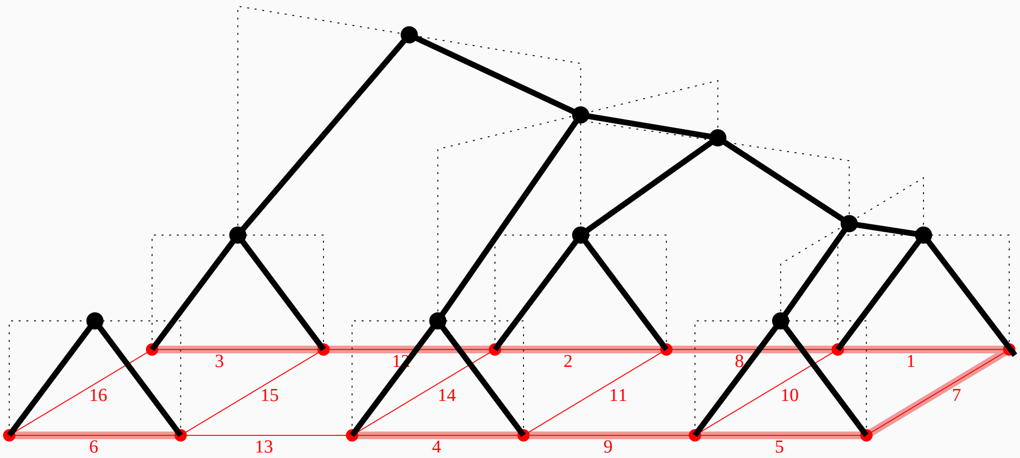
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



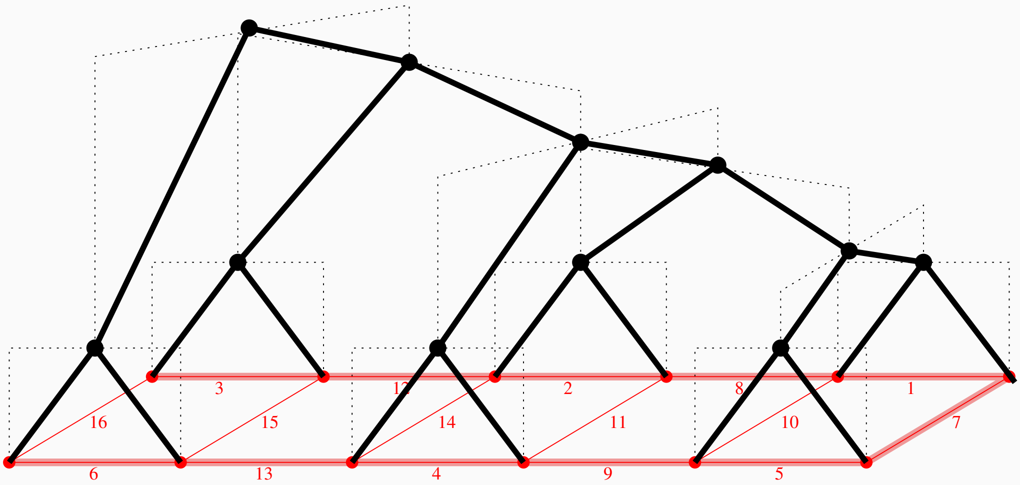
MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



MST (**red**) and BPH (**black**) in construction

MST and BPH: introduction



MST (**red**) and BPH (**black**)

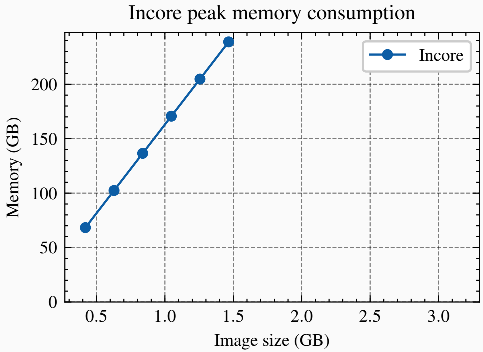
Out-of-core algorithms: introduction

Problem

- When the image exceeds a certain size
  - data cannot fit in the main memory
  - usual sequential algorithm fails to produce a result
- Example: biological images

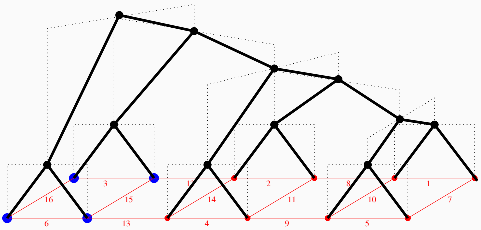
Solution - Out-of-core algorithms

- Produce the same result as the usual algorithms
- Minimize the size of the data structures in memory

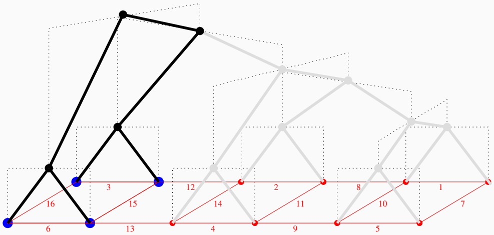


Out-of-core BPH: formal problem statement

- *select*$(X, \mathcal{H})$
  - arg. 1: X is a subset of V
  - arg. 2: $\mathcal{H}$ is a hierarchy
  - result: the hierarchy made of every region of $\mathcal{H}$ that hits X



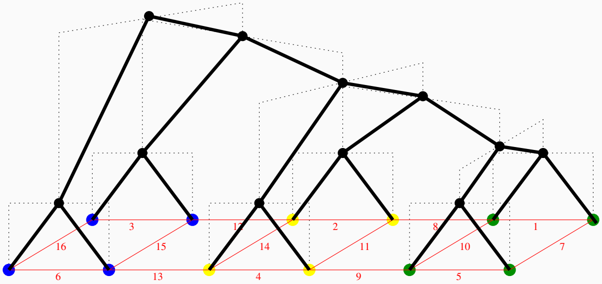
$\mathcal{H}$ (black) and X (blue)



select$(X, \mathcal{H})$

Distributed hierarchies

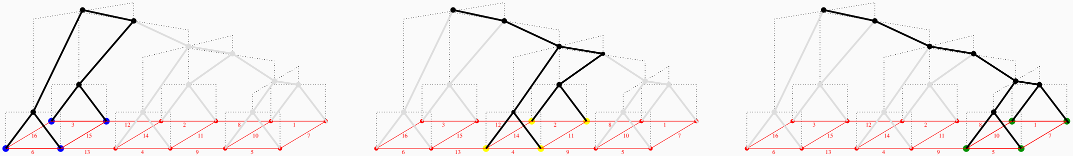
- The distribution of $\mathcal{H}$ over a partition $\mathbf{P}$ of the ground of $\mathcal{H}$ is
  - $\{select\,(R, \mathcal{H}) \mid R \in \mathbf{P}\}$.



$\mathcal{H}$ (black) and $\mathbf{P}$ (blue, yellow, green)

Distributed hierarchies

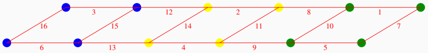
- The distribution of $\mathcal{H}$ over a partition $\mathbf{P}$ of the ground of $\mathcal{H}$ is
  - $\{select\,(R, \mathcal{H})\ \mid R \in \mathbf{P}\}$.



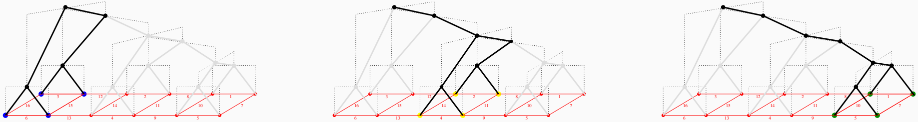
Distribution of $\mathcal{H}$ over $\mathbf{P}$

Out-of-core BPH framework

Given:



Find:

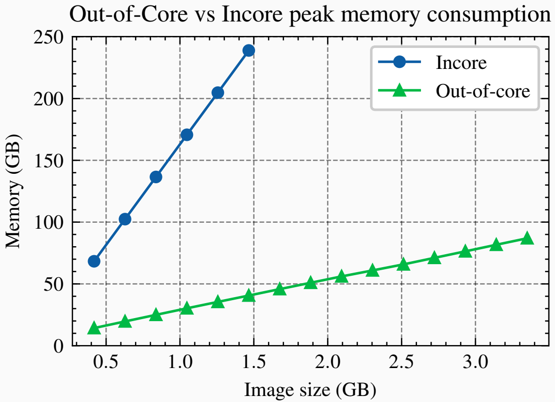


- Compute the distribution of the binary partition hierarchy $\mathcal{H}$ of $(G, \prec)$ over a partition of its ground set
    - Without computing $\mathcal{H}$
    - Where each computation step requires a limited amount of memory

Experimental Result

Peak memory comparison

- Beyond 1.5GB, the incore algorithm cannot produce results
- Out-of-core algorithm's slope is 6.7 times lower than the incore one



Out-of-Core Representation: What do we do with this?

Objective

- Once the distribution of a binary partition hierarchy is available, post-process it in an out-of-core manner to obtain useful information for applications:
  - **Regional attributes**
  - (Hierarchical) Watershed
  - Quasi flat zone hierarchies
  - Connected image filtering
  - Etc.

Attribute in Out-of-Core Context

Definition: Attribute

An *attribute on* $\mathcal{H}$ is a mapping A associating an attribute value (scalar, boolean) $A(R)$ to every region R of $\mathcal{H}$.

Definition: Causal Partition

A *causal partition* is a series $(S_0, \ldots, S_k)$ of *slices* such that

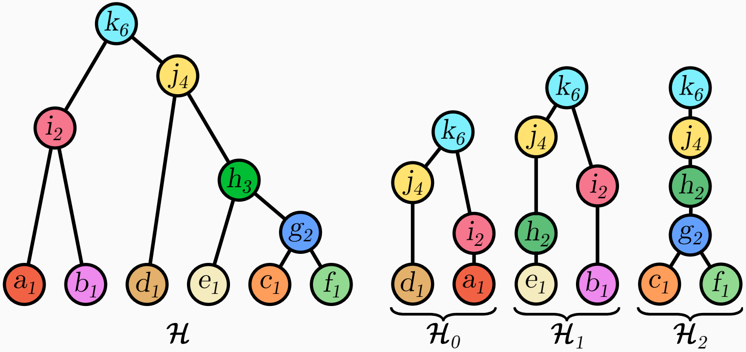
- $\{S_0, \ldots, S_k\}$ is a partition of V
- S_i is only adjacent to S_{i-1} and to S_{i+1}

Definition: Distribution of an Attribute

Given an attribute A on $\mathcal{H}$ and $\delta_{\mathcal{H}}$ the distribution of $\mathcal{H}$ over a causal partition $(S_0, \cdots, S_k)$, we define the distribution of A over $\delta_{\mathcal{H}}$ as the series $\delta_A = (A_{\mathcal{B}_0}, \cdots, A_{\mathcal{B}_k})$ such that for any i in $\{0, \ldots, k\}$ and any R in $\mathcal{B}_i$, we have $A(R) = A_{\mathcal{B}_i}(R)$.

Example of Distribution of Area Attribute

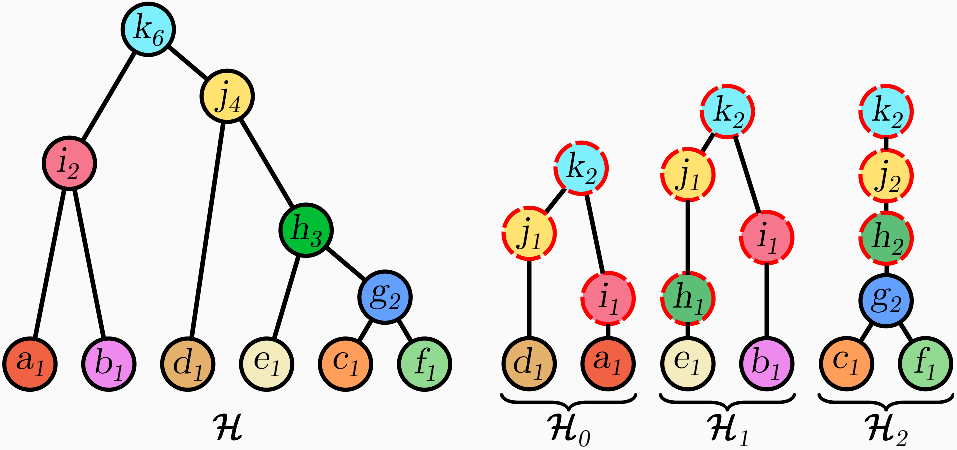
Let $\mathcal{H}$ be a binary partition hierarchy and its distribution on the causal partition $(\{a, d\}, \{b, e\}, \{c, f\})$



Area: definition

The area of a region is equal to the sum of the area of vertices in this regions.

Problem: We cannot use regular algorithm to independently compute area on each local hierarchy



Problem statement

Problem

- Algorithms for in core (classical) attributes computation fails to produce a result as they run out of memory
- The information present in one single element of the distribution is not sufficient to compute a correct attribute value as the attribute of a region is a global value

Solution - Out-of-core algorithms

- Produce the same result as the usual algorithms
- Minimize the size of the data structures in memory
- Working on the elements of the distribution should yield the same result as working on the global hierarchy

OOC attribute computation

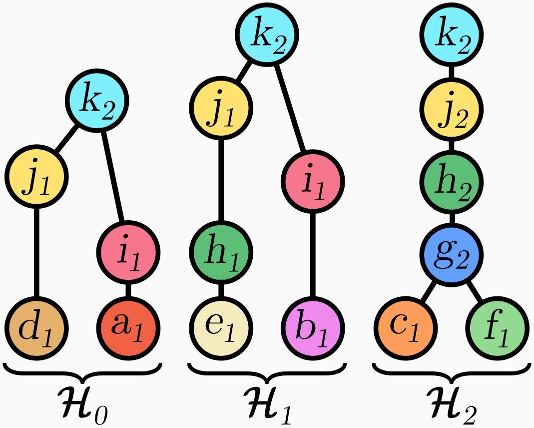
OOC attribute computation: overview

3 Steps Workflow

1. Compute a *partial attribute* value locally on each tree of the distribution;
2. *Propagate* partial attribute values from neighboring trees in the causal direction, *i.e.*, from slices of lower indices to those of higher indices;
3. Back-propagate the “correct” attribute values in the anti-causal direction, *i.e.*, from slices of higher indices to those of lower indices. At the end of this step, all the attribute values of all the trees in the distribution are correct.

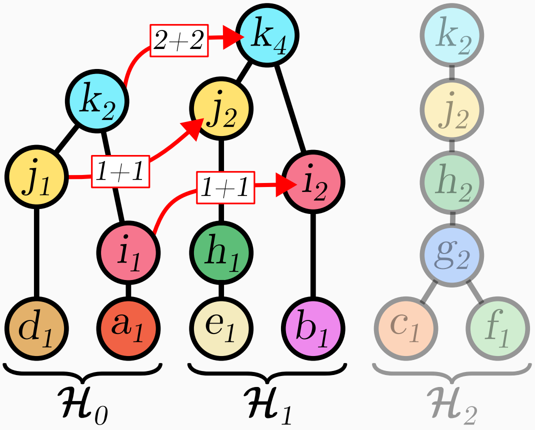
Area attribute: local initialization

1. Compute partial area attribute locally with established algorithm



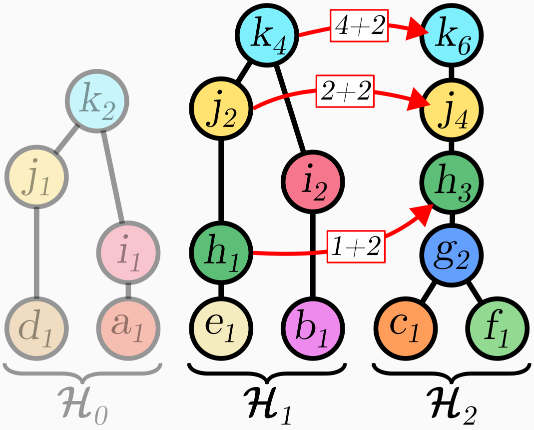
Area attribute: propagate values #1

2. Causal traversal: **Merge** attributes of $(\mathcal{H}_0, \mathcal{H}_1)$ on $\mathcal{H}_1$ using $+$ operation

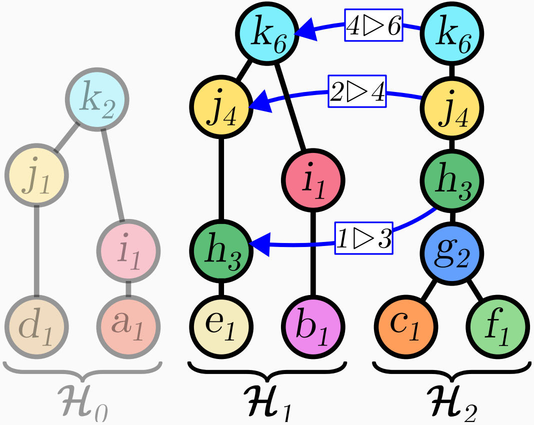


Area attribute: propagate values #1

2. Causal traversal: **Merge** attributes of $(\mathcal{H}_1, \mathcal{H}_2)$ on $\mathcal{H}_2$ using $+$ operation



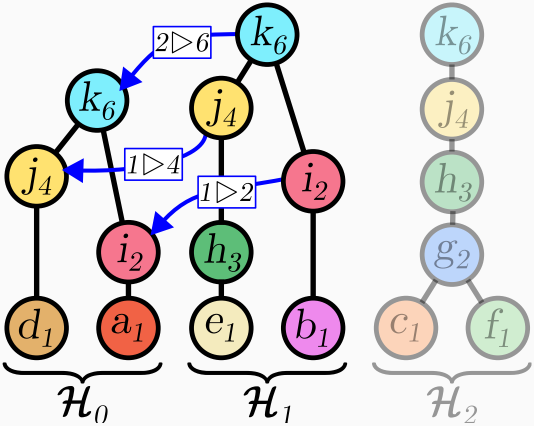
3. Anti-causal traversal: **Merge** attributes of $(\mathcal{H}_2, \mathcal{H}_1)$ on $\mathcal{H}_1$ using $\triangleright$ operator



$^* a \triangleright b = b$

Area attribute: propagate values #2

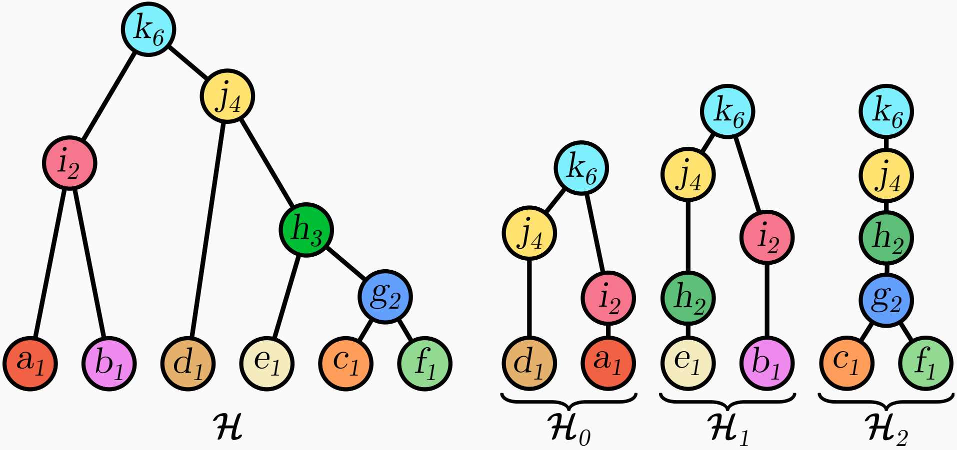
3. Anti-causal traversal: **Merge** attributes of $(\mathcal{H}_1, \mathcal{H}_0)$ on $\mathcal{H}_0$ using $\triangleright$ operator



$^* a \triangleright b = b$

Area attribute

Attribute area on local hierarchies is equal to the expected result as computed on $\mathcal{H}$: **we successfully computed the distribution of the area**.



Pseudo-code and complexity

An efficient algorithm to compute attributes

- **Merge** has a linear complexity with respect to the total number of regions in the two hierarchies
- $O(k)$ calls to **Merge**, where k is the number of parts onto which the data are split

Algorithm 1: PROPAGATE

Data: The distribution $(\mathcal{B}_0, \ldots, \mathcal{B}_k)$ of a BPH; a series $(A_{\mathcal{B}_0}, \ldots, A_{\mathcal{B}_k})$ of locally initialized attributes; and a binary operator $\oplus$.

Result: A new series $(A^{\downarrow}_{\mathcal{B}_0}, \ldots, A^{\downarrow}_{\mathcal{B}_k})$.

```
A↑_{B_0} := A_{B_0}
foreach i from 1 to k do
    A↑_{B_i} := MERGE(B_{i−1}, B_i, A↑_{B_{i−1}}, A_{B_i}, ⊕)
A↓_{B_k} := A↑_{B_k}
foreach i from k − 1 to 0 do
    A↓_{B_i} := MERGE(B_{i+1}, B_i, A↓_{B_{i+1}}, A↑_{B_i}, ◁)
return (A↓_{B_0}, ..., A↓_{B_k})
```

$A^{\uparrow}_{\mathcal{B}_0} := A_{\mathcal{B}_0}$

foreach *i from* 1 *to* k **do**

  $A^{\uparrow}_{\mathcal{B}_i} := \text{MERGE}(\mathcal{B}_{i-1}, \mathcal{B}_i, A^{\uparrow}_{\mathcal{B}_{i-1}}, A_{\mathcal{B}_i}, \oplus)$

$A^{\downarrow}_{\mathcal{B}_k} := A^{\uparrow}_{\mathcal{B}_k}$

foreach *i from* $k-1$ *to 0* **do**

  $A^{\downarrow}_{\mathcal{B}_i} := \text{MERGE}(\mathcal{B}_{i+1}, \mathcal{B}_i, A^{\downarrow}_{\mathcal{B}_{i+1}}, A^{\uparrow}_{\mathcal{B}_i}, \triangleleft)$

return $\left(A^{\downarrow}_{\mathcal{B}_0}, \ldots, A^{\downarrow}_{\mathcal{B}_k}\right)$

A Generic Methodology

Various attributes can be computed depending on the local initialization and $\oplus$ the operator.

Attribute	Operator $\oplus$
Area	$+$
Volume	$+$
Height	max
Topological height	max
Minima	min / $\wedge$
Rightmost vertex	max
Number of Minima*	$+$

This methodology can be extended to:

- Bounding boxes
- Mean grayscale value
- Determine watershed edge

** requires a non local pre-processing relying on Propagate*

Main contributions

Main contributions

- Efficient algorithms for the out-of-core computation of the distribution of Binary Partition Hierarchy

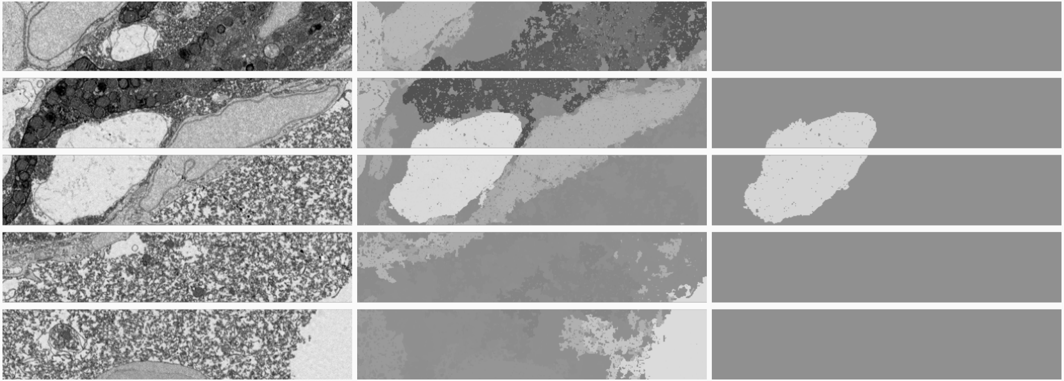
  Join, Select, and Insert, Efficient Out-of-core Algorithms for Hierarchical Segmentation Trees, Lefèvre, J., Cousty, J., Perret, B., Phelippeau, H., DGMM22
- A global scheme for computing attributes for a distribution of Binary Partition Hierarchy under out-of-core constraint
- Efficient and easily implementable algorithms

Perspectives

Perspectives

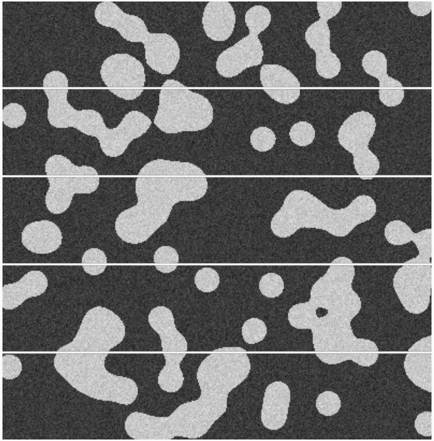
- Out-of-core connected filtering
- Out-of-core (seeded) watershed
- Extend this work to the computation of more complex attributes such as extinction values, persistences, smallest common ancestors, etc.

Attribute filtering

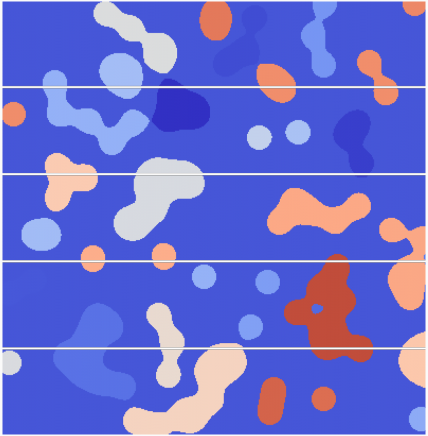


Partitioned grayscale image — Out-of-core area filtering ($a < \theta$) — Out-of-core volume filtering ($v < \theta_1 \lor v > \theta_2$)

Out-of-core Segmentation



Noisy blobs



Out-of-core seeded Watershed

Questions?

`https://github.com/PerretB/Higra-distributed`
`(pip install higra)`

Experiment

Execution time comparison

- Beyond 1.5GB, the incore algorithm cannot produce results
- A trade off can be found between number of slices and execution time
- ≈ 50% of computation time is dedicated to IO