Planting, Growing, and Pruning Trees: Connected Filters Applied to Document Image Analysis

Guillaume Lazzara, Thierry Géraud, Roland Levillain

EPITA Research and Development Laboratory (LRDE)

April 8, 2014



Reviewer 1:

I wasn't overly impressed with this paper until I saw Figure 9.

イロト イポト イヨト イヨト

Reviewer 1:

I wasn't overly impressed with this paper until I saw Figure 9.

イロト イポト イヨト イヨ

Reviewer 1:

I wasn't overly impressed with this paper until I saw Figure 9.

イロト イポト イヨト イヨ



We know that mathematical morphology can often look impressive...

...yet, today you just need to understand \leq and \subset ...

G. Lazzara, T. Géraud, R. Levillain (EPITA) Connected Filters Applied to Document Image Analysis 2014-04-08 3

ヘロト ヘ戸ト ヘヨト ヘヨ

Hum...

We know that *mathematical morphology* can often look impressive...

C. Links between $(\Theta, \overleftarrow{\bullet})$ and $(\Theta, \overleftarrow{\bullet})$

The nodes of $(\dot{\Theta}, \dot{\blacktriangleleft})$ which are preserved in $(\ddot{\Theta}, \ddot{\blacktriangleleft})$ are the sup/max-generators of I, i.e., the valued connected components $K \in \Theta$ which contribute effectively to the (re)construction of I via their associated cylinder function CK (see Formulae (13) and (17)). This property can however be expressed without directly considering the relations between I and the cylinder functions induced by Θ .

Property 6: Let $K = (X, v) \in \Theta$. We have

$$(K \in \ddot{\Theta}) \Leftrightarrow ((K \in \bigwedge^{\leq} \dot{\Theta}) \lor (K \neq \bigsqcup \bigwedge^{\leq} K^{\downarrow}))$$
 (28)

Proof: First note that (Ω, \bot) satisfies Formula (28). Let us now suppose that $K \neq (\Omega, \bot)$. If $K = (X, v) \in \Lambda^{\leq} \dot{\Theta}$, then for all $x \in X$, we have I(x) = v. If $K \neq | | \bigwedge^{\leq} K^{\downarrow}$, $K = | | \wedge^{\leq} K^{\downarrow}$, then for each $x \in X$, there exists $K' \in$ and the result follows.

(29) (Θ, \triangleleft) is a upper-piecewise lattice.

Proof: Let $K = (X, v) \in \Lambda^{\leq} \Theta$. It derives from Property 2 that (K^{\uparrow}, \leq) is a lattice. Since for any $x \in X$ where (X, \triangleleft) is a lattice, $(x^{\uparrow}, \triangleleft)$ is still a lattice, (Θ, \triangleleft) is a upper-piecewise lattice.

As a corollary, we have the following property, related to the structure of the equivalence classes of \sim_{θ} .

Property 8: Let (V, \leq) be a lower-piecewise lattice. Let $K \in \Theta$, then

> $([K]_{\sim a}, \trianglelefteq)$ is a lower-semilattice. (30)

> > < □ > < 同 > < 回 > < 回

Proof: Let K = (X, v). Let $K' = (Y, u) \in \Lambda^{\leq} \Theta$ such that $Y \subseteq X$. From Property 7, $(K'^{\uparrow}, \triangleleft)$ is a lattice. Moreover we have $[K]_{\sim_{\theta}} \subseteq K'^{\dagger}$. As (V, \leq) is a lower-piecewise lattice, (u^{\downarrow}, \leq) is a lattice. Let $(X, v_1), (X, v_2) \in [K]_{\sim a}$. We then, there exists $x \in X$ such that I(x) = v. The fact that have $X \subseteq \lambda_{v_1 \vee \leq v_2}(I)$, and then, from Property (P3), $X \in I$ $K \in \Theta$ then derives from Formula (13). If $K \notin \Delta^{\trianglelefteq} \Theta$ and $C[\lambda_{v_1 \vee \leqslant v_2}(I)]$. Consequently, we have $(X, v_1 \vee \leqslant v_2) \in [K]_{\sim_{\theta}}$,

Hum...

We know that *mathematical morphology* can often look impressive...

C. Links between $(\Theta, \overleftarrow{\bullet})$ and $(\Theta, \overleftarrow{\bullet})$

The nodes of $(\dot{\Theta}, \dot{\blacktriangleleft})$ which are preserved in $(\ddot{\Theta}, \ddot{\blacktriangleleft})$ are the sup/max-generators of I, i.e., the valued connected components $K \in \Theta$ which contribute effectively to the (re)construction of I via their associated cylinder function CK (see Formulae (13) and (17)). This property can however be expressed without directly considering the relations between I and the cylinder functions induced by Θ .

Property 6: Let $K = (X, v) \in \Theta$. We have

$$(K \in \ddot{\Theta}) \Leftrightarrow ((K \in \bigwedge^{\leq} \dot{\Theta}) \lor (K \neq \bigsqcup \bigwedge^{\leq} K^{\downarrow}))$$
 (28)

Proof: First note that (Ω, \bot) satisfies Formula (28). Let us now suppose that $K \neq (\Omega, \bot)$. If $K = (X, v) \in \Lambda^{\leq} \dot{\Theta}$, then for all $x \in X$, we have I(x) = v. If $K \neq | | \Lambda^{\leq} K^{\downarrow}$, $K = | | \wedge^{\leq} K^{\downarrow}$, then for each $x \in X$, there exists $K' \in$ and the result follows.

(29) (Θ, \triangleleft) is a upper-piecewise lattice.

Proof: Let $K = (X, v) \in \Lambda^{\leq} \Theta$. It derives from Property 2 that (K^{\uparrow}, \leq) is a lattice. Since for any $x \in X$ where (X, \triangleleft) is a lattice, $(x^{\uparrow}, \triangleleft)$ is still a lattice, (Θ, \triangleleft) is a upper-piecewise lattice.

As a corollary, we have the following property, related to the structure of the equivalence classes of \sim_{θ} .

Property 8: Let (V, \leq) be a lower-piecewise lattice. Let $K \in \Theta$, then

> $([K], ..., \triangleleft)$ is a lower-semilattice. (30)

> > • • • • • • • • • • • • •

3

Proof: Let K = (X, v). Let $K' = (Y, u) \in \Lambda^{\leq} \Theta$ such that $Y \subseteq X$. From Property 7, $(K'^{\uparrow}, \triangleleft)$ is a lattice. Moreover we have $[K]_{\sim_{\theta}} \subseteq K'^{\dagger}$. As (V, \leq) is a lower-piecewise lattice, (u^{\downarrow}, \leq) is a lattice. Let $(X, v_1), (X, v_2) \in [K]_{\sim u}$. We then, there exists $x \in X$ such that I(x) = v. The fact that have $X \subseteq \lambda_{v_1 \vee \leq v_2}(I)$, and then, from Property (P3), $X \in I$ $K \in \Theta$ then derives from Formula (13). If $K \notin \Delta^{\trianglelefteq} \Theta$ and $C[\lambda_{v_1 \vee \leqslant v_2}(I)]$. Consequently, we have $(X, v_1 \vee \leqslant v_2) \in [K]_{\sim_{\theta}}$,

...yet, today you just need to understand \leq and \subset ...

Fig. 9



Sample uses of connected operators. Top: input images; Bottom: filtered images (results).

・ロト ・回ト ・ヨト ・ヨト

ъ

43643

Evangelization from the Church of Mathematical Morphology :-)

Regarding...

... Mathematical Morphology (MM)

Refresh your vision of MM \rightarrow forget ε and δ !

... Connected Filters

Powerful, simple, and well-suited to DIA.

... Methodology

Advocate gray-level morphological strategies (vs approaches based on "binarize first").

・ 同下 ・ ヨト ・ ヨ

Regarding...

... Mathematical Morphology (MM) Refresh your vision of MM \rightarrow forget ε and δ !

... Connected Filters

Powerful, simple, and well-suited to DIA.

... Methodology

Advocate gray-level morphological strategies (vs approaches based on "binarize first").

.

Regarding...

... Mathematical Morphology (MM)

Refresh your vision of MM \rightarrow forget ε and δ !

... Connected Filters

Powerful, simple, and well-suited to DIA.

... Methodology

Advocate gray-level morphological strategies (vs approaches based on "binarize first").

Regarding...

... Mathematical Morphology (MM)

Refresh your vision of MM \rightarrow forget ε and δ !

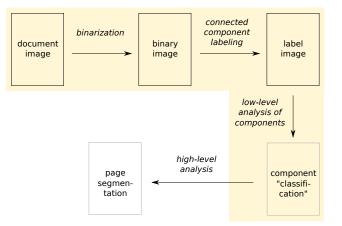
... Connected Filters

Powerful, simple, and well-suited to DIA.

... Methodology

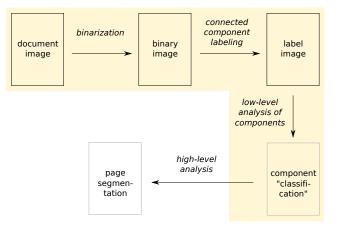
Advocate gray-level morphological strategies (vs approaches based on "binarize first").

Departing From this Typical DIA Workflow



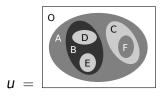
・ロト ・回ト ・ヨト ・ヨト

Departing From this Typical DIA Workflow

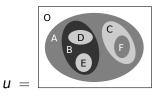


Starting by binarization is hell!

イロト イポト イヨト イヨ

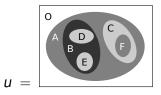


A connected component of the set $[u \le 1] = \{ p \in \mathcal{D}, u(p) \le 1 \}$ is included (so \subset) into a component of the set $[u \le 2]$

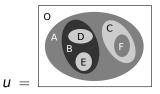


A connected component of the set $[u \le 1] = \{ p \in \mathcal{D}, u(p) \le 1 \}$

is included (so \subset) into a component of the set $[u \leq 2]$

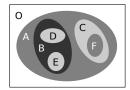


A connected component of the set $[u \le 1] = \{ p \in \mathcal{D}, u(p) \le 1 \}$ is included (so \subset) into a component of the set $[u \le 2]$

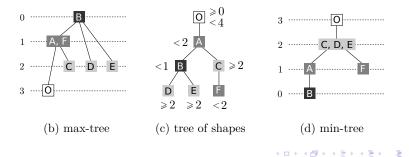


A connected component of the set $[u \le 1] = \{ p \in \mathcal{D}, u(p) \le 1 \}$ is included (so \subset) into a component of the set $[u \le 2]$

Components and Trees



(a) image



An interesting class of filters:

- Not based on structuring elements (so not like ε or δ)
- Considering all the connected components obtained by thresholding the image.
- Don't shift contours; don't create new ones.
- Intuitive, powerful, and efficient.
- Can be implemented as tree filtering.

Min/Max-Tree Implementation [Berger et al., 2007]

```
FIND-ROOT(X)
    if zpar(x) = x then return x
2
        else {zpar(x) \leftarrow FIND-ROOT(zpar(x)) ; return zpar(x) }
COMPUTE-TREE(f)
    for each p, zpar(p) \leftarrow undef
    R \leftarrow \text{REVERSE-SORT}(f) // maps \mathcal{R} into an array
    for each p \in R in direct order
4
        parent(p) \leftarrow p; zpar(p) \leftarrow p
        for each n \in \mathcal{N}(p) such as zpar(n) \neq undef
6
            r \leftarrow \text{FIND-BOOT}(n)
            if r \neq p then { parent(r) \leftarrow p ; zpar(r) \leftarrow p }
8
    DEALLOCATE(zpar)
    return pair(R, parent) // a ''parent'' function
9
CANONIZE-TREE(parent, f)
    for each p \in R in reverse order
1
2
        q \leftarrow parent(p)
3
        if f(parent(q)) = f(q) then parent(p) \leftarrow parent(q)
    return parent // a ''canonized'' parent function
```

tree computation (no code missing!)

イロト イ押ト イヨト イヨト

Min/Max-Tree Implementation [Berger et al., 2007]

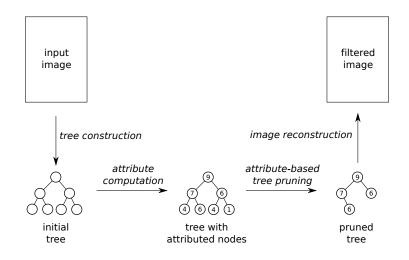
```
FIND-ROOT(X)
    if zpar(x) = x then return x
2
        else {zpar(x) \leftarrow FIND-ROOT(zpar(x)) ; return zpar(x) }
COMPUTE-TREE(f)
    for each p, zpar(p) \leftarrow undef
   R \leftarrow \text{REVERSE-SORT}(f) // maps \mathcal{R} into an array
   for each p \in R in direct order
3
4
        parent(p) \leftarrow p; zpar(p) \leftarrow p
        for each n \in \mathcal{N}(p) such as zpar(n) \neq undef
6
            r \leftarrow \text{FIND-BOOT}(n)
7
            if r \neq p then { parent(r) \leftarrow p ; zpar(r) \leftarrow p }
8
    DEALLOCATE(zpar)
    return pair(R, parent) // a ''parent'' function
9
CANONIZE-TREE(parent, f)
    for each p \in R in reverse order
1
2
        q \leftarrow parent(p)
3
        if f(parent(q)) = f(q) then parent(p) \leftarrow parent(q)
    return parent // a ''canonized'' parent function
```

tree computation (no code missing!)

イロト イ押ト イヨト イヨト

image filtering \rightarrow add about 10 lignes of code...

Connected Operators as Tree Filtering



A B + A B +
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A
 A

→ E > < E</p>

Various trees leading to various operators. Pruning a max-tree Algebraic opening. Pruning a min-tree Algebraic closing. Pruning a tree of shapes Grain filter.

Structural vs Algebraic Openings



Initial image.

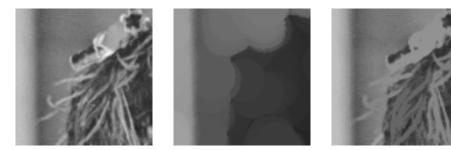


Structural opening with a disk (r = 15).



Algebraic opening $(\lambda = \pi r^2).$

Structural vs Algebraic Openings



Initial image.

Structural opening with a disk (r = 15).

Algebraic opening $(\lambda = \pi r^2).$

Application: Filtering Everything But Boxes





イロト イ押ト イヨト イヨト

Application: Showing Filtered Lines



G. Lazzara, T. Géraud, R. Levillain (EPITA) Con

・ロト ・聞 ト ・ ヨト ・ ヨトー

Application: An Image Featuring Almost Only Text



G. Lazzara, T. Géraud, R. Levillain (EPITA)

2014-04-08 1

Connected Filters: Conclusion

- Benefits
 - Non destructive (preserve contours).
 - Sound and strong mathematical properties [Soille, 2004, Najman and Talbot, 2010].
 - Take into account all components.
 - Really intuitive to use.
 - Very extensible (many attributes).
 - Efficient.
- Applications in Document Image Analysis.
 - Page segmentation.
 - Text identification.
 - Object recognition.
 - "Smart" binarization.
 - Etc.

Implementation: The Olena Platform

Code and tools available in Olena, a free software image processing platform.

```
http://olena.lrde.epita.fr
```

Milena

A generic and efficient C++ image processing library [Levillain et al., 2010].

Scribo

A framework for Document Image Analysis [Lazzara et al., 2011].

イロン イ理 とくほう くほ

Thank You!



イロン イロン イヨン イヨン

æ

Berger, C., Géraud, T., Levillain, R., Widynski, N., Baillard, A., and Bertin, E. (2007). Effective component tree computation with application to pattern recognition in astronomical imaging. In Proceedings of the IEEE International Conference on Image Processing (ICIP), volume 4, pages 41-44. Lazzara, G., Levillain, R., Géraud, T., Jacquelet, Y., Marquegnies, J., and Crépin-Leblond, A. (2011). The SCRIBO module of the Olena platform: a free software framework for document image analysis. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR).

< □ > < 同 > < 回 > < 回

Levillain, R., Géraud, T., and Najman, L. (2010).
 Why and how to design a generic and efficient image processing framework: The case of the Milena library.
 In Proceedings of the IEEE International Conference on Image Processing (ICIP), pages 1941–1944.

Najman, L. and Talbot, H., editors (2010). Mathematical Morphology—From Theory to Applications. ISTE & Wiley.

Soille, P. (2004). Morphological Image Analysis: Principles and Applications. Springer, 2 edition.

イロト イ押ト イヨト イヨト