Web Page Segmentation Based on the Hough transform and Vision Cues

Tingting Wei[1] Yonghe Lu[2*] Xuanjie Li[3] and Jinglun Liu[4]
Sun Yat-Sen University, Guangzhou, China
[1] School of Information Science and Technology E-mail: tingtingwei2011@126.com
[2] School of Information Management E-mail: luyonghe@mail.sysu.edu.cn **Corresponding author**
[3] School of Information Management E-mail: lixuanjie@qq.com
[4] School of Software E-mail: liujinglunapply@163.com

***Abstract*—Web page segmentation has extensive value in web applications. As the traditional method which only based on Document Object Model (Dom) tree poorly reflects the actual semantic structure of a page, the vision-based measures which attempt to understand the perception of user have attracted great concern in recently. Moreover, the visual page layout structuring is more suitable to suggest a semantic partitioning of a page. Vision-based Page Segmentation (VIPS) algorithm is a notable technique which improves the situation that the visual effect is not consistent with the corresponding DOM tree of web pages. However, due to the increasing complicated structure of web pages and ever-changing of web design, the rules in VIPS become numerous and are no longer fully applicable. To alleviate the shortcoming of VIPS, this paper introduces Hough transform in image processing and takes advantage from DOM trees and visual cues. Our proposed method first extracts the visual separator in web pages according to the perceptive of web designers, then adjusts the segmented information blocks by Hough transformation in the hope of enhancing the VIPS algorithm compatibility and improving the performance of information extraction in applications. The quality of our proposed method is evaluated by subjective and objective measurements, and the experiment results show that this method has come to an anticipant result which even better than the classical one.**

I. INTRODUCTION

As the development of internet, identifying and extracting distinct information elements from the Web has increasingly become difficult. Web pages as the largest sources on the Web, besides the main content texts, a web page usually contains a bouquet of other topic unrelated elements such as navigation, user comments, text ads, legal disclaimers etc. Therefore, identifying the web page structure and segmenting these distinct elements into different parts is very important for web information extraction.

In recent years many information extraction techniques have been proposed and have obtained a certain effect in practice. They show their talents and make these applications more intelligent. On the whole, all the measures can be grouped into four classes: statistical learning theory based measures, template-based measures, HTML DOM tree structure based measures, and vision-based measures. However, the first three measures either have great limitations in visual assessment or cannot be used in large scale web pages. Compared with them, vision-based measures show more effectiveness to directly deal with the web pages shown to the user by image processing. As a result, vision-based measures have attracted great concern. At present, Vision-based Page Segmentation Algorithm (namely VIPS) [1] is the represent of modern measures that is applied widely in information extraction and have achieved marked achievements. VIPS is inspired by human visual system, it improves the situation that the visual effect is not consistent with the corresponding DOM tree of web pages. However, due to the increasing complicated structure of web pages and ever-changing of web design, the rules in VIPS become numerous and are no longer fully applicable.

To alleviate the shortcoming of VIPS, on the basis of image processing technology and characteristics of web pages, this paper introduces Hough transform in image processing and takes advantage from DOM trees and visual cues. We investigate the characteristic of separator of web pages, extract the visual separator in web pages according to the perceptive of web designers, then adjusts the segmented information blocks by Hough transformation in the hope of enhancing the VIPS algorithm compatibility and improving the performance of information extraction in applications.

This paper is organized as follows. Section 2 presents the related works. Section 3 details our algorithm based on Hough transformation and vision cues. Section 4 illustrates the fundamental knowledge and shows an overview of our proposed approach. The experimental results are reported in Section 5. Section 6 summarizes our contributions and concludes the paper.

II. RELATED WORKS

In the field of web information extraction, a lot of researches have been done for it. We will review several lines of related works which associated with our proposed method in this section.

One line of these methods is focus on the DOM trees. The classical approach is proposed by Valter [2] which can automatically extract data from large web sites. This method based on a set of sample HTML pages belonging to the same class, when run on these pages, it compared the HTML codes

of the two pages, then inferred a common structure and a wrapper, and used that to extract the source dataset. Wang [3] proposed DSE (data-rich section extraction) method which can carry out the desired filtering task by identifying data-rich sections in HTML pages, so as to improve the extraction performance, but it mainly processes the web pages generated on the fly by querying a database server and placing the results into a predefined page structure. Liu [4] proposed MDR (mining data records) algorithm based on two observations. Gupta [5] counted the proportion of linked/non-linked words, and compared it to a predefined threshold then identify the topic text area. However, it needs the manpower to adjust the parameter for the best results and also cannot deal with images. Feng [6] proposed a framework of web page analysis with coordinate trees, which can divide the pages by space relationships and their locations, but it also relied on the rules. Chakrabarti [7] gave an algorithm that is based on formulating an appropriate optimization problem on weighted graphs, where the weights capture if two nodes in the DOM tree should be placed together or apart in the segmentation. They learnt the weights from manually labeled data in a principled manner.

The other one line is based on visual representation. Cai [1, 8] proposed a vision-based page segmentation method (VIPS) which simulates how a user understands web layout structure based on his visual perception. It took full account of characteristics of web pages such as the size and style of types, background colors, and blank areas, meanwhile laid down the rules to divided page into blocks, which was independent to underlying documentation representation such as HTML and works well even when the HTML structure is far different from layout structure. Later, there are many works are carried out on the basis of the VIPS. Weng [9] firstly exploited the VIPS algorithm to segment a query result page into a Visual Block tree, then identify data records in the page based on its Visual Block tree by using the techniques proposed in [10]. Wu [11] proposed a method that extracted a number of features to represent a web page based on web mining and computer vision techniques, which argued that the visual complexity of web pages can affect user experience. It also employed the VIPS to analysis web page structure. Besides VIPS, Cao [8] presented a segmentation method for web page analysis using shrinking and dividing, which divided the web page into sub-images (basic blocks) according to the dividing zones which are the spaces between blocks, then the shrinking technology is employed for better result. But it is difficult to find the desired dividing conditions due to the increasingly complexity structure of web pages.

III. FUNDAMENTALS

In this section, we introduce the VIPS algorithm and give a procedure of separator detection based on the Progressive Probabilistic Hough Transform, which are the basis of our approach.

A. The VIPS Algorithm

The VIPS algorithm relied on heuristic rules to enhance the block extraction process and then put all the extracted blocks into a pool for visual separator detection. Then the weights of these separators are set based on the certain rules. We present the procedure of VIPS from [1] and [12] as follows.

The segmentation process is illustrated in Fig. 1. It has three steps: block extraction, separator detection and content structure construction. First, DOM structure and visual information, such as position, background color, font size, font weight, etc., are obtained from a web browser. Then, from the root nodes of the DOM tree, the visual block extraction process is started to extract visual blocks of the current level from the DOM tree based on visual cues. Every DOM node is checked to judge whether it forms a single block or not. If not, its children will be processed in the same way. When all blocks of the current level are extracted, they are put into a pool. Visual separators among these blocks are identified and the weight of a separator is set based on properties of its neighboring blocks. After constructing the layout hierarchy of the current level, each newly produced visual blocks is checked to see whether or not it meets the granularity requirement. If no, this block will be further partitioned. After all blocks are processed, the final vision-based content structure for the web page is outputted.

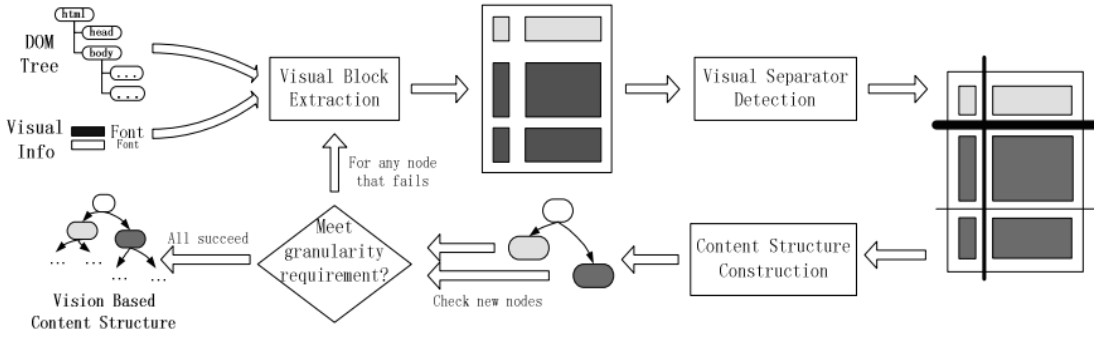


Fig. 1 The procedure of vision-based page segmentation algorithm

Although the top-down dividing measure is fast but VIPS needs prior knowledge and relies on many heuristic rules. Due to the increasing complicated structure of web pages and ever-changing of web design, the rules in VIPS become numerous and are no longer fully applicable. In this paper, we combine VIPS with Hough Transform which we will illustrate in the subsequent section to improve separator detection such that obtain more accurate results.

B. *Progressive Probabilistic Hough Transform*

The first Hough Transform (namely HT), introduced and patented by Paul Hough in 1962 [6], was based on the line equation in the slope-intercept form. Princen et al. [7] formalized HT as a hypothesis testing process. The structure of HT when described as generically as possible is [5]:

1. Some *evidence* is extracted from the input.
2. For each piece of the evidence, *accumulators* corresponding to the *hypotheses* that are supported by that evidence are incremented. Possible hypotheses are represented by an N-dimensional *parameter space* of accumulators.
3. Probable hypotheses are detected as peaks in the parameter space.

The simplest case of HT is detecting straight lines, which is the reason we employ in this paper. However, the main disadvantage of the standard HT was its large storage and computational requirements. To improve the standard HT and satisfy different information needs, a lot of reformations of HT have been proposed (see [13] for an overview of the Hough Transform). In our approach, we intend to employ the improved algorithm Progressive Probabilistic Hough Transform (namely PPHT) [14] to detect separator in web pages, this algorithm is ideally suited for real-time applications with a fixed amount of available processing time, since voting and line detection is interleaved. The most salient features are likely to be detected first. These features are very suitable for processing large-scale even with complicated structure web pages. Below we will present the PPHT in detail according to [14].

Firstly, it selects a new point at random from the input images (do not duplicate random choices), then transform the point and update the accumulator. Secondly, check if the highest peak in the accumulator that was modified by the new pixel is higher than threshold, if not then goto the first step. Thirdly, look along a corridor specified by the peak in the accumulator, and find the longest segment of pixels either continuous or exhibiting a gap not exceeding a given threshold. Finally, if the line segment is longer than the minimum length adds it into the output list, goto the first step.

We apply the PPHT to get a list of line segmentation of web pages, which not only can improve the processing efficiency but also achieve a better result.

IV. THE OVERALL PROCEDURE

A. *Overview of Our Approach*

Our solution employs the VIPS and incorporates the advantages of the PPHT on content extraction. Below we will first show two example pages (http://news.qq.com/), comparing our VIPS result and the integrated way in Fig. 2 and Fig. 3 respectively, then analysis the comparative results.

In Fig. 2, after several iterations, the final block is VB1-2-1-1-2-2-1, which is marked with red rectangle, but this block can be further segmented in several sub-blocks. There has two potential problems: (1) the false semantic block. Many blocks are formed up with noisy only, but they remain in the extracted block; (2) the granularity problem. Multiple data sub-blocks form up a large data block, since they are adjacent to each other. To solve these problems, we apply the PPHT to further detect and divide the blocks.

Fig. 2 The sample web page extracted by VIPS algorithm

Fig. 3 The sample web page extracted by our proposed algorithm

From Fig. 3 which results from the PPHT measure, we can see that the block VB1-2-1-1-2-2-1 still can be segmented into sub-blocks, and there is obvious line separator to separate each block.

From the above examples, we can see that the integrated way can successfully identify the separators among different blocks in the web page, while VIPS only fails.

After the detection of separators, from the previous introduction, we note that the last step of VIPS is to construct the content structure according to the detected separators and their set weights. The construction process starts from the separators with the lowest weight and the blocks beside these separators are merged to form new virtual blocks. This process iterates till separators with maximum weights are met. To consistent with the quantization range of DoC of each new block, it needs to linear normalized the weights of separators in each iteration. The normalized is defined as follows.

$$normalizedValue = \frac{SW - \min W}{\max W - \min W} * 10 + 1 \qquad (1)$$

Where SW is the weight of separator in current iteration, $minW$ and $maxW$ are the minimum and maximum weight among all separators respectively. The separator detection algorithm is described in Algorithm 1.

B. Procedure

In this section, we give an overall procedure of our algorithm (namely HoughVIPS) in Fig. 4 and outline the process steps in detail as below.

1. Web pages preprocessed. The page is first passed through an HTML parser that creates a DOM tree representation of the web page.

2. Web pages extracted.

2.1 Pre-define the *Permitted Degree of Coherence* (PDoC) and extract a semantic related parts set O;

2.2. Employ the VIPS to detect the separators and resulting a set S_1;

2.3. Employ the PPHT to further detect the separators and return a set S_2. To obtain better results, we adjust the high and low threshold according to empirical experiment in the edge detection phase, and the peak value in the accumulator when performing the transformation.

2.4. Merge S_1 and S_2, and setting weights for all separators.

2.5. Looping through the content structure construction until it meets the common requirement for DoC is that DoC > PDoC, if PDoC is pre-defined.

3. Output the extracted results.

Algorithm1 The separator detection algorithm

Input: The height of web page *pageHeight*, the visual structure *visualStructure*.

Output: Separators and their corresponding weights set *S*.

Procedure normalizeSeparatorsMinMax ()

1. *getAllSeparators (visualStructure, S)*; /*get original *S**/
2. *maxSep.start = 0*;
3. *maxSep.end = pageHeight*; /* get the largest separator */
4. *S.add (maxSep)*; /*add to *S**/
5. *maxSep.weight = 40*; /*set the weight as 40*/
6. *S.sort()*; /*sort the weight in S*/
   /*get the minimum weight in *S**/
7. *minWeight = separators.get(0).weight*;
   /*get the maximum weight in *S**/
8. *maxWeight = separators.get(separators.size()-1).weight*;
9. Loop
   for each $S_i \in S$
   /*normalized the weight*/
   normalizedValue = (separator.weight - minWeight) / (maxWeight - minWeight) *10 + 1;
   /*convert to the DoC value*/
   separator.normalizedWeight = getDoCValue ((int) Math.ceil (normalizedValue));
10. Return *S*

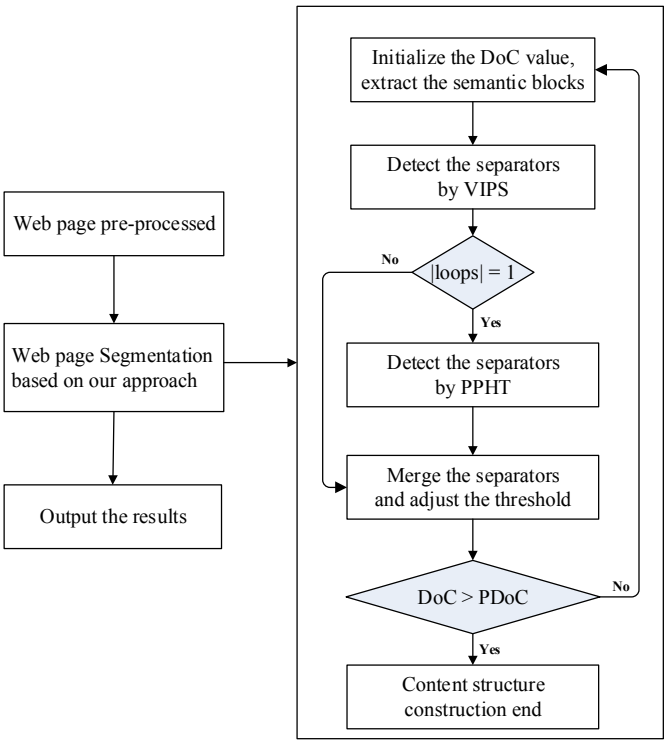


Fig. 4 The HoughVIPS algorithm

V. EXPERIMENTAL EVALUATION

In this section, we first present the process of determining the parameters of the PPHT method, then evaluate our approach and compare it with the classical one by the way of controlling variables. We implement the algorithm in Eclipse Luna and Visual Studio 2010, and employ some Java open-source tools such as NekoHTML, CSSBox and OpenCv. The platform of the experiment is a PC with 2.6Ghz processor, 8G RAM, and Window X7.

We manually constructed a set of 3957 segments obtained from 150 Chinese web pages, of which 87 pages are from CWT70th2[1] dataset, and other 63 pages from a wide range of sources, news websites, Wikipedia, blog, forum and navigable websites.

C. *Separators extracted based on PPHT*

Adaptive determine parameters of Edge detection. To apply the PPHT in separator extracting, it needs to make the web page to a gray one since the input of PPHT is a binary image which resulting from an edge detector. In this experiment, Canny is selected as our edge detection algorithm. To reduce the number of fake edges, Canny set two parameters t_{high} and t_{low} denote the high and low threshold of edge detection respectively. The parameter t_{high} controls the initial point of strong edge, but if this value is set too high there may generate incoherent line. The t_{low} is used to edge linkage. We employ the adaptive functional of OpenCV to auto-adjust these two parameters, the adaptive results are shown in Fig. 5. The t_{ratio} denotes the ratio of t_{high} and t_{low}. In general, t_{ratui} is suggested as 0.4 [15].

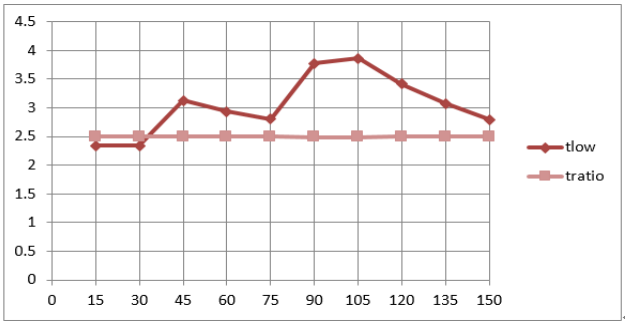
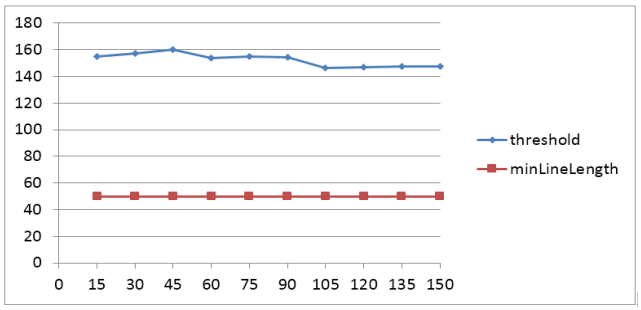


Fig. 5 Adaptive determine parameters of Edge detection

In Fig. 5, the abscissa represents the accumulator of web pages. We can see from that t_{low} is range from 2 to 4, and t_{ratio} achieves a relative stable value 2.5, which is close to the suggested value.

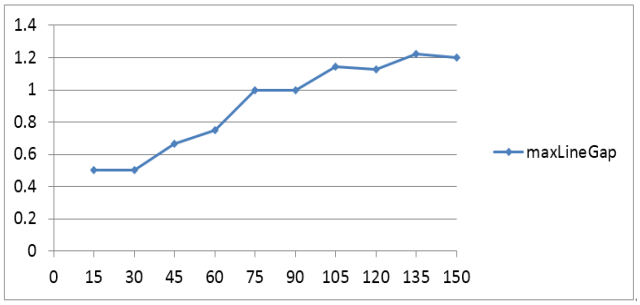
PPHT parameters determination. In PPHT, there are three important parameters, the peak threshold *threshold*, the minimum length of line *minLineLength* and the gap between two lines *maxLineGap*. The parameter *threshold* is the significance level at which lines are accepted; *minLineLength* is used to check if the line segment is longer than it then add it into output list and *maxLineGap* is used to judge if two line segments either continuous or exhibiting a gap can be merged as a line or not. In this experiment, we manually adjust these parameters and output their values when we obtain the desirable segmentation results. The experimental results are shown in Fig. 6. The abscissa represents the accumulator of web pages; all three parameters are counted as the average of the best situation on all web pages.

From Fig. 6, the significance level *threshold* is range around 150; the minimum length of line is almost get a constant value 50 and the gap range from 0.5 to 1.3. The results indicate that these parameters show little changes on different amount of web pages. We note that the *threshold* is proportional to the *minLineLength*, since the longer of line the more pixels obtained in accumulating procedure, such that the greater probability of the same line that the pixels source from. In our experiment, the *threshold*, *minLineLength* and *maxLineGap* are decided to be 150, 50, and 1.1 respectively.



(a) *Threshold* and *minLineLength* experimental results

[1] http://www.cwirf.org/



(b) *MaxLineGap* experimental results

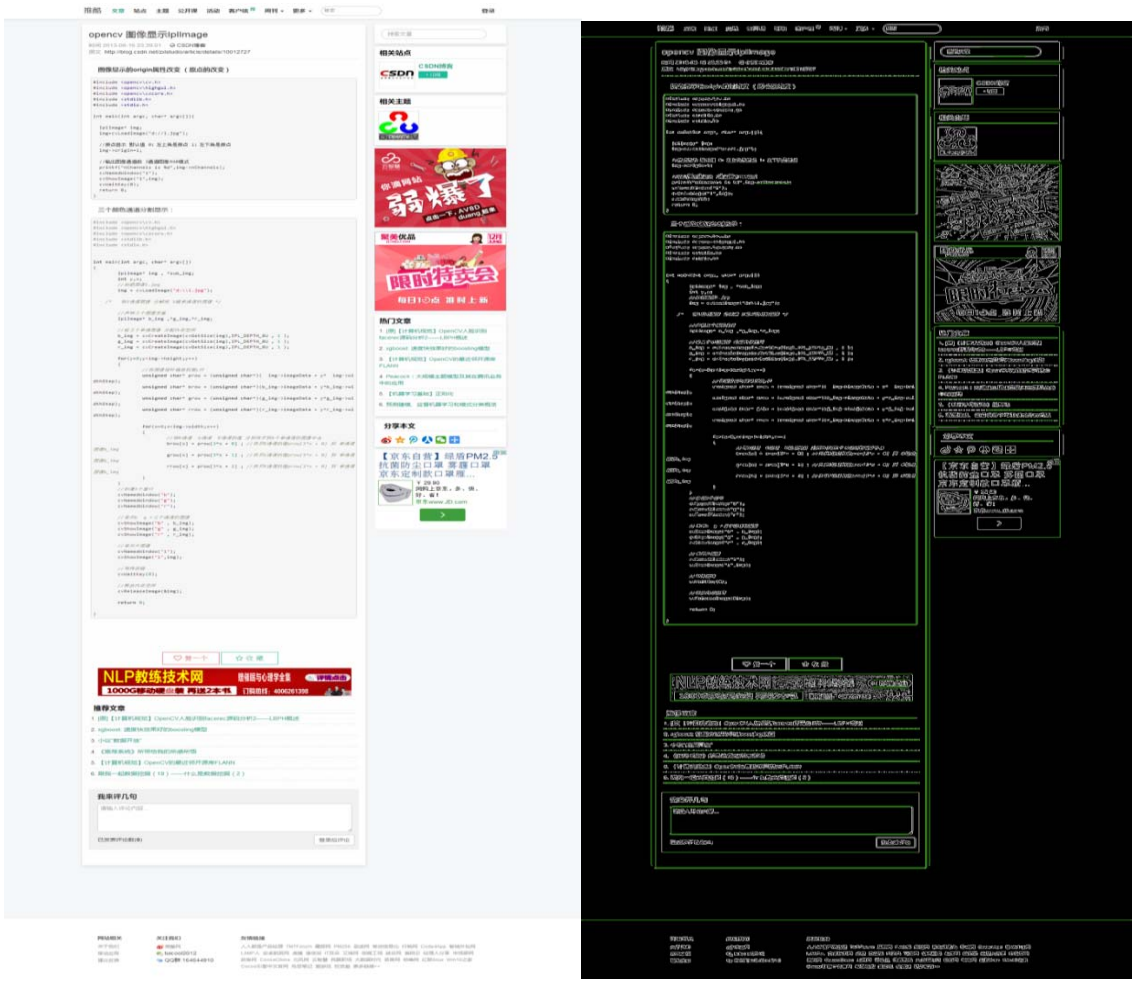
Fig. 6 PPHT parameters estimate

Separator extraction based on PPHT. Similar to VIPS algorithm, in this experiment, we extract separators including horizontal separators and vertical separators. Fig. 7 shows an example of separator extraction for a web page (http://www.tuicool.com/articles/UVZ7Jf.2014-3-1), before and after separators extraction. The separators marked with green line in the right figure. From Fig. 7, we can see that the PPHT method can effectively extract all the separator in that web page. However, the VIPS needs to pre-defined PDoC value to determine the granularity of segmentation in this phrase. This may cause the granularity problem. Multiple data sub-blocks form up a large data block since they are adjacent to each other. While PPHT can detect separators include those are added by designer, which is aware of the perspective of the designer, making it possible to use separators and content extraction algorithms efficiently and accurately.

D. Evaluation of segmentation

In this section, we evaluate our proposed approach (namely HVIPS) and compare it to VIPS algorithm. Below, we first describe some experimental settings.

Evaluation measures. As the content structure detection has the nature of subjectivity and uncertainty, we assess the results by human judgments. Similar to [1], four volunteers in our experiment are asked to judge the results based on below criteria in Table 1.



(a) Original web page (b) Separator extraction results

Fig. 7 PPHT experimental results

TABLE I
THE SUBJECTIVE EVALUATION INDEX OF SEGMENTATION

Criteria	Description
Perfect	The separated blocks are consistent with human judges, the error rate not exceeding 5% and the content is close to original texts.
Satisfactory	The error rate range between 5% and 10%, and the content is almost close to original texts.
Fair	The error rate exceeds 10%.
Bad	Get the wrong content structure.

TABLE II
EVALUATION OF THE HVIPS AND VIPS ALGORITHMS

Human Judgment	User1		User2		User3		User4		Average percentage	
	VIPS	HVIPS	VIPS	HVIPS	VIPS	HVIPS	VIPS	HVIPS	VIPS	HVIPS
Perfect	82	90	81	85	92	95	81	91	56%	60.17%
Satisfactory	56	51	59	56	45	46	57	53	36.17%	34.33%
Fair	9	7	9	8	13	9	11	5	7%	4.83%
Bad	3	2	1	1	0	0	1	1	0.83%	0.67%

Experimental results. We carry our algorithms and compare the segmentations generated by them with those obtained manually. The results are shown in Table 2. As can be seen, there have 92.17% (56%+36.17%) semantic content structures correctly detected in VIPS algorithm, while this figure raise to 94.5% (60.17%+34.33%) in HVIPS. In addition, the rate of "fair" and "bad" of HVIPS is much less than the one in VIPS. Note that the HVIPS algorithm is helpful for promoting the "satisfactory" into "perfect" extraction; it is able to obtain more "perfect" results, because HVIPS can effectively extract the separator which uses a very thin image representing a line, while VIPS cannot handle this situation. For HVIPS, those "fair" and "bad" pages, the major reason is similar to those described in [1], the browser provides wrong position information so that our algorithm cannot get the correct content structure.

In addition, we use the standard measures of precision, recall and F1 in machine learning to evaluate the performance of our algorithm and compart it to the VIPS, and the results are shown in Table3. The PDoC is decided as the largest value 10, which will make the more granular segmentation of web pages, and ensure that the partition granularity is less than the manually divided one such that facilitate the assessment.

TABLE III
EVALUATION OF THE HVIPS AND VIPS ALGORITHMS

Algorithm	Precision	Recall	F1
VIPS	88.35%	83.93%	0.861
HVIPS	89.64%	86.33%	0.880

We summarize the experimental results in Table 3 as follows.

1. Our proposed method HVIPS achieves more accurate segmentation results compared to VIPS. Since VIPS algorithm does not consider the semantic relationship among blocks, it cannot detect the blocks that are incorrectly constructed, and which usually results from the design mistakes made by designers. For example, if the advertisement area is embedding into the topic text block, VIPS may treat these two different content blocks as the only one. While HVIPS can accurately dividing these two semantic blocks from the perspective of users, and which has no effect on the mistakes of the designers.

2. From the results, we note that the HVIPS helps improve the recall of segmentation. The major reason is that the HVIPS can divide the large blocks into many semantic independent sub-blocks, while VIPS algorithm is limited by the value of PDoC and cannot further segment those semantic blocks.

We take an example web page [2] that is belonging to a topic-driven page to show the performance of our algorithm. We show the segmentation result in Fig. 8. We can see that the HVIPS can separate the blocks that comprise by many sub-blocks that have different semantic meaning, which result in improving the recall of this algorithm.

Execution time. On our experiment platform, the execution time for each page is always less than 0.4 second.

[2] http://baike.baidu.com/link?url=7lEKI155-kG-iNgdrM_QuNOJdQXmxAy_ArmX8K_GcX_U92yJKjgtBIUZ9H1q-iy7IIPkKwCrnmZjcyo-w-EAf_

Fig. 8 Segmentation results of a page

VI. Conclusions and Future Work

Conclusions. In this paper, we integrate the vision-based page segmentation method (VIPS) with the Progressive Probabilistic Hough Transform (PPHT) in image processing for better web pages segmentation. We take advantage from the PPHT on detecting line, which can compensate the shortcoming of VIPS. For example, it can effectively extract the separator which uses a very thin image representing a line, while VIPS cannot handle this situation. In addition, it can also divide the large blocks into many semantic independent sub-blocks, which is aware of the perspective of designers. While VIPS algorithm is limited by the value of PDoC and cannot further segment those semantic blocks. More importantly, our algorithm is efficient. The experimental results show that our proposed method improves the level of satisfaction on web page segmentation, also it helps improve recall. Overall, we validate that the PPHT is effective to separator detection in the field of web page segmentation.

Outlook and future work. The approach presented in this paper is based on the existing work and considers complementary aspects to solve the segmentation task. In further, we will pay more attention into the properties of the separators, and image pre-processing for the better result. We also want to discuss the influence of the mentioned parameters on different web pages. Another direction is to investigate the use of our technique in some fields such as duplicate detection, information retrieval and so on.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC Grant No: 71373291). This work also was supported by the National High Technology Research and Development Program of China (863 Program) under Grant 2012AA101701.

References

[1] D. Cai, S. Yu, J.-R. Wen, W.-Y. Ma, Vips: a vision-based page segmentation algorithm, in, Microsoft technical report, MSR-TR-2003-79, 2003.
[2] V. Crescenzi, G. Mecca, P. Merialdo, Roadrunner: Towards automatic data extraction from large web sites, in: VLDB, 2001, pp. 109-118.
[3] J. Wang, F.H. Lochovsky, Data-rich section extraction from HTML pages, in: Web Information Systems Engineering, 2002. WISE 2002. Proceedings of the Third International Conference on, IEEE, 2002, pp. 313-322.
[4] B. Liu, R. Grossman, Y. Zhai, Mining data records in Web pages, in: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 601-606.
[5] A. Herout, M. Dubská, J. Havel, Review of Hough Transform for Line Detection, in: Real-Time Detection of Lines and Grids, Springer, 2013, pp. 3-16.
[6] H.P. VC, Method and means for recognizing complex patterns, in, Google Patents, 1962.
[7] J. Princen, J. Illingworth, J. Kittler, Hypothesis testing: a framework for analyzing and optimizing Hough transform performance, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 16 (1994) 329-341.
[8] D. Cai, X. He, W.-Y. Ma, J.-R. Wen, H. Zhang, Organizing WWW images based on the analysis of page layout and web link structure, in: Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on, IEEE, 2004, pp. 113-116.
[9] D. Weng, J. Hong, D.A. Bell, Automatically Annotating Structured Web Data Using a SVM-Based Multiclass Classifier, in: Web Information Systems Engineering–WISE 2014, Springer, 2014, pp. 115-124.
[10] D. Weng, J. Hong, D.A. Bell, Extracting data records from query result pages based on visual features, in: Advances in Databases, Springer, 2011, pp. 140-153.
[11] O. Wu, W. Hu, L. Shi, Measuring the visual complexities of Web pages, ACM Transactions on the Web (TWEB), 7 (2013) 1.
[12] D. Cai, S. Yu, J.-R. Wen, W.-Y. Ma, Extracting content structure for web pages based on visual representation, in: Web Technologies and Applications, Springer, 2003, pp. 406-417.
[13] P. Mukhopadhyay, B.B. Chaudhuri, A survey of Hough Transform, Pattern Recognition, 48 (2015) 993-1010.
[14] J. Matas, C. Galambos, J. Kittler, Progressive probabilistic hough transform, (1998).
[15] J. Canny, A computational approach to edge detection, Pattern Analysis and Machine Intelligence, IEEE Transactions on, (1986) 679-698.