Meaningful Disjoint Level Lines Selection

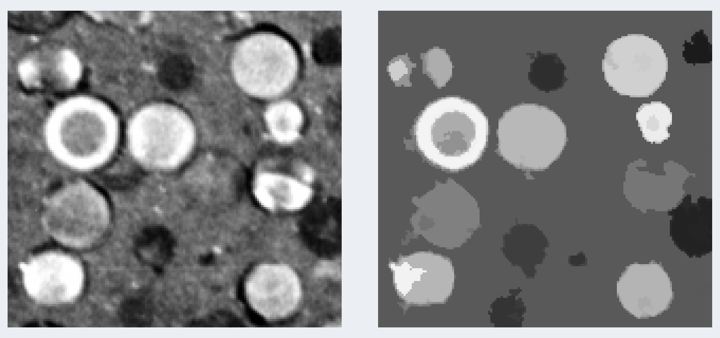
Yongchao Xu[1,2], Edwin Carlinet[1,2], Thierry Géraud[1], Laurent Najman[2]

[1]EPITA Research and Development Laboratory (LRDE), France

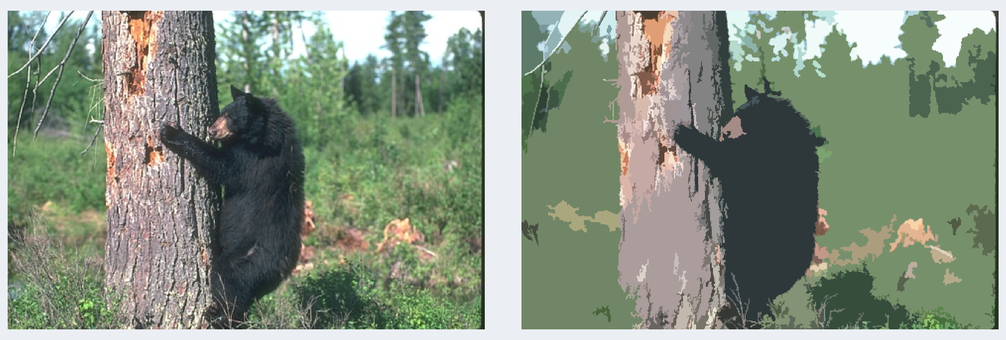
[2]Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge (LIGM), ESIEE Paris, France

{firstname.lastname}@lrde.epita.fr, l.najman@esiee.fr

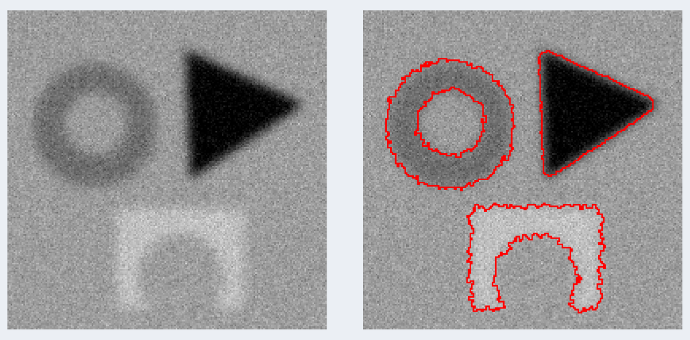
Tree of shapes $\mathcal{T}$ [1, 2]: a versatile tool for many applications



Shape filtering (ICPR 2012).



Energy-driven simplification (ICIP 2013).



Object segmentation (ICIP 2012).

Hierarchical image segmentation (ISMM 2013).

These results are from the PhD work [3] supervised by T. Géraud & L. Najman available in http://www.lrde.epita.fr/wiki/User:Xu ⟶

At a glance

Motivation

- Significant contours of objects ⇔ segments of level lines [1]
- Inclusion relationship ⇒ tree of shapes $\mathcal{T}$ [2]: a versatile representation
- The knowledge of tree structure is fundamental for a deep tree analysis

Problem

- The number of shapes is about as large as the number of pixels

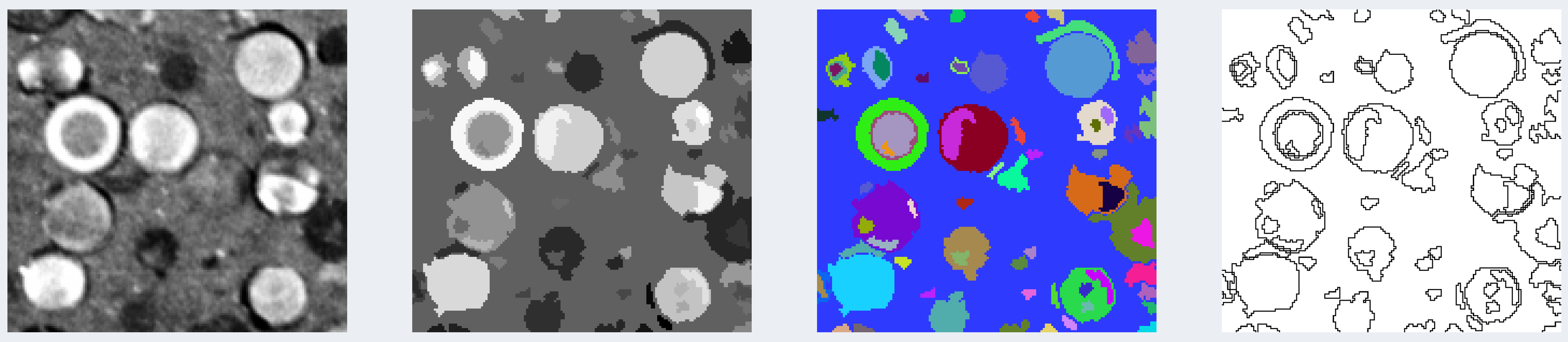
Objective

- Select a subset of level lines representing the main tree structure

Contribution

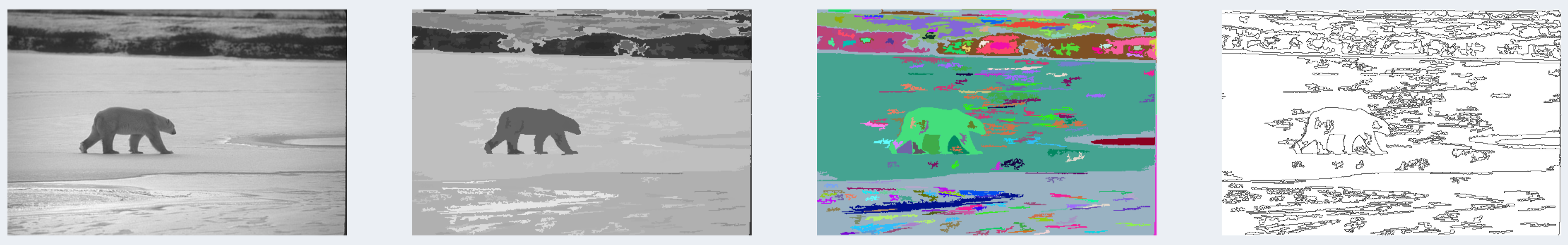
- An efficient algorithm for extracting meaningful and disjoint level lines
- A simplified image providing an intuitive idea about main tree structure

Some effective results



Input f 9944 level lines. Output f' 72 level lines. Randomly colorized f'. Extracted level lines.

Selection based on decreasing order of circularity



Input f 45578 level lines. Output f' 220 level lines. Randomly colorized f'. Extracted level lines.

Selection using decreasing order of mean gradient ∇

Basic idea

Select a subset of meaningful and disjoint level lines from the tree of shapes $\mathcal{T}$ to represent the main tree structure; Two main ideas:

1. $\forall \mathcal{N} \in \mathcal{T}$, find its lowest ancestor shape $\mathcal{N}'$: Smallest Enclosing Shape $\mathrm{SES}(\mathcal{N})$, such that $\mathcal{N} \subseteq \mathcal{N}', \partial\mathcal{N}' \cap \partial\mathcal{N} = \emptyset$.
2. $\forall \mathcal{N} \in \mathcal{T}$ in some *Order*, select $\mathcal{N}$ if it is not deactivated by any descendant, and none of $[\mathcal{N} \rightsquigarrow \mathrm{SES}(\mathcal{N})[$ is selected, then deactivate $[\mathcal{N} \rightsquigarrow \mathrm{SES}(\mathcal{N})[$.

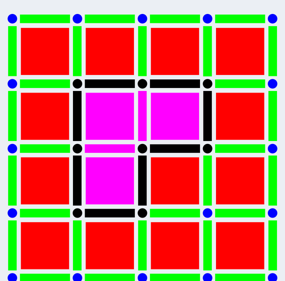
Algorithm overview: three main steps

1. Tree of shapes construction: use the union-find-based algorithm in [4] to compute the set of all level lines.
2. SES computation: bottom-up traversal updating based on the nodes' depth
3. Level lines selection: sequential test based on the status of $[\mathcal{N} \rightsquigarrow \mathrm{SES}(\mathcal{N})[$

Smallest Enclosing Shape (SES) computation

The algorithm in [4] works on Khalimsky grid $\mathcal{K}_\Omega$. A shape is represented by a 2-face; *parent*: inclusion relationship; getCanonical: canonical element.

```
COMPUTE_SES(parent, S, depth)
foreach x in K_Ω do SES(x) ← getCanonical(x)
foreach 2-face x in reverse order of S do
    foreach 0 and 1-face e in x̄ do
        if depth(e) < depth(SES(x)) then
            SES(x) ← getCanonical(e)
    q ← parent(x);
    if depth(SES(x)) < depth(SES(q)) then
        SES(q) ← SES(x)
return SES
```

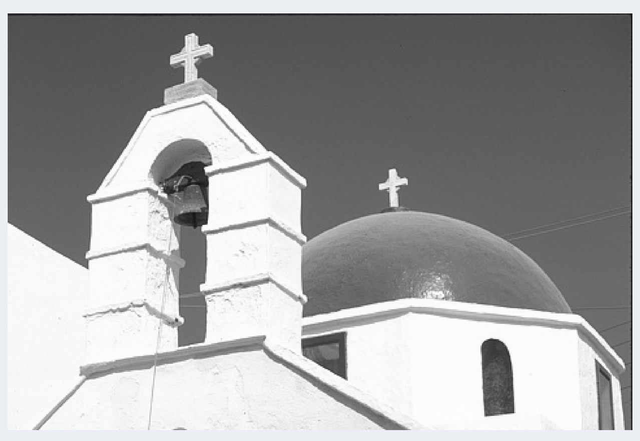


Khalimsky grid: 0-faces (small disks), 1-faces (strips), and 2-faces (squares).

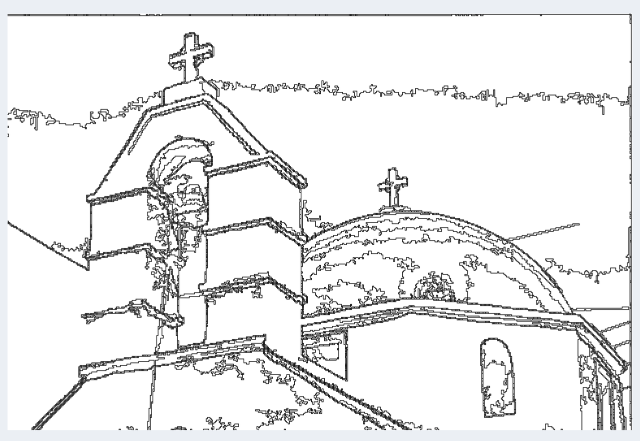
Final disjoint level lines S' selection

```
SELECT_LEVEL_LINES(parent, SES, Order)
foreach x in K_Ω do status(x) ← Null;
S' = ∅;
foreach canonical element x in Order do
    if status(x) ≠ Unactive then
        y ← parent(x);
        while y ≠ SES(x) and status(y) ≠ Active do  y ← parent(y);
        if y = SES(x) then
            status(x) ← Active;
            S' ← S' ∪ {x};
            y ← parent(x);
            while y ≠ SES(x) do  status(y) ← Unactive; y ← parent(y);
return S'
```

Comparison with different selection orders



Input image; 35990 level lines;



Every 15 levels; 99 level lines.

Bottom-up selection; 121 level lines.

Top-down selection; 89 level lines.

Decreasing of mean ∇; 86 level lines.

References

[1] V. Caselles, B. Coll, and J. Morel, "**Topographic maps and local contrast changes in natural images**," *International Journal of Computer Vision*, vol. 33, no. 1, pp. 5–27, 1999.

[2] P. Monasse and F. Guichard, "**Fast computation of a contrast-invariant image representation**," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.

[3] Y. Xu, "**Tree-based shape spaces: Definition and applications in image processing and computer vision**," *PhD Thesis, Univ. Paris-Est, Marne-la-Vallée, France*, Dec 2013.

[4] T. Géraud, E. Carlinet, S. Crozet, and L. Najman, "**A quasi-linear algorithm to compute the tree of shapes of nD images**," in *Proc. of ISMM*, LNCS 7883, pp. 98–110, 2013.

21st International Conference on Image Processing (ICIP) Paris, France -- October 27-30, 2014