

# How to Modify the Tree of Shapes of an Image: Connected Operators Without Gradient Inversion

---

Julien Mendes Forte<sup>(1)</sup>, Nicolas Passat<sup>(2)</sup>, Yukiko Kenmochi<sup>(1)</sup>

(1) Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France

(2) Université de Reims Champagne-Ardenne, CRESTIC, Reims, France

RFIAP 2024, Lille, July 1st 2024



# Preliminaries

---

## Image filtering

- Filter: modifies the content of an image  
→ denoising, enhancement, simplification. . .
- Connected filters: do not create contours

# Connected filters

## Image filtering

- Filter: modifies the content of an image  
→ denoising, enhancement, simplification. . .
- Connected filters: do not create contours

## Connected filters: advantages

- Avoid blurring effects
- Preserve the structure of the image

# Connected filters

## Image filtering

- Filter: modifies the content of an image  
→ denoising, enhancement, simplification. . .
- Connected filters: do not create contours

## Connected filters: advantages

- Avoid blurring effects
- Preserve the structure of the image

## Connected filters: paradigm

- Standard filters act at the scale of **pixels**
- Connected filters act at the scale of **flat zones**

# Connected filters & morphological trees

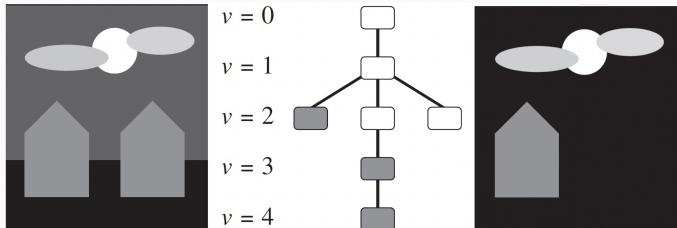
## Filtering paradigm

Image → Tree → Pruned tree → Filtered image

# Connected filters & morphological trees

## Filtering paradigm

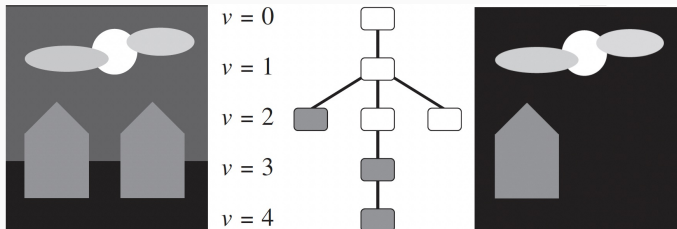
Image  $\rightarrow$  Tree  $\rightarrow$  Pruned tree  $\rightarrow$  Filtered image



# Connected filters & morphological trees

## Filtering paradigm

Image  $\rightarrow$  Tree  $\rightarrow$  Pruned tree  $\rightarrow$  Filtered image



## Morphological trees

- Component-tree (Max-tree, Min-tree)
- Tree of shapes



# Connected filters & tree of shapes

## Tree of shapes (aka level-line tree)

- Models an image as a topographic map
- Nodes = isocontour regions / Edges = nesting relation

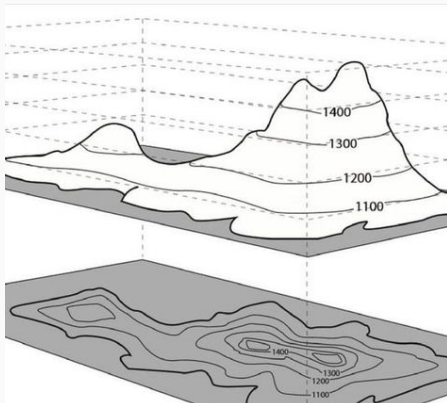
Filtering an image via its tree of shapes: relevant but not trivial!

# Connected filters & tree of shapes

## Tree of shapes (aka level-line tree)

- Models an image as a topographic map
- Nodes = isocontour regions / Edges = nesting relation

Filtering an image via its tree of shapes: relevant but not trivial!



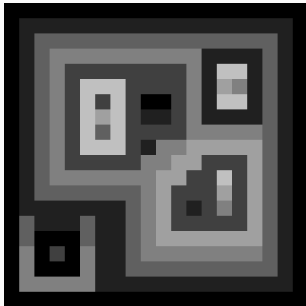
# Introduction

---

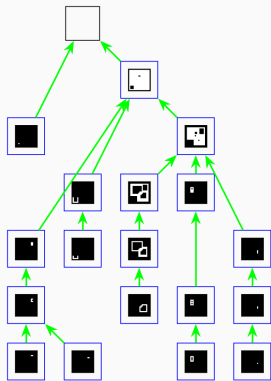
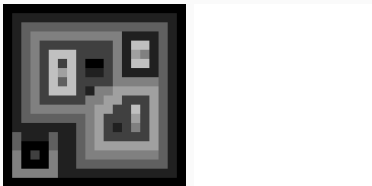
# Context and framework

## Support, value, image

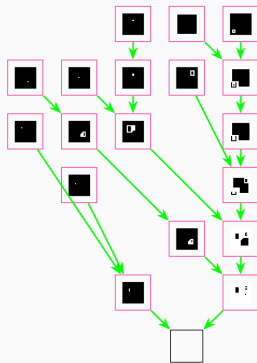
- Discrete support  $\mathbb{U} = \mathbb{Z}^n$  ( $n \geq 2$ )
- Values  $\mathbb{V} \simeq \mathbb{Z}$
- Image = function  $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$
- The Jordan-Brouwer property holds



# Morphological trees: Component trees

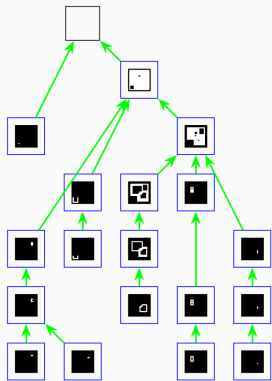
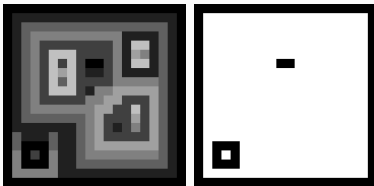


Max-tree

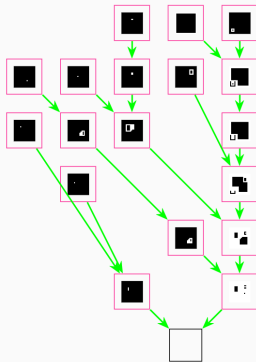


Min-tree

# Morphological trees: Component trees

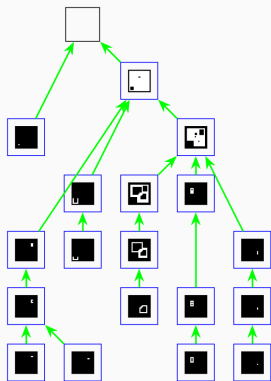
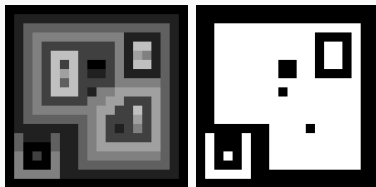


Max-tree

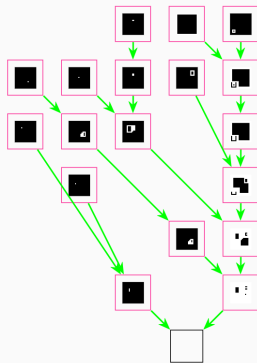


Min-tree

# Morphological trees: Component trees

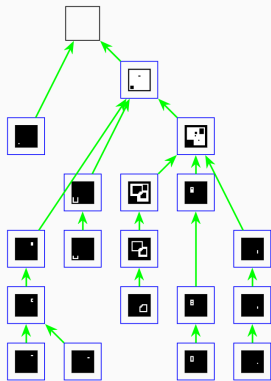
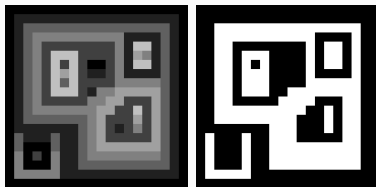


Max-tree

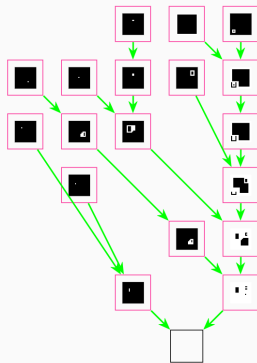


Min-tree

# Morphological trees: Component trees



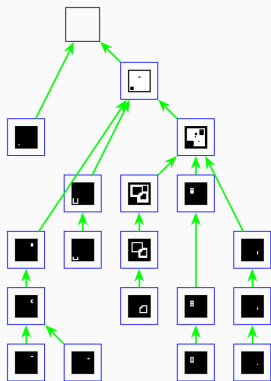
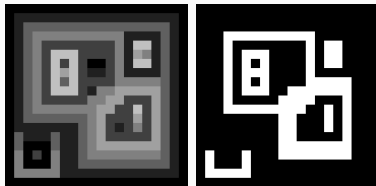
Max-tree



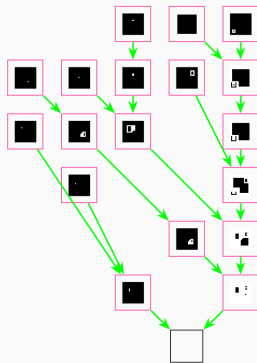
Min-tree



# Morphological trees: Component trees

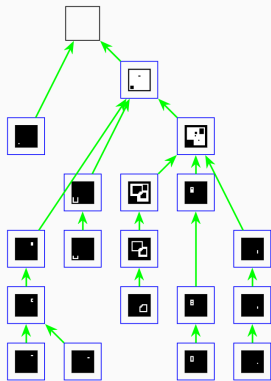
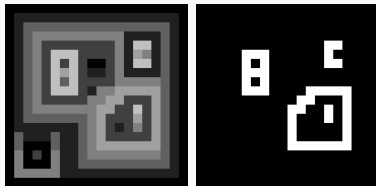


Max-tree

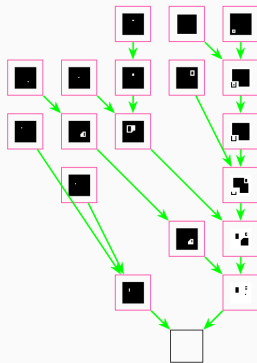


Min-tree

# Morphological trees: Component trees

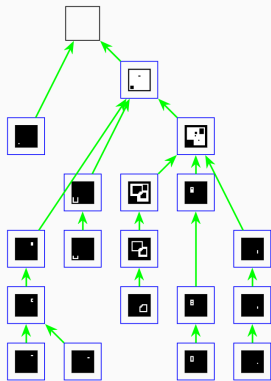
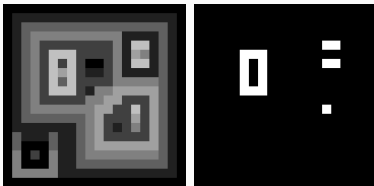


Max-tree

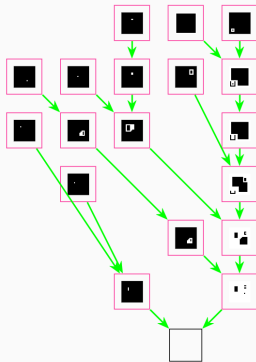


Min-tree

# Morphological trees: Component trees

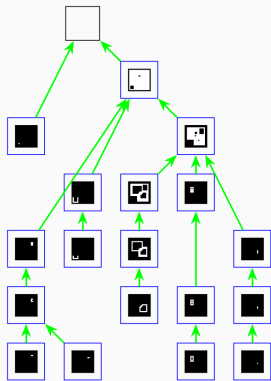
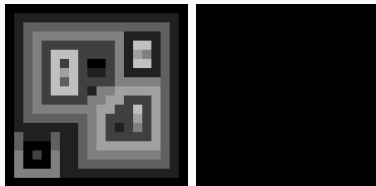


Max-tree

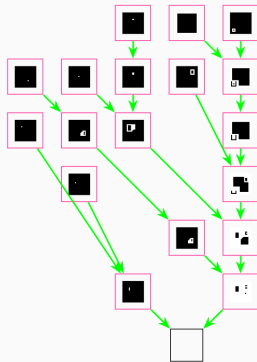


Min-tree

# Morphological trees: Component trees

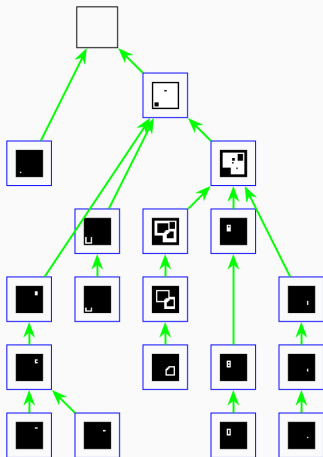


Max-tree

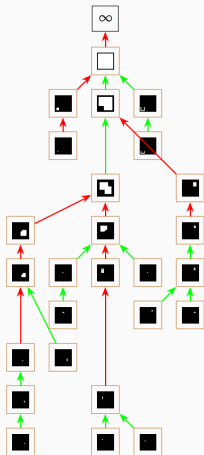


Min-tree

# Morphological trees: the tree of shapes



# Morphological trees: the tree of shapes



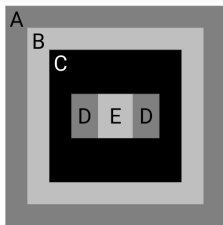
## Tree of shapes

- represents an image through the “nested” relationship of its level lines
- can be computed in quasi-linear time

## Applications: connected filtering

- Filtering
- Segmentation
- Simplification

# Modifying the tree of shapes: issue



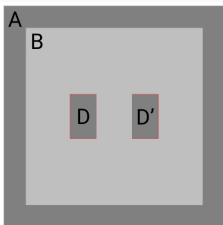
Image



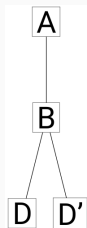
ToS



Modified ToS



Reconstruction



ToS of reconstruction



# Objective

How to:

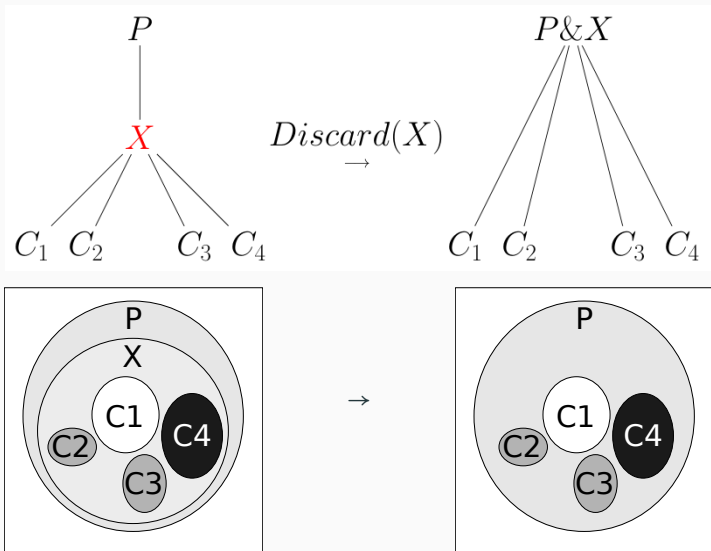
- preserve the **signs of the gradients** and
- preserve a **valid** tree of shapes

as a result of a **modification**?

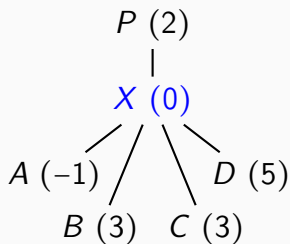
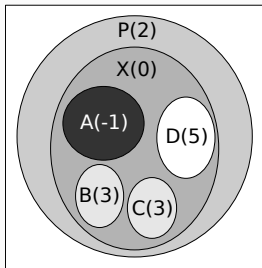
# Modification rules

---

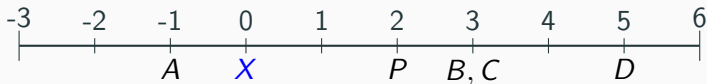
# Atomic operation: discarding a node



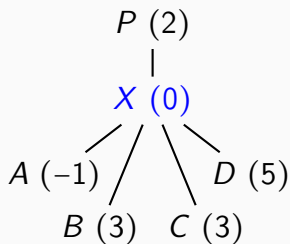
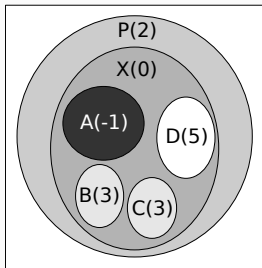
## Modification rules: in a node's neighborhood (1/3)



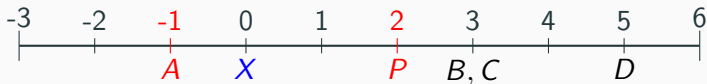
Suppose that we want to set the altitude of  $X$  to 2.



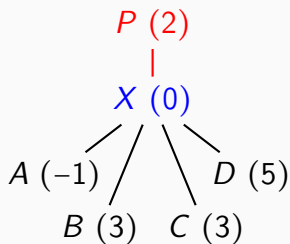
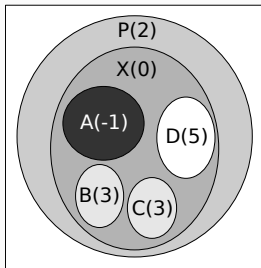
## Modification rules: in a node's neighborhood (1/3)



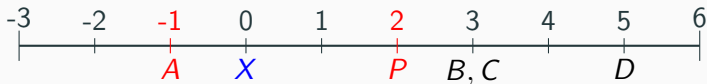
Suppose that we want to set the altitude of X to 2.



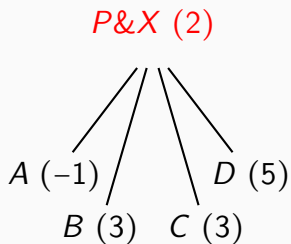
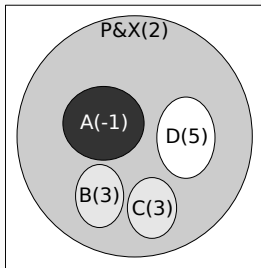
## Modification rules: in a node's neighborhood (1/3)



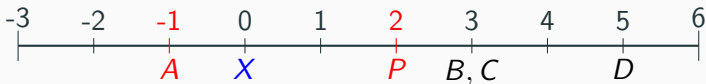
Suppose that we want to set the altitude of X to 2.



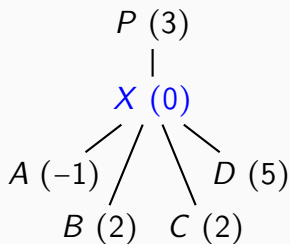
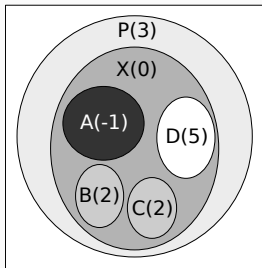
## Modification rules: in a node's neighborhood (1/3)



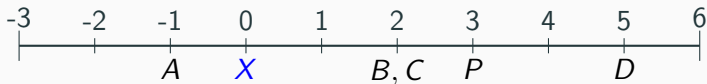
Suppose that we want to set the altitude of X to 2.



## Modification rules: in a node's neighborhood (2/3)

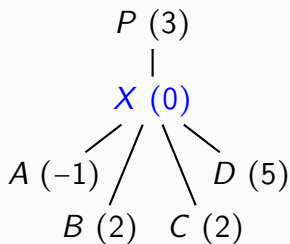
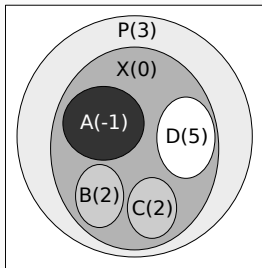


Suppose that we want to set the altitude of X to 2.

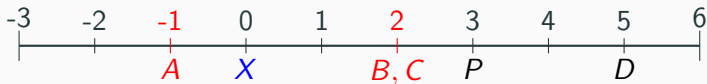




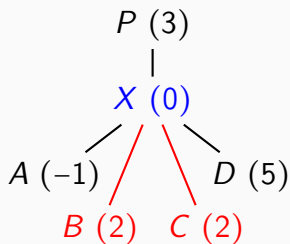
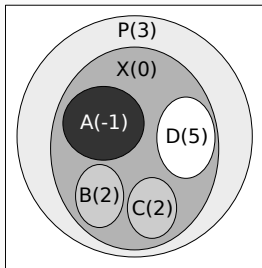
## Modification rules: in a node's neighborhood (2/3)



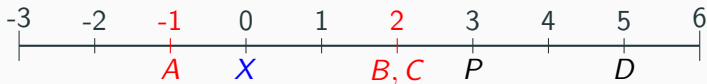
Suppose that we want to set the altitude of X to 2.



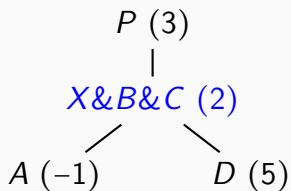
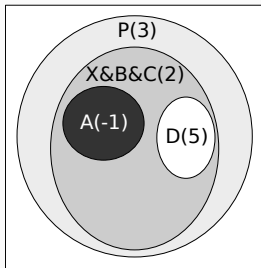
## Modification rules: in a node's neighborhood (2/3)



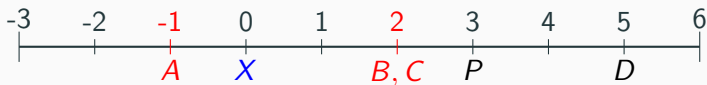
Suppose that we want to set the altitude of X to 2.



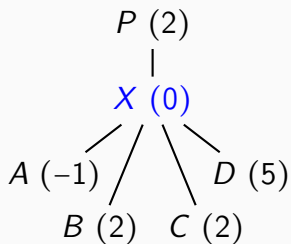
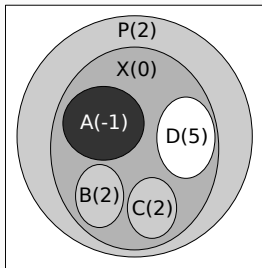
## Modification rules: in a node's neighborhood (2/3)



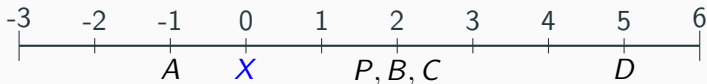
Suppose that we want to set the altitude of  $X$  to 2.



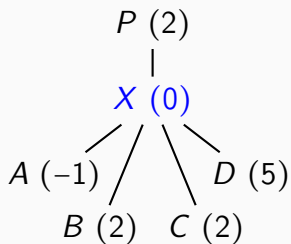
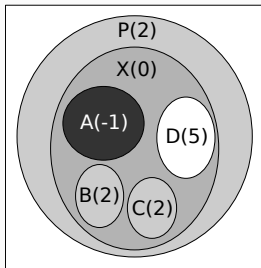
## Modification rules: in a node's neighborhood (3/3)



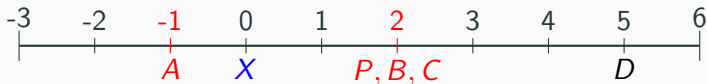
Suppose that we want to set the altitude of  $X$  to 2.



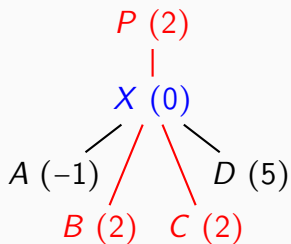
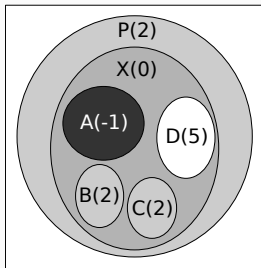
## Modification rules: in a node's neighborhood (3/3)



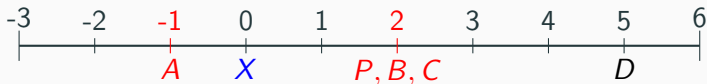
Suppose that we want to set the altitude of  $X$  to 2.



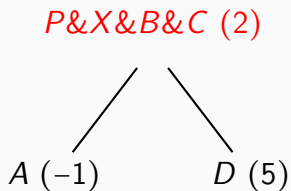
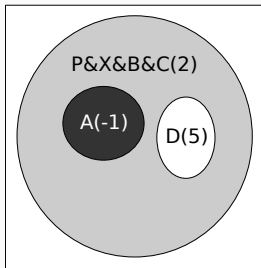
## Modification rules: in a node's neighborhood (3/3)



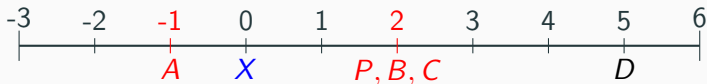
Suppose that we want to set the altitude of X to 2.



## Modification rules: in a node's neighborhood (3/3)



Suppose that we want to set the altitude of  $X$  to 2.



## Modification rules: globally

How to generalize the modification rules to **any** desired altitude?



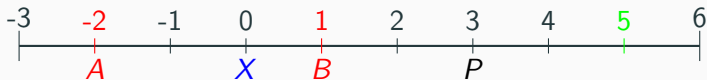
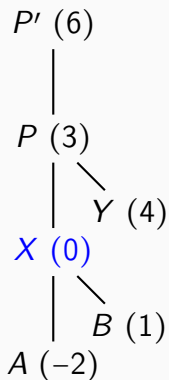
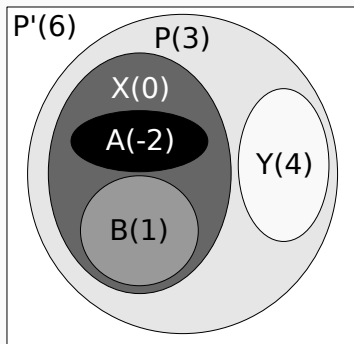
## Modification rules: globally

How to generalize the modification rules to **any** desired altitude?

We apply the **local modification** until the **desired altitude** is reached.

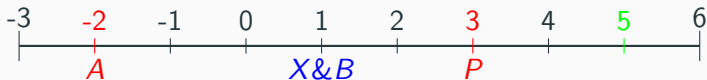
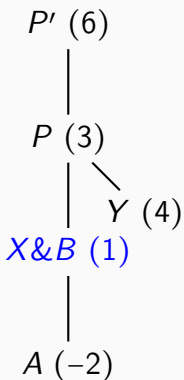
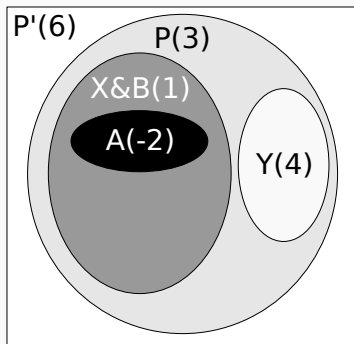
## Modification rules: globally

If we move X to altitude 5:



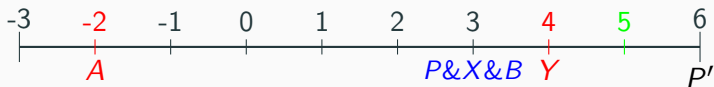
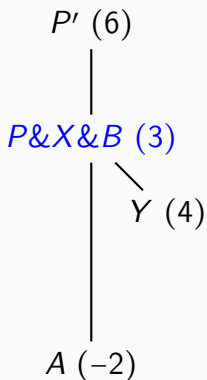
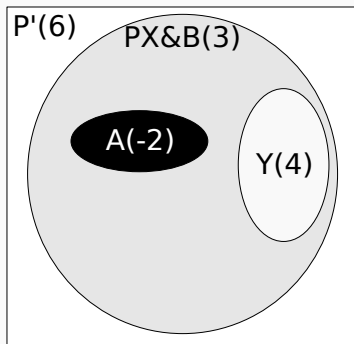
## Modification rules: globally

If we move X to altitude 5:



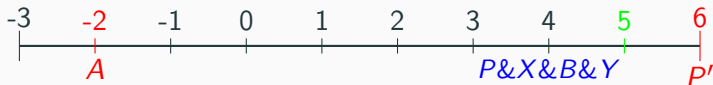
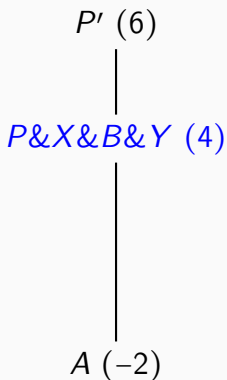
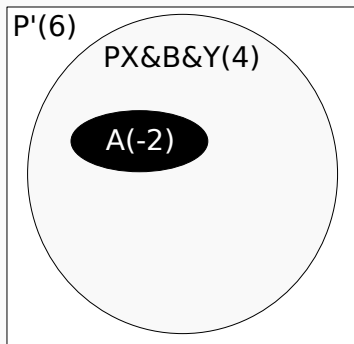
## Modification rules: globally

If we move X to altitude 5:



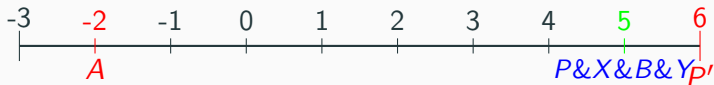
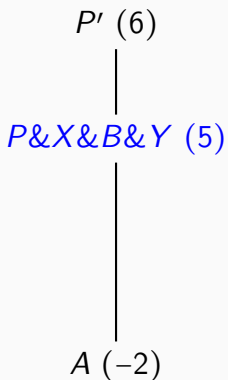
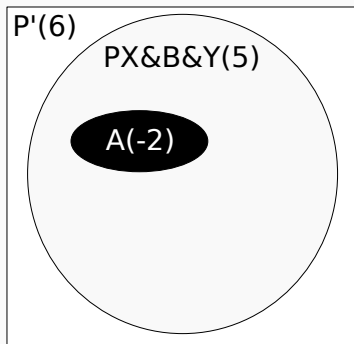
## Modification rules: globally

If we move X to altitude 5:



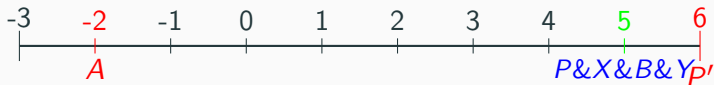
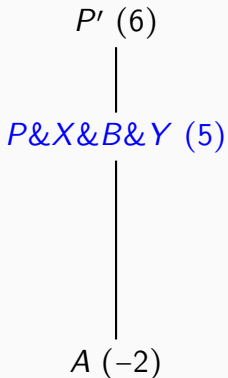
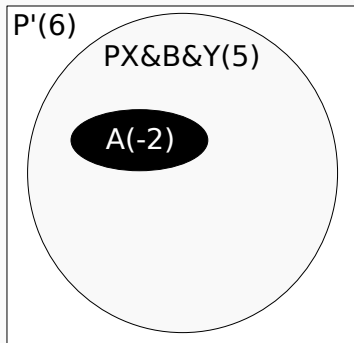
## Modification rules: globally

If we move X to altitude 5:



## Modification rules: globally

If we move X to altitude 5:

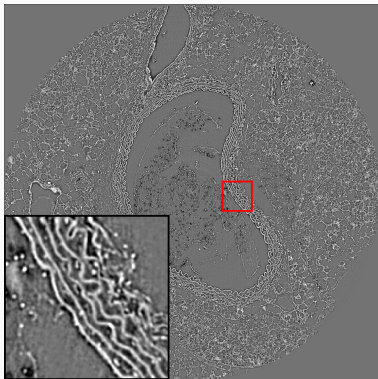


# Applications

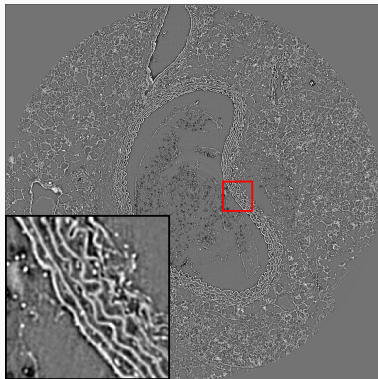
---



# Image simplification: mean connected filtering

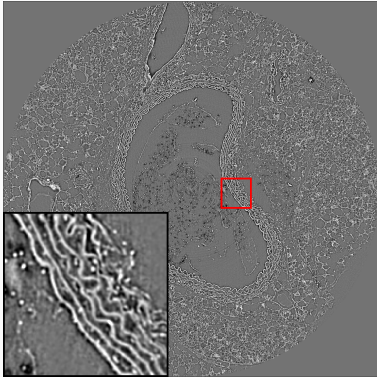


Biological image  
 $4.2 \cdot 10^6$  pixels,  
ToS:  $1.2 \cdot 10^6$  nodes

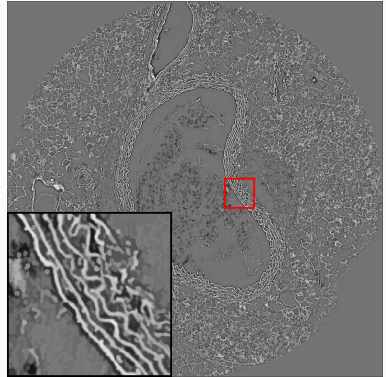


$n = 10$   
SSIM= 0.910  
ToS:  $2.3 \cdot 10^5$  nodes

# Image simplification: mean connected filtering

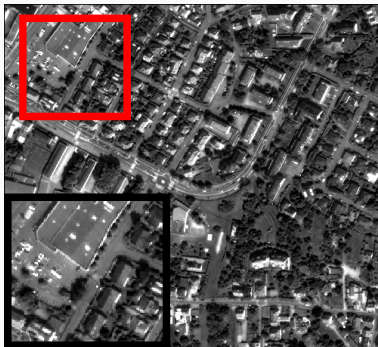


Biological image  
 $4.2 \cdot 10^6$  pixels,  
ToS:  $1.2 \cdot 10^6$  nodes



$n = 10$   
SSIM= 0.910  
ToS:  $2.3 \cdot 10^5$  nodes

# Image filtering: area opening



Satellite image



Proper part  $< 3$

# Image filtering: area opening



Satellite image



Proper part  $< 3$

## Image filtering: area opening



Satellite image



Proper part  $< 10$

## Conclusion

---

## Contributions

- Tree of shapes modification
- Preservation of the structure while modifying the image
- Preservation of the gradient sign

## Perspectives

- Tree of shapes simplification
- Connected operators
- Topological denoising
- Simplification of other trees (e.g. topological tree of shapes)

Thanks for your attention!

If you want to know more, ask this guy:



Julien Mendes Forte

PhD candidate, GREYC, Caen

[julien.mendes-forte@unicaen.fr](mailto:julien.mendes-forte@unicaen.fr)