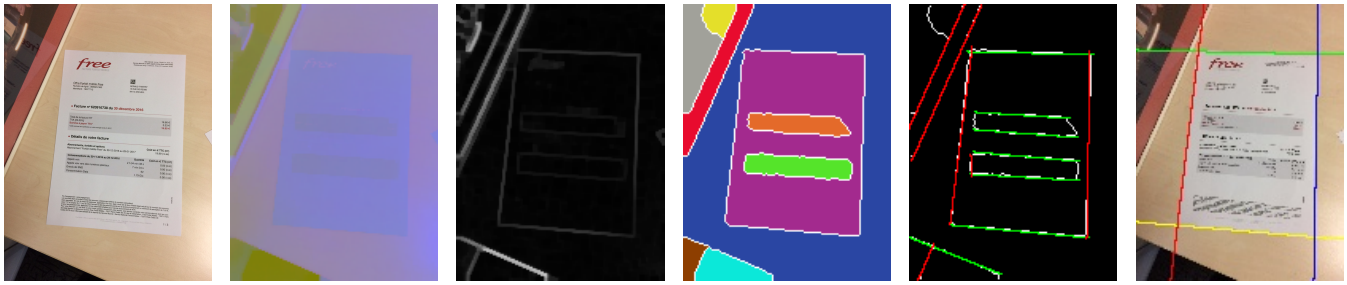
REAL-TIME DOCUMENT DETECTION IN SMARTPHONE VIDEOS

Élodie Puybareau, Thierry Géraud †

EPITA Research and Development Laboratory (LRDE)
14-16, rue Voltaire
F-94270, Le Kremlin-Bicêtre, France



(a) Input. (b) La^*b^* filtered. (c) Gradient. (d) Basins. (e) Chunks. (f) Decision.

Fig. 1: Main steps of the document detection method.

ABSTRACT

Smartphones are more and more used to capture photos of any kind of important documents in many different situations, yielding to new image processing needs. One of these is the ability of detecting documents in real time on smartphones' video stream while being robust to classical defects such as low contrast, fuzzy images, flares, shadows, etc. This feature is interesting to help the user to capture his document in the best conditions and to guide this capture (evaluating appropriate distance, centering and tilt). In this paper we propose a solution to detect in real time documents taking very few assumptions concerning their contents and background. This method is based on morphological operators which contrasts with classical line detectors or gradient based thresholds. The use of such invariant operators makes our method robust to the defects encountered in video stream and suitable for real time document detection on smartphones.

***Index Terms*—** Image processing, Document detection, Mathematical morphology, Real-time video processing.

1. INTRODUCTION

The increasing use of dematerializing process raises new problematic for image processing tools. Especially, the ability of taking a picture with a phone of an important document and uploading it or store it is now unavoidable. Studies addresses the issue of mobile-captured documents [1, 2] and challenges such as in the annual "SmartDoc" competition at ICDAR aim at improving this feature and databases are easily available [3]. The first step of this application is the ability of correctly detect the presence of a document in the camera video stream, in real time. It can easily be tricky due to lack of light, presence of noise or objects, flare and shadows etc. Some attempts already exist for real time document detection on mobile devices [4, 5] but suffer from limitations due to hard constraints, or are not adapted to every kind of documents.

In the field of document image processing and analysis, mathematical morphology [6, 7, 8] is useful and can give effective results. This is the case for filtering with connected operators [9] or for document detection with a tree-based representation [10] (in the context of the "SmartDoc" competition at ICDAR 2015 [11] where this method reaches 1st place). We have chosen an approach based on mathematical morphology tools for several reasons. First, we want to be robust to the presence of noise, that is prominent in the context of videos from mobile device. Second, we want to be rather insensitive with respect to the image contrast (the dynamics of colors in some videos can be very low when the videos acquired in a low-light environment, or when the shadow of the user is cast on the document). Third, we want a "generic" method, since we have no information about the document contents and about the background. Mathematical morphology tools help classical line detectors to recover lines and so document boundaries candidates. A decision has then to be taken to identify the real boundaries. The method described in this document aims at detecting the boundaries of a document in every frame of a video acquired by a smartphone or a tablet in real time. The key

† 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.

features of this method are that we rely on information gathered from both regions and contours; we take very few assumptions about the document; we are robust to major defects such as noise, low-light, shadows, specular blooms (flares), defocus, and motion blur and we detect documents in real-time.

2. METHOD DESCRIPTION

The four main steps of the method are the following:

– reduce size and change of color space (Fig. 1b, Sec. 2.1);
– segment the image into regions (Fig. 1d, Sec. 2.2);
– extract line chunks from region contours (Fig. 1e, Sec. 2.3);
– find the document boundaries (Fig. 1f, Sec. 2.4).

In the following, a square-shaped structuring element of size $s \times s$ is denoted by $B_{\square s}$. We developped out method using home-made tools and OpenCV 3 [12].

2.1. Pre-Processing

The first step is rather trivial. For a detection method to run in real-time on a smartphone, we need to reduce the amount of data to process. From a classical smartphone video input such as in Fig. 1a (with 1280×720 frames), we reduce each frame to an image of about 180×100 pixels (with a linear interpolation). We then convert this image to La^*b^* space, which is known to better map the human psychovisual distances between colors. In addition, the components in this space are very convenient to process, since it separates the luminance from chrominance information. We then process the La^*b^* image: a morphological closing with $B_{\square 7}$ is applied on the luminance (L component), and a morphological erosion with $B_{\square 3}$ is applied on the 'a' component. These two operators regularize the image, and partly remove the text contained in the document, as it can be seen in Fig. 1b. Remark that the text looks actually like a texture since the size of the input image has been drastically reduced.

2.2. Segmentation into Regions

The aim of this step is to obtain a segmentation of the image into regions, i.e., an image of labels with one distinct label per region. The separation of regions is materialized by a set of pixels, which corresponds to all region contours.

Gradient. On each component of La^*b^*, we compute the morphological thick gradient, difference between a dilation and an erosion $\nabla = \delta - \varepsilon$ with $B_{\square 3}$, which actually is a "magnitude" of gradient so a scalar image. The three gradients are then summed up together: $\nabla = \nabla(L) + \nabla(a^*) + \nabla(b^*)$. It is depicted in Fig. 1c. Last, we apply a morphological closing, also with $B_{\square 3}$, so that very small minima are removed. We thus filter out spurious regional minima whose shape is included in a 3×3 square, and those whose dynamics [13] is lower than 3. This operation is important to reduce the number of basins (regions) obtained by the watershed algorithm. Visually, there is no noticeable difference in the gradient image before and after applying the closing operator.

Watershed. A watershed transform is applied on the filtered gradient. We have chosen to obtain a "thick" watershed, that is, a watershed line defined as a set of pixels. We run the Meyer's algorithm based on a priority queue [14] (see [15] to get a deeper insight about this transform). The resulting watershed image is a label image, meaning that a pixel value is an integer representing a region; pixels of the watershed line (contours of regions) have one particular label, and every region has a different label.

2.3. Set of Segments Extraction

The result of the watershed transform is a mere image, meaning that the contours are not primitives (such as mathematical segments). From the contours in the watershed image, we want to obtain two sets of segments, i.e., line chunks: one set of rather horizontal segments, and one set of rather vertical ones. If the document is not so close to the camera, it appears very small in the image, so finding its boundaries requires to choose a rather permissive segment detector and consider all segments, even small ones. In addition, if we want to detect documents that does not appear about aligned with the image borders, we cannot discard diagonal segments. Eventually, we rely on the classical Hough transform [16] (replacable by its probabilistic version [17] or by the line segment detector [18]), which is illustrated in Fig. 1e. To obtain a set of segments, we have several steps described hereafter.

Hough transform and chunks On the binary watershed line image (contours are white on a black background), we run the classical Hough transform. The output of this transform is a set of lines, traversing the image, so they do not have endpoints. Hough lines are cut into chunks, keeping only the closest pixels to the watershed line. During this step, we also compute and store for every chunk its length l, its mean distance d to the watershed line (average distance of all its pixels), its orientation: horizontal (actually "rather horizontal", that is, with a slope between $-45°$ and $45°$) or vertical (otherwise); the orientation is displayed respectively in green and red in Fig. 1e, the list of regions (watershed basins) present in both sides of the chunk and the variation of saturation Δs from one side of the chunk to the other side. To compute the two last items, a distinction is made between horizontal and vertical chunks. The labels of the regions separated by the watershed line are read in the watershed images at row (resp. column) ± 2 for horizontal (resp. vertical) case.

Chunk selection. Since the Hough transform can produce several lines to represent the same aligned part of the watershed line, we often end up with several chunks at the same place. The aim of this final step is to remove redundant chunks, while retaining the "best" one. For that, we first sort the chunks with an increasing energy U defined as:

$$U = d/\sqrt{l+1}, \tag{1}$$

A low value of U means that a chunk is both long and close to the watershed line. Then, for each chunk, if there is no "similar" chunk already selected, select the current chunk. The similarity criterion of two chunks is based on the inclusion degree of their dilated. The result is depicted in Fig. 1e.

2.4. Identification of Document Boundaries

In Fig. 2, we can see that a document can be over-segmented at the end of the previous step; it is then composed of several

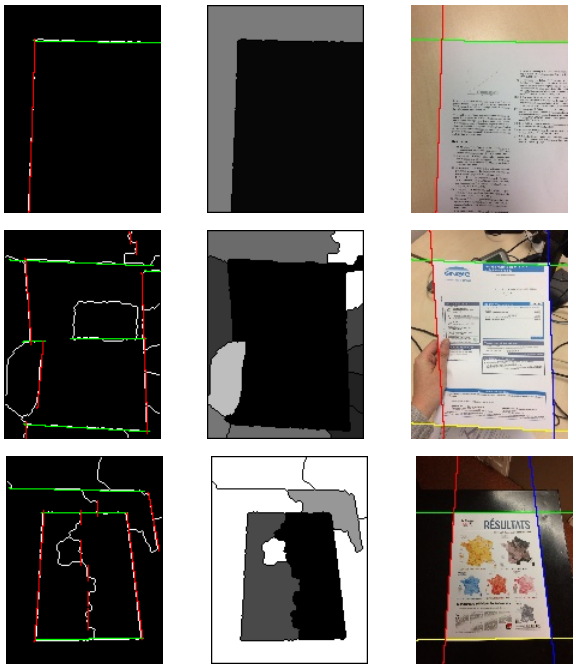
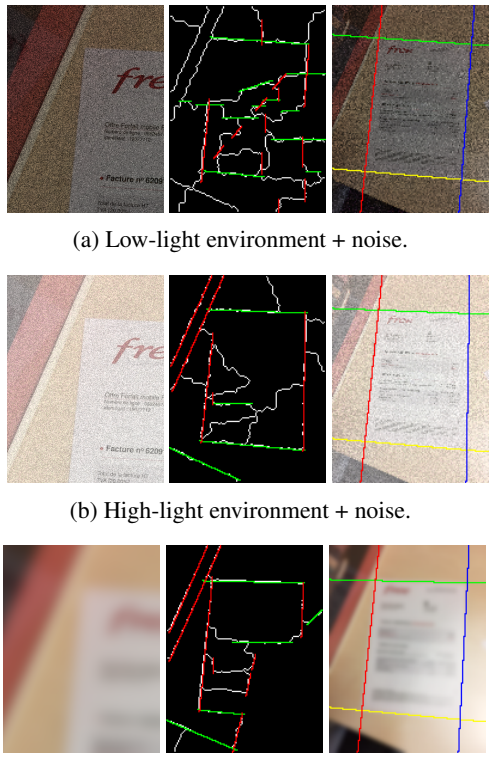


Fig. 2: Some results (the middle column depicts the average saturation of the basins).

basins / regions. We thus cannot just select as the document the "best" region; we need a decision process to recognize the appropriate segments that form the document boundary. From the previous step, we have selected a set of horizontal and vertical chunks that can be good candidates to boundaries. The idea we propose is to find the most "relevant" consistent path, made of segments.

Four categories of segments. First we split the two categories of segments into four categories: top, bottom, left and right. Assuming that the document color is less saturated than the background, and using the variation of saturation Δs from one side of the segment to the other one (computed during the segments extraction step, see Section 2.3), we can decide that an horizontal (resp. vertical) segment is a top segment if $\Delta s < 0$ (resp.left) or bottom segment if $\Delta s > 0$ (resp. right). For instance, in the case of the middle row in Fig. 2, there are three long vertical segments (in red). $\Delta s < 0$ for the two left ones (higher on left then on the right) and $\Delta s > 0$ for the right one (lower on left than on the right).

Getting pairs and sequences. From these four categories, we extract lists of potential pairs of segments: left-top, top-right, right-bottom, and bottom-left. We try all combinations under two restrictions: **1.** a very simple geometric constraint: in the left-top case for instance, we only retain the pair if the center of the left segment is on the left and below the center of the top segment (an equivalent simple rule applies for each of the three other cases); **2.** and a regional coherency constraint: in the left-top case, we have to find the same region on the right of the left-segment and below the top-segment. We then group pairs of segments into sequences of segments, starting from each possible side (left, top, right, and bottom) and discarding sub-sequences. Setting that the energy U_{seq} of a sequence is the sum of the energy U of its segments (Eq. (1) in Section 2.3), we retain the sequence having the lowest energy. The decision on the running example is given in Fig. 1f.



(a) Low-light environment + noise.

(b) High-light environment + noise.

(c) Defocus + flare.

Fig. 3: Examples of the robustness of our method. From left to right: detail of the input image, chunks, final decision.

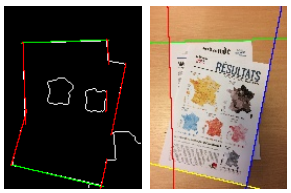
3. RESULTS

3.1. Qualitative results

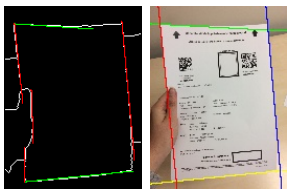
We have proposed a method to detect documents in video frames. This method has two major advantages: we gather many information from regions and contours, and we are robust to many defects such as noise or contrast variations thanks to the invariants of our mathematical morphology framework [19, 20]. In Fig. 3, blurred, noisy and illumination variation cases are correctly handled. Other qualitative results are given in Fig. 2 and show that even in tedious cases we can detect document. In Fig. 4, some failure cases are depicted. These failure cases are due to the superposition of documents, making decision hazardous and distorted (with non-straight boundaries) document. We acquired pictures of documents of different types (magazines, bills, train ticket, etc.) using an iPhone 4.

3.2. Quantitative results

We rely on the evaluation of the ICDAR2015 SmartDoc Competition [11], with databases and associated groundtruth [21] and compared ourselves with some methods using this dataset to obtain a reproducible quantitative evaluation. We applied our method on 4 datasets provided by the competition and compared the results with the top method of the challenge (*LRDE*),



(a) Two documents: we do not use in the final decision process the fact that two opposite boundaries shall be parallel.



(b) Bended document: the top boundary line is not well adjusted to the document boundary curve.

Fig. 4: Some cases of failure.

Method	set 01	set 02	set 03	set 04	runtime
Xu *et al.* [23]	0.997	0.987	0.999	0.994	>1min
LRDE [11]	0.987	0.977	0.989	0.984	>1min
Leal *et al.* [22] best	0.961	0.944	0.965	0.930	0.43s
SmartDoc avg. [11]	*0.946*	*0.903*	*0.938*	*0.812*	*?*
Leal *et al.* [22] fastest	0.921	0.849	0.909	0.840	0.10s
Our	0.905	0.936	0.859	0.903	**0.04s**

Table 1: Quantitative results (Jaccard coefficient) of automatic document detection methods applyied on 4 datasets. "SmartDoc avg." corresponds to the average of the methods of the challenge.

the methods of Leal *et al.* [22] and the method of Xu *et al.* [23]. *LRDE's* relies on two Tree of Shapes [24] computed on the La^*b^* components of each frame. Each node has an energy based on its quadrilateral fitting and the inclusion of lines or images. Candidates objects are supposed to have the highest energies in the two trees, and final selection is performed using the location of the selection in the previous frame. *Xu et al.'s* is an improvement of *LRDE*. A post-processing is applied for the videos suffering from partial occlusions or superposition of documents. The four corners are re-estimated to make the quadrilateral fit the best the contents inside the detected shape. In *Leal et al.'s* , input images are down-sampled and a Geodesic Object Proposal is applied using six seeds and signed geodesic distance transform on each foreground/background seed result. Candidates are post-processed using erosion, dilation and polygonal simplification algorithm [25]. The best candidate is chosen according to its size and shape. The authors made experiments with different downsampling, influencing runtime and quality of results. We show the result of their best performance in quality (Leal *et al.*'s best, downscaled for 1/4) and in runtime (Leal *et al.*'s fastest, downscaled at 1/8).

Quantitative results on SmartDoc database (see Tbl 1) shows that even if the Jaccard index (not the most appropriate but the reference of the challenge) of our method is lower than the one of the other methods, it remains globally better than the average of the SmartDoc challenge and than the fastest method of Leal *et al.*, with an overall average of 0.901 against 0.899 and 0.880. We can observe that our method has better results than SmartDoc's average on the set 04 and 02. These sets have a light background, with low contrast especially for 04. Our method can handle these cases while the others need enough contrast. Moreover, our method is the fastest. The main error cases is the presence of the border of tables that can be preferred to the border of the document it the document is too small, or a border is missing. While we are in this case, our method cannot give 4 points as asked by the evaluation protocol, so we rejected the result yielding a higher penalty considering 0 points found. Our methods is the fastest as it was designed to process sequences in real-time. As shown by Leal *et al.* [22] and Tbl. 1, an important simplification of the image leading to a shorter runtime yields not quite as good results. Moreover, our application case is not the SmartDoc dataset, but it was the opportunity to have a quantitative estimation of our performances. Our method has not been optimized on this dataset unlike the other methods, and has few hypothesis: documents are centered, take most of the part of the frames, without big rotations and with legible text. We did not expect our method to be as efficient as our previous one [11], but still works well in practical cases.

4. DISCUSSION AND CONCLUSION

Our method gives rather good results for real-time document detection but suffers from limitations due to the objective of our development: our method was designed to detect documents (no specified types) in mobile device video flow just before photo capture. Unfortunately, we cannot rely on some (very easy-to-use and would-be-powerful) information, because they are not reliable to universally detect documents. Indeed, we cannot assume that: the document does not touch the border of the image (see the top row in Fig. 2); it is centered; his surface takes the major part of the image; his contents is homogeneous in color (see the bottom row in Fig. 2); the background, outer part of the document in the image, is uniform (see the middle row in Fig. 2); two successive boundary segments are close (in the middle row in Fig. 2, the selected left boundary, the one on the top, is far from the bottom one, due to the presence of a hand); the document is the most textured part of the image; the most salient contours are given by the document boundary (it is not the case for the gradient image in Fig. 1c).

We have tried to take some benefits from other information: the document is less saturated than the background, and it is spatially coherent. Yet we believe that the three following constraining objectives shall be removed: detect small documents (actually documents looking small in the image, for instance because they are far from the camera, could be wrongly detected if there is a border of table); detect documents having an orientation around $\pm 45°$; detect documents that are not centered in the image. Indeed, we observed that, without these constraints, the method we propose can be more robust. First, a better line detector (step of extraction of a set of lines from the watershed line) can be set up. We can filter the watershed line image to get two new images: one with horizontal parts only, and one with vertical parts only. A probabilistic Hough transform applied on both images works very well, thanks to the facts that the document boundaries are not small (the document shall not be small), and that the boundary orientations are in the range $\pm 20°$ around $0°$ and $90°$ (the document is rather aligned with the image). Second, considering that the document is present in a significant part of the middle of the image helps a lot the final step of boundaries identification. Last, another interesting idea is to rely on salient lines [26] or on saliency maps [27].

References

[1] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey," *Intl. J. on Document Analysis and Recognition*, vol. 7, no. 2, pp. 84–104, 2005.

[2] D. Esser, K. Muthmann, and D. Schuster, "Information extraction efficiency of business documents captured with smartphones and tablets," in *Proc. of the ACM Symposium on Document Engineering (DocEng)*, 2013, pp. 111–114.

[3] S. S. Bukhari, F. Shafait, and T. M. Breuel, "The IUPR dataset of camera-captured document images," in *Proc. of the Intl. Workshop on Camera Based Document Analysis and Recognition (CBDAR)*, ser. Lecture Notes in Computer Science, vol. 7139. Springer, 2011, pp. 164–171.

[4] N. Skoryukina, D. P. Nikolaev, A. Sheshkus, and D. Polevoy, "Real time rectangular document detection on mobile devices," in *Proc. of the SPIE 7th Intl. Conf. on Machine Vision (ICMV)*, vol. 9445, 2015, pp. 1–6.

[5] K. Bulatov *et al.*, "Smart IDReader: Document recognition in video stream," in *Proc. of the Intl. Workshop on Camera Based Document Analysis and Recognition (CBDAR)*, 2017, pp. 39–44.

[6] J. Serra, *Image Analysis and Mathematical Morphology*. London: vol. 1 & vol. 2, Academic Press, 1982 & 1988.

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

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

[9] G. Lazzara, T. Géraud, and R. Levillain, "Planting, growing and pruning trees: Connected filters applied to document image analysis," in *Proc. of the IAPR Intl. Conf. on Document Analysis Systems (DAS)*, 2014, pp. 36–40.

[10] E. Carlinet and T. Géraud, "MToS: A tree of shapes for multivariate images," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5330–5342, December 2015.

[11] J. Burie *et al.*, "ICDAR 2015 competition on smartphone document capture and OCR (SmartDoc)," in *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1161–1165.

[12] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, "Real-time computer vision with OpenCV," *Communications of the ACM*, vol. 55, no. 6, pp. 61–69, June 2012.

[13] Y. Xu, T. Géraud, and L. Najman, "Connected filtering on tree-based shape-spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1126–1140, 2016.

[14] F. Meyer, "Watershed topographic distance and watershed lines," *Signal Processing*, vol. 38, no. 1, pp. 113–125, 1994.

[15] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: Thinnings, shortest path forests, and topological watersheds," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 925–939, 2010.

[16] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Computer Vision, Graphics and Image Processing*, vol. 44, pp. 87–116, 1988.

[17] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic Hough transform," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119–137, 2000.

[18] R. G. Von Gioi, J. Jakubowicz, J. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[19] V. Caselles, B. Coll, and J. Morel, "Topographic maps and local contrast changes in natural images," *International Journal of Computer Vision*, vol. 33, no. 1, pp. 5–27, 1999.

[20] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur, *A Theory of Shape Identification*, ser. Lecture Notes in Mathematics. Springer, 2008, vol. 1948.

[21] J. Chazalon, M. Rusiñol, J.-M. Ogier, and J. Lladós, "A semi-automatic groundtruthing tool for mobile-captured document segmentation," in *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR)*, 2015, pp. 621–625.

[22] L. R. Leal and B. L. Bezerra, "Smartphone camera document detection via geodesic object proposals," in *Proc. of the IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2016, pp. 1–6.

[23] Y. Xu, E. Carlinet, T. Géraud, and L. Najman, "Hierarchical segmentation using tree-based shape spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 457–469, 2017.

[24] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Trans. on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.

[25] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[26] Y. Xu, T. Géraud, and L. Najman, "Salient level lines selection using the Mumford-Shah functional," in *Proceedings of the 20th IEEE International Conference on Image Processing (ICIP)*, Melbourne, Australia, September 2013, pp. 1227–1231.

[27] M. Ôn Vũ Ngọc, J. Fabrizio, and T. Géraud, "Saliency-based detection of identy documents captured by smartphones," in *Proc. of the IAPR Intl. Workshop on Document Analysis Systems (DAS)*, 2018, to appear. [Online]. Available: http://publications.lrde.epita.fr/movn.18.das