A new morphological hierarchical representation of images

Application to text segmentation in document and natural images

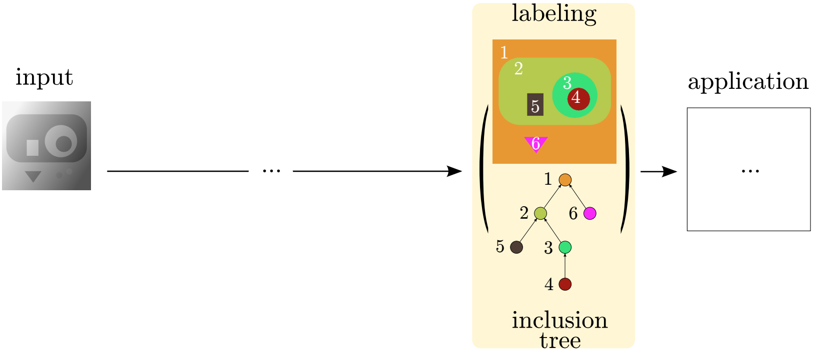
Thierry Géraud   Lê Duy Huynh   Yongchao Xu

EPITA Research and Development Laboratory (LRDE), France

firstname.lastname@lrde.epita.fr

GT GeoDis 2016, Massilia

Objective



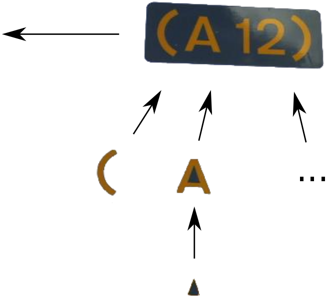
Objective:

to compute a (useful / easy to use) hierarchical representation,

which structures the image contents by inclusion

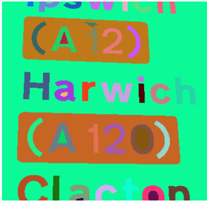
Illustration

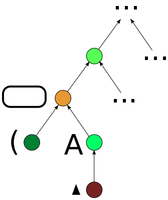
input



object inclusion

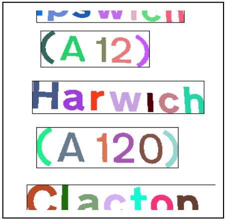
Illustration





input $\rightsquigarrow$ label image $+$ incl. tree

$\Rightarrow$



Application #1: Text detection in natural images

Qualitative evaluation (from *Robust Reading* at ICDAR):

input | labeling (+ incl. tree) | result

Quantitative evaluation:

in A Morphological Hierarchical Representation with Application to Text Segmentation in Natural Images (submitted to ICPR 2016)

Used in the competition *SmartDoc* at ICDAR 2015:


A detention center for illegal immig

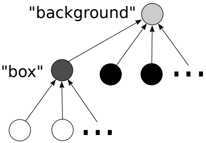
IMMIGRATION

AN ILLEGAL ALIEN PROBLEM—IN MEXICO

IN THE ECONOMIC FOOD CHAIN of the Americas, the U.S. is not the only country with an undocumented-worker crisis.

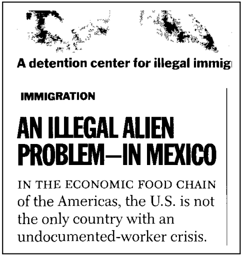
input image

$\rightsquigarrow$



tree (+label image)

$\Rightarrow$


A detention center for illegal immig

IMMIGRATION

AN ILLEGAL ALIEN PROBLEM—IN MEXICO

IN THE ECONOMIC FOOD CHAIN of the Americas, the U.S. is not the only country with an undocumented-worker crisis.

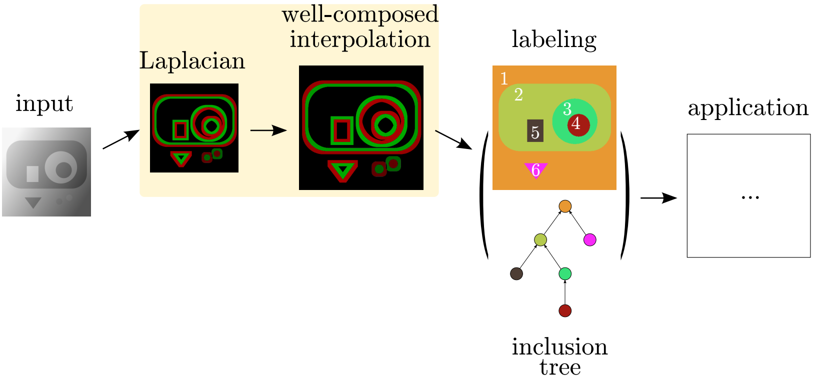
output

So What?

The point is:

it is *not trivial* to get such inclusion structures

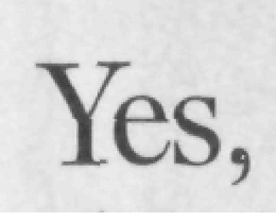
Outline of the Proposed Solution



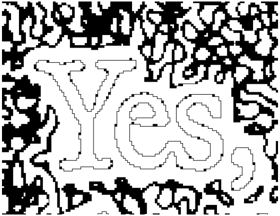
based on a Laplace operator *and* a well-composed interpolation

(so there's some topological considerations...)

Zero-crossings of the Laplace operator $\Delta u = u_{xx} + u_{yy}$:

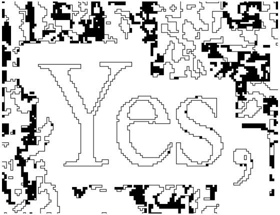


Original image



LoG 17×17

Morphological version →

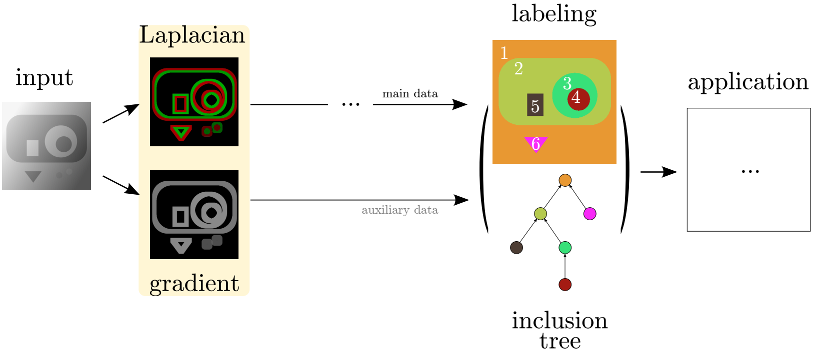


Morphological Laplace Operator

Morphological Laplace Operator: $\Delta_B = (\delta_B - \mathrm{id}) - (id - \varepsilon_B)$

- few sensitive to
  - contrast changes
  - uneven illumination
  - the structuring element B
- zeros stick to data :-) but...
  - we have "spurious" zeros (Pb #1)
  - there is no "trivial" inclusion (Pb #2) :-(

Solving Pb #1



we compute $\nabla_B = \delta_B - \varepsilon_B$ to *ignore* spurious $\Delta_B = 0$

(not detailed here)

Solving Pb #2

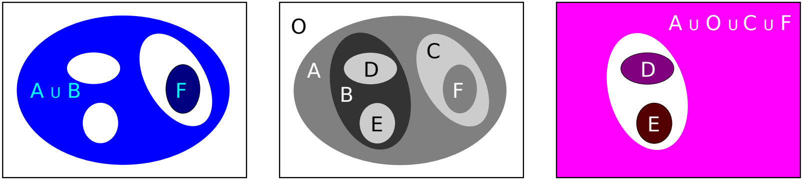
Rationale:

- we are interested in zero-crossings of the Laplace operator
- we want to get a partition of the image (the label image)
- we want the regions to arrange into an inclusion tree
- we do not want to favor a given contrast
- last we want to be fast...

$\rightsquigarrow$ *we want the 0-level lines given by the "tree of shapes" of* $\mathrm{sign}(\Delta_B u)$

Given an image $u : \mathcal{D} \subset \mathbb{R}^2 \to \mathbb{R}$

- lower level sets: $\forall\,\lambda,\ [\,u < \lambda\,] = \{\,x \in \mathcal{D} \mid u(x) < \lambda\,\}$
- upper level sets: $\forall\,\lambda,\ [\,u \geq \lambda\,] = \{\,x \in \mathcal{D} \mid u(x) \geq \lambda\,\}$



a lower level set — u — a upper level set

with the cavity-fill-in operator Sat:

- lower shapes: $\mathcal{S}_{<}(u) = \{\,\mathrm{Sat}(\Gamma);\ \Gamma \in \mathcal{CC}([\,u < \lambda\,])\,\}_{\lambda}$
- upper shapes: $\mathcal{S}_{\geq}(u) = \{\,\mathrm{Sat}(\Gamma);\ \Gamma \in \mathcal{CC}([\,u \geq \lambda\,])\,\}_{\lambda}$

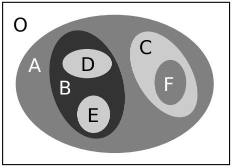
Tree of Shapes and Tree of Lines

We have an inclusion *tree of shapes*:

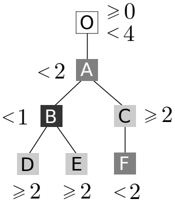
- $\mathfrak{S}(u) = \mathcal{S}_<(u) \cup \mathcal{S}_\geq(u)$,

and we have the contours of shapes:

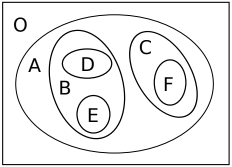
- $\{\, \partial\Gamma;\ \Gamma \in \mathfrak{S}(u) \,\}$ called *level lines*.



an image u



its tree of shapes $\mathfrak{S}(u)$

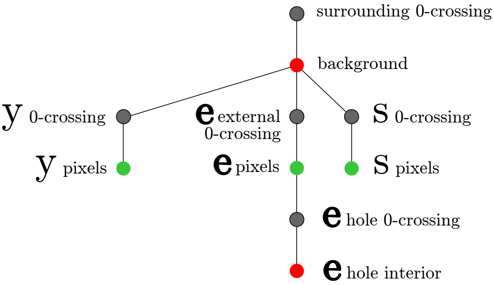


its level lines

The blue pill: so we have an inclusion *tree of level lines*...

Illustration

$\Delta_{\square}(u)$ colorized

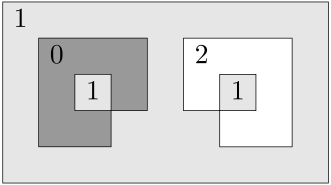


$\mathfrak{S}(\,\mathrm{sign}(\Delta_{\square}(u))\,)$

The red pill: in the discrete settings we face some difficulties...

1	1	1	1	1	1	1	1	1
1	0	0	0	1	2	2	2	1
1	0	1	0	1	2	1	2	1
1	0	0	1	1	1	2	2	1
1	1	1	1	1	1	1	1	1

an image



its level lines

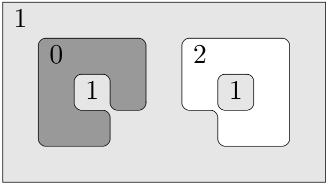
what is the inclusion of shapes, i.e., of level lines?

consider the left inner square at 1, is it inside the 0s?

A solution in the *continuous setting*:

1	1	1	1	1	1	1	1	1
1	0	0	0	1	2	2	2	1
1	0	1	0	1	2	1	2	1
1	0	0	1	1	1	2	2	1
1	1	1	1	1	1	1	1	1

an image



its level lines

considering an *upper semi-continuous* function

level lines are Jordan curves and an inclusion tree exists

A solution in the *discrete setting*:

1	1	1	1	1	1	1	1	1
1	0	0	0	1	2	2	2	1
1	0	1	0	1	2	1	2	1
1	0	0	1	1	1	2	2	1
1	1	1	1	1	1	1	1	1

an image

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	1
1	1	0	0	0	0	0	1	1	2	2	2	2	2	2	2	1
1	1	0	1	1	1	0	1	1	2	2	2	2	2	2	2	1
1	1	0	1	1	1	0	1	1	2	2	2	1	2	2	2	1
1	1	0	1	1	1	1	1	1	2	2	2	2	2	2	2	1
1	1	0	0	0	1	1	1	1	1	1	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

its max-interpolated

It is *as if* we use c8 (resp. c4) for upper (resp. lower) shapes...

A solution in the *discrete setting*:

1	1	1	1	1	1	1	1	1
1	0	0	0	1	2	2	2	1
1	0	1	0	1	2	1	2	1
1	0	0	1	1	1	2	2	1
1	1	1	1	1	1	1	1	1

an image

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	1
1	1	0	0	0	0	0	1	1	2	2	2	2	2	2	2	1
1	1	0	1	1	1	0	1	1	2	2	2	2	2	2	2	1
1	1	0	1	1	1	0	1	1	2	2	2	1	2	2	2	1
1	1	0	1	1	1	1	1	1	2	2	2	2	2	2	2	1
1	1	0	0	0	1	1	1	1	1	1	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

its max-interpolated

we do have Jordan curves and an inclusion tree *but* we are **not self-dual**

we do **not** want to favor a particular contrast $\rightsquigarrow$ solution discarded!

A **self-dual** solution in the digital setting:

1	1	1	1	1	1	1	1	1
1	0	0	0	1	2	2	2	1
1	0	1	0	1	2	1	2	1
1	0	0	1	1	1	2	2	1
1	1	1	1	1	1	1	1	1

an image

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	1	1	1	2	2	2	2	2	1	1
1	1	0	1	1	1	0	1	1	1	2	1	1	1	2	1	1
1	1	0	1	1	1	0	1	1	1	2	1	1	1	2	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	2	1	1
1	1	0	0	0	1	1	1	1	1	1	1	2	2	2	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

a $\mathfrak{F}$-interpolated

interpolation obtained thanks to a front propagation method $\mathfrak{F}$

$\leadsto$ How to make *n*D functions WC in a self-dual way (ISMM 2015)

About the $\mathfrak{F}$-Interpolation

9	11	15
7	1	13
3	5	3

an image u

9	9	11	11	15
8	8	8	8	13
7	7	1	8	13
7	7	5	8	8
3	5	5	5	3

the $\mathfrak{F}$-interpolation of u

This *n*D well-composed interpolation is invariant by:

- contrast changes,
- classical rotations and symmetries,
- inversion (so it is self-dual).

...

About the $\mathfrak{F}$-Interpolation

9	11	15
7	1	13
3	5	3

an image u

9	9	11	11	15
8	8	8	8	13
7	7	1	8	13
7	7	5	8	8
3	5	5	5	3

the $\mathfrak{F}$-interpolation of u

Actually:

- this interpolation is a *digitally well-composed* image,
- it has a "purely self-dual" tree of shapes.
  Links btw the morphological ToS and WC gray-level images (ISMM 2015)
- its level sets are Jordan curves.

Recap

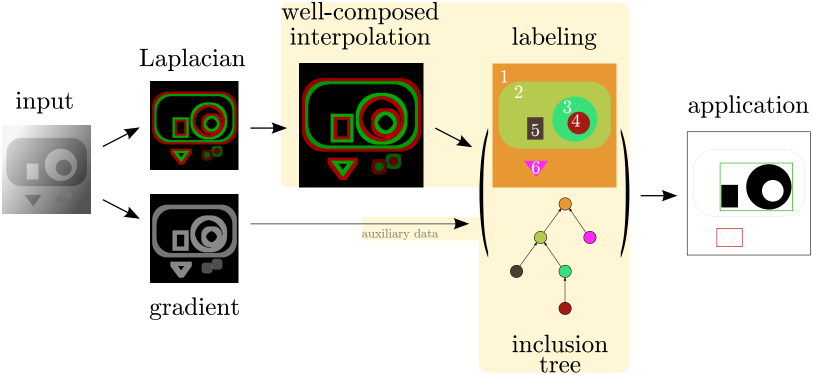
Original image

Morphological Laplacian $\rightsquigarrow$ WC

Hierarchical repr. (lab.+tree)

App. (here node selection and grouping)

Computing the Hierarchical Structure



(Label image + tree) computation.

Hierarchical Structure Computation

```
LABELING(Δ□^wc, ∇□)

for all p do
    label(p) ← 0
    isContour(p) ← false
ℓ ← 0;

for all p do
    if label(p) ≠ 0 then continue
    CONTOURIZE(parent*, ℓ*, isContour*, p, ∇□)          // * means input/output
    label(p) ← ℓ
    Q.push(p)
    while not Q.is_empty() do
        q ← Q.pop()
        for all n ∈ 𝒩(q) do
            if label(n) = 0 and Δ□^wc(p)×Δ□^wc(n) ≥ 0 then
                label(n) ← ℓ
                Q.push(n)
            else
                isContour(n) ← true

return (label, parent)
```

This is a simple front propagation blob labeling (yellowish) algorithm!
with contour elimination and inclusion tree (parent) computation

Hierarchical Structure Computation

Properties:

- linear time complexity,
- parallelizable,
- no $\times 4$ factor: we can *emulate* the $\mathfrak{F}$-interpolation!
- easy to code.

Conclusion

- A hierarchical representation:
  - data simplification,
  - easy to use.
- Emphasis put on inclusion:
  - useful for some applications.
- From theory to practical results:
  - self-duality of cross-sections in digital topology...
  - some effective results.

Bibliography

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

S. Crozet and T. Géraud, "A first parallel algorithm to compute the morphological tree of shapes of *n*D images," in *Proc of ICIP*, pp. 2933–2937, 2014. [PDF]

N. Boutry, T. Géraud, and L. Najman, "On making *n*D images well-composed by a self-dual local interpolation," in *Proc of DGCI*, vol. 8668 of *LNCS*, pp. 320–331, Springer, 2014. [PDF]

T. Géraud, E. Carlinet, S. Crozet, "Self-duality and digital topology: Links between the morphological tree of shapes and well-composed gray-level images," in *Proc. of ISMM*, vol. 9082 of *LNCS*, pp. 573–584, Springer, 2015. [PDF]

N. Boutry, T. Géraud, and L. Najman, "How to make *n*D functions well-composed in a self-dual way," in *Proc. of ISMM*, vol. 9082 of *LNCS*, pp. 561–572, Springer, 2015. [PDF]

L.D. Huynh, Y. Xu, T. Géraud, "A morphological hierarchical representation with application to text segmentation in natural images," *Submitted to ICPR*, 2016. [PDF]