A Goal-Oriented Verification-based Approach for Target Text Line Extraction from a Document Image Captured by a Pen Scanner

Zhen-Long BAI and Qiang HUO
Department of Computer Science and Information Systems,
The University of Hong Kong, Pokfulam Road, Hong Kong, China
(Email: zlbai@csis.hku.hk, qhuo@csis.hku.hk)

Abstract

In this paper, we present a goal-oriented verification-based approach for target text line extraction from a document image captured by a pen scanner. Given a binary image, a series of processing steps are invoked adaptively, guided by the text line verification result in the preceding step. Each step adopts a strategy that is most effective for dealing with the problem concerned. Consequently, the target text line can be extracted in a more efficient and reliable way depending on the nature of the captured image. The effectiveness of the above approach is confirmed by a benchmark test.

1. Introduction

The work reported in this paper deals with the problem of how to extract a *target text line* from a document image captured by a pen scanner used in C-Pen products [4]. Given document images as shown in Figs.1(a)-(d), the target text line is defined to be the informative text line with complete characters by ignoring the possible fragmented characters (e.g., Figs.1(b)(c)), or non-text components (e.g., the underline in Fig.1(d)). Such an extracted text line will be fed to a Chinese/English OCR engine (e.g., [3]) for character recognition. Ideally, the text line finder should also be able to reject any garbage image that does not contain a text line with complete characters (e.g., Fig.1(e)), or the one that can not be recognized reliably by the OCR engine involved.

Although there exist many decomposition algorithms for document images captured by a traditional flatbed scanner (e.g., [6] and references therein), none of them can fully achieve the goal for our specific application here, mainly because of the following unique problems in C-Pen images. They are: 1) A C-Pen image includes only very limited text that makes threshold setting difficult for many existing algorithms; 2) The scanned text line might be skewed, undulated, and even curved, that gives trouble for text line finding; 3) Fragmented-characters often exist in C-Pen images, that brings difficulty to target text line extraction; 4) Some images should be rejected. Simple strategies can hardly address all the above issues, while complicated strategies are often less efficient. Our first attempt to solve the above problem using a complicated bottom-up approach was reported in [1]. As a continuation of the previous work, in this paper, we present a more efficient and reliable solution by integrating multiple strategies adaptively and intelligently.

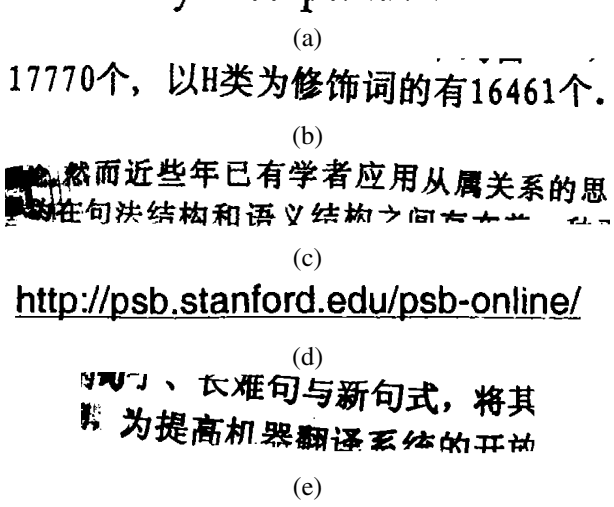

Figure 1. Examples of C-Pen images

2. Our Verification-Based Approach

2.1. Overall Architecture

The overall architecture of our approach is shown in Fig.2, and is explained briefly in the following along with the description of motivation of relevant designs.

Given a binary document image captured by a pen scanner, a series of preprocessing steps are performed as detailed in [1]. It is observed that most images scanned naturally by an experienced user have a simple layout, with just one text line (e.g., Fig.1(a)), or with straight text lines. It is well known that "X-Y cut approach" is effective for dealing with "Manhattan-layout" document images [5]. So, a similar *top-down approach* by horizontal pixel projection is first used to form the possible text lines. This is fol-

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

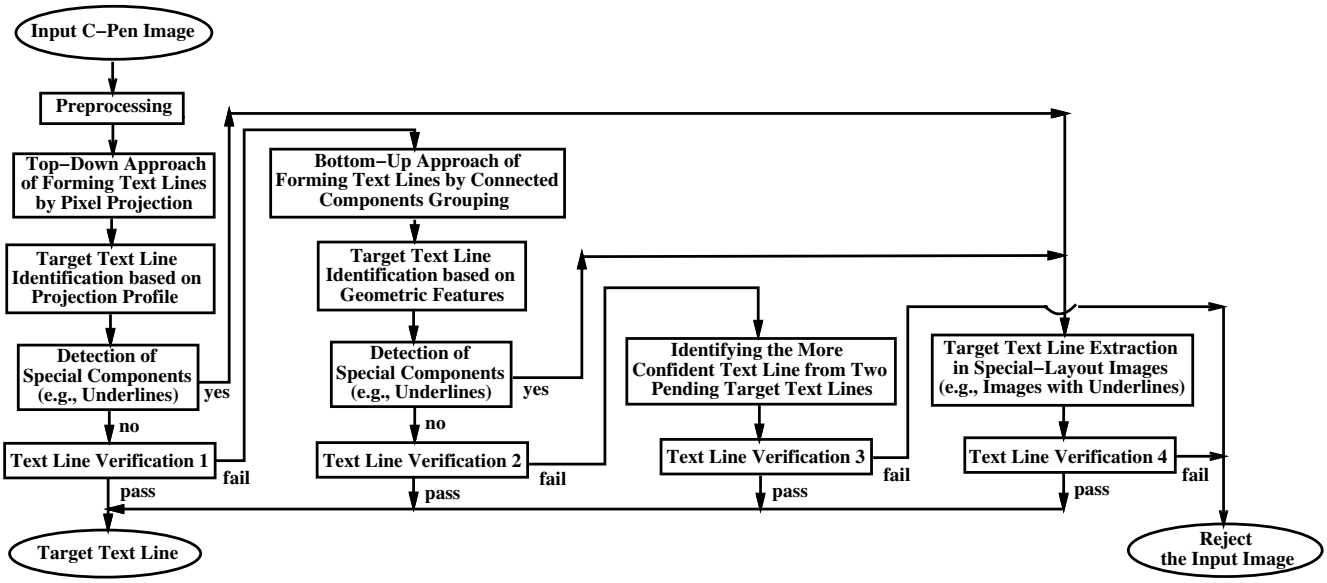

Figure 2. The overall architecture of verification-based approach for target text line extraction.

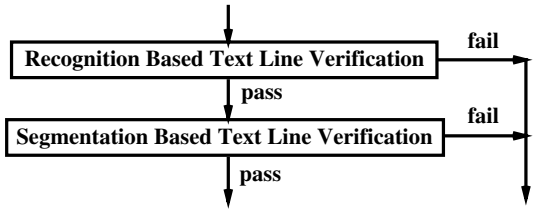

Figure 3. Flow chart of text line verification.

lowed by a step of target text line identification using the projection profile information. The hypothesized target line is fed to a processing module to detect whether there exist some special components, e.g., underlines. If yes, a target text line extraction module for special-layout images is invoked. Otherwise, a *text line verification* step (labelled as "Text Line Verification 1" in Fig.2) is invoked to judge whether the result is confident enough. The flow chart of verification module is shown in Fig.3, where a recognition-based text line confidence measure (CM), V_i^r (cf.[1]), and two segmentation based text line CMs, $V_i^{s_h}$ and $V_i^{s_w}$ (to be explained later), are used for verification purpose. Most of the above mentioned simple-layout images without underlines are expected to pass this verification step, thus the target text line can be obtained directly in this stage. If the above verification fails, a series of more complicated processing steps will be invoked.

In addition to simple-layout images, images with a more complex layout may be captured by a novice user or even by an experienced user scanning the text casually. Such examples are illustrated in Figs.1(b)(c), where more than one text lines exist, including the one formed by fragmented characters. Typically, these images can not pass the "Text Line Verification 1", thus will be fed to the next stage for further processing, that includes the steps, 1) forming text lines by connected components (CC) grouping, and 2) identification of the target text line based on geometric features. This *bottom-up approach* is largely the same as what we described in [1]. The above hypothesized target text line is checked again to detect whether there exist some special components, e.g., underlines. If yes, the target text line extraction module for special-layout images is invoked again. Otherwise, after the above processing steps, another verification step (labelled as "Text Line Verification 2" in Fig.2) is performed. If pass, the target text line is extracted in this stage; otherwise, next processing stage is invoked.

If a reliable decision can not be made even after the above processing stages, we try to identify, in this stage, the more confident text line from two pending target text lines extracted in the previous two stages. The text line CM used for this purpose is $V_i^r + V_i^{s_w} + V_i^{s_h}$, namely a sum of three component CMs. Followed by a further verification step labelled as "Text Line Verification 3" in Fig.2, a decision can be made whether the target text line can be extracted in this stage, or the input image is rejected.

As for the above mentioned modules for the detection of special components, and the target text line extraction in special-layout images with underlines (e.g., Fig.1(d)), we offer a solution in [2] for underline detection and removal using multiple strategies. In this stage, after a final verification step labelled as "Text Line Verification 4" in Fig.2, we either output the extracted target text line, or reject the input image.

In the following, we elaborate on details of segmentation based text line CM and four verification modules.

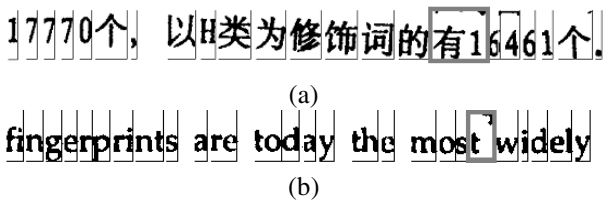

Figure 4. Effects of segmentation-based CMs due to (a) Assumption 1, (b) Assumption 2.

2.2. Segmentation-Based Text Line CM

Usually, if a text line includes some extra components or misses some components, positions of character segmentation of this text line will also change. Character segmentation result can thus inform us somehow whether a target text line is wrongly formed or not. This motivates us to develop segmentation based text line CMs. To do so, we make the following assumptions: 1) **Assumption 1**: For a normal text line, a single character's width is not greater than the text line's height; 2) **Assumption 2**: For a normal text line, a single character's height is not greater than the text line's height. In addition to character recognition result, our OCR engine [3] can also provide character segmentation information that is more reliable than the segmentation result obtained purely by vertical pixel projection. For the kth character image block hypothesized by our OCR engine in a text line L_i with a height h_i^p, let's use $w_b(k)$ and $h_b(k)$ to denote its width and height respectively. Then we define two block-level CMs as follows:

$$CM_i^{s_w}(k) = \begin{cases} 1, \text{if } w_b(k) \leq \lambda_1 \cdot h_i^p; \\ 0, \text{otherwise}; \end{cases}$$

$$CM_i^{s_h}(k) = \begin{cases} 1, \text{if } h_b(k) \leq \lambda_2 \cdot h_i^p; \\ 0, \text{otherwise}. \end{cases}$$

Here, text line height h_i^p is defined and estimated as follows: We sort heights of all N_i segmented blocks in the text line in descending order; then h_i^p is set to be the jth height, where $j = 30\%N_i$. Such an estimation of h_i^p is robust for curved text lines (e.g., Fig.1(b)) as well as text lines with small noisy components nearby (e.g., Figs.4(a)(b)). As for two control parameters λ_1 and λ_2, we set $\lambda_1 = 1$ and $\lambda_2 = \frac{4}{3}$ in practice. The setting of $\lambda_1 < \lambda_2$ is due to the fact that most single character's width is often smaller than its height. Based on the above, we define two text line level CMs as follows:

$$V_i^{s_w} = \frac{1}{N_i}\sum_{k=1}^{N_i} CM_i^{s_w}(k), \quad V_i^{s_h} = \frac{1}{N_i}\sum_{k=1}^{N_i} CM_i^{s_h}(k) \ .$$

The power of segmentation-based CMs is illustrated in Fig.4. In Fig.4(a), the highlighted block has a $CM_i^{s_w} = 0$, and the whole text line has a $V_i^{s_w} = 0.95$. In Fig.4(b), the highlighted block has a $CM_i^{s_h} = 0$, and the whole text line has a $V_i^{s_h} = 0.97$. The above CM values inform us that the two highlighted blocks are wrongly segmented, and the two text lines may be wrongly formed. However, for these two text lines, they both have a high recognition based text line CM, $V_i^r = 1$. The recognition based CM alone for these two examples does not work. Segmentation-based CMs can offer additional information to make a better decision.

2.3. Four Text Line Verification Modules

There are in total four verification modules in our approach. In each verification module, the recognition based text line CM defined in [1] is the most important one because the final goal is to recognize the target text line. The decision rule for kth verification module, where $k = 1, 2, 3, 4$, is defined as follows:

$$D_k = \begin{cases} Pass, \text{if } V_i^r \geq R_k \text{ and } V_i^{s_w} \geq SW_k \text{ and } V_i^{s_h} \geq SH_k; \\ Fail, \text{ otherwise}; \end{cases}$$

where R_k, SW_k and SH_k are three threshold parameters. In the following, we give the setting of these control parameters that work in practice.

In the module of *Text Line Verification 1*, we set $R_1 = 1$, $SW_1 = 0.80$, $SH_1 = 1$. It means that if all characters are recognized confidently, and above 80% segmented blocks are normal according to **Assumption 1**, and all segmented blocks are normal according to **Assumption 2**, the target text line can be extracted confidently; otherwise, further processing is needed. In the module of *Text Line Verification 2*, we set $R_2 = 0.9$, $SW_2 = 0.8$, $SH_2 = 0.8$. In the module of *Text Line Verification 3*, we set $R_3 = 0.8$, $SW_3 = 0$, $SH_3 = 0$. In the module of *Text Line Verification 4*, we set $R_4 = 0.66$, $SW_4 = 0$, $SH_4 = 0$.

In the above setting of control parameters, stricter thresholds are used in the earlier stage of verification. This is for reducing the risk of outputting wrong results prematurely without a careful check by using some more complicated strategies. Actually, segmentation-based CMs are not used in verification modules 3 and 4. This is for reducing the negative effects caused by possible inaccurate assumptions in some special cases, such as text lines with single or a small number of large characters. In the last two verification modules, lower thresholds are used for R_3 and R_4 to reduce the false rejection rate.

3. Experiments and Results

To show how our approach works, Fig.5 illustrates results of target text line extraction from the images in Fig.1. All target text lines are successfully extracted. We have also performed a benchmark test to verify the efficacy of our approach. To form a testing set, we collected totally 1287 document images by using C-Pen 10. All images are scanned from printed English/Chinese books, journals,

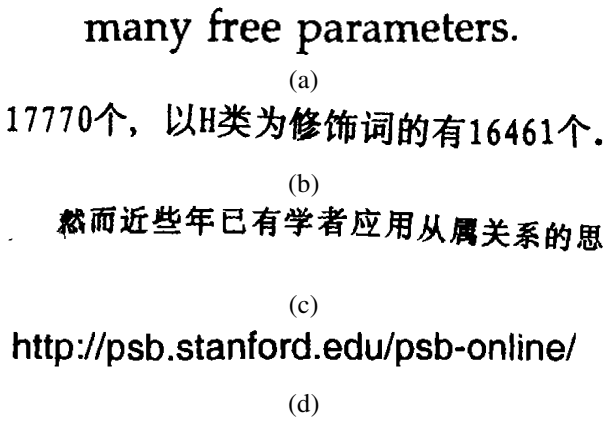

Figure 5. Results of target text line extraction from images in Fig.1.

newspapers, etc. We label manually these C-Pen images into 3 types. A *Type-1* image has only a single pure text line (e.g., Fig.1(a)). *Type-2* images are those with more than one text line (e.g., Figs.1.(b)(c)). *Type-3* images are garbage images that should be rejected (e.g., Fig.1(e)). Since special-layout images with underlines (e.g., Fig.1(d)) are not included in this testing set, the module of "Detection of Special Components" in Fig. 2 is bypassed in our experiments here. Detailed results for dealing with images with underlines are reported in a companion paper [2].

Table 1 summarizes the benchmark results for *positive* input images that include target text lines. All control parameters are set as the ones described in previous sections. Several observations can be made from the results. Firstly, for simple-layout images of Type-1, most of them (72.6%) can pass *text line verification 1* (labelled as TLV1 in Table 1), thus corresponding target text lines can be obtained in the most efficient way. Among the remaining images that need go through more complicated processing stages, 19.7% can pass *text line verification 2* (labelled as TLV2 in Table 1), and 7.7% can pass *text line verification 3* (labelled as TLV3 in Table 1). No image is falsely rejected. Secondly, for more complex layout images of Type-2, a good percentage of them (41.5%) can still be resolved by TLV1, while 51.6% can pass TLV2 and 6.5% can pass TLV3. Only one image is falsely rejected. Thirdly, among all images passed the verification, only 5 Type-2 images are identified by visual inspection to get wrong target lines. This gives an overall 99.5% accuracy of target text line extraction for positive images. As for garbage rejection capability, our approach can only reject 55.2% among 212 *Type-3* garbage images.

Table 1. Benchmark test results for positive input images. Number in () indicates the absolute number of relevant images.

Image Type	Pass TLV1	Pass TLV2	Pass TLV3	False Rej.	Correct Output
Type-1 (827)	72.6% (600)	19.7% (163)	7.7% (64)	0% (0)	100% (827)
Type-2 (248)	41.5% (103)	51.6% (128)	6.5% (16)	0.4% (1)	98.0% (243)
Total (1075)	65.4% (703)	27.1% (291)	7.4% (80)	0.1% (1)	99.5% (1070)

4. Discussions and Conclusion

In this paper, we have proposed a goal-oriented verification-based approach for target text line extraction from a document image captured by a pen scanner. Given the binary image, a series of processing steps are invoked adaptively, guided by the text line verification result in the preceding step. Each step adopts a strategy that is most effective for dealing with the problem concerned. Consequently, the target text line can be extracted in a more efficient and reliable way depending on the nature of the captured image. The above benchmark results demonstrate that our approach can achieve successfully the goal of target text line extraction for positive images. However, our approach can not perform very well for garbage rejection. For the pen scanner based applications of our interest, this technical limitation is not so serious for us to design a good user interface. From the usability point of view, once the user is in the loop of human-machine interaction, the goal of garbage rejection can be easily achieved by the user's inspection of the captured image. The problem caused by a poorly captured image can be simply resolved by a more careful re-scanning of the text line. Of course, improving capabilities of automatic garbage verification and rejection can offer opportunities to design a more intelligent user interface, thus can always be a topic for future researches. As a final remark, it is our belief that the framework proposed in this paper is general enough to be applied to other document image analysis and recognition applications.

References

[1] Z. L. Bai and Q. Huo, "An approach to extracting the target text line from a document image captured by a pen scanner," in *Proc. ICDAR-2003*, Edinburgh, August 2003, pp.I-76-80.

[2] Z. L. Bai and Q. Huo, "Underline detection and removal in a document image using multiple strategies," submitted to *ICPR-2004*, Cambridge, August 2004.

[3] Z. D. Feng and Q. Huo, "Confidence guided progressive search and fast match techniques for high performance Chinese/English OCR," in *Proc. ICPR-2002*, pp.III-89-92.

[4] http://www.cpen.com/

[5] G. Nagy and S. Seth, "Hierarchical representation of optically scanned documents," in *Proc. ICPR-1984*, 1984, pp.347-349.

[6] G. Nagy, "Twenty years of document image analysis in PAMI," *IEEE Trans. on PAMI*, Vol. 22, No. 1, pp.38-62, 2000.