Morphology-Based Hierarchical Representation with Application to Text Segmentation in Natural Images

Lê Duy Huỳn, Yongchao Xu, Thierry Géraud

Morphology-Based Hierarchical Representation with Application to Text Segmentation in Natural Images

Lê Duy Huỳnh, Yongchao Xu, Thierry Géraud †
EPITA Research and Development Laboratory (LRDE)
14-16 rue Voltaire, FR-94270 Le Kremlin-Bicêtre, France
Email: firstname.lastname@lrde.epita.fr

***Abstract*—Many text segmentation methods are elaborate and thus are not suitable to real-time implementation on mobile devices. Having an efficient and effective method, robust to noise, blur, or uneven illumination, is interesting due to the increasing number of mobile applications needing text extraction. We propose a hierarchical image representation, based on the morphological Laplace operator, which is used to give a robust text segmentation. This representation relies on several very sound theoretical tools; its computation eventually translates to a simple labeling algorithm, and for text segmentation and grouping, to an easy tree-based processing. We also show that this method can also be applied to document binarization, with the interesting feature of getting also reverse-video text.**

***Index Terms*—Mathematical Morphology; Tree of Shapes; Non-linear Laplace Operator; Text Segmentation; Document Binarization.**

I. Introduction

With the dramatic increase of images and video acquired with mobile devices, content-based analysis techniques have received a great deal of attention over the last few years; this is in particular the case of text detection. In this paper, we focus on text segmentation, that is, finding candidate components for text characters in natural images.

Text localization methods are often classified into sliding-windows-based and connected-components-based approaches. For the former approaches, a classifier is used to determine if a window contains text, which can be SVM [1], AdaBoost [2], or Convolutional Neural Networks (CNN) [3]. Such methods are relatively expensive due to the number of windows to take into account. The latter approaches consider as text candidates connected components extracted from the image thanks to, e.g., keypoints [4], the Stroke Width Transform (SWT) [5], the Toggle Mapping Morphological Segmentation (TMMS) [6], or the Maximally Stable Extrema Region (MSER) [7]. Actually many methods in ICDAR 2015 "Robust Reading" competition are based on MSER or on Extremal Regions (ER) [8]. Let us mention that a non-window-based text detection method, using Fully Convolutional Networks (FCN) to make pixels-to-pixels text prediction, has just been proposed in [9]. Last, we shall notice that some very recent methods have put the emphasis on getting good run-time performances [10], [4], [11]. For a larger bibliography, the reader can refer to [12], [6], and [13].

We propose a hierarchical image representation based on the morphological Laplace operator, also called morphological Laplacian, with an application that segments text characters (so it is a component-based approach), and groups them into text boxes / lines thanks to two kinds of spatial relations: adjacency and inclusion. The contributions of this paper are the following:

- a hierarchical (i.e., tree-based) representation of the image contents, where adjacency between components is related to inclusion;
- a character segmentation method which is a good trade-off between efficiency (linear time complexity) and quality (with a competitive F-score);
- an efficient grouping of characters into text boxes, taking fully advantage of the tree structure;
- an illustration on another application (document binarization) of the capabilities of the proposed tree-based representation.

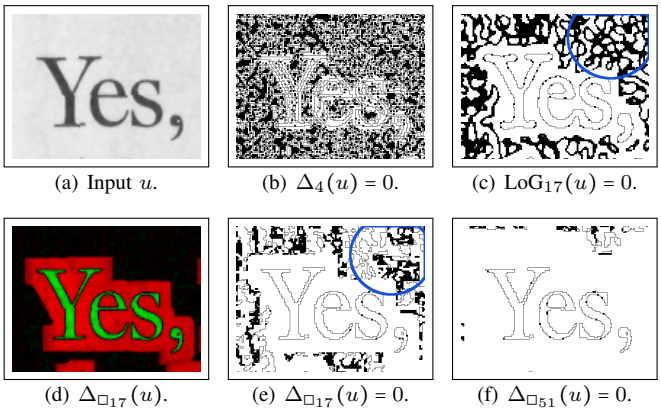
As compared to many methods of the literature, the method that we propose features many properties: it is invariant to contrast inversion (so we also extract reverse-video text without any special processing); it is invariant to contrast change; it is invariant to scale and rotation; and it handles a large variety of scripts (Latin, Hebrew, Chinese, etc.). Note that we only use the hierarchical representation to extract text candidates; although we filter out some irrelevant components, we do not include a false positive elimination step (whereas many methods do). This paper does not contain an evaluation at text-line level of a complete text detection pipeline. Yet we give some quantitative results of text segmentation, which reveals that our method outperforms some widely used text component extraction methods.

In Sec. II we recall the theoretical background in mathematical morphology and digital topology that is used in our approach. In Sec. III we detail the different aspects of our approach: the hierarchical Laplacian-based representation and how we rely on it to segment text. In Sec. IV we proceed to experiments and show that we compete with classical component-based text segmentation methods. Last we conclude and give perspectives in Sec. V.

II. Theoretical Background

This section gives a brief introduction to the theoretical tools involved in the proposed method. The impact of these tools is to obtain strong properties without making the method more complicated, and actually, they allow for simplification.

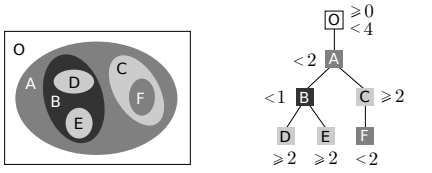
† This work has been conducted in the context of the MOBIDEM project, part of the "Systematic Paris-Region" and "Images & Network" Clusters (France). This project is partially funded by the French Government and its economic development agencies.



(a) Input u. (b) $\Delta_4(u) = 0$. (c) $\mathrm{LoG}_{17}(u) = 0$.

(d) $\Delta_{\square_{17}}(u)$. (e) $\Delta_{\square_{17}}(u) = 0$. (f) $\Delta_{\square_{51}}(u) = 0$.

Fig. 1: Zero-crossing contours of different Laplace operators: (b) and (c) come from classical linear operators; (e) and (f) come from the morphological operators. On (d), the scalar morphological Laplacian is depicted with positive and negative values tinted resp. in green and red.



(a) A simple image. (b) Its tree of shapes.

Fig. 2: Representation of an image by its tree of shapes.

(a) An image. (b) A well-composed interpolation.

Fig. 3: Having a well-composed interpolation (right) of an image (left) ensures that level sets contours (depicted in black) are Jordan curves.

A. Morphological Laplace Operator

In order to detect text in images, many methods first look for candidate regions for characters. A seminal method, based on contour detection, is to consider the 0-crossings of a discrete Laplace operator (denoted by Δ): given a gray-level image u, the contours of interest are given by $\Delta u = u_{xx} + u_{yy} = 0$. This method is interesting for several reasons: 1) it is a very simple approach; 2) it provides closed contours; 3) labeling the components of the image having the same sign, resp. positive and negative, gives a segmentation; 4) it is self-dual, i.e., it processes dark objects and bright ones in the same way.

The simplest discretization of this linear operator relies on a cross-shaped convolution kernel. Yet this elementary operator is very sensitive to noise, so many 0-crossings arise as it can be seen in Figure 1(b). To get rid of this problem, one can rely on a larger kernel, e.g., by considering the approximate given by the Laplacian of Gaussian (LoG) operator. Unfortunately, its smoothing effect alter the localization of contours, as illustrated by Figure 1(c).

An elementary morphological Laplace operator has been defined in [14] by $\Delta_{\mathcal{N}} = (\delta_{\mathcal{N}} - \mathrm{id}) - (id - \varepsilon_{\mathcal{N}})$, relying on the elementary dilation (δ) and erosion (ε) morphological operators. A natural extension of the elementary operator uses a structuring element B to replace the neighborhood $\mathcal{N}$; it is depicted in Figure 1(d) with B being a centered square (denoted by $\square$) of size 17×17. Although it has the same "simplification strength" as the linear LoG version, one can see when comparing the resulting 0-crossings (LoG in Figure 1(c) vs. morphological in Figure 1(e)) that the morphological non-linear version features a much higher fidelity to actual object contours than the linear version. Furthermore, when increasing the size of the structuring element, 0-crossing contours keep a strong fidelity to data, as illustrated in Figure 1(f) with B now being a 51×51 square.

One can also observe that the salient object contours are curiously very stable—they are not altered—when the size of the structuring element (the "morphological kernel") increases. From Figure 1(e) to Figure 1(f), the contours of the "Yes" word remain the same, whereas spurious non-interesting contours disappear. Furthermore, the size of the structuring element B, a 51×51 square, is much larger than the character thickness (note that the input image u has 130×100 pixels). It means that obtaining salient contours thanks to the morphological Laplace operator hardly depends on the size of B. Despite of this great advantage, the morphological Laplacian has been almost never used in the literature [15], [16].

B. Tree of Shapes

The tree of shapes is a morphological self-dual representation of an image; see [17] for history, implementation, and references. This tree encodes the inclusion of the level sets, i.e., the connected components obtained by thresholding. An illustration is given on a very simple image by Figure 2.

Given an image $u : X \to \mathbb{Z}$ and any scalar $\lambda \in \mathbb{Z}$, the lower level sets are defined as $[u < \lambda] = \{x \in X \mid u(x) < \lambda\}$, and the upper level sets as $[u \geq \lambda] = \{x \in X \mid u(x) \geq \lambda\}$, with $X = \mathbb{Z}^2$ for a 2D image. Considering the connected components of these sets (obtained by the $\mathcal{CC}$ operator), and using the cavity-fill-in operator (denoted by Sat), the tree of shapes of an image u is defined by: $\mathfrak{S}(u) = \{\mathrm{Sat}(\Gamma) \mid \Gamma \in \mathcal{CC}([u < \lambda]) \cup \mathcal{CC}([u \geq \lambda])\}_\lambda$. Such a tree is called self-dual since many self-dual operators can be derived from this tree [18].

C. Well-Composed Sets and Maps

A sub-class of sets defined on the cubical grid, called well-composed, has been proposed in [19], where all connectivities are equivalent, thus avoiding many topological problems. This notion of well-composedness has been extended in [20] from sets to functions: a gray-level image u is well-composed if any set $[u \geq \lambda]$ is well-composed. A straightforward characterization of well-composed gray-level images is that every 2×2 block of pixels values $(a, d)(c, b)$ should verify: $\mathrm{intvl}(a, b) \cap \mathrm{intvl}(c, d) \neq \varnothing$, where $\mathrm{intvl}(v, w) = [\![\min(v, w), \max(v, w)]\!]$.

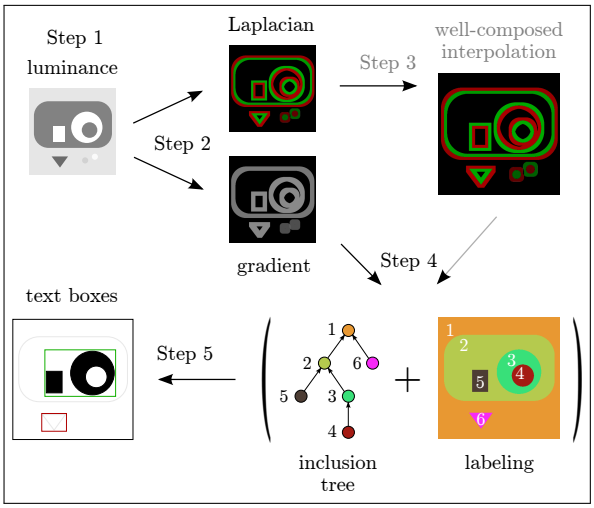


Fig. 4: Overview of the proposed method: hierarchical representation (Steps 1 to 4) and application (Step 5).

Well-composed gray-level images are interesting because every contour of any level set component is a Jordan curve.

To get a well-composed image from a primary image, one can compute an interpolation of the primary image that is well-composed. Figure 3 gives an example of an image which is not well-composed, but whose interpolation is well-composed. In [21], the authors have proposed a very simple method to get a self-dual interpolation, that makes sense when considering the tree of shapes of an image.

III. Description of the Proposed Method

In this section, we focus both on the proposed morphological hierarchical representation and how it can be used to segment text in images.

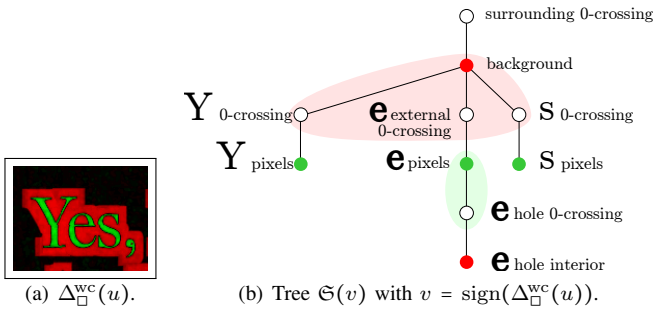
A. Method Overview

The method that we propose to segment text in natural images is very simple; put very shortly, text components are selected among the 0-crossing lines of the Laplace operator of the gray-level input image. Yet, the very "classical" scheme of considering the 0-crossings obtained with the linear operator as object contours does not work well in practice (see Figures 1(b) and 1(c)). So let us explain what the different steps of our method are, and let us give the reasons why their combination (depicted in Figure 4) is effective.

The four first steps aim at computing a hierarchical representation of the input image, and the final step is an example of how this hierarchical representation can serve to segment text and extract text lines.

Step 1. We first start by taking the luminance of the input color image. With the resulting gray-level image, we have lost color information but we observed in our experiments that it almost never negatively affects text retrieval.

Step 2. With a 11×11 square structuring element, we compute its morphological dilation $\delta_\square$ and erosion $\varepsilon_\square$ to directly deduce two images. First we get the morphological gradient $\nabla_\square = \delta_\square - \varepsilon_\square$, which is a scalar thick gradient, used later to discard contours that are not enough contrasted. Second we get the morphological Laplace operator $\Delta_\square = \delta_\square + \varepsilon_\square - 2\,\mathrm{id}$.



(a) $\Delta^{\mathrm{wc}}_\square(u)$. (b) Tree $\mathfrak{S}(v)$ with $v = \mathrm{sign}(\Delta^{\mathrm{wc}}_\square(u))$.

Fig. 5: The inclusion tree is the tree of shapes of the Laplacian sign image: positive and negative regions are respectively green and red nodes of the ToS, and 0-crossing regions are white nodes.

The text character boundaries are expected to belong to the 0-crossing contours of this non-linear operator; actually they are, and their localization is precise (as shown later in Table I).

Step 3. We compute a particular interpolated image, $\Delta^{\mathrm{wc}}_\square$, of the Laplacian image $\Delta_\square$, having 4 times more pixels than the original. (We will see in Sec. III-D that this step can be emulated, so it is cost-free.) This resulting image is well-composed, meaning that the boundaries of every components of any threshold set are Jordan curves. Thus the 0-crossings (precisely the external 1D contours of the 0-crossings) are simple closed curved, so they cannot have the shape of a '8'. As a consequence, they are disjoint, and this set of curves can be organized in an inclusion tree. That contrasts with the 0-crossings depicted in Figure 1, where the Laplacian images are not well-composed; it is especially visible within the blue circles in Figures 1(c) and 1(e), where we cannot say what are the inclusion relationships between (white) regions.

Step 4. Due to the fact that $\Delta^{\mathrm{wc}}_\square$ is well-composed, the regions delimited by the 0-crossings can be labeled very efficiently (by the classical blob labeling algorithm), and their inclusion tree is built. In addition, many 0-crossings are discarded on the fly during the labeling, because they are not contrasted enough (we use $\nabla_\square$), or because they do no satisfy some geometrical criteria (when they are too small for instance). The resulting "tree + label image" are depicted on the bottom-right part of Figure 4.

This 4th step can be seen as the computation of the tree of shapes of the "sign of the Laplacian" image, as depicted by Figure 5. This image, $v = \mathrm{sign}(\Delta^{\mathrm{wc}}_\square(u))$, is a ternary-valued image (with pixels set to -1, 0, or 1). To compute its tree of shapes, we run the classical queue-based "blob labeling" algorithm, which is very efficient. (Note that actually we do not want regions representing null values in the final tree, so we group nodes corresponding to 0-crossings with their parent; it is displayed by the light colored backgrounds in Fig. 5(b).)

Step 5 (application). Last we group components together to form text boxes. For that, we only consider the bottom of this tree (the leaves and sometimes their parent): for each component, we search spatially in the label image what are their left and right components to be grouped into a text box. In this step, we highly take advantage of the tree structure: it allows very easily to discard many regions as non-text, and

```
1  LABELING(Δ□^wc, ∇□)
2  for all p do label(p) ← 0, isContour(p) ← false
3  ℓ ← 0
4  for all p do
5      if label(p) ≠ 0 then continue;
6      (parent, ℓ′, ℓ) ← CONTOURIZE(ℓ, p, ∇□, isContour);
7      label(p) ← ℓ′, Q.push(p);
8      while not Q.is_empty() do
9          q ← Q.pop();
10         for all n ∈ N(q) do
11             if label(n) = 0 and Δ□^wc(p) × Δ□^wc(n) ≥ 0 then
12                 label(n) ← ℓ′, Q.push(n);
13             else isContour(n) ← true;
14 return (parent, label)
```

Algorithm 1: Blob labeling and tree computation.

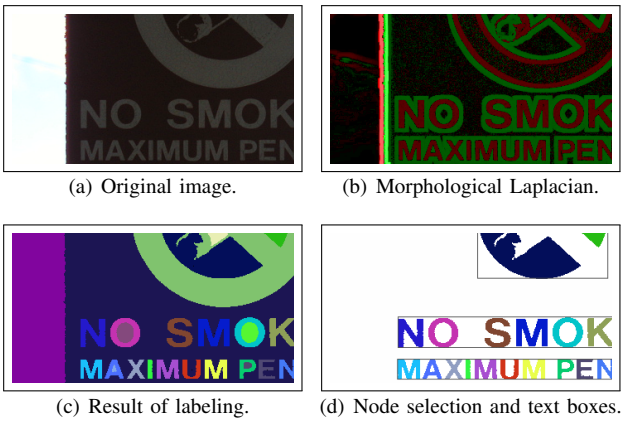
to determine if a leaf region is a character hole or a plain character.

The key features of this method are the following: **1.** It runs very fast since the processing chain is very simple and since all operations have a linear time complexity (see Sec. III-E); **2.** The proposed method, based on the morphological Laplace operator, outperforms more "classical" component-based methods that select candidate regions for characters (see Sec. IV); **3.** The fact that regions form an inclusion tree, thanks to the well-composedness property, allows for powerful decision taking when grouping regions into text boxes.

B. Computing the Hierarchical Representation (Step 4)

The algorithm of the labeling process and tree creation, Step 4 in Fig. 4, is depicted in Algo. 1. '*label*' is the label image to compute, '*parent*' is an array encoding the parenthood relationship of the inclusion tree (e.g., having $parent(l_1) = l_2$ means the region with label l_1 is included in the region with label l_2), Q is a queue of pixels, *isContour* is an auxiliary binary image, and ℓ is the current label.

We browse the pixels in raster scan order (main loop, line 4). When we reach an unlabeled pixel p, we follow the contour of the unlabeled region, which is a hole in the label image, thanks to the *isContour* image. This is performed in the CONTOURIZE routine, line 6. The routine CONTOURIZE computes on the fly the bounding box of the hole and the average of gradient's magnitude of its contour. If the bounding box is too tiny or if the gradient magnitude is very low (meaning respectively that a 0-crossing contour is due to noise or occurs in a flat region), we will not create a new node / a new label for this hole region. With p_{-1} being the pixel just before p in the raster scan order (p_{-1} is thus guaranteed to be labeled), the routine CONTOURIZE performs: $\ell' \leftarrow \ell + 1$ and $parent(\ell') \leftarrow label(p_{-1})$ if the hole region is a new region, otherwise $\ell' \leftarrow label(p_{-1})$ and this spurious region will be ignored (because merged with its parent). Eventually ℓ' is the label value to label the considered (yet unlabeled) hole region. For that, we initialize a queue-based propagation, line 7, and we proceed to the classical "blob labeling" algorithm, lines 8 to 13. The blob/region to be labeled is characterized by the connected set of pixels having the same Laplacian sign as p or being null (Cf. Fig. 5). During this propagation, we update the auxiliary *isContour* image (line 13) to memorize the contours of the holes included in this region.



(a) Original image. (b) Morphological Laplacian. (c) Result of labeling. (d) Node selection and text boxes.

Fig. 6: Illustration of the proposed method: mathematical morphology tools are contrast-invariant so we successfully deal with low-contrasted data (note that the Laplacian image (b) has been lightened to be readable).

C. Application: Grouping Components into Text Lines (Step5)

The leaves (and sometimes their parents) of the resulting tree are then grouped together to form text line candidates (Step 5 in Figure 4). We only consider roughly horizontal words, containing at least two characters. Thanks to the Laplacian 0-crossings inclusion, the characters of a same word belong to the same background; this implies that they have the same parent in the tree structure. As a consequence, the grouping process can be performed efficiently: the only candidate regions for characters to be grouped into text lines are siblings in the tree structure. Starting from each tree leaf, we thus use a classical search in the image space to group siblings (some additional geometric information such as region height and maximal inter-distance between regions are also used to control the grouping process). Note that we know when a leaf is a character hole, because both left and right neighbor regions are its grand-parent in the inclusion tree (the background region being the parent); we then consider its parent node (its background being the character). An illustration is given in Figure 6.

D. Space Optimization

A drawback of this method seems to be the need of multiplying by 4 the number of pixels (when computing the well-composed interpolation of the Laplacian image, and then proceeding to the labeling, and the final grouping). Actually we do **not** need to duplicate the number of pixels: we just can do as if there were an interpolation. Indeed we can emulate that the Laplacian image is well-composed, during the labeling and the contour browsing. Considering Fig. 4, it means that Step 3 is useless and that all the processing chain is performed on images having the same size as the input image.

E. Complexity Analysis

The morphological gradient and Laplacian rely on a dilation and erosion using a square structuring element, which can be efficiently implemented thanks to a 2-pass (horizontal then vertical) incremental (heap-based) process. In addition this

(a) Some results.

(b) A case of failure.

Fig. 7: Qualitative results using "ICDAR 2015 Robust Reading" DB: input (left), labeling (middle), final boxes (right).

TABLE I: Text segmentation comparison.

Method	Recall	Precision	F-score	Consistency
SWT [22]	0.464192	0.8861	0.609232	0.505042
ER [11]	0.613059	0.892023	0.629221	0.726689
TMMS [6]	**0.784568**	0.7522	**0.768043**	0.791303
Our	0.636168	**0.933058**	**0.756528**	**0.849754**

local process is easily parallelizable, and eventually it has a linear time complexity w.r.t. the number of pixels. The blob labeling process (see Algo. 1) has also a linear time complexity: every pixels are only visited once with the main 'for' loop and the queue-based propagation, and browsing the 0-crossings contours is also limited by the number of pixels. Last the grouping process, dealing with very few nodes of the tree and browsing a few pixels of the label image, is trivially linear.

IV. Experimental Results

A. Quantitative Results on Text Segmentation

We have evaluated the proposed method of text segmentation in the context of task 2 of Challenge 2 in ICDAR 2015 "Robust Reading" competition. The dataset contains 233 natural images with focused scene texts. The ground truth of text segmentation results is available. Some qualitative results are given by Figure 7; they include reverse-video, uneven illumination, fancy fonts, blur, and different text sizes.

We have compared our method with three popular methods for generating text candidate regions: Stroke Width Transform (SWT) [22] with the implementation provided by https://

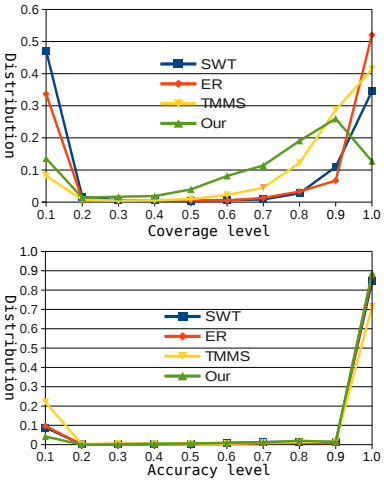


Fig. 8: Evaluation based on coverage and accuracy [23].

sites.google.com/site/roboticssaurav/strokewidthnokia, text detection based on Extremal Regions (ER) [7], [11] (implemented in OpenCV), and Toggle Mapping Morphological Segmentation (TMMS) [6]. The first two methods are widely used as the first step of many state-of-the-art pipelines. For fair comparison, we compare the performance of text candidate region generation of the four methods, that is text segmentation, and we discard the rest of the pipeline (mainly false positive removal). For that, we only consider generated regions that touch the ground truth (GT) texts. We use the evaluation scheme proposed by [24], [25], based on the recall and precision scores in terms of pixels; we also compute a consistency value measuring how much ground-truth text components are split into several pieces. The results are given by Table I; our method achieves a competitive recall with a high precision. Figure 8 depicts in detail how each method behaves w.r.t. all the ground-truth texts in the dataset: the plots illustrate the distribution of the segmented text components at different coverage level (left) and accuracy level (right). The coverage (resp. accuracy) represents the percentage of the matched surface between the GT and a detection object with respect to the GT (resp. detection) surface; see [23] for details. One can see that our method covers the ground-truth texts at relatively high coverage levels (mostly distributed between 50% to 100%), which is not the case of the other methods.

B. Qualitative Interpretation of Results

Many methods extract objects thanks to what can be interpreted as "local thresholding". For instance, for Sauvola's binarization approach (see [26] for a multiscale version), the thresholds computed at pixel-level do not vary a lot for the same object (given than the window used to compute thresholds has to be large). Another examples are the MSER and ER approaches [7], where objects come from the image level sets. As acknowledged in [4], there is only a partial match between the boundaries of actual objects and the contours of level sets (called level lines). Due to the presence of uneven illumination, these classical approaches and many others just fail to properly detect the object boundaries. Conversely, these boundaries belong to the 0-crossings of the morphological

(a) Original image. (b) Self-dual binarization.

Fig. 9: Text segmentation with our method can be seen as a binarization technique, providing also reverse-video text.

Laplace operator. Last, let us recall that we cannot have T-junctions in 0-crossings. As a consequence, when an artifact interferes with text characters, that can lead to the case of failure of our method as depicted in Figure 7 (b).

C. Applying the Method to Document Binarization

The method proposed in this paper has been applied to binarize documents in the challenge #2 (Smartphone OCR) of "ICDAR 2015 Competition on Smartphone Document Capture and OCR (SmartDoc)" [27]. We ranked 2nd in this competition among 8 contestants, with a character accuracy of 95.85% (note that the winner has taken advantage of the redundancy of documents in the test set to correct each individual document). Relying on the Laplace operator has turned out to be robust for the binarization of both blurred text and low-contrasted text. In addition, our method is self-dual since it naturally handles the same way the case of dark objects over bright background and the opposite case, as depicted in Figure 9.

V. Conclusion

In this paper we have presented a hierarchical representation of the image contents based on the inclusion of the 0-crossings of the morphological Laplace operator. Thanks to the well-composedness property, we guarantee that this tree-based representation exists, and we have given an algorithm with linear time complexity to compute this representation (from our first experiments with an ordinary computer, it runs in about 0.2s on a 1M Pixel image). We have explained how to rely on this representation to segment text lines in natural images, and we have shown that it competes with classical methods of text candidate extraction. We also have applied an about similar scheme to document image binarization, which has been used in the ICDAR 2015 SmartDoc competition. As a perspective, we intend to integrate our text segmentation approach in a text detection pipeline—thus including false positives detection—to get an end-to-end evaluation. Another perspective is to make the component grouping step be able to handle multi-oriented text, such as in [6]. In addition we plan to study quantitatively the robustness of the morphological Laplace operator when involved in various applications.

Acknowledgment

The authors want to thank Jonathan Fabrizio and Ana Stefania Calarasanu for their help, and the reviewers for their comments.

References

[1] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D.J. Wu, and A.Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR)*, 2011, pp. 440–445.

[2] J.J. Lee, P.H. Lee, S.W. Lee, A. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *Proc. of ICDAR*, 2011, pp. 429–434.

[3] T. Wang, D.J. Wu, A. Coates, and A.Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. of the Intl. Conf. on Pattern Recognition (ICPR)*, 2012, pp. 3304–3308.

[4] M. Bušta, L. Neumann, and J. Matas, "FASText: Efficient unconstrained scene text detector," in *Proc. of ICCV*, 2015, pp. 1206–1214.

[5] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2963–2970.

[6] J. Fabrizio, M. Robert-Seidowsky, S. Dubuisson, S. Calarasanu, and R. Boissel, "TextCatcher: A method to detect curved and challenging text in natural scenes," *IJDAR*, pp. 1–19, 2016.

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[8] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. of ICDAR*, 2015, pp. 1156–1160.

[9] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. of CVPR*, 2016, pp. 4159–4167.

[10] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2558–2567.

[11] L. Neumann and J. Matas, "Real-time lexicon-free scene text localization and recognition," *IEEE Trans. on PAMI*, vol. 38, no. 9, pp. 1872–1885, 2016.

[12] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, 2015.

[13] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.

[14] L.J. van Vliet, I.T. Young, and G.L. Beckers, "An edge detection model based on non-linear laplace filtering," in *Proc. of PRAI*, 1988, pp. 63–73.

[15] P. Soille, Ed., *Morphological Image Analysis—Principles and Applications*, Springer-Verlag, 2nd edition, 2004.

[16] L. Najman and H. Talbot, Eds., *Mathematical Morphology—From Theory to Applications*, ISTE Ltd and John Wiley & Sons Inc, 2010.

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

[18] Y. Xu, T. Géraud, and L. Najman, "Connected filtering on tree-based shape-spaces," *IEEE Trans. on PAMI*, vol. 38, no. 6, pp. 1126–1140, 2016.

[19] L.J. Latecki, U. Eckhardt, and A. Rosenfeld, "Well-composed sets," *Comp. Vis. and Image Understanding*, vol. 61, no. 1, pp. 70–83, 1995.

[20] L.J. Latecki, "3D well-composed pictures," *Graphical Models and Image Processing*, vol. 59, no. 3, pp. 164—-172, 1997.

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

[22] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2963–2970.

[23] S. Calarasanu, J. Fabrizio, and S. Dubuisson, "Using histogram representation and Earth Mover's Distance as an evaluation tool for text detection," in *Proc. of ICDAR*, 2015, pp. 221–225.

[24] S. Calarasanu, J. Fabrizio, and S. Dubuisson, "What is a good evaluation protocol for text localization systems? Concerns, arguments, comparisons and solutions," *Ima. Vis. Comp.*, vol. 46, pp. 1–17, 2016.

[25] S. Calarasanu, J. Fabrizio, and S. Dubuisson, "From text detection to text segmentation: a unified evaluation scheme," in *Proc. of Intl. Workshop on Robust Reading (IWRR, in conjunction with ECCV)*, 2016, available online.

[26] G. Lazzara and T. Géraud, "Efficient multiscale Sauvola's binarization," *IJDAR*, vol. 17, no. 2, pp. 105–123, 2014.

[27] J.C. Burie and J. Chazalon *et al.*, "ICDAR2015 competition on smartphone document capture and OCR (SmartDoc)," in *Proc. of ICDAR*, 2015, pp. 1161–1165.