# Morphological Hierarchical Image Decomposition Based on Laplacian 0-Crossings

Lê Duy HUỲNH, Yongchao XU, Thierry GÉRAUD

EPITA Research & Development Laboratory (LRDE)

ISMM, Fontainebleau, May 2017

# Overview

Text detection method presented in ICPR [huynh et al. ICPR 2016]:

input | label image | detected text boxes


The underlying structure is

- a hierarchical representation,
- based on 0-crossings of Laplacian,
- **constructed with quasi-linear time complexity**.

$\longrightarrow$ Computation of Tree of Shapes of Laplacian sign (ToSL)

# Outline

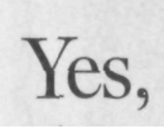

1. Theoretical Background
   - Morphological Laplace Operator
   - Well-Composed Images
   - Tree of Shapes
2. Computation of Tree of Shapes of Laplacian sign
   - ToSL Construction
   - Optimized ToSL Construction
3. Conclusion

# Morphological Laplace Operator

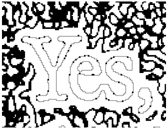
The morphological Laplace operator: $\Delta_{\square} = \delta_{\square} + \varepsilon_{\square} - 2id$

deformation of contour ↓

closed and precise contour ↓

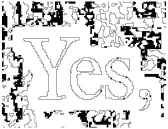



$u$



$LoG_{17}(u) = 0$

$\Delta_{\square}(u)$

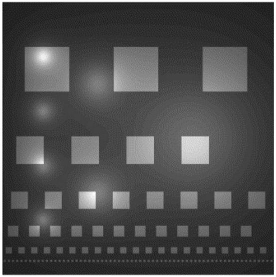


$\Delta_{\square}(u) = 0$

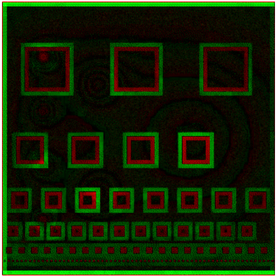
The morphological Laplace operator is **simple**, **self-dual**, and provides **closed contour**.

# Morphological Laplace Operator

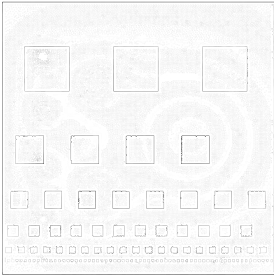
The morphological Laplace operator: $\Delta_\square = \delta_\square + \varepsilon_\square - 2id$



$u$



$\Delta_\square(u)$



$\Delta_\square(u) = 0$

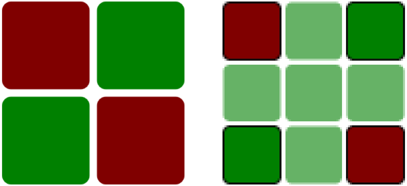
The morphological Laplace operator is **robust to uneven illumination**.

# Well-composed Images

In 2D well-composed images:

- All connectivities are equivalent [ Latecki JMIV 1998].
- There are no “critical configurations”: (  or ).

$\Rightarrow$ No connectivity ambiguity. All contours are Jordan curves



A transformation that removes critical configurations make an image well-composed

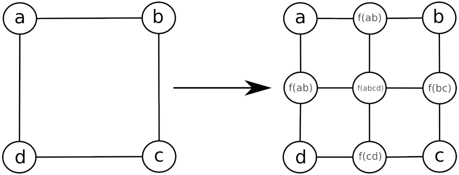
# Well-composed Interpolation

How to obtain a well composed image:

- Modification of the pixel values: modify images topology.
- Interpolation: image has 4 times number of pixels.

Interpolation methods:

- Local interpolation (e.g., by min, max, median operator).
- Non-local interpolation (e.g. [Boutry et al. ISMM 2015].



# Tree of Shapes (ToS)

- A self-dual fusion of min-tree and max-tree

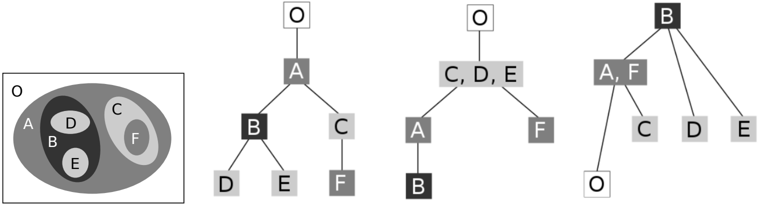



image ToS min-tree max-tree

# Tree of Shapes (ToS)

- An inclusion tree of level lines

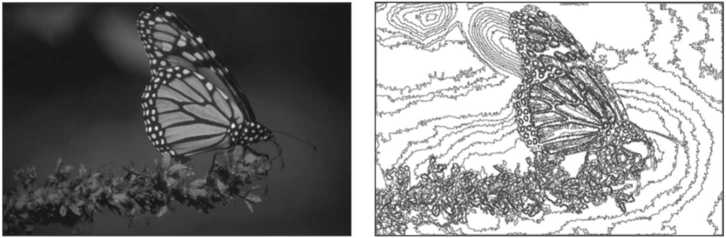
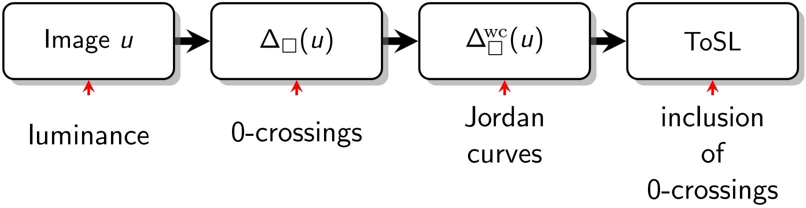
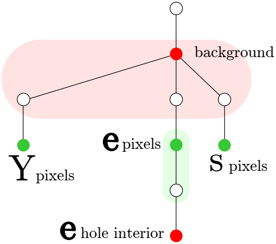


Image and its level lines (every 5 levels)

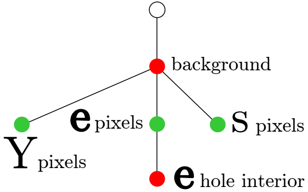
# Tree of Shapes of Laplacian sign (ToSL)




$\Delta^{\mathrm{wc}}_{\square}(u)$.




$\mathfrak{S}(\mathrm{sign}(\Delta^{\mathrm{wc}}_{\square}(u)))$.

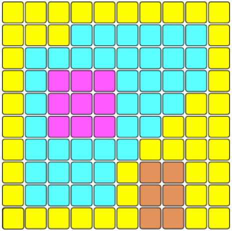


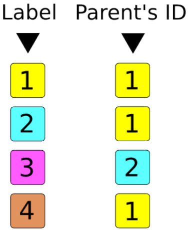

Simplified $\mathfrak{S}$, called ToSL.

■ ■ □: positive, negative, and zeroes of $\Delta$

# ToSL construction

ToSL is represented by a label map and a parent table:



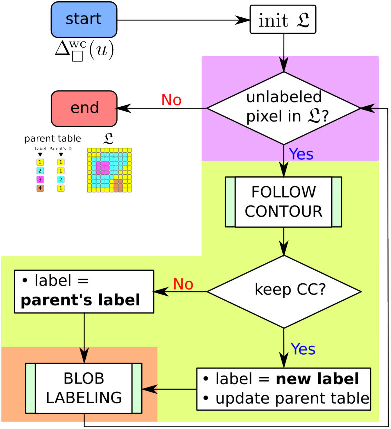



Label map $\mathfrak{L}$ + a parent table

The root node:

- is defined as the sign of median of all points on the contour,
- has itself as its parent.

# ToSL construction - Implementation




Main steps:

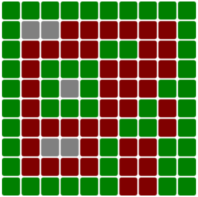
1. Init an empty labeling map $\mathfrak{L}$
2. Scan $\mathfrak{L}$ for unlabeled pixel
3. Get properties $\mathcal{P}$ of unlabeled CC [1], assign a label (new or its parent's label), and update parentArray
4. Label CC. Continue step 2 until a fully labeled map is obtained

[1] it could be contour length, gradient magnitude, height, width...

# An example: construction of ToSL with interpolation

Compute the morphological Laplacian $\Delta_{\square}(u)$

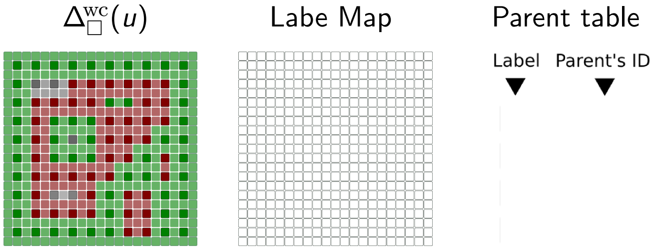
$\Delta_{\square}(u)$



: positive, negative, and zeroes of $\Delta$

# An example: construction of ToSL with interpolation

Compute $\Delta^{\mathrm{wc}}_{\square}(u)$, and create an empty label map $\mathfrak{L}$

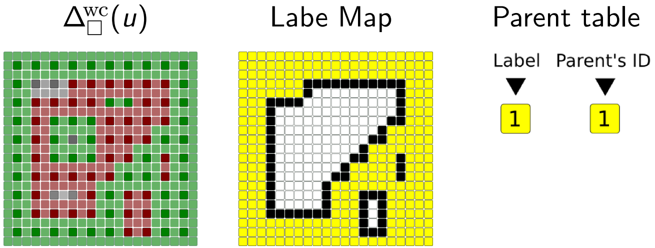



□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# An example: construction of ToSL with interpolation

Label first CC and mark inner border



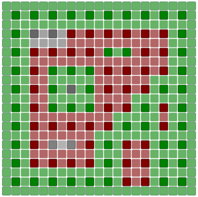

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

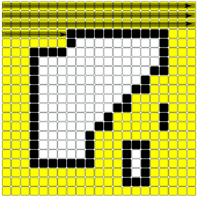
# An example: construction of ToSL with interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region

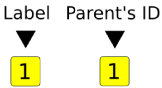
$\Delta^{\mathrm{wc}}_{\square}(u)$



Labe Map



Parent table



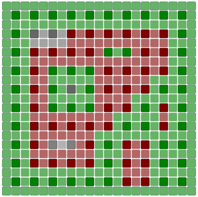

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

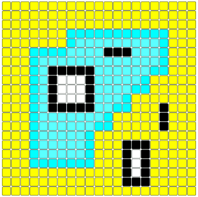
# An example: construction of ToSL with interpolation

Label second CC and mark inner border

$\Delta^{\mathrm{wc}}_{\square}(u)$



Labe Map



Parent table

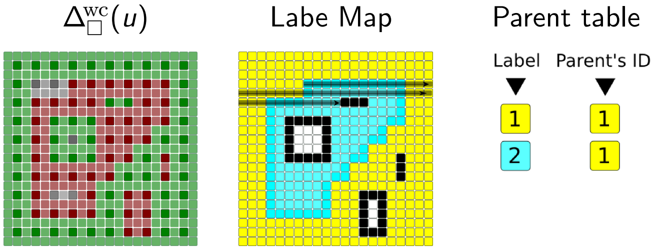
| Label | Parent's ID |
|---|---|
| 1 | 1 |
| 2 | 1 |

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# An example: construction of ToSL with interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region



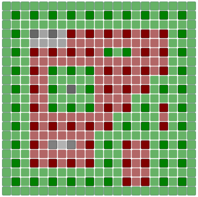

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

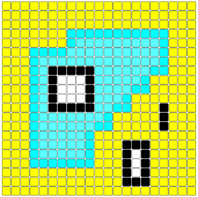
# An example: construction of ToSL with interpolation

Filter out third CC which is small

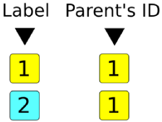
$\Delta^{\text{wc}}_{\square}(u)$



Labe Map



Parent table

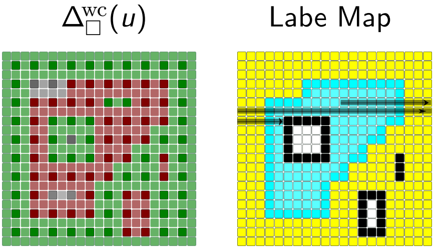



□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# An example: construction of ToSL with interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region




Parent table

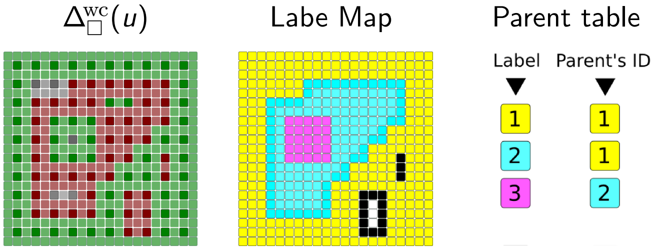
| Label | Parent's ID |
|---|---|
| 1 | 1 |
| 2 | 1 |

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# An example: construction of ToSL with interpolation

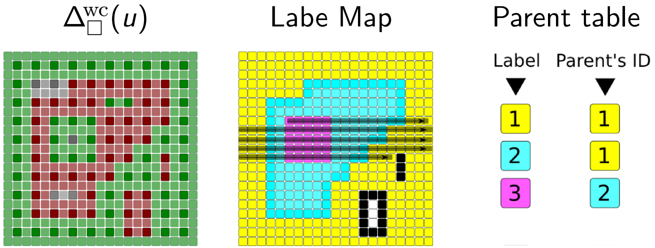
Label forth CC




□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels
■ ■ ■: positive, negative, and zeroes of $\Delta$

# An example: construction of ToSL with interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region



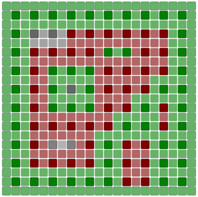

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

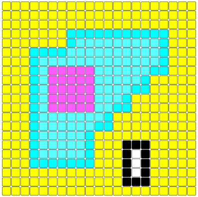
# An example: construction of ToSL with interpolation

Filter out fifth CC which is small

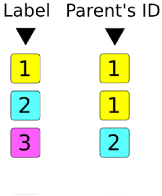
$\Delta^{\text{wc}}_{\square}(u)$



Labe Map



Parent table

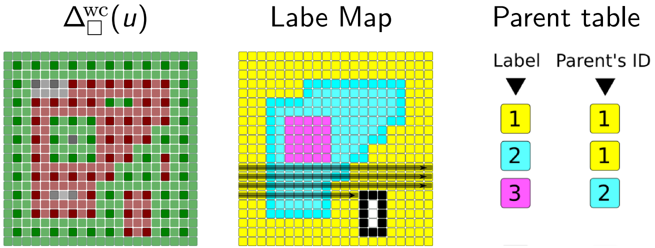



□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# An example: construction of ToSL with interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region

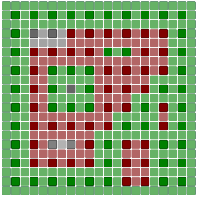



□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels
■ ■ ■: positive, negative, and zeroes of $\Delta$

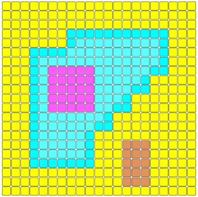
# An example: construction of ToSL with interpolation

Label sixth CC

$\Delta^{\mathrm{wc}}_{\square}(u)$



Labe Map



Parent table

| Label | Parent's ID |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 1 |

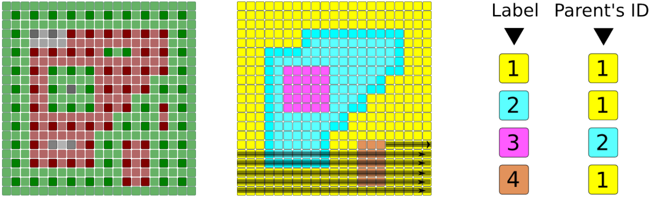
□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# An example: construction of ToSL with interpolation

Scan $\mathfrak{L}$ for unlabeled pixel, there is no unlabeled pixels

$\Delta_{\square}^{\mathrm{wc}}(u)$ Labe Map Parent table



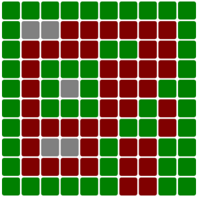

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

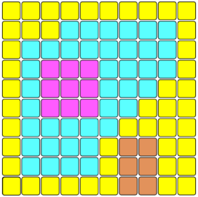
# An example: construction of ToSL with interpolation

Go back to original space

$\Delta_{\square}(u)$



Labe Map



Parent table

| Label | Parent's ID |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 1 |

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# A particular Well-composed Non-local interpolation

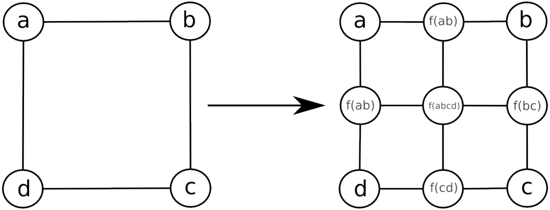
Our non-local interpolation:

- Only take into account the sign of $\Delta_{\square}(u)$,
- New pixels are calculated by:

$$f(a_1, a_2..., a_n) = \begin{cases} sign(a_1) & sign(a_1) = sign(a_2)... = sign(a_n) \\ \chi & \exists a_h, a_k; sign(a_h) \neq sign(a_k) \end{cases}$$

where $\chi$ is the sign of outside CC

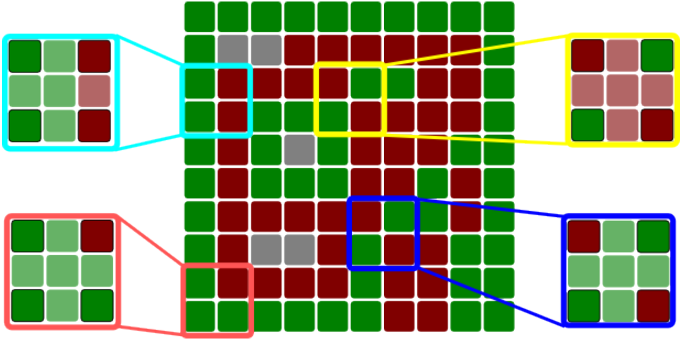
$\longrightarrow$ $f$ is self-dual, symmetrical, and in-between.



# A particular Well-composed Non-local interpolation

This interpolation:

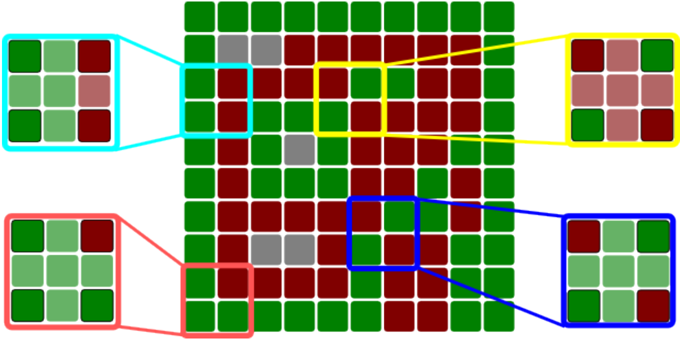
- Follows the same scheme as our ToSL construction,
- Its topological behavior is deterministic,
- It could be easily emulated.



# A particular Well-composed Non-local interpolation

Output of this non-local interpolation on original map:

- At critical configurations (▞ or ▚):
  - Two pixels labeled first are connected ($\mathbb{N}_8$),
  - Others two are separated ($\mathbb{N}_4$).
- At other configurations, $\mathbb{N}_8$ or $\mathbb{N}_4$ are equivalent.



*$\mathbb{N}_x$: x-connected neighborhood

# Modification

Some small modification is needed (red) so that we use:

- $\mathbb{N}_8$ for normal pixels,
- $\mathbb{N}_4$ for pixels marked as border.

$\longrightarrow$ No need to process 4 times the number of image pixels.

```
LABELING (Δ□,∇□)
forall p do
    𝔏(p) ← 0;
    border(p) ← undef;
nlabels ← 1;
forall p do
    if 𝔏(p) ≠ 0 then
        continue;
    if p = p0 then
        ℓ ← 1;
        parent[ℓ] = 1;
    else
        𝒫 ←
        FOLLOW_CONTOUR(p);
        if evaluate(𝒫) then
            nlabels ← nlabels + 1;
            ℓ ← nlabels;
            parent[ℓ] ← 𝔏(p−1);
        else
            ℓ ← 𝔏(p−1);
    BLOB_LABELING (p,ℓ);
return parent, 𝔏;
```

```
BLOB_LABELING (p,ℓ)
𝔏(p) ← ℓ; Q.push(p);
while Q is not empty do
    q ← Q.pop(), ℕ ← ℕ4;
    if border(q) = undef then
        ℕ ← ℕ8; /*optimized*/
    forall n ∈ ℕ(q) do
        if 𝔏(n) = 0 and
        Δ□(p) × Δ□(n) ≥ 0 then
            𝔏(n) ← ℓ; Q.push(n);
        else
            border(n) ← a;
```

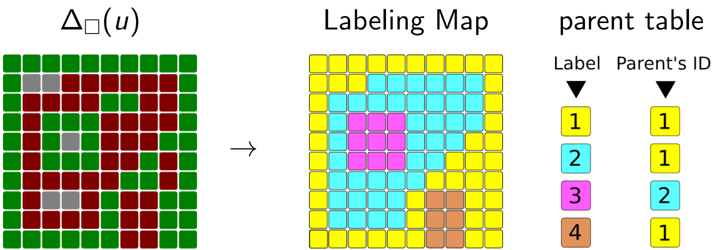
```
FOLLOW_CONTOUR (p)
𝒫.init(); border(p) ← ā; Q.push(p) ;
while Q is not empty do
    q ← Q.pop(); 𝒫.update(q);
    forall n ∈ ℕ4 (q) do
        if border(n) = a then
            Q.push(n); border(n) ← ā;
return 𝒫;
```

*$\mathbb{N}_x$: x-connected neighborhood

# Optimized ToSL Construction

This optimization allows us to:

- directly compute ToSL from $\Delta_{\square}(u)$,
- obtain the same topology as by using interpolation method.




: different labels

: positive, negative, and zeroes of $\Delta$

# Conclusion

We have presented a morphological hierarchical image decomposition based on morphological Laplacian that:

- **is computed with linear time complexity**,
- allows objects extraction with precis contour,
- performs well in presence of uneven illumination.


input label image detected text boxes

Some result of text detection method using ToSL [huynh et al. ICPR 2016]

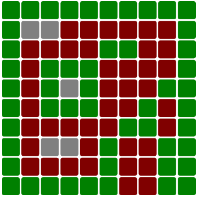
# Questions and Answers

Thanks for your attention!

# A backup slide: Construction of ToSL without interpolation

From the Morphological Laplacian $\Delta_{\square}(u)$

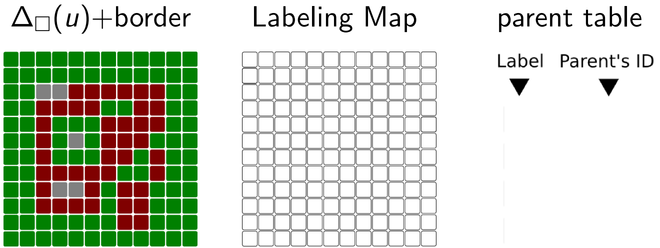
$\Delta_{\square}(u)$



■ ■ ■: positive, negative, and zeroes of $\Delta$

# A backup slide: Construction of ToSL without interpolation

Compute $\Delta^{\text{wc}}_{\square}(u)$ create an empty labeling map $\mathfrak{L}$

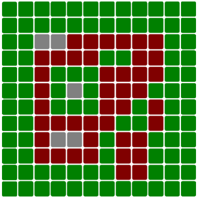



□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

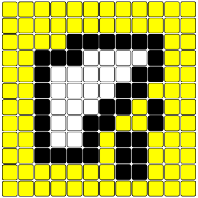
■ ■ ■: positive, negative, and zeroes of $\Delta$

Label first CC and mark inner border

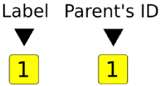
$\Delta_{\square}(u)$+border



Labeling Map



parent table



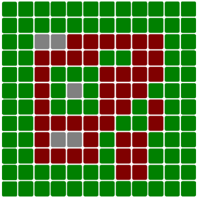

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

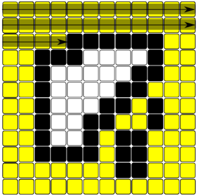
# A backup slide: Construction of ToSL without interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region

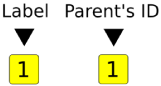
$\Delta_{\square}(u)$+border



Labeling Map



parent table



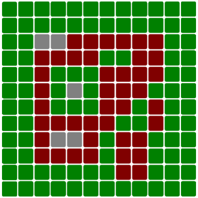

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

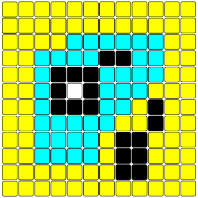
# A backup slide: Construction of ToSL without interpolation

Label second CC and mark inner border

$\Delta_{\square}(u)$+border



Labeling Map



parent table

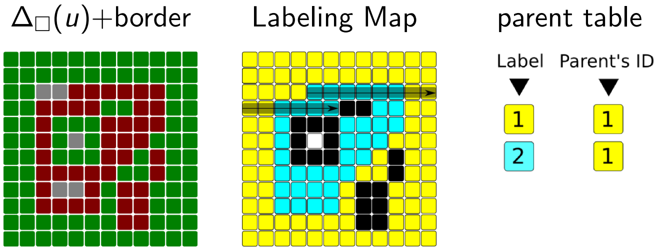
| Label | Parent's ID |
|---|---|
| 1 | 1 |
| 2 | 1 |

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of Δ

# A backup slide: Construction of ToSL without interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region



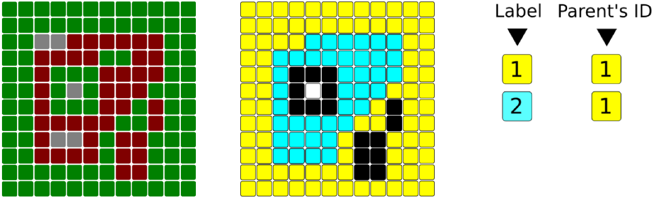

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# A backup slide: Construction of ToSL without interpolation

Filter out third CC which is small

$\Delta_{\square}(u)$+border      Labeling Map      parent table

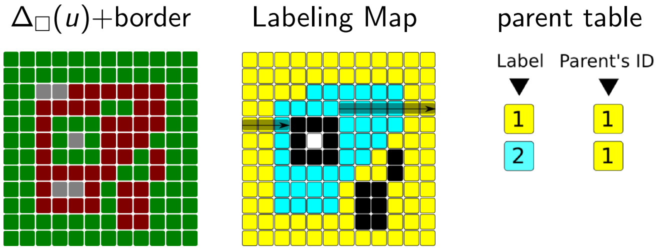



□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# A backup slide: Construction of ToSL without interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region

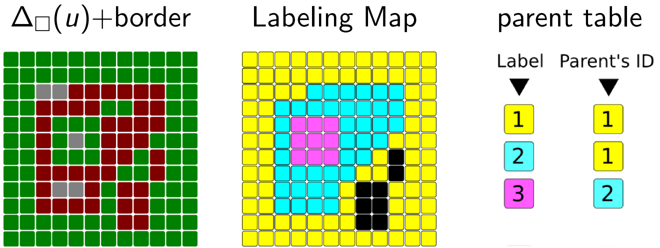



□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# A backup slide: Construction of ToSL without interpolation

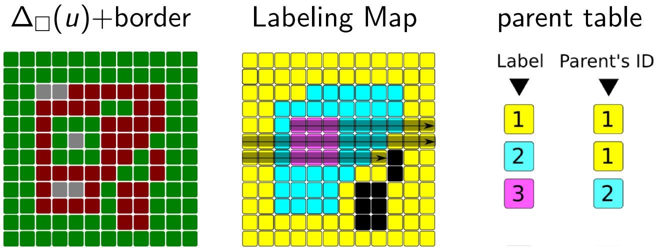
Label forth CC




□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region



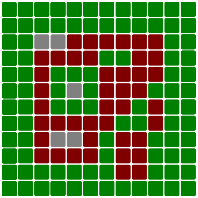

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

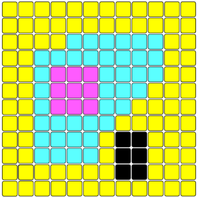
# A backup slide: Construction of ToSL without interpolation

Filter out fifth CC which is small

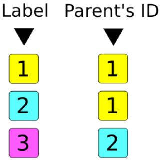
$\Delta_{\square}(u)$+border



Labeling Map



parent table



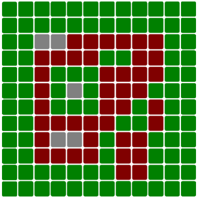

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

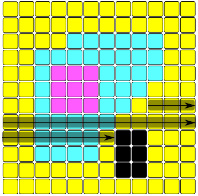
# A backup slide: Construction of ToSL without interpolation

Scan $\mathfrak{L}$ for unlabeled pixel. Check properties of new region

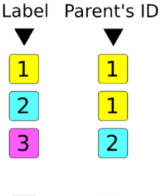
$\Delta_{\square}(u)$+border



Labeling Map



parent table



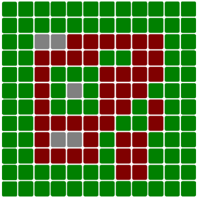

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

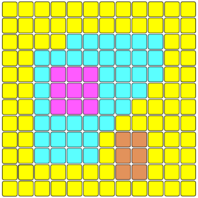
# A backup slide: Construction of ToSL without interpolation

Label sixth CC

$\Delta_{\square}(u)$+border



Labeling Map



parent table

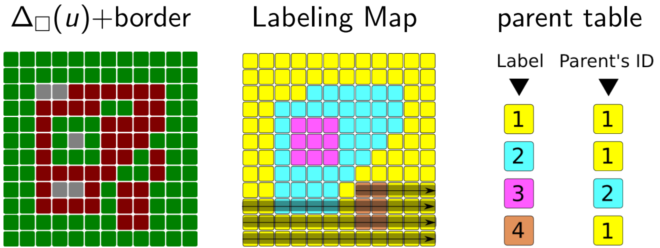
| Label | Parent's ID |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 1 |

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

# A backup slide: Construction of ToSL without interpolation

Scan $\mathfrak{L}$ for unlabeled pixel, there is no unlabeled pixels



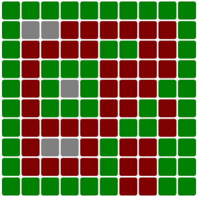

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$

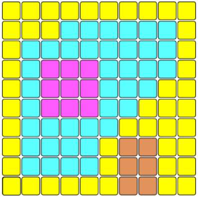
# A backup slide: Construction of ToSL without interpolation

Return to original space

$\Delta_{\square}(u)$



Labeling Map



parent table

| Label | Parent's ID |
| --- | --- |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 1 |

□:unlabeled, ■: unlabeled marked border, ■ ■ ■ ■: different labels

■ ■ ■: positive, negative, and zeroes of $\Delta$