Selection of relevant nodes from component-trees in linear time

Nicolas Passat, Benoît Naegel

Selection of relevant nodes
from component-trees in linear time

Nicolas Passat[*][(a)], Benoît Naegel[(b)]

(a) Université de Strasbourg, LSIIT, UMR CNRS 7005, France
(b) Université Nancy 1, LORIA, UMR CNRS 7503, France
passat@unistra.fr, benoit.naegel@loria.fr

Abstract. Component-trees associate to a discrete grey-level image a descriptive data structure induced by the inclusion relation between the binary components obtained at successive level-sets. This article presents a method to extract a subset of the component-tree of an image enabling to fit at best a given binary target selected beforehand in the image. A proof of the algorithmic efficiency of this method is proposed. Application examples related to the extraction of drop caps from ancient documents emphasise the usefulness of this technique in the context of assisted segmentation.

Key words: component-tree, image analysis, grey-level images.

1 Introduction

The *component-tree* (also known as *dendrone* [1], *confinement tree* [2] or *max-tree* [3]) is a graph-based structure which models some characteristics of a grey-level image by considering its binary level-sets obtained from successive thresholding operations. Component-trees have been involved, in particular, in the development of morphological operators [3,4].

By definition, they are particularly well-suited for the design of methods devoted to process grey-level images based on hypotheses related to the topology (connectedness) and the specific intensity (locally/globally minimal or maximal) of structures of interest. Based on these properties, component-trees have been involved in various kinds of image processing methods [1,2,3,5,6,7,8].

Several works related to component-trees have been devoted to enable their efficient computation [2,3,4,9]. In particular, the ability to compute them in (quasi-)linear time opens the way to the development of interactive and efficient segmentation methods.

The design of interactive segmentation methods is an active research field (see, *e.g.*, [10] for a recent survey). This dynamism is justified by (*i*) the increasing necessity to analyse images in a large spectrum of application fields, (*ii*) the difficulty to develop fully automatic segmentation methods, and (*iii*) the importance to develop segmentation methods as tools for assisting the user by explicitly using his expertise.

Some of interactive segmentation methods aim at correcting a rough segmentation initially performed in a manual fashion. In this article, we focus on this kind of issue, especially in the (frequent) case where the structures of interest to be segmented are the ones of extremal intensities. In this context, component-trees can be of high usefulness.

Based on these considerations, this article is devoted to answer the following problem: Let I be a grey-level image, let T be its component-tree, and let G be a binary object defined on the same domain as I (assumed to be a rough segmentation of I); how can we determine a part of T (and thus of I) which enables to fit at best G with the lowest computational cost? This "best" approximation can, in particular, be considered from a quantitative point of view, *i.e.*, by finding a solution minimising the amount of false positives/negatives.

The article is organised as follows. In Sec. 2, definitions related to the notion of component-tree are recalled. In Sec. 3, definitions and notations related to the considered issue are proposed. In Sec. 4, some solutions to the proposed problem are described and their linear algorithmic cost is established. Algorithmic considerations are proposed in Sec. 5. An application to the segmentation of drop caps from ancient documents is proposed in Sec. 6, in order to illustrate the relevance of the method and its actual usefulness in real image analysis applications. Conclusions will be found in Sec. 7.

2 Component-tree

Let $n \in \mathbb{N}^*$. Let us consider an adjacency relation on the discrete grid defined by $\mathbb{Z}^n$, for instance, the $2n$- or the $(3^n - 1)$-adjacency. Let $X \subseteq \mathbb{Z}^n$ be a non-empty set of $\mathbb{Z}^n$.

We say that two points $x, y \in X$ are connected (in X), and we note $x \sim_X y$, if there exists a sequence $(x_k)_{k=1}^t$ $(t \geq 1)$ of elements of X such that $x_1 = x$, $x_t = y$ and x_k, x_{k+1} are adjacent for all $k \in [\![1, t-1]\!]$. Note that $\sim_X$ is an equivalence relation on X. The connected components of X are the elements of the quotient set $X/\sim_X$ (noted $C[X]$ in the sequel). We say that X is connected if $C[X] = \{X\}$.

Let $E \subset \mathbb{Z}^n$ be a finite connected set. Let $\bot \leq \top \in \mathbb{Z}$ and $V = [\![\bot, \top]\!]$. A discrete grey-level image I can be defined as a function $I : E \rightarrow V$ (we also note $I \in V^E$).

For any $v \in V$, we define the thresholding function $X_v : V^E \rightarrow \mathcal{P}(E)$ (where $\mathcal{P}(E) = \{Y \mid Y \subseteq E\}$) by $X_v(I) = \{x \in E \mid v \leq I(x)\}$ for all $I \in V^E$.

For any $v \in V$, and any $X \subseteq E$, we define the cylinder function $C_{X,v} : E \rightarrow V$ by $C_{X,v}(x) = v$ if $x \in X$ and $\bot$ otherwise. A discrete image $I \in V^E$ can then be expressed as

$$I = \bigvee_{v \in V} C_{X_v(I),v} = \bigvee_{v \in V} \bigvee_{X \in C[X_v(I)]} C_{X,v} \tag{1}$$

where $\bigvee$ is the pointwise supremum for the sets of functions.

Let $\mathcal{K} = \bigcup_{v \in V} C[X_v(I)]$ be the set of all the connected components obtained from the different thresholdings of I at values $v \in V$. The inclusion relation $\subseteq$ is then a partial order on $\mathcal{K}$. Let $v_1 \leq v_2 \in V$. Let $B_1, B_2 \subseteq E$ be the binary images defined by $B_k = X_{v_k}(I)$ for $k \in \{1, 2\}$. Let $C_2 \in C[B_2]$ be a connected component of B_2. Then, there exists a (unique) connected component $C_1 \in C[B_1]$ of B_1 such that $C_2 \subseteq C_1$.

Based on these properties, it can be easily deduced that the Hasse diagram of the partially ordered set $(\mathcal{K}, \subseteq)$ is a tree (*i.e.*, a connected acyclic graph), and more precisely
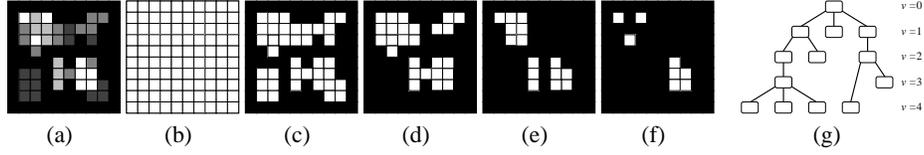
Fig. 1. (a) A grey-level image $I : [\![0,9]\!]^2 \rightarrow [\![0,4]\!]$ (from 0, in black, to 4, in white). (b–f) Threshold images $X_v(I)$ (white points) for v varying from 0 (b) to 4 (f). (g) The component-tree of I. Its levels correspond to increasing thresholding values v. The root (*i.e.*, the upper node located at the level $v = 0$) corresponds to the support $E = [\![0,9]\!]^2$ of the image.

a rooted tree, the root of which is the supremum $X_\perp(I) = E$. This tree is called the *component-tree* of I.

Definition 1 *Let $I \in V^E$ be a grey-level image. The* component-tree *of I is the rooted tree $T = (\mathcal{K}, L, R)$ such that:*

(i) $\mathcal{K} = \bigcup_{v \in V} C[X_v(I)]$
(ii) $L = \{(X, Y) \in \mathcal{K}^2 \mid Y \subset X \wedge \forall Z \in \mathcal{K}, Y \subseteq Z \subset X \Rightarrow Y = Z\}$
(iii) $R = \max(\mathcal{K}, \subseteq) = X_\perp(I) = E$

The elements of $\mathcal{K}$ (resp. of L) are the nodes *(resp. the* edges*) of T. The element R is the* root *of T. For any $N \in \mathcal{K}$, we set $ch(N) = \{N' \in \mathcal{K} \mid (N, N') \in L\}$; $ch(N)$ is the set of the* children *of the node N in T. An example of component-tree is illustrated in Fig. 1.*

Each node of $\mathcal{K}$ is a binary connected component distinct from all the other nodes. However, such a connected component can be an element of $C[X_v(I)]$ for several (successive) values $v \in V$. For each $X \in \mathcal{K}$, we set $m(X) = \max\{v \in V \mid X \in C[X_v(I)]\} = \min_{x \in X}\{I(x)\}$. We then consider that X is "associated" to the value $m(X)$, *i.e.*, to the highest value of V which generates this connected component.

The following definition, establishing "correlation scores" between a node and a given binary object, will be useful in the sequel of the article.

Definition 2 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $N \in \mathcal{K}$ be a node of T. Let $G \subseteq \mathbb{Z}^n$ be a binary object. We set $p^*(N, G) = |(N \setminus \bigcup_{N' \in ch(N)} N') \cap G|$, and $n(N, G) = |N \setminus G|$. The value $n(N, G)$ is the number of points of N which do not belong to G. The value $p^*(N, G)$ is the number of points of N which belong to G and which do not belong to any children of N.*

Remark 3 *When building the component-tree of I, it is possible to store, at each node $N \in \mathcal{K}$, the set of points $E_N = N \setminus \bigcup_{N' \in ch(N)} N'$. This leads, in particular, to an algorithmically useful partition $\{E_N\}_{N \in \mathcal{K}}$ of E. In such conditions, for a given binary object $G \subseteq \mathbb{Z}^n$, the computation of all the $p^*(N, G)$ and $n(N, G)$ $(N \in \mathcal{K})$ can obviously be performed in linear time $O(|E|)$. In the sequel, we will assume that $p^*(N, G)$ and $n(N, G)$ have been computed and are then available for every node $N \in \mathcal{K}$.*

3 Purpose

Component-trees can be used to develop image processing/analysis procedures based on filtering or segmentation strategies. Such procedures generally consist of determining a subset $\widehat{\mathcal{K}} \subseteq \mathcal{K}$ among the nodes of the component-tree $T = (\mathcal{K}, L, R)$ of a considered image $I : E \rightarrow V$.

When performing segmentation, the (binary) resulting image $I_s \subseteq E$ is defined as the union of the nodes of $\widehat{\mathcal{K}}$, *i.e.*, as

$$I_s = \bigcup_{N \in \widehat{\mathcal{K}}} N \tag{2}$$

In this context, the determination of the nodes to preserve is a complex issue, which can be handled by considering attributes (*i.e.*, qualitative or quantitative information related to each node) to characterise the nodes of interest. An alternative solution is to search the set of nodes $\widehat{\mathcal{K}} \subseteq \mathcal{K}$ which enables to generate a binary object being as similar as possible to a given binary target (*e.g.*, an approximate segmentation obtained from a manual process). In the sequel of the article, we focus on this specific issue, which can be formalised as an optimisation problem.

Problem to solve Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $G \subseteq E$ be a binary image. Let d be a (pseudo-)distance on $\mathcal{P}(E)$. How can we compute a set of nodes $\widehat{\mathcal{K}} \subseteq \mathcal{K}$ such that $d(\bigcup_{N \in \widehat{\mathcal{K}}} N, G)$ is minimal, *i.e.*, such that the best binary object which can be built from $\mathcal{K}$ is as close as possible to G? More formally, the problem can be summarised as a minimisation problem, consisting of determining

$$\widehat{\mathcal{K}} = \arg \min_{\mathcal{K}' \in \mathcal{P}(\mathcal{K})} \{d(\bigcup_{N \in \mathcal{K}'} N, G)\} \tag{3}$$

An intuitive solution for determining a useful (pseudo-)distance d is to consider the amount of false positives/negatives induced by $\bigcup_{N \in \mathcal{K}'} N$ w.r.t. the considered binary object of interest G.

Definition 4 *Let $\alpha \in [0, 1]$. Let $d^\alpha : \mathcal{P}(E) \times \mathcal{P}(E) \rightarrow \mathbb{R}^+$ be the function defined by*

$$d^\alpha(X, Y) = \alpha.|X \setminus Y| + (1 - \alpha).|Y \setminus X| \tag{4}$$

The pseudo-distance[1] d^α constitutes a good similarity criterion between binary objects. Note that $d^0(X, Y) = |Y \setminus X|$ (resp. $d^1(X, Y) = |X \setminus Y|$), *i.e.*, $d^0(X, Y)$ (resp. $d^1(X, Y)$) is the amount of false negatives (resp. false positives) in X w.r.t. Y.

In the next sections, we will consider this (pseudo-)distance. It will be established that it leads to algorithmically efficient processes, and satisfactory applicative results.

---

[1] The function d^α is actually not a distance since $d^\alpha(X, Y) = d^\alpha(Y, X)$ if and only if $\alpha = 1/2$, $d^\alpha(X, Y) = 0 \Leftrightarrow X = Y$ if and only if $\alpha \in]0, 1[$, and d^α does not satisfy, in general, the triangle inequality.

4 Theoretical study

4.1 Preliminary properties

The following property directly derives from the definitions of Sec. 2.

Property 5 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $N \in ch(E)$. Let $\mathcal{K}_N = \{N' \in \mathcal{K} \mid N' \subseteq N\}$. Let $I_{|N} \in V^N$ be the grey-level image corresponding to the restriction of I to the node N. The Hasse diagram $(\mathcal{K}_N, L_N)$ of the partially ordered set $(\mathcal{K}_N, \subseteq)$ enables to define the component-tree $T_N = (\mathcal{K}_N, L_N, N)$ of $I_{|N}$ which is actually a subtree of T. Note in particular that $\{E\} \cup \{\mathcal{K}_N\}_{N \in ch(E)}$ is a partition of $\mathcal{K}$, while $\{(E, N)\}_{N \in ch(E)} \cup \{L_N\}_{N \in ch(E)}$ is a partition of L.*

Definition 6 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $x \in E$. We set $\mathcal{K}_x = \{N \in \mathcal{K} \mid x \in N\}$, $\mathcal{K}_x \subseteq \mathcal{K}$ is the subset of all the nodes of $\mathcal{K}$ which contain x.*

Since $E \in \mathcal{K}$, the following property is obvious, while the next one derives from the structure of the component-tree.

Property 7 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $x \in E$. Then, $\mathcal{K}_x$ is non-empty.*

Property 8 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $x \in E$. Then, $(\mathcal{K}_x, \subseteq)$ is a completely ordered set.*

Definition 9 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $\mathcal{G} : \mathcal{P}(\mathcal{K}) \to \mathcal{P}(E)$ be the function defined by $\mathcal{G}(\mathcal{K}') = \bigcup_{N \in \mathcal{K}'} N$ for all $\mathcal{K}' \subseteq \mathcal{K}$. We set $Q = \mathcal{G}(\mathcal{P}(\mathcal{K})) = \{\mathcal{G}(\mathcal{K}')\}_{\mathcal{K}' \subseteq \mathcal{K}}$, Q is the set of all the binary objects which can be generated from the subsets of nodes of $\mathcal{K}$.*

Although there exist $2^{|\mathcal{K}|}$ distinct subsets $\mathcal{K}'$ of $\mathcal{K}$, most of these subsets generate a same binary object of E, more formally, we have $|Q| \leq |\mathcal{P}(\mathcal{K})|$ (and generally $|Q| \ll |\mathcal{P}(\mathcal{K})|$).

Property 10 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let Q be the set of the objects which can be generated from the subsets of nodes of $\mathcal{K}$. Let $Q \in Q$. Then, we have*

$$C[Q] = \min_{\subseteq} \mathcal{G}^{-1}(Q) \tag{5}$$

Less formally, the set of the connected components of Q is actually a subset of nodes of $\mathcal{K}$ which is included in any other subset of nodes $\mathcal{K}'$ of $\mathcal{K}$ generating Q. Such sets $\mathcal{K}'$ are then redundant (they contain in particular some nodes which are included in other nodes, and then useless for the generation of Q).

4.2 Main properties

Smallest superset / largest subset In this subsection, we first focus on a specific case of the considered issue, which consists of finding a subset of nodes of the component-tree of an image I such that the object generated by these nodes is *included in* (resp. *includes*) the binary target G and is the *largest* (resp. the *smallest*) one verifying this property. This problem is equivalent to consider a pseudo-distance d which only takes into account the amount of false negatives (resp. false positives) w.r.t. G.

The next property establishes that there exists a (unique) solution to this problem.

Property 11 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let Q be the set of the objects which can be generated from the subsets of nodes of $\mathcal{K}$. Let $G \subseteq E$. Then there exist $G^+, G^- \in Q$ such that*

$$G^+ = \min_{\subseteq} \{Q \in Q \mid G \subseteq Q\} \tag{6}$$

$$G^- = \max_{\subseteq} \{Q \in Q \mid Q \subseteq G\} \tag{7}$$

Proof If $G = \emptyset$, by setting $G^+ = \emptyset \in Q$, we are done. Let us now suppose that $G \neq \emptyset$. For any $x \in G$, we set $N_x = \min_{\subseteq} \mathcal{K}_x$. Let $G^+ = \bigcup_{x \in G} N_x$. Then we have $G^+ \in Q$ and $G \subseteq G^+$. Let $Q' \in Q$ such that $G \subseteq Q'$. Let $y \in G^+$. If $y \in G$, then we have $y \in Q'$. Let us now suppose that $y \in G^+ \setminus G$. Then, there exists $x \in G$ such that $y \in N_x$. Since $x \in G \subseteq Q'$, there exists $N \in \mathcal{K}_x$ such that $N \subseteq Q'$. But then, we have $y \in N_x \subseteq N \subseteq Q'$. Consequently, we have $Q \subseteq Q'$, and thus $G^+ = \min_{\subseteq} \{Q \in Q \mid G \subseteq Q\}$.
Let $G^- = \bigcup_{N \in \mathcal{K} \wedge N \subseteq G} N$. We have $G^- \in Q$ and $G^- \subseteq G$. Let $Q' \in \{Q \in Q \mid Q \subseteq G\}$. Let us suppose that there exists $x \in Q' \setminus G^-$. In particular, we have $x \in G$. There exists $N_x \in \mathcal{K}_x$ such that $N_x \subseteq Q'$. If $N_x \subseteq G$ then we have $x \in N_x \subseteq G^-$: contradiction. If $N_x \nsubseteq G$ then we have $Q' \nsubseteq G$: contradiction. Consequently, for all $x \in Q'$, we have $x \in G^-$, and thus $G^- = \max_{\subseteq} \{Q \in Q \mid Q \subseteq G\}$. $\square$

We define now two functions which enable to compute these solutions G^+ and G^- (Def. 12, Props. 13 and 14) and we show that they authorise a computation in linear time w.r.t. the size (*i.e.*, the number of nodes) of the component-tree of the considered image I or the size of the support E of this image (Prop. 15).

Definition 12 *Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $G \subseteq E$. Let $\mathcal{F}^+, \mathcal{F}^- \in \mathcal{P}(\mathcal{K})^{\mathcal{K}}$ be the functions recursively defined, for all $N \in \mathcal{K}$, by*

$$\mathcal{F}^+(N) = \begin{cases} \{N\} & \text{if } p^*(N, G) \neq 0 \\ \bigcup_{N' \in ch(N)} \mathcal{F}^+(N') & \text{if } p^*(N, G) = 0 \end{cases} \tag{8}$$

$$\mathcal{F}^-(N) = \begin{cases} \{N\} & \text{if } n(N, G) = 0 \\ \bigcup_{N' \in ch(N)} \mathcal{F}^-(N') & \text{if } n(N, G) \neq 0 \end{cases} \tag{9}$$

In particular, if $ch(N) = \emptyset$, we have $\bigcup_{N' \in ch(N)} \mathcal{F}^-(N') = \bigcup_{N' \in ch(N)} \mathcal{F}^+(N') = \emptyset$, which guarantees the termination of these recursive definitions. The function $\mathcal{F}^+$ (resp. $\mathcal{F}^-$) provides, for any node $N \in \mathcal{K}$, the subset of nodes of the subtree of T having N for root, which enables to generate the set $(G \cap N)^+$ (see Eq. (6)) (resp. $(G \cap N)^-$ (see Eq. (7))) for the restriction of the image I to N.

Property 13 *Let $\sigma \in \{+, -\}$. Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $G \subseteq E$. Then we have*

$$C[G^\sigma] = \mathcal{F}^\sigma(E) \tag{10}$$

Proof Let $X, Y \in \mathcal{F}^\sigma(E)$. By definition, we have $X, Y \in \mathcal{K}$. Moreover, if $X \neq Y$, it obviously comes that $X \cap Y = \emptyset$. Consequently, there exists $Q \in Q$ such that $\mathcal{F}^\sigma(E) = C[Q]$. By induction from the definition of $\mathcal{F}^+(E)$ (resp. $\mathcal{F}^-(E)$), we easily deduce that $\bigcup_{N \in \mathcal{F}^+(E)} N = \bigcup_{N \in \mathcal{K} \wedge p^*(N,G) \neq 0} N$ (resp. $\bigcup_{N \in \mathcal{F}^-(E)} N = \bigcup_{N \in \mathcal{K} \wedge n(N,G)=0} N$). In particular, it follows that $\bigcup_{N \in \mathcal{F}^+(E)} N \in \{Q \in Q \mid G \subseteq Q\}$ (resp. $\bigcup_{N \in \mathcal{F}^-(E)} N \in \{Q \in Q \mid Q \subseteq G\}$). Let $N \in \mathcal{F}^+(E)$. Let $y \in G$ such that $y \in N$ and $y \notin \bigcup_{N' \in ch(N)} N'$ (such a point y exists as $p^*(N, G) \neq 0$). Then, $N = \min_{\subseteq} \mathcal{K}_y$, and since $y \in G^+$, we must have $N \subseteq G^+$. Consequently, we have $\bigcup_{N \in \mathcal{F}^+(E)} N \subseteq G^+$, and then $\bigcup_{N \in \mathcal{F}^+(E)} N = G^+$ and $\mathcal{F}^+(E) = C[G^+]$. Let $x \in G^- \setminus \bigcup_{N \in \mathcal{F}^-(E)} N$. Then, there exists $N \in \mathcal{K}_x$ such that $N \subseteq G^-$. As $x \notin \bigcup_{N \in \mathcal{F}^-(E)} N$, we have $N \notin \mathcal{F}^-(E)$, and in particular, $n(N, G) \neq 0$. But then, there exists $y \in N$ such that $y \notin G$, and thus, $G^- \not\subseteq G$: contradiction. Consequently, we have $G^- \subseteq \bigcup_{N \in \mathcal{F}^-(E)} N$, and then $G^- = \bigcup_{N \in \mathcal{F}^-(E)} N$ and $\mathcal{F}^-(E) = C[G^-]$. □

The following property immediately derives from Prop. 13.

Property 14 *Let $\sigma \in \{+, -\}$. Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $G \subseteq E$. Then we have*

$$G^\sigma = \bigcup_{N \in \mathcal{F}^\sigma(E)} N \tag{11}$$

Property 15 *Let $\sigma \in \{+, -\}$. Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $G \subseteq E$. Then $C[G^\sigma]$ (and thus G^σ) can be computed with a linear algorithmic complexity $O(\max\{|\mathcal{K}|, |E|\})$, w.r.t. the number of nodes of the tree or the size of the image.*

Proof From the definition of $\mathcal{F}^\sigma(E)$, it is easily proved that each node is processed at most once. For each one of these $O(|\mathcal{K}|)$ processed nodes, one equality (related to $p^*(N, G)$ or $n(N, G)$, which are assumed to be precomputed, see Remark 3) is tested, and the status of the node ("in" or "out of" the result $\mathcal{F}^\sigma(E)$) is possibly modified. These two operations have a constant algorithmic complexity $O(1)$. The whole process then presents a linear complexity $O(|\mathcal{K}|)$. The generation of G^σ from $\mathcal{F}^\sigma(E)$ can be performed by modifying, for each node N of $\mathcal{K}$ and for each point x of N (these points being stored in E_N for each node N, see Remark 3) the status of x to indicate that it belongs to G^σ. This process then presents an algorithmic complexity $O(|E|)$. Hence the result holds. □

General case We now focus on the general case of the problem stated in Sec. 3, which consists of finding a set of nodes $\widehat{\mathcal{K}}$ of the component-tree of an image I verifying Eq. (4), for the pseudo-distance d^α proposed in Def. 4. The purpose is then to find the best compromise (according to a chosen weight $\alpha \in [0, 1]$) between the amount of false positives and false negatives w.r.t. a binary target G.

Since the set Q of the objects which can be generated from the subsets of nodes of a component-tree is finite, there necessarily exists a solution to this problem. Hereafter, we show that such a solution (Def. 16) can be computed in linear time w.r.t. the size (*i.e.*, the number of nodes) of the component-tree of the considered image I or the size of the support E of this image (Props. 19 and 20).

Definition 16 *Let $\alpha \in [0, 1]$. Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $G \subseteq E$. Let $\prec \in \{<, \leq\}$. Let $\mathcal{F}^\alpha : \mathcal{K} \to \mathcal{P}(\mathcal{K})$ and $c^\alpha : \mathcal{K} \to \mathbb{R}^+$ be the functions recursively cross-defined, for all $N \in \mathcal{K}$, by*

$$(\mathcal{F}^\alpha(N), c^\alpha(N)) = \begin{cases} (\{N\}, \alpha.n(N, G)) & \text{if } \alpha.n(N, G) \prec (1 - \alpha).p^*(N, G) + \sum_{N' \in ch(N)} c^\alpha(N') \\ (\bigcup_{N' \in ch(N)} \mathcal{F}^\alpha(N'), (1 - \alpha).p^*(N, G) + \sum_{N' \in ch(N)} c^\alpha(N')) & \text{otherwise} \end{cases} \tag{12}$$

In particular, if $ch(N) = \emptyset$, we have $\bigcup_{N' \in ch(N)} \mathcal{F}^\alpha(N') = \emptyset$ (which guarantees the termination of these recursive definitions), and $\sum_{N' \in ch(N)} c^\alpha(N') = 0$. The function $\mathcal{F}^\alpha$ provides, for any node $N \in \mathcal{K}$, the subset of nodes of the subtree of T having N for root, which enables to generate the set $(G \cap N)^\alpha$ (see Eq. (13), below) for the restriction of the image I to N. The function c^α provides the cost (w.r.t. d^α) of this best solution.

Definition 17 *Let $\alpha \in [0, 1]$. Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let Q be the set of the objects which can be generated from the subsets of nodes of $\mathcal{K}$. Let $G \subseteq E$. We define $G^\alpha \in Q$ as*

$$G^\alpha = \bigcup_{N \in \mathcal{F}^\alpha(E)} N \tag{13}$$

From a reasoning similar to (and actually simpler than) the one of Prop. 13, we have the following result.

Property 18 *Let $\alpha \in [0, 1]$. Let $I \in V^E$ be a grey-level image. Let $G \subseteq E$. Then we have*

$$\mathcal{F}^\alpha(E) = C[G^\alpha] \tag{14}$$

Property 19 *Let $\alpha \in [0, 1]$. Let $I \in V^E$ be a grey-level image. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let Q be the set of the objects which can be generated from the subsets of nodes of $\mathcal{K}$. Let $G \subseteq E$. Then, we have*

$$d^\alpha(G^\alpha, G) = c^\alpha(E) = \min_{Q \in Q}\{d^\alpha(Q, G)\} \tag{15}$$

Proof Let us suppose that $ch(E) = \emptyset$. Then we have $Q = \{\emptyset, E\}$, $d^\alpha(\emptyset, G) = (1 - \alpha).p(E, G)$ and $d^\alpha(E, G) = \alpha.n(E, G)$. If $\alpha.n(E, G) \prec (1 - \alpha).p^*(E, G) + \sum_{N \in ch(E)} c^\alpha(N)$, *i.e.*, if $\alpha.n(E, G) \prec (1 - \alpha).p(E, G)$, then $\mathcal{F}^\alpha(E) = \{E\}$, $c^\alpha(E) = \alpha.n(E)$ and thus we have $d^\alpha(G^\alpha, G) = d^\alpha(E, G) = c^\alpha(E) = \min_{Q \in Q}\{d^\alpha(Q, G)\}$. If $\alpha.n(E, G) \not\prec (1 - \alpha).p(E, G)$, then we have $\mathcal{F}^\alpha(E) = \emptyset$, $c^\alpha(E) = (1 - \alpha).p(E, G) = (1 - \alpha).p^*(E, G)$ and thus, $d^\alpha(G^\alpha, G) = d^\alpha(\emptyset, G) = c^\alpha(E) = \min_{Q \in Q}\{d^\alpha(Q, G)\}$. Consequently, the property is true whenever $ch(E) = \emptyset$. Let us now suppose that $ch(E) \neq \emptyset$ and that the property holds for any $N \in ch(E)$ (w.r.t. $I_{|N}$, T_N and $G \cap N$, instead of I, T and G, see Prop. 5). Note that

$\min_{Q \in Q}\{d^\alpha(Q, G)\} = \min\{d^\alpha(E, G), \min_{Q \in Q \setminus \{E\}}\{d^\alpha(Q, G)\}\}$, while $d^\alpha(E, G) = \alpha.|E \setminus G| = \alpha.n(E)$, and $\min_{Q \in Q \setminus \{E\}}\{d^\alpha(Q, G)\} = \min_{Q \in Q \setminus \{E\}} \alpha.|Q \setminus G| + (1 - \alpha).|G \setminus Q|$. Note also that $\{Q \cap N\}_{N \in ch(E)}$ is a partition of Q whenever $Q \neq E$ while $\{G \setminus \bigcup_{N \in ch(E)} N\} \cup \{G \cap N\}_{N \in ch(E)}$ is a partition of G (by omitting the possibly empty subsets). If $Q \neq E$, we have $d^\alpha(Q, G) = \alpha.|Q \setminus G| + (1 - \alpha).|G \setminus Q| = \alpha.|\bigcup_{N \in ch(E)}(Q \cap N) \setminus G| + (1 - \alpha).|((G \setminus \bigcup_{N \in ch(E)} N) \cup \bigcup_{N \in ch(E)}(G \cap N)) \setminus Q| = \sum_{N \in ch(E)} \alpha.|(Q \cap N) \setminus G| + (1 - \alpha).|(G \setminus \bigcup_{N \in ch(E)} N) \setminus Q| + \sum_{N \in ch(E)}(1 - \alpha).|(G \cap N) \setminus Q| = \sum_{N \in ch(E)}(\alpha.|(Q \cap N) \setminus (G \cap N)| + (1 - \alpha).|(G \cap N) \setminus (Q \cap N)|) + (1 - \alpha).|G \setminus \bigcup_{N \in ch(E)} N| = \sum_{N \in ch(E)}(\alpha.|(Q \cap N) \setminus (G \cap N)| + (1 - \alpha).|(G \cap N) \setminus (Q \cap N)|) + (1 - \alpha).p^*(E)$. From the above partition properties, it then comes that $\min_{Q \in Q \setminus \{E\}}\{d^\alpha(Q, G)\} = \min_{Q \in Q \setminus \{E\}}\{\sum_{N \in ch(E)}(\alpha.|(Q \cap N) \setminus (G \cap N)| + (1 - \alpha).|(G \cap N) \setminus (Q \cap N)|) + (1 - \alpha).p^*(E)\} = (1 - \alpha).p^*(E) + \sum_{N \in ch(E)} \min\{\alpha.|(Q \cap N) \setminus (G \cap N)| + (1 - \alpha).|(G \cap N) \setminus (Q \cap N)|\} = (1 - \alpha).p^*(E) + \sum_{N \in ch(E)} d^\alpha(Q \cap N, G \cap N) = (1 - \alpha).p^*(E) + \sum_{N \in ch(E)} c^\alpha(N)$, by induction hypothesis. Consequently, $\min_{Q \in Q}\{d^\alpha(Q, G)\} = \min\{\alpha.n(E), (1 - \alpha).p^*(E) + \sum_{N \in ch(E)} c^\alpha(N)\}$, and the result follows by induction from Def. 16. $\square$

Property 20 *Let $\alpha \in [0, 1]$. Let $I \in V^E$ be a grey-level image. Let $G \subseteq E$. Then $\mathcal{F}^\alpha(E) = C[G^\alpha]$ (and thus G^α) can be computed with an algorithmic complexity $O(\max\{|\mathcal{K}|, |E|\})$, linear w.r.t. the number of nodes of the tree or the size of the image.*

Proof The proof is similar to the proof of Prop. 15. The only difference lies in the fact that the set of conditions to be tested ($\alpha.n(N) \prec (1 - \alpha).p^*(N) + \sum_{N' \in ch(N)} c^\alpha(N')$) requires at most $|\mathcal{K}|$ comparison operations ($\prec$) and $4.|\mathcal{K}|$ arithmetic operations ($., +, -$), while the computation of all the terms $c^\alpha(.)$ involves (at most) the value $c^\alpha(N')$ only once for any $N' \in \mathcal{K}$, leading to less than $|\mathcal{K}|$ additions in the set of all the $\sum$ terms. Such supplementary operations then do not increase the algorithmic complexity $O(|\mathcal{K}|)$ of the computation of $\mathcal{F}^\alpha(E)$ by comparison to $\mathcal{F}^\sigma(E)$. Hence the result holds. $\square$

Remark 21 *The set of nodes $\mathcal{F}^\alpha(E)$ and its associated binary object G^α enable to minimise $d^\alpha(., G)$, and thus to obtain an optimal solution to the issue considered in this work. However, $\mathcal{F}^\alpha(E)$ and G^α are generally not unique. To illustrate this assertion, let us consider the trivial case where $G = \emptyset$ (resp. $G = E$) and $\alpha = 0$ (resp. $\alpha = 1$). Obviously, in such a case, any set of nodes and any associated binary object minimise $d^0(., G)$ (resp. $d^1(., G)$), which is always equal to 0. However, the way to define $\prec$ in Eq. (12) enables to break this non-determinism by choosing to favour the smallest ($<$) or the largest ($\leq$) solution (w.r.t. the inclusion relation $\subseteq$) among all the possible ones. In particular, if $\prec$ is set to $<$ (resp. to $\leq$) we have $\mathcal{F}^+ = \mathcal{F}^0$ (resp. $\mathcal{F}^- = \mathcal{F}^1$) (the easy proof of this assertion is left to the reader).*

5 Algorithmics

From the above study, which provides an answer to the question stated in Sec. 1, we can derive the method described in Alg. 1. (For the sake of readability, this algorithm, which is intrinsically recursive, is described in an iterative fashion.)

Algorithm 1 - Segmentation method

Input
$I : E \rightarrow V$ (image to be segmented);
$G \subseteq E$ (rough segmentation of I)
$\alpha \in [0, 1]$ (weight parameter for false positives/negatives)
$\prec \in \{<, \leq\}$ (order involved in the cost minimisation formula)
Output
$G^{\alpha} \subseteq E$ (final segmentation of I)

Algorithm
1 - *Component-tree computation*
$T = (\mathcal{K}, L, R)$ (component-tree of I)
for all $N \in \mathcal{K}$ **do**
$\quad E_N = N \setminus \bigcup_{N' \in ch(N)} N'$
$\quad p^*(N, G) = |E_N \cap G|$
$\quad n(N, G) = |N \setminus G|$
end for
2 - *Cost minimisation*
for $v = \top$ to $\bot$ **do**
$\quad$ **for all** $N \in \mathcal{K}$ such that $m(N) = v$ **do**
$\quad\quad$ **if** $\alpha.n(N, G) \prec (1 - \alpha).p^*(N, G) + \sum_{N' \in ch(N)} c^{\alpha}(N')$ **then**
$\quad\quad\quad c^{\alpha}(N) = \alpha.n(N, G)$
$\quad\quad\quad \mathcal{F}^{\alpha}(N) = \{N\}$
$\quad\quad$ **else**
$\quad\quad\quad c^{\alpha}(N) = (1 - \alpha).p^*(N, G) + \sum_{N' \in ch(N)} c^{\alpha}(N')$
$\quad\quad\quad \mathcal{F}^{\alpha}(N) = \bigcup_{N' \in ch(N)} \mathcal{F}^{\alpha}(N')$
$\quad\quad$ **end if**
$\quad$ **end for**
**end for**
3 - *Result computation*
$G^{\alpha} = \bigcup_{N \in \mathcal{F}^{\alpha}(E)} N$

In its general form, the method corresponds to Def. 16, which solves the general case considered in Sec. 4.2. In the specific case where $\alpha = 0$ and $\prec = <$ (resp. $\alpha = 1$ and $\prec = \leq$), the method corresponds to Eq. (8) (resp. Eq. (9)) in Def. 12, which solves the specific case of the smallest result including (resp. the largest result included in) the rough segmentation, considered in Sec. 4.2.

6 Application example: Assisted segmentation of drop caps

As established in the previous sections, given an image, a rough segmentation, and a parameter controlling the trade-off between false positives and false negatives, it is possible to compute, in linear time, the best segmentation composed by the connected components stored in the component-tree of the image. Based on Alg. 1, an interactive segmentation software tool has been developed, and applied to the extraction of drop caps textural parts from ancient documents [11]. These drop caps are issued from the
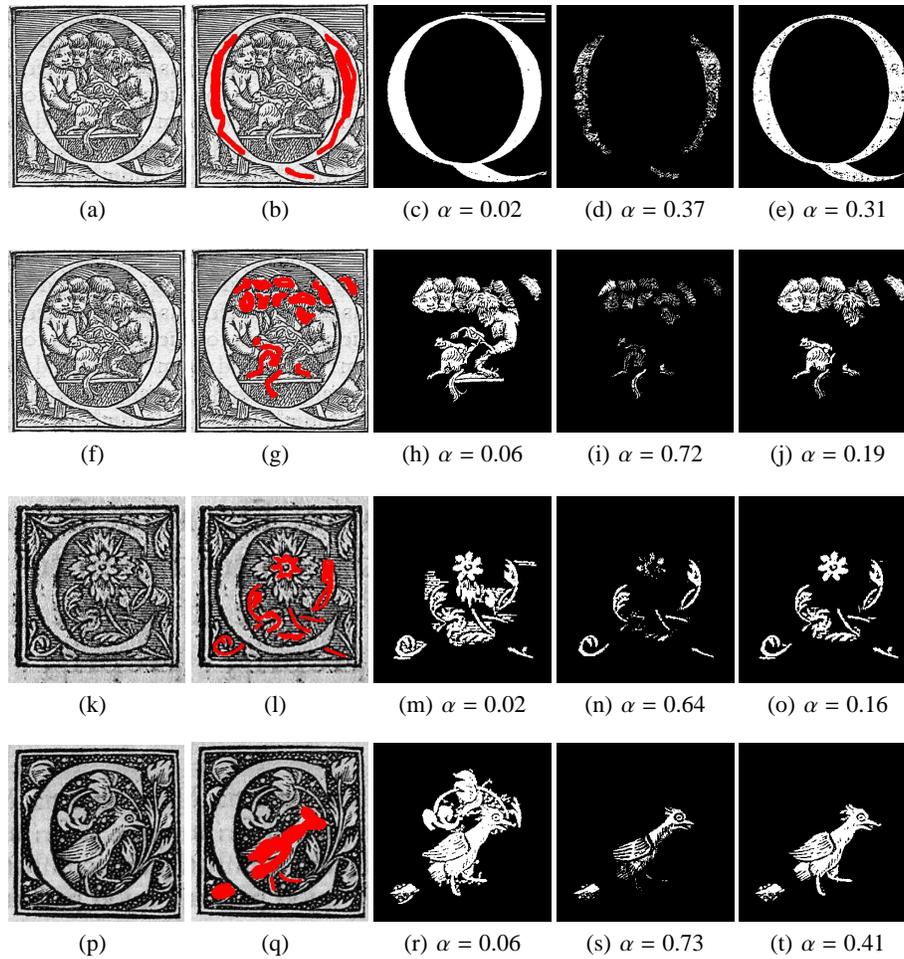
Fig. 2. Segmentation process. First column: initial images. Second column: user-defined rough segmentations (in red). Third to fifth columns: examples of proposed segmentation results for different values of α (fifth column: best obtained segmentation).

Madonne database OLDB (Ornamentals Letters DataBase), which consists of more than 6000 grey-scale graphical decorative initials extracted from archival documents[2]. Drop caps images are composed of a letter (uppercase) part and textural parts. They are noisy and contain artifacts such as superimposed text, coming from neighbouring book pages. The size of these images varies from 150×150 to 750×750 pixels. In this study, we are interested in the fast extraction of objects belonging to the textural (background)

part of the drop cap (these objects are of extremal intensity, which is compliant with the requirements of the method). The components are then used afterwards as query objects in a system for drop caps retrieval and indexation.

The segmentation protocol, exemplified in Fig. 2 is the following one. Given a drop cap image, (first column), the user proposes a manually-defined rough segmentation (second column). He can then choose the most satisfactory segmentation by simply interactively tuning the α value (third to fifth column) between 0 and 1 in a threshold-like fashion. For each chosen α, the segmentation is computed on the fly, in real time.

7 Conclusion

In this article, it has been established that the component-tree structure can be used to compute, in linear time, a binary object which fits at best (w.r.t. false positives/negatives criteria) a given binary target which can be assumed to identify structures of interest in a digital grey-level image.

Based on this result a segmentation method has been proposed and successfully applied to the case of document analysis, emphasising the relevance of the approach.

In a further extended version of this work, it will be shown that this interactive segmentation method can be optimised from both time and space point of views, by establishing in particular the increasing property of the results w.r.t. the α values. Additional applications to medical images will also be proposed.

References

1. Hanusse, P., Guillataud, P.: Sémantique des images par analyse dendronique. In: RFIA 1991. Volume 2. (1991) 577–588
2. Mattes, J., Demongeot, J.: Efficient algorithms to implement the confinement tree. In: DGCI 2000. Volume 1953 of LNCS., Springer (2000) 392–405
3. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Transactions on Image Processing **7**(4) (1998) 555–570
4. Breen, E., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding **64**(3) (1996) 377–389
5. Jones, R.: Connected filtering and segmentation using component trees. Computer Vision and Image Understanding **75**(3) (1999) 215–228
6. Mattes, J., Richard, M., Demongeot, J.: Tree representation for image matching and object recognition. In: DGCI 1999. Volume 1568 of LNCS., Springer (1999) 298–312
7. Ouzounis, G., Wilkinson, M.: Mask-based second-generation connectivity and attribute filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6) (2007) 990–1004
8. Urbach, E., Roerdink, J., Wilkinson, M.: Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(2) (2007) 272–285
9. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Transactions on Image Processing **15**(11) (2006) 3531–3539
10. McGuinness, K., O'Connor, N.: A comparative evaluation of interactive segmentation algorithms. Pattern Recognition **43**(2) (2010) 434–444
11. Naegel, B., Wendling, L.: Combining shape descriptors and component-tree for recognition of ancient graphical drop caps. In: VISAPP 2009. Volume 2. (2009) 297–302