BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023
http://excel.fit.vutbr.cz

Vision-based Web Page Segmentation

František Maštera*

Abstract

There are many methods with various approaches to the web page segmentation. FitLayout offers a suite of tools for evaluation of such methods for their evaluation and further development. The aim of this paper is to extend this suite with an implementation of another publicly available method. For this goal, the Cormier et al. segmentation method has been chosen and integrated into the FitLayout code-base. The implementation has been tested to prove its plausibility against the original publication and was put to further evaluation to analyze its attributes and behavior. As a result, the newly integrated method can be used for further research, which can be built on the evaluation results found in this paper.

*xmaste02@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

WPS (Web Page Segmentation) focuses on segmenting the web page to more semantically coherent parts. These parts can be more easily processed for document indexing or data mining, or they can be used by assistive technologies to aid visually impaired users.

The problem of WPS has been tackled by many publications, presenting their own approaches. The FitLayout framework offers tools for their evaluation and research, such as web page rendering, result visualization or creating persistent data sets. It also contains a suite of already implemented WPS methods, on which these tools can be used.

The goal of this paper is to extend this suite by another publicly available WPS method. For this goal, the Cormier et al. [1] WPS method has been chosen. Furthermore, the integration of the method to the FitLayout is tested and further evaluated to analyze its strength, weaknesses and and how it reacts to certain parameter settings.

2. Integration & Pipeline

As seen in the Figure 1, in order for the method to segment the web page, it needs the web page itself. This is handled by the tools offered by the FitLayout framework by either loading already existing web page from the RDF repository, or by rendering a new one from the given URL. Either way, the input page, along with the customizable parameters, is then handed over to the `CormierProvider`, which implements the FitLayout's `ArtifactService` interface for transforming a certain artifact type to a different one – in this case it's the web page being transformed into a tree of segments, using the `CormierSegmentation` class, encapsulating the method's pipeline.

The method then takes the screenshot from the page, as it's all the method needs for creating the segmentation. This screenshot is then converted to 2 matrices containing probabilities that each pixel of the image contains a locally significant horizontal or vertical edge. This process will be further explained in the following section 3. The final step passes the probabilities to the module for splitting the segments, which uses them for deciding where the segmenting lines will be. Result of that is the final segmentation. Section 4 will present more details about its creation.

3. Locally Significant Edges

The goal of this process is to get the baseline information based on which the following stages could decide where to segment the web page. This can be achieved by applying the standard edge strength using the Sobel operator on the web page screenshot, as seen in Figure 2 with vertical edges. By itself, this information isn't perfect though. Higher values are obtained in areas with higher contrast – often text. At the same time, more subtle edges and back-

ground transitions, which are often a good hint for segmentation, result in much lower values.

This problem is tackled by calculating the final probability from the difference between the edge strength of each pixel and the edge strength distribution of pixels in its neighborhood. This way, the faint lines get higher probability, as long as stand out from their surroundings. The neighborhood also doesn't include the pixels in the line along the pixel to prevent lines from lowering the value of the pixels on them. It's also split in half, each processed separately.The one with higher probability is then favored to improve the detection of lines which separate different types of content (e.g. textured images from plain background).

As a result, the final stage in Figure 2 displays the formerly faint line now having much higher probability of later being used for segmentation.

4. Final Segmentation

The output, displayed in the Figure 3, is a hierarchical structure of segments, forming an X-Y tree. The root node always covers the entire page. Children nodes are created by splitting the parent node, in a different direction with each level. Finally, the leaf nodes are segments which can't be split any further.

To decide where each node will be split, the method iterates over all possible lines in the current level's direction of splitting and picks the one with the highest probability of being semantically significant. This probability is based on a portion of the pixels in the line which meet the required local significance. This process the repeats itself for the node for as long as there are lines which have this probability high enough. There's also a minimum size for each segment.

5. Output Preview

The Figure 4 shows an output example, on which the hierarchical structure can be clearly observed, as the method starts from main sections (navigation bar, featured items, ...), which are then structured further (featured items are split into separate items, which then have separate details). In this example specifically, the neighborhood split into 2 parts (described in section 3) helps a lot, as the images have often a different texture from the plain background.

Imperfections of the method can be observed in this example as well, such as inconsistencies in segmenting similar elements differently (see the buttons on the right bottom corner of each item in the featured category), or accidental segmentation of the bottom center item due to the logo forming a strong line or due to the long text in the top banner.

6. Testing & Evaluation

The final section shows a set of graphs with the evaluation results. The X-axis of all of them displays the similarity of the method's output to some other segmentation results, interpreted using the extended BCubic F_1-score [2].

The outputs of the integrated method have been compared against the outputs of a publicly available implementation to prove their *plausibility*. The graph in Figure 5a shows that the most outputs were similar, with about 10% being completely different. These are mostly failed segmentations by one of the implementations, or cases of the first level of segmentation being in different directions.

The other graphs show similarity against a crowdsourced data set of ground truth segmentations. [3] This measurement is for defining the *quality* of the segmentation. The cause of the mostly mixed results seen in Figure 5b is mostly a different approach to creating segments, as well as worse results in pages with less favorable visual cues.

The method offers a wide set of parameters, of which the most effective are displayed by the graph in Figure 5c. The quality of the segmentation has been measured for multiple value combinations of these parameters. The graphs display the performance of each of them, with mean values represented with the cross marks.

Overall, the parameters can affect various aspects of the method such as the segmentation granularity, the intensity of the effect of the locally significant edges, or even the overall time needed for the segmentation. The best-performing values improve the quality of the method's results by over 20% on average.

7. Conclusions

In this paper, the FitLayout framework has been extended with a new WPS method. This method's plausibility has been properly tested and it's attributes have been evaluated in detail. The output of this paper offers further research of the WPS methods using the method's implementation, which can be based on the presented evaluation results.

Acknowledgements

I would like to thank my supervisor doc. Ing. Radek Burget, Ph.D. for his help with this paper.

References

[1] Michael Cormier. *Computer Vision on Web Pages: A Study of Man-Made Images*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 8 2018.

[2] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12:461–486, 8 2009.

[3] Johannes Kiesel, Florian Kneist, Lars Meyer, Kristof Komlossy, Benno Stein, and Martin Potthast. Web page segmentation revisited: Evaluation framework and dataset. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 3047–3054, New York, NY, USA, 10 2020. Association for Computing Machinery.