Image and Vision Computing 28 (2010) 1443–1451

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

A general framework for tree-based morphology and its applications to self-dual filtering

Alla Vichik [a], Renato Keshet [b,*,1], David Malah [a]

[a] Electrical Engineering Department, Technion–Israel Institute of Technology, Haifa 32000, Israel
[b] Hewlett Packard Laboratories – Israel, Technion City, Haifa 32000, Israel

ARTICLE INFO

Article history:
Received 8 July 2008
Received in revised form 13 April 2009
Accepted 25 August 2009

Keywords:
Complete inf-semilattices
Self-dual operators
Tree representation of images

ABSTRACT

This paper presents a tree-based framework for producing self-dual morphological operators, based on a tree-representation complete inf-semilattice (CISL). The idea is to use a self-dual tree transform to map a given image into the above CISL, perform one or more morphological operations there, and map the result back to the image domain using the inverse tree transform. We also present a particular case of this general framework, involving a new tree transform, the Extrema-Watershed Tree (EWT). The operators obtained by using the EWT in the above framework behave like classical morphological operators, but in addition are self-dual. Some application examples are provided: pre-processing for OCR and dust and scratch removal algorithms, and image denoising. We also explore first steps towards obtaining tree transforms that induce a CISL on the image domain as well.

1. Introduction

An operator ψ is called *self-dual* when $\psi(-f) = -\psi(f)$ for any input image f. In contrast to linear signal processing, where practically every operator is self-dual, morphological operators usually come in dual pairs – dilation/erosion, opening/closing, etc. This is a challenge for some applications, like image denoising, where self-duality is an expected property. As a consequence, the design of self-dual morphological operators is an active research field.

One of the main approaches for producing self-dual morphological operators is by using a tree transform, which represents an image by means of a tree. For instance, Salembier and Garrido proposed a *Binary Partition Tree* for hierarchical segmentation in [9,11], and a *tree of shapes* was proposed by Monasse and Guichard [14,15] (see also [16,17]). These tree representations are usually used for performing *connected* filtering operations on an image; however, they do not yield non-connected operators, such as erosions, dilations or openings by a structuring element.

In [3] (see also [4]) a new complete inf-semilattice (CISL), called the shape-tree semilattice, was introduced. This semilattice provides non-connected morphological operations, based on the above-mentioned "tree of shapes." As a consequence, self-dual erosions and openings were obtained. Similar operators had been developed earlier on the so-called Reference Semilattices (introduced in [5], and further studied by Heijmans and Keshet in [7]); however, they require a reference image, which somewhat limits the usage of these operators. The self-dual operators in the shape-tree semilattice provide erosions and openings without the need for a reference image.

In this paper, which is an extension to the conference paper [1], we present a general framework for tree-based morphological image processing, which generalizes the shape-tree operators. This framework yields a set of new morphological operators (erosion, dilation, opening, etc.), for each given tree representation of images. The heart of the proposed approach is a novel complete inf-semilattice of tree representations of images. Because many of the properties of the tree are inherited by the corresponding operators, the choice of the tree representation is of high importance. We focus mostly on self-dual trees, which represent dark and bright elements equally.

A particular case of the proposed framework is also presented, based on a novel tree representation, the Extrema-Watershed Tree (EWT). Following the general framework, we derive self-dual morphological operators from the EWT. Examples of applications discussed here are pre-processing for OCR (Optical Character Recognition) algorithms, de-noising of images, and pre-processing for dust and scratch removal.

2. Theoretical background

2.1. Complete inf-semilattices

A complete inf-semilattice (CISL) is a partially-ordered set $\mathscr{S}$, where the non-empty infimum operation $(\wedge)$ is always well-

* Corresponding author. Tel.: +972 54 5927970.
E-mail addresses: alla.vichik@gmail.com (A. Vichik), renato.keshet@hp.com (R. Keshet), malah@ee.technion.ac.il (D. Malah).

[1] Renato Keshet is also an adjunct lecturer at the EE Dep., Technion–Israel Institute of Technology, Israel.

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.08.008

defined (but the supremum $\vee$ is not necessarily so). The theory of mathematical morphology on complete semilattices was introduced in [6,7], and is an almost-straightforward extension of the traditional morphology on complete lattices. It mathematically supports intuitive observations, such as the fact that erosions (when defined[2] as maps that commute with non-empty infima) are naturally extended from complete lattices to CISLs, whereas dilations are not universally well-defined on CISLs.

On the other hand, some results may not be necessarily intuitive. The main ones are as follows: (a) it is always possible to associate an opening γ to a given erosion ε by means of $\gamma(x) = \bigwedge\{y \mid \varepsilon(y) = \varepsilon(x)\}$, (b) even though the adjoint dilation δ is not universally well-defined, it is always well-defined for elements on the image of $\mathscr{S}$ by ε, and (c) $\gamma = \delta\varepsilon$. The closing $\varepsilon\delta$ is only partially defined.

2.2. Rooted trees and their corresponding CISL

This section reviews basic graph theory notions (given in [19, Chapter 1]), including the natural partial ordering on rooted trees, which provide them with a CISL structure.

A *graph* is a pair of sets $G = (V, E)$ such that $E \subseteq [V]^2$, that is, the elements of E are 2-element subsets of V. A *path* is a non-empty graph $P = (V, E)$ of the form: $V = \{x_0, x_1, \ldots, x_k\}$, $E = \{x_0x_1, x_1x_2, \ldots, x_{k-1}x_k\}$, where the x_i are all distinct. The points x_0 and x_k are the end points of the path. A graph G is *connected* if any two vertices are end points of some path in G. A *connected component* of a graph G is a maximal connected sub-graph of G.

A *cycle* is a graph consisting of a path with end points x_0 and x_k, where $k \geqslant 2$, plus the edge x_0x_k. A graph not containing any cycles is called a *forest*. A connected forest is called a *tree* (thus, a forest is a graph whose connected components are trees). In a tree, any two vertices are joined by a unique path.

Sometimes it is convenient to consider one vertex of a tree as special; such a vertex is then called the *root* of this tree. A tree with a fixed root is a rooted tree. Choosing a root r in a tree t imposes the following partial ordering on $V(t)$: $x \preceq_t y \iff x \in rty$, where rty is the unique path in t that connects y to the root. Note that $(V, \preceq_t)$ is a CISL, where r is the least element, and the maximal elements are the *leafs* of t. The infimum between vertices is the nearest common ancestor vertex.

We say that a tree t_1 is smaller than another tree t_2 if $t_1 \subseteq t_2$.

3. Tree semilattices

This section presents the proposed general framework for tree-based morphological image processing (introduced in [2]). This framework enables the definition of new morphological operators that are based on tree representations. The proposed image processing scheme is shown is Fig. 1.

3.1. CISL of tree representations

The heart of the proposed approach is a novel complete inf-semilattice of tree representations of images.

Let L be an arbitrary set of "labels," and $t = (V, E)$ a rooted tree, with root r, such that $V \subseteq L$. Therefore, t is a tree of labels. Moreover, let $\mathbb{E}$ may be an Euclidean space or a discrete rectangular grid within an image area, and $M : \mathbb{E} \to V$ be an image of vertices, mapping each point in $\mathbb{E}$ to a vertex of t.

Definition 3.1 (*Tree representation*). The structure $T = (t, M)$ shall be called a *tree representation*. The set of all tree representations

[2] In complete lattices, erosions can also be defined by means of adjuctions [8].

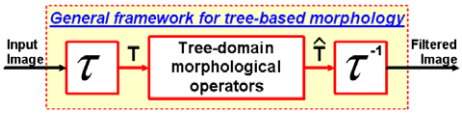

Fig. 1. Proposed tree-based morphology. An image is transformed into a tree representation, morphologically processed in a complete inf-semilattices of tree representations, and then transformed back to the images domain.

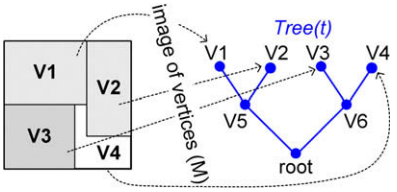

Fig. 2. An example of image tree representation.

associated with the label set L and with the root r shall be denoted by $\mathscr{T}_r^L$.

Fig. 2 depicts an example of a tree representation.

3.1.1. Partial ordering

Consider the following relation between tree representations: for all $T_1 = (t_1, M_1)$ and $T_2 = (t_2, M_2)$ in $\mathscr{T}_r^L$,

$$T_1 \leqslant T_2 \iff \begin{cases} t_1 \subseteq t_2 \quad \text{and} \\ M_1(x) \preceq_{t_2} M_2(x) \quad \forall x \in \mathbb{E}, \end{cases} \tag{1}$$

where $\subseteq$ is the usual graph inclusion and $\preceq_{t_2}$ is the partial ordering of vertices within the tree t_2 (see Section 2.2).

Proposition 3.1. *The above tree relation $\leqslant$ is a partial ordering on $\mathscr{T}_r^L$.*

Proof. It is easy to see that $\leqslant$ is *reflexive*. Now, if $T_1 \leqslant T_2$ and $T_2 \leqslant T_1$ then $t_1 \subseteq t_2$ and $t_2 \subseteq t_1$, therefore $t_1 = t_2 \triangleq t$. Moreover, $M_1(x) \preceq_t M_2(x)$, and $M_2(x) \preceq_t M_1(x)$, $\forall x \in \mathbb{E}$, and therefore $M_1(x) = M_2(x)$, $\forall x \in \mathbb{E}$, which means that $\leqslant$ is *anti-symmetric*. Finally, assume that $T_1 \leqslant T_2$ and $T_2 \leqslant T_3$. Then besides $t_1 \subseteq t_3$, we have $M_1(x) \preceq_{t_2} M_2(x)$, and $M_2(x) \preceq_{t_3} M_3(x)$, $\forall x \in \mathbb{E}$. But since $t_2 \subseteq t_3$, we can also write $M_1(x) \preceq_{t_3} M_2(x)$, $\forall x \in \mathbb{E}$. Therefore, $M_1(x) \preceq_{t_3} M_3(x)$, $\forall x \in \mathbb{E}$, and $\leqslant$ is *transitive*. □

An example of this partial ordering is shown in Fig. 3.

A relevant question at this point is what are the infimum and supremum operators related to the above partial ordering and whether they are well-defined for any subset of tree representations. In other words, is $(\mathscr{T}_r^L, \leqslant)$ a lattice or a semilattice, and if so, complete or not?

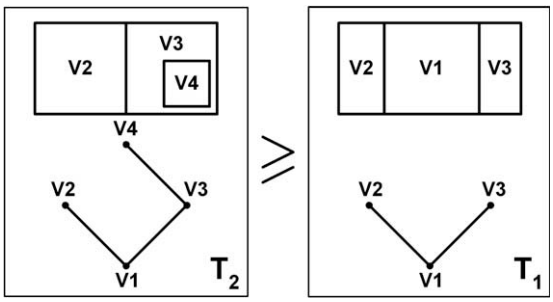

Fig. 3. An example of order of trees. The tree representation T_2 is larger than T_1 because the tree t_1 is included in tree t_2, and all pixels in T_1 are mapped to labels that are smaller or equal to those in T_2.

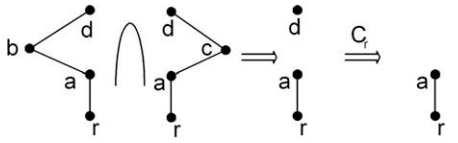

Fig. 4. An example of a tree intersection that does not yield a tree.

3.1.2. Infimum

First notice that, for two graphs g_1 and g_2, the intersection $g_1 \cap g_2$ is the infimum graph. However, if g_1 and g_2 are trees, then unfortunately $g_1 \cap g_2$ is *not* necessarily a tree. This is because the resulting graph may be disconnected, in which case $g_1 \cap g_2$ is a forest, but not a tree. For instance, consider the two trees $t_1 = (\{r,a,b,d\}, \{ra, ab, bd\})$ and $t_2 = (\{r,a,c,d\}, \{ra, ac, cd\})$ as shown in Fig. 4. The intersection is given by $t_1 \cap t_2 = (\{r,a,d\}, \{ra\})$, which has two connected component: $(\{r,a\}, \{ra\})$ and $(\{d\}, \{\})$. Notice, however, that the connected component that contains the root r is the largest tree with root r that is included both in t_1 and in t_2. That is, the subtree containing r is the infimum between the two trees regarding the inclusion order.

This is true in general. That is, if we define $C_r(\cdot)$ to be the operator that extracts from a given graph (and thus from a given rooted tree in particular) its connected component containing the root r, then $C_r(t_1 \cap t_2)$ is the infimum $t_1 \wedge t_2$ between the trees t_1 and t_2.

Let us turn now to the infimum of images of vertices. The element $M_1(x) \curlywedge_{t_1 \wedge t_2} M_2(x)$ would be the natural candidate for the infimum of the two vertices $M_1(x)$ and $M_2(x)$. However, it may occur that either $M_1(x)$ or $M_2(x)$ (or both) do not belong to its set of vertices $V(t_1 \wedge t_2)$. For instance, consider the trees t_1 and t_2 as in the example above, and assume that $M_1(x) = b$ and $M_2(x) = c$ as shown in Fig. 5. Neither $M_1(x)$ nor $M_2(x)$ belongs to $V(t_1 \wedge t_2) = \{r,a\}$.

The infimum is obtained by first "projecting" each vertex $M_1(x)$ and $M_2(x)$ to $V(t_1 \wedge t_2)$, and then taking the infimum of the projected vertices on the tree $t_1 \wedge t_2$. By "projection" of a vertex v onto a subtree $t' \subseteq t$ we mean finding the vertex w in $V(t')$ that is the closest to v in the path connecting v to r inside it t. For instance, in the above example, the projection of both $M_1(x) = b$ and $M_2(x) = c$ onto $t_1 \wedge t_2$ is a. Recalling that vtr represents the path linking v to r in the tree t, the projection $P_{t \to t'}(v)$ of v onto the subtree t' could be also defined as the leaf of $vtr \cap t'$.

In summary, we get the following proposition.

Proposition 3.2. *Consider the non-empty collection of tree representations $\{T_i\}_i = \{(t_i, M_i)\}_i$. Let $\hat{t} = C_r(\bigcap_i t_i)$, where $C_r(\cdot)$ stands for the connected graph containing r. Also, let $\hat{M} = (\curlywedge_{\hat{t}})_i P_{t_i \to \hat{t}}(M_i)$, where $P_{t \to \hat{t}}(\cdot)$ means the projection onto $\hat{t}$, i.e., the leaf of $(\cdot)tr \cap \hat{t}$. Then the tree representation infimum is given by*

$$\bigwedge_i T_i = (\hat{t}, \hat{M}). \tag{2}$$

Proof. In general, an element $\hat{X}$ is the infimum of $\{X_i\}$ iff (1) $\hat{X} \leqslant X_i,\ \forall i$ and (2) any X^* that is smaller or equal to any X_i is also smaller or equal to $\hat{X}$. Let us then first show that $(\hat{t}, \hat{M})$ is smaller or equal to all T_i. First $\hat{t} = C_r(\bigcap_i t_i) \subseteq (\bigcap_i t_i) \subseteq t_i,\ \forall i$. Also, $\hat{M} = (\curlywedge_{\hat{t}})_i P_{t_i \to \hat{t}}(M_i) \preceq_{\hat{t}} P_{t_i \to \hat{t}}(M_i) \preceq_{\hat{t}} M_i,\ \forall i$. And since, for all i, $\hat{t} \subseteq t_i$, then $\hat{M} \preceq_{t_i} M_i$. Therefore, $(\hat{t}, \hat{M}) \leqslant T_i,\ \forall i$.

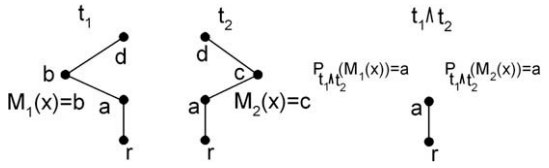

Fig. 5. An example of the projection of vertices $M_1(x)$ and $M_2(x)$ onto the subtree $t_1 \wedge t_2$.

Now assume that $T^* = (t^*, M^*)$ is smaller or equal to all T_i. Then $t^* \subseteq (\bigcap_i t_i)$. But t^* is a tree with root r, so it is also smaller than the largest tree with root r that is included in $\bigcap_i t_i$. Which is the connected component of $\bigcap_i t_i$ that contains r, i.e., $C_r(\bigcap_i t_i)$. So, $t^* \subseteq C_r(\bigcap_i t_i) = \hat{t}$. Moreover, for all x, $M^*(x)$ is a vertex in t^*; but $t^* \subseteq \hat{t}$, so $M^*(x)$ is a vertex of $\hat{t}$. Let v be the largest vertex in the path $M^*(x) t_1 M_1(x)$ that belongs to $\hat{t}$. Since $M^*(x) \preceq_{t_1} M_1(x)$, then v is also the largest vertex in the path $r t_1 M_1(x)$ that belongs to $\hat{t}$. Because $\hat{t} \subseteq t_1$, we have that $r t_1 M_1(x) \cap \hat{t}$ is the path $r\hat{t}v$, so v is the leaf of $r t_1 M_1(x) \cap \hat{t}$, which is given by $P_{t_1 \to \hat{t}}[M_1(x)]$ by definition. So, $M^*(x) \preceq_{\hat{t}} P_{t_1 \to \hat{t}}[M_1(x)]$, and similarly $M^*(x) \preceq_{\hat{t}} P_{t_i \to \hat{t}}[M_i(x)],\ \forall i$. Therefore $M^*(x) \preceq_{\hat{t}} (\curlywedge_{\hat{t}})_i P_{t_i \to \hat{t}}[M_i(x)] = \hat{M}(x)$. The conclusion is that $T^* \leqslant (\hat{t}, \hat{M})$. □

3.1.3. Supremum

Now, let us find out what the supremum of the tree representations is. First, notice that the union $t_1 \cup t_2$ of two trees t_1 and t_2 is always connected, but one cannot assure that it does not contain loops. Therefore, the union is not necessarily a tree. Worse, there does not necessarily exist a smallest tree that is larger than the union. For instance, if $t_1 = (\{r,a,c\}, \{ra, ac\})$ and $t_2 = (\{r,b,c\}, \{rb, bc\})$, then $t_1 \cup t_2$ is equal to the graph $(\{r,a,b,c\}, \{ra, ac, rb, bc\})$, which is not a tree, and there does not exist a tree that contains it. Therefore, there does not exist a supremum of tree representations if the union of their trees is not a tree. Now, suppose that the union is a tree; let us focus on the image of vertices. Now both $M_1(x)$ and $M_2(x)$ do belong to $V(t_1 \cup t_2)$, but their supremum in $t_1 \cup t_2$ may not exist. For instance, if $t_1 = (\{r,a\}, \{ra\})$ and $t_2 = (\{r,b\}, \{rb\})$, then $t_1 \cup t_2$ is a tree; however, if $M_1(x) = a$ and $M_2(x) = b$ for some x, then $[M_1 \curlyvee_t M_2](x)$ does not exist.

In summary, we obtain the following.

Proposition 3.3. *Let $\{T_i\} = \{(t_i, M_i)\}$ be a collection of tree representations. If $\bar{t} \triangleq \bigcup_i t_i$ is a tree, and $(\curlyvee_{\bar{t}})_i M_i$ exists, then the supremum $\bigvee_i T_i$ is given by $(\bar{t}, (\curlyvee_{\bar{t}})_i M_i)$. Otherwise, the supremum does not exist.*

Proof. First, assume that $\bar{t} \triangleq \bigcup_i t_i$ is a tree and that $(\curlyvee_{\bar{t}})_i M_i$ exists. Then, $(\bar{t}, (\curlyvee_{\bar{t}})_i M_i)$ is the supremum, since $\bar{t}$ is the smallest tree with root r that contain all t_i, and $(\curlyvee_{\bar{t}})_i M_i$ is the smallest image of vertices that is larger than all M_i. Now, if $\bar{t} \triangleq \bigcup_i t_i$ is not a tree, then there is not a tree containing all t_i, and therefore the supremum does not exist. And if $(\curlyvee_{\bar{t}})_i M_i$ does not exist, then there is not a image of vertices that is the smallest majorant of $\{M_i\}$. □

In conclusion, $(\mathscr{T}_r^L, \leqslant)$ is a CISL. The least element is $T_0 \triangleq (t_0, M_0)$, where $t_0 \triangleq (\{r\}, \{\})$, and $M_0(x) \equiv r, \forall x \in \mathbb{E}$.

3.2. Fixed tree

Let us now focus on the particular case where all tree presentations involved in an infimum or supremum operation have a common tree associated with them:

Proposition 3.4. *Let $\{T_i = (t, M_i)\}$ be a collection of tree representations with a common tree t. In this case,*

$$\bigwedge_i T_i = (t, \curlywedge_t \{M_i\}), \tag{3}$$

and

$$\bigvee_i T_i = (t, \curlyvee_t \{M_i\}), \tag{4}$$

where $\curlywedge_t$ and $\curlyvee_t$ are the pointwise infimum and supremum associated to vertex order $\preceq_t$, respectively. Notice that $\curlyvee_t \{M_i\}$ may not always exist.

The situation where the set of tree representations share the same tree is what one encounters when defining flat erosions and dilations on the complete inf-semilattice of tree representations. The flat erosion can be defined as the operator ε given by

$$\varepsilon_B(T) \triangleq \bigwedge_{b\in B} T_{-b} = \bigwedge_{b\in B}(t, M_{-b}), \tag{5}$$

where b is a structuring element, and $M_{-b}(x) = M(x+b)$. It is easy to verify that the above operator is indeed an erosion on $\mathscr{T}_r^L$.

Using Proposition 3.4, one obtains that

$$\varepsilon_B(T) = (t, \curlywedge_t\{M_{-b} \,|\, b \in B\}). \tag{6}$$

As mentioned in Section 2.1, concerning complete inf-semilattices, one can associate to any given erosion ε an opening γ (and, in fact, any morphological operator that is derived from compositions of erosions and openings, such as the internal gradient, dark top-hat transform, and skeletons). Furthermore, the adjoint dilation δ exists, and, even though it is not well defined for all complete inf-semilattice elements, it is always well-defined for elements that are mapped by the erosion ε, and $\gamma = \delta\varepsilon$.

In the case of the above tree-representation flat erosion, the adjoint dilation is given by

$$\delta_B(T) = (t, \curlyvee_t\{M_b \,|\, b \in B\}). \tag{7}$$

We also define the tree-representation reconstruction of T from a marker $\overline{T} = (t, \overline{M}) \leqslant (t, M) = T$ as the infinite iteration of the conditional dilation

$$\delta_B(\overline{T}\,|\,T) \triangleq (t, \curlyvee_t\{\overline{M}_b \curlywedge_t M \,|\, b \in B\}). \tag{8}$$

The conditional dilation $\delta_B(\overline{T}\,|\,T)$ is indeed a dilation for fixed T, and variable $\overline{T}$ smaller or equal to T (see Corollary 3.1 at the end of Section 3.3). Notice that it is always well defined, since it consists of a supremum of bounded elements. This definition is compatible with the novel approach for designing reconstruction operators, proposed by Ronse in [21]. In fact, (8) is a particular case of the last equation in Section 4 of [21]. We also note that, for reconstruction, one would typically use an *extensive* structuring element B (i.e., one that contains the origin of $\mathbb{E}$).

3.3. Threshold sets

The image of vertices M in a tree representation $T = (t, M)$ can be equivalently represented by the collection of sets $\{R_T(v)\}$, $v \in V(t)$, given by

$$R_T(v) \triangleq \{x \in \mathbb{E} \,|\, M(x) \succeq_t v\}. \tag{9}$$

This is because one can obtain M back from $\{R_T(v)\}$ and t by

$$M(x) = \curlyvee_t\{v \in V(t) \,|\, x \in R_T(v)\}. \tag{10}$$

Proposition 3.5. *Let $\{R(v)\}$, $v \in V(t)$, be an arbitrary collection of sets indexed by the vertices of a rooted tree t.* This collection corresponds to a tree representation (t, M), *where M is given by* (10) (replacing R_T with R), iff:

1. $R(r) = \mathbb{E}$,
2. $v \preceq_t v' \Rightarrow R(v) \supseteq R(v')$, and
3. if v and v' are not comparable in t, then $R(v) \cap R(v') = \emptyset$.

Proof. $\{R(v)\}$ corresponds to the tree representation (t, M), where M is computed via (10) with R_T replaced by R, when (a) M is well defined, and (b) $R_T(v) = R(v)$ where $\{R_T(v)\}$ are the thresholds sets computed from M via (9). Conditions 1 and 3 are sufficient for (10) to be well defined (the former ensures that $\{v \in V(t) \,|\, x \in R_T(v)\}$ is not empty, and the latter that it has a supremum). Computing $R_T(v)$ for this M yields $R_T(v) = \{x \in \mathbb{E} \,|\, \curlyvee_t\{v' \in V(t) \,|\, x \in R(v')\} \succeq_t v\} = \bigcup_{v' \succeq v} R(v')$. Condition 2 is then sufficient to yield $R_T(v) = R(v)$. Conversely, condition 1 is necessary, because, whatever M is, $M(x)$ must be larger than or equal to r for all x. Condition 2 is also necessary because if $v \preceq_t v'$, then $M(x) \succeq_t v$ implies $M(x) \succeq_t v'$, which, according to (9), leads to $R(v) \supseteq R(v')$. And condition 3 is necessary because $M(x)$ cannot be larger than two non comparable vertices at the same time. □

The following propositions indicate that the set of threshold sets of tree representations provide an alternative way of characterizing tree morphology on fixed trees.

Proposition 3.6. *Let T_1 and T_2 be two tree representations sharing the same rooted tree t. Then $T_1 \leqslant T_2$ iff $R_{T_1}(v) \subseteq R_{T_2}(v)$, $\forall v \in V(t)$. In addition, if $\{T_i\}$ is a collection of tree representations sharing the same tree t, then $\hat{T} = \bigwedge_i T_i$ iff $R_{\hat{T}}(v) = \bigcap_i R_{T_i}(v)$, $\forall v \in V(t)$. Moreover, assuming that $\bigvee_i T_i$ exists, then $\widetilde{T} = \bigvee_i T_i$ iff $R_{\widetilde{T}}(v) = \bigcup_i R_{T_i}(v)$, $\forall v \in V(t)$.*

Proof. $M_1(x) \preceq_t M_2(x) \iff (M_1(x) \succeq_t v \;\Rightarrow M_2(x) \succeq_t v)$, $\forall v \iff R_{T_1}(v) \subseteq R_{T_2}(v)$, $\forall v$. This proves the equivalence relationship between the partial ordering and the inclusion of threshold sets. The inclusion ordering leads then to intersection and union as infimum and supremum, respectively. □

Notice that the intersection of threshold sets preserves the three conditions of Proposition 3.5, which corroborates the fact that the infimum is always well-defined. The union, on the other hand, does not preserve Condition 3 (even though it does preserve the other two), which is aligned with the fact that the supremum is sometimes not well-defined.

Proposition 3.7. *Consider the flat erosion $\varepsilon_B(T)$ of a tree representation T. Then, for all $v \in V(t)$:*

$$R_{\varepsilon_B(T)}(v) = R_T(v) \ominus B, \tag{11}$$

where $(.) \ominus B$ is the conventional binary erosion by the s.e. B.

Proof.

$$\begin{aligned} R_T(v) \ominus B &= \{x \in \mathbb{E} \,|\, M(x) \succeq_t v\} \ominus B = \bigcap_{b\in B}\{x \in \mathbb{E} \,|\, M(x) \succeq_t v\}_{-b} \\ &= \bigcap_{b\in B}\{x - b \in \mathbb{E} \,|\, M(x) \succeq_t v\} \\ &= \bigcap_{b\in B}\{y \in \mathbb{E} \,|\, M(y+b) \succeq_t v\} \\ &= \{y \in \mathbb{E} \,|\, M_{-b}(y) \succeq_t v, \forall b \in B\} \\ &= \{y \in \mathbb{E} \,|\, \curlywedge_t\{M_{-b}(y) \,|\, b \in B\} \succeq_t v\} = R_{\varepsilon_B(T)}(v). \end{aligned} \tag{12}$$ □

Proposition 3.7 suggests an alternative algorithm for computing the erosion. For any v, (a) compute $R_T(v)$, (b) compute $R_{\varepsilon_B(T)}(v) = R_T(v) \ominus B$, and (c) assign $\ell(v)$ to all points within $R_{\varepsilon_B(T)}(v) \setminus \bigcup_{v \prec v'} R_{\varepsilon_B(T)}(v')$.

Proposition 3.8. *Assume that the flat dilation $\delta_B(T)$ of a tree representation T exists. Then, for all $v \in V(t)$:*

$$R_{\delta_B(T)}(v) = R_T(v) \oplus B. \tag{13}$$

Proof. Assuming that $\delta_B(T)$ exists, the proof is similar to that of Proposition 3.7. □

Proposition 3.9. *Consider the conditional dilation $\delta_B(\overline{T}\,|\,T)$ of a marker $\overline{T}$ inside a mask T, both with the same rooted tree t. Then, for all $v \in V(t)$:*

$$R_{\delta_B(\overline{T}\,|\,T)}(v) = R_T(v) \cap [R_{\overline{T}}(v) \oplus B]. \tag{14}$$

Proof. According to Proposition 3.6, $R_{\overline{T}_b \curlywedge_t T}(v) = R_{\overline{T}_b}(v) \cap R_T(v)$. Furthermore, $R_{(\curlywedge_t)_{b\in B}[\overline{T} \quad b \curlywedge_t T]}(v) = \bigcup_{b\in B}[R_{\overline{T}_b}(v) \cap R_T(v)] = [\bigcup_{b\in B} R_{\overline{T}_b}(v)] \cap R_T(v) = R_T(v) \cap [R_{\overline{T}}(v) \oplus B]$. □

One can verify that, as in the case of the flat erosion, Eq. (14) preserves the three conditions of Proposition 3.5. Drawing once more the analogy with Ronse's work, we note that (14) above is a particular case of Eq. (25) in [21].

Notice that the conditional dilation at the right side of (14) is indeed a dilation for fixed R_T and for $R_{\overline{T}}$ included in R_T. This leads to the following corollary.

Corollary 3.1. *$\delta_B(\cdot|T)$ is a dilation on the set of tree representations, with rooted tree t, that are smaller or equal to T.*

4. Image processing on tree semilattices

Now that morphology on the tree representation domain has been established, we can turn to our ultimate goal, which is to process a given grayscale image f.

4.1. General approach

Let us assume that f is an integer-valued function on $\mathbb{E}$, i.e., $f \in \mathrm{Fun}(\mathbb{E}, \mathbb{Z})$. Moreover, let τ be an operator that transforms f into a pair (T, ℓ), where $T = (t, M) \in \mathscr{T}_r^L$ is a tree representation, and $\ell : L \to \mathbb{Z}$ is a function that maps labels into graylevels. The tree transformation τ should be invertible, and the inversion be given by $[\tau^{-1}(T, \ell)](x) = \ell(M(x))$, $\forall x \in \mathbb{E}$. We propose the following approach for processing f, using the CISL of tree representations:

1. Compute $\tau(f) = (T, \ell) = ((t, M), \ell)$,
2. Perform one or more morphological operations on $T = (t, M)$ to obtain a processed tree representation $\hat{T} = (t, \hat{M})$.
3. Transform $(\hat{T}, \ell)$ back into a new image $\hat{f} \in \mathrm{Fun}(\mathbb{E}, \mathbb{Z})$, using

$$\hat{f}(x) = \tau^{-1}(\hat{T}, \ell) = \ell(\hat{M}(x)). \tag{15}$$

If the morphological operation in step 2 above is the erosion ε_B, then (15) becomes

$$\hat{f}(x) = \ell(\curlywedge_t\{M_{-b}(x) \,|\, b \in B\}). \tag{16}$$

4.2. Particular cases

4.2.1. Flat zones

In order for the tree transform to be invertible, τ should be such that it assigns a common label to each flat zone of f. This is because τ^{-1} maps each label to a single graylevel. This suggests that special attention should be paid to the flat zones of f.

One way of addressing the flat zones of a given image is by considering its Regional Adjacency Graph (RAG). The RAG is a graph, where V is the set of all flat zones of the image, and E contains all pairs of flat zones that are adjacent to each other.

A spanning tree is a sub-graph of a RAG that should, obviously, be a tree, and have the same vertex set V as the RAG. A spanning tree creates a hierarchy in the RAG, defining father/son relationships between adjacent flat zones.

The proposed morphological scheme is of particular interest when t is a spanning tree of the RAG. In this case, the associated morphological operators do not create new gray/color values.

4.2.2. Self-dual trees

Of particular interest are self-dual trees, which treat bright and dark "objects" in the same way. Let us call a tree transform τ self-dual when $\{R_{\tau(f)}(v)\} = \{R_{\tau(-f)}(v')\}$. According to Proposition 3.7, if one can characterize "objects" in an image by thresholding a tree representation (i.e., the "objects" are the sets $\{R_T(v)\}$), then the corresponding tree-domain erosion performs simultaneous "shrinking" of all these image "objects." In particular, if the tree representation is self-dual, then all "objects" are shrunk regardless of whether they are darker or brighter than their background.

4.2.3. Examples

One particular group of (non self-dual) trees are the Max- and Min-Trees [10]. When a tree vertex is always brighter (resp., darker) than its sons, as in the Max-Tree (resp., Min-Tree), the infimum operation always changes the gray level to the local minimum (resp., maximum), which is precisely what the traditional grayscale erosion (resp., dilation) does. In other words, for these trees, the proposed tree approach becomes the traditional grayscale mathematical morphology (resp., its dual version).

More interesting particular cases are the Boundary Topographic Variation (BTV) Tree (see [2]), which is spanning tree of the RAG, built using a minimal topographic distance criterion. Another one is the shape-tree defined in [18] and the resulting semilattice defined in [3,4]. Both provide self-dual morphological operators, based on some inclusion criterion. The shape-tree is not a spanning tree of the RAG, but its nodes are actually the threshold sets of such a spanning tree.

5. Extrema-watershed tree

Based on the general framework of Section 3, all that is needed in order to obtain a new set of morphological operators is a given tree representation. In this section, we explore a particular case of the proposed framework, using a novel self-dual tree representation, which we call the *Extrema-Watershed Tree* (EWT).

5.1. EWT construction

The EWT is a particular case of "Binary Partitioning Tree" [11]; in particular, the proposed representation is built using a particular case of the iterative merging process presented by Salembier, Garrido and Garcia in [13], as follows.

Input all the extrema of a given image (i.e., all regions associated to a local minimum or maximum) into a list, sorted by increasing area.[3] Also, initiate the EWT by setting each flat zone as a leaf vertex. The main loop for the computation of the EWT is as follows: Take the first extremum from the ordered list (the one with smallest area), and merge it with the adjacent neighbor that is the closest one in terms of graylevels. Then, set the merged region as the parent vertex of the above two regions (the extremum and its neighbor) in the EWT. Select the graylevel of the non-extremum region to be the graylevel of the new merged region. Finally, check whether the newly merged region and all its neighbors are extrema, and insert those that are into the sorted list (in their corresponding place, according to the listing order). This loop runs until the list has just one element, which then becomes the EWT root.

Fig. 6 illustrates the computation of the EWT. Consider the image in Fig. 6(a), which contains two extrema with the same area: v_1 and v_3. The first step of the procedure, shown in Fig. 6(b), consists of merging v_1 with v_2, since the difference in graylevel between v_1 and v_2 is smaller than the one between v_3 and v_4. This merger produces a new flat zone – v_5, with the same graylevel as v_2 – which is a new extremum in the image. In the next step, shown in Fig. 6(c),

[3] If two extrema have the same area, input first the one that has the smallest grayscale distance to its closest neighbor.

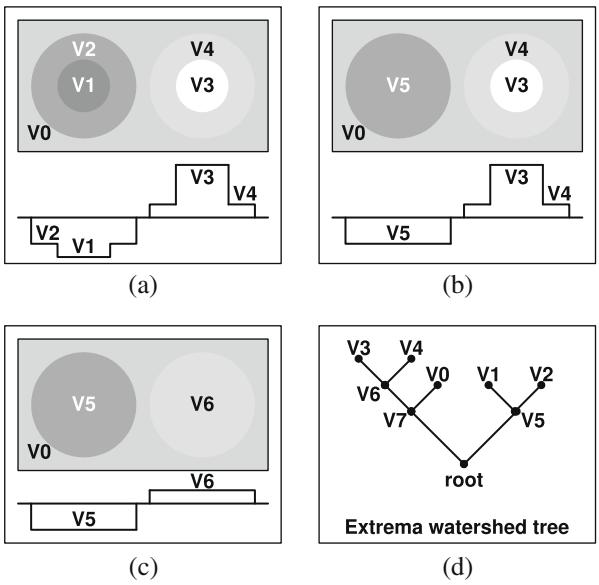

Fig. 6. Example of the EWT computation. (a) Input image, (b) first merging step, (c) second merging step, and (d) the final EWT.

the extremum v_3 is merged with v_4 to create v_6. The procedure continues until all extrema (old and new) are merged. Fig. 6(d) shows the final EWT.

The EWT was loosely inspired by a Watershed segmentation procedure (hence the name), where minima of the gradient image (which include the extrema of the original image) serve as starting point for a region growing procedure, which eventually produce a partition (segmentation) of the image. In [2], evidence of the segmentation characteristics of the EWT are illustrated. One important difference between the EWT construction procedure and Watershed algorithms is the former being computed on the flat zones of the original image, as opposed to the gradient image. In this respect, the EWT more closely resembles the *flat-zone approach* for segmentation [12].

By construction, the EWT is self-dual, since it does not involve the polarity of local contrasts, thus an image f and its "inverse" $-f$ yield the same merging results, and thus identical tree representations, up to an inverted ℓ function. As a consequence, dark and bright elements are processed in the same way, which is the goal of self-dual processing.

5.2. EWT morphological operators

Following the approach described in Section 4, we derive EWT morphological operators by computing morphological operators in the EWT-representation domain (i.e., the tree transform τ is set to the EWT). Fig. 7 shows the result of the EWT erosion and EWT opening. These operators inherit the self-duality property of the EWT; notice that very small features are removed, whereas the larger ones shrunk, in a self-dual manner. The overall brightness of the picture does not change; in particular, the picture does not become darker, which is what usually happens after a standard erosion or opening.

6. Application examples

The EWT has many potential applications; in this section we list just a few.

One application that requires image simplification is pre-processing for OCR (Optical Character Recognition). We have chosen a specific OCR algorithm, used for recognition of license plate numbers, that was developed in [20]. This algorithm uses a mask for each digit and looks for the best correlation among these masks with an image. The algorithm also outputs a confidence grade, which can be used for comparing algorithms. Any noise that exists in the image degrades the correlation value and interferes with the recognition. Consider the example license plate shown in Fig. 8(a), which has been artificially corrupted with blobs of different sizes. Without pre-processing the algorithm fails to read the correct number. Several different algorithms (including linear filtering and traditional grayscale morphology) have been applied to this image. In order to compensate for the lack of duality in classical morphology, we have also compared the EWT with the "quasi-self-dual" opening–closing by reconstruction operator. The only algorithms that cause the algorithm to correctly read the number were the median filter, the quasi-self-dual filter and the EWT opening by reconstruction (see Fig. 8(b), (c), and (d), respectively). The confidence grades associated with the EWT pre-processed image were higher than those for the median filter and the quasi-self-dual filter. Further details on this experiment can be found in [2].

Another example uses opening by reconstruction as an initial step for an application that removes dust and scratches from images. The elements filtered by the opening by reconstruction are completely extracted, including their edges. This enables one to extract candidates for dust and scratch removal, without corrupting their shapes. The proposed operation is a EWT top-hat filter. Fig. 9 shows an example. Subsequent steps (not considered here) can then make further analysis of the image in order to decide which candidates should be removed. We have compared the proposed approach to linear and median filters. Subjective and objective criteria were used. The subjective criterion is the overall corruption of the candidate shapes. The objective criterion is the measured energy of the filtered images. The EWT performed better in both criteria. On the one hand, for the relevant structuring

Fig. 7. (a) Original image, (b) EWT erosion by square SE 5×5, and (c) EWT opening by square SE 5×5.

Fig. 8. (a) Input image, artificially corrupted, (b) filtered with a median filter, (c) filtered with regular "quasi-self-dual" opening by reconstruction, and (d) filtered by the EWT-based opening by reconstruction, using circle SE of radius 4.

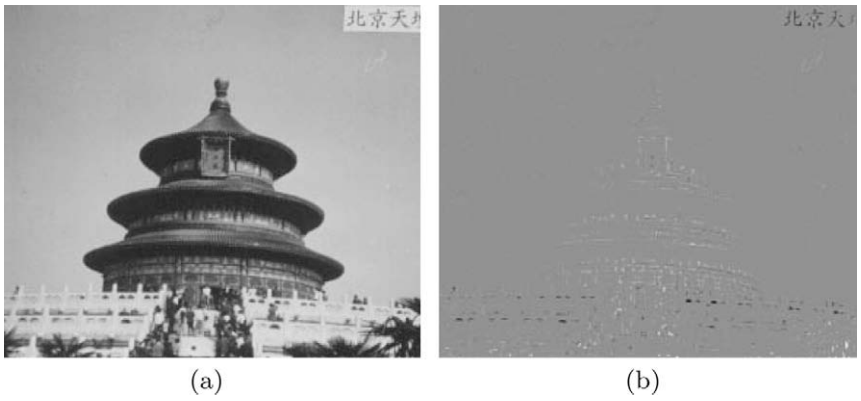

Fig. 9. Top hat, using cross SE 3×3, as a pre-processing stage for dust and scratch removal. (a) Original image and (b) top hat by reconstruction based on EWT.

elements, the energy of the EWT filtered image was lower than for the linear and median filters. On the other hand, the linear and median filters do not completely extract the artifacts, as can be seen in Fig. 10 for the "cross" structuring element.

7. Towards tree-based image semilattices

We have proposed in Section 3.1 a CISL of tree representations, and its use for image processing, which, as seen in Section 6, can be

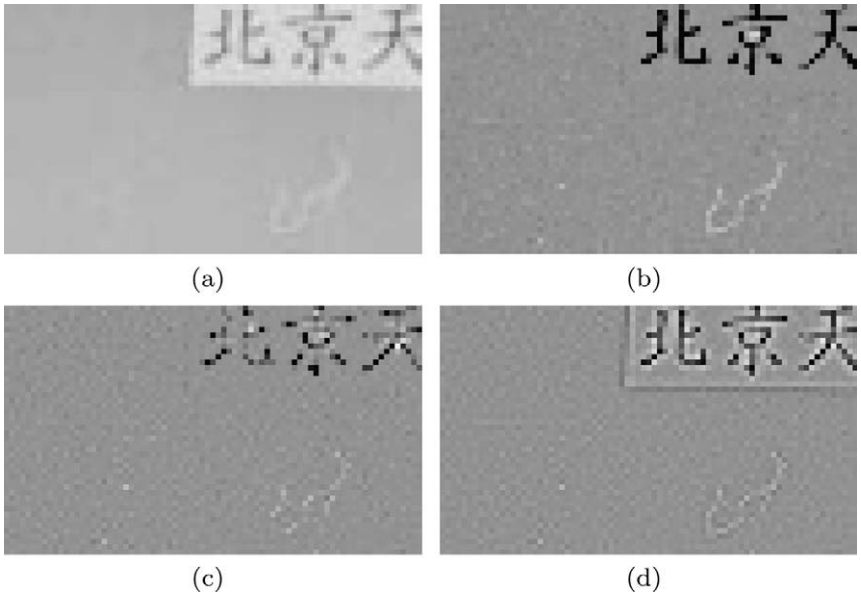

Fig. 10. Detail view of Fig. 9. Top hat, using cross SE 3×3, as a pre-processing stage for dust and scratch removal. (a) Original image, (b) top hat by reconstruction based on EWT, (c) top hat using median, and (d) top hat using an averaging filter. Notice that the EWT top-hat is the only result that faithfully extract both the hair-like artifact as well as the fonts. This simplifies a subsequent artifact removal process. The median produces "broken lines," whereas the averaging-based filter introduces fringes/halos as well as unnecessary candidates (the rectangle borders, in this case).

very useful. On the other hand, what we would really like is the CISL of tree representation to induce a CISL *in the image domain*. That is, we would like, for instance, the composite operation of $\tau^{-1}\varepsilon\tau$ to be an erosion in the image domain. However, that is not guaranteed. If one fixes the label function ℓ (that is, make it independent from τ), then the partial ordering in the tree-representation domain does induce a partial ordering for images, for any τ; however, the infimum operation is not guaranteed to be well-defined. In fact, the EWT proposed here, in its current form, does not yield a CISL in the image domain. On the other hand, the *shape tree* does induce a CISL in the image domain, as shown in [3]. In this section, we explore first steps towards designing tree transforms τ that ensure an image CISL.

Let $\mathscr{F}$ be a set, $(\mathscr{T},\leqslant)$ be a CISL, and consider the maps $\tau:\mathscr{F}\to\mathscr{T}$ and $\tau^{-1}:\mathscr{T}\to\mathscr{F}$, such that $\tau^{-1}\tau$ is equal to the identity. Let $\mathscr{F}^*$ be the image of $\mathscr{F}$ through τ. Moreover, let $\overline{\mathscr{T}}^*$ be the closure of $\mathscr{T}^*$ by the CISL infimum, that is, $\overline{\mathscr{T}}^*$ is the set of elements in $\mathscr{T}$ that can be written as $\curlywedge\{\tau f_i\}$ for some collection $\{f_i\}$ in $\mathscr{F}$. Define a partial order on $\mathscr{F}$ as follows: $f_1\sqsubseteq f_2\iff\tau f_1\leqslant\tau f_2$.

Proposition 7.1. *$(\mathscr{F},\sqsubseteq)$ is a CISL with infimum $\sqcap\{f_i\}$ given by $\tau^{-1}(\wedge\{\tau f_i\})$ iff $\tau\tau^{-1}$ is an opening on $\overline{\mathscr{T}}^*$.*

Proof. First, let us show that if $\tau\tau^{-1}$ is an opening on $\overline{\mathscr{T}}^*$, then $(\mathscr{F},\sqsubseteq)$ is a CISL. Consider a set $\{f_i\}$ of elements in $\mathscr{F}$, and the element given by $\tau^{-1}(\wedge\{\tau f_i\})$. We have $\tau\tau^{-1}(\wedge\{\tau f_i\})\preceq\wedge\{\tau f_i\}\preceq\tau f_i,\ \forall i$, therefore $\tau^{-1}(\wedge\{\tau f_i\})\sqsubseteq f_i,\ \forall i$. Furthermore, if some arbitrary element g satisfies, $g\sqsubseteq f_i,\ \forall i$, then $\tau g\leqslant\tau f_i,\ \forall i$, which leads to $\tau g\leqslant\wedge\{\tau f_i\}$. Applying $\tau\tau^{-1}$ to both sides, and using $\tau\tau^{-1}\tau=\tau$, yield $\tau g\leqslant\tau\tau^{-1}(\wedge\{\tau f_i\})$, and therefore $g\sqsubseteq\tau^{-1}(\wedge\{\tau f_i\})$. The conclusion is that $\tau^{-1}(\wedge\{\tau f_i\})$ is the infimum of $\{f_i\}$, and since it is well-defined, $(\mathscr{F},\sqsubseteq)$ is a CISL.

Now let us prove the opposite direction. Assume $(\mathscr{F},\sqsubseteq)$ is a CISL with the infimum $\tau^{-1}(\wedge\{\tau(\cdot)\})$. Then $\tau^{-1}(\wedge\{\tau f_i\})\sqsubseteq f_i,\ \forall i$, that is, $\tau\tau^{-1}(\wedge\{\tau f_i\})\leqslant\tau f_i,\ \forall i$, which in turn leads to $\tau\tau^{-1}(\wedge\{\tau f_i\})\leqslant\wedge\{\tau f_i\}$. In other words, $\tau\tau^{-1}$ is *anti-extensive* for elements in $\overline{\mathscr{T}}^*$. Moreover, since $\tau^{-1}\tau=\mathrm{id}$, we have $\tau\tau^{-1}\tau\tau^{-1}=\tau\tau^{-1}$, which means that $\tau\tau^{-1}$ is also *idempotent*. Finally, assume that $\wedge\{\tau f_i\}\leqslant\wedge\{\tau g_i\}$; then $\wedge\{\tau f_i\}\leqslant\tau g_i,\ \forall i$, and therefore $\tau\tau^{-1}(\wedge\{\tau f_i\})\leqslant\tau g_i,\ \forall i$, which leads to $\tau^{-1}(\wedge\{\tau f_i\})\sqsubseteq g_i,\ \forall i$. But this means that $\tau^{-1}(\wedge\{\tau f_i\})\sqsubseteq\sqcap\{g_i\}=\tau^{-1}(\wedge\{\tau g_i\})$. Therefore $\tau\tau^{-1}(\wedge\{\tau f_i\})\leqslant\tau\tau^{-1}(\wedge\{\tau g_i\})$, and we get that $\tau\tau^{-1}$ is also *increasing* for elements in $\overline{\mathscr{T}}^*$. □

Let us focus now on the case of tree representations. $\mathscr{F}$ is the set of functions, $(\mathscr{T},\leqslant)$ is the CISL of tree representations, and τ is a tree transform, where ℓ is fixed – i.e., the label function ℓ does not depend on τ, so that, instead of $\tau^{-1}(T,\ell)$, we will write simply $\tau^{-1}(T)$ in this subsection.

The best situation is, of course, when $\mathscr{T}^*=\mathscr{T}$, and then $\overline{\mathscr{T}}^*$ also must be equal to $\mathscr{T}$. In this case $\tau\tau^{-1}$ is the identity operator on $\overline{\mathscr{T}}^*$, since any element in $\overline{\mathscr{T}}^*$ can be written as $\tau\tilde{f}$ for some $\tilde{f}\in\mathscr{F}$, thus $(\tau\tau^{-1})\tau\tilde{f}=\tau(\tau^{-1}\tau)\tilde{f}=\tau\tilde{f}$. Being the identity, $\tau\tau^{-1}$ is an opening, and this ensures that $(\mathscr{F},\sqsubseteq)$ is a CISL. However, this situation does not occur for tree representations; one usually can replace the tree t in a representation $T=(t,M)$ by another tree t' (e.g., any t' such that $t'\supset t$), and the reconstruction $\tau^{-1}(T)$ will not change. That is, τ is not bijective, and therefore $\mathscr{T}^*\subset\mathscr{T}$.

Another good situation is the case when $\overline{\mathscr{T}}^*=\mathscr{T}^*\subset\mathscr{T}$, where we still obtain $\tau\tau^{-1}=\mathrm{id}$ as above. But this case is not common either: Suppose we have $T_1=(t,M_1)$ and $T_2=(t,M_2)$, and that $M_1\curlywedge_t M_2$ spans only a subset t' of the tree t. Then we get that $\tau\tau^{-1}[(t,M_1\curlywedge_t M_2)]$ will probably not return $(t,M_1\curlywedge_t M_2)$; instead it could return $(t',M_1\curlywedge_t M_2)<(t,M_1\curlywedge_t M_2)$ – if τ is designed appropriately.

Rather than the above ideal situations, it is more likely for one to be able to design τ such that $\tau\tau^{-1}$ is an opening different from the identity. A good strategy could be the following. First, fix the label function ℓ a priori (make it independent from τ). Now design τ such that (a) $\tau\tau^{-1}$ preserves the image of vertices, i.e., if $(t',M')=\tau\tau^{-1}[(t,M)]$, then $M'=M$, and (b) the mapping $t\mapsto t'$ is an opening w.r.t. the inclusion order. This is not necessarily a trivial task, and this subject is still under investigation.

8. Conclusion

We have presented a general framework for producing new morphological operators that are compatible to given tree representations. Furthermore, a useful particular case is provided, based on a new tree representation, the Extrema Watershed Tree. The resulting morphological erosion and opening operators were applied to a number of application examples, giving better results in comparison to other filtering techniques, including classical morphological filtering. In general, EWT-based filtering performs well in tasks suitable for classical morphological filtering, especially when self-duality is required. We have also presented first steps towards the design of tree representations that induce a CISL in the image domain.

Acknowledgments

We would like to thank the reviewers for their thorough reports, which we believe helped significantly to improve the article.

References

[1] A. Vichik, R. Keshet, D. Malah, Self-dual morphology on tree semilattices and aplications, in: Mathematical Morphology and its Applications to Image and Signal Processing, Proc. of ISMM'2007, 2007, pp. 49–60.
[2] A. Vichik, Self-dual morphological operators based on tree representations of images, Masters Thesis, EE Dept., Technion-IIT, Israel, 2006. Available from: <http://sipl.technion.ac.il/new/Research/Publications/Graduates/Alla_Vichik/Alla_main_revised_f.pdf>.
[3] R. Keshet, Shape-tree semilattice, Journal of Mathematical Imaging and Vision 22 (May) (2005) 309–331.
[4] R. Keshet, Adjacency lattices and shape-tree semilattices, Image & Vision Computing, Special Issue on ISMM05, vol. 25, No. 4, April 2007.
[5] R. Keshet (Kresch), Mathematical morphology on complete semilattices and its applications to image processing, Fundamenta Informaticae 41 (January) (2000) 33–56.
[6] R. Keshet (Kresch), Extension of morphological operations to complete semilattices and its applications to image and video processing, mathematical morphology and its applications to image and signal processing, in: Proc. of ISMM'98, June 1998, pp. 35–42.
[7] H.J.A.M. Heijmans, R. Keshet, Inf-semilattice approach to self-dual morphology, Journal of Mathematical Imaging and Vision 17 (July) (2002) 55–80.
[8] J. Goutsias, H.J.A.M. Heijmans, Fundamenta MorphologicæMathematicæ, Fundamenta Informaticæ 41 (January) (2000) 1–2.
[9] P. Salembier, J. Llach, L. Garrido, Visual segment tree creation for MPEG-7 description schemes, Pattern Recognition 35 (March) (2002) 563–579.
[10] Ph. Salembier, L. Garrido, Connected operators based on region-tree prunning, in: Proc. of the 15th International Conference on Pattern Recognition (ICPR'00), vol. 3, September 2000, pp. 367–370.
[11] Ph. Salembier, Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, IEEE Transactions on Image Processing 9 (April) (2000) 561–576.
[12] J. Crespo, R.W. Schafer, J. Serra, C. Gratin, F. Meyer, The flat zone approach: a general low-level region merging segmentation method, Signal Processing 62 (1997) 37–60.
[13] Ph. Salembier, L. Garrido, D. Garcia, Auto-dual connected operators based on iterative merging algorithms, IEEE Transactions on Image Processing 7 (April) (1998) 4.
[14] P. Monasse, F. Guichard, Fast computation of a contrast-invariant image representation, IEEE Transactions on Image Processing 9 (2000) 860–872.
[15] P. Monasse, F. Guichard, Scale-space from a level lines tree, Journal of Visual Communication and Image Representation 11 (2000) 224–236.
[16] C. Ballester, V. Caselles, P. Monasse, The tree of shapes of an image, ESAIM: Control, Optimization and Calculus of Variations 9 (2003) 1–18.
[17] V. Caselles, P. Monasse, Grain filters, Journal of Mathematical Imaging and Vision 17 (3) (2002) 249–270.
[18] P. Monasse, Morphological representation of digital images and application to registration, Universit Paris IX-Dauphine, Ph.D. Thesis, 2000.

[19] R. Diestel, Graph Theory, Electronic ed., Springer, New York, 2000.
[20] S. Vichik, R. Sandler, A. Rosen, Moving car license plate recognition, Semestrial project, Technion-IIT, 1999. Available from: <http://www.cs.technion.ac.il/Labs/Isl/Project/Projects_done/cars_plates/finalreport.htm>.
[21] C. Ronse, Reconstructing masks from markers in non-distributive lattices, Applicable Algebra in Engineering, Communication and Computing 19 (1) (2008) 51–85.