Natural Language Inspired Approach for Handwritten Text Line Detection in Legacy Documents*

Vicente Bosch Campos
Inst. Tec. de Informática
Univ. Politécnica Valencia
Valencia - Spain
vbosch@iti.upv.es

Alejandro Héctor Toselli
Inst. Tec. de Informática
Univ. Politécnica Valencia
Valencia - Spain
ahector@iti.upv.es

Enrique Vidal
Inst. Tec. de Informática
Univ. Politécnica Valencia
Valencia - Spain
evidal@iti.upv.es

Abstract

Document layout analysis is an important task needed for handwritten text recognition among other applications. Text layout commonly found in handwritten legacy documents is in the form of one or more paragraphs composed of parallel text lines. An approach for handwritten text line detection is presented which uses machine-learning techniques and methods widely used in natural language processing. It is shown that text line detection can be accurately solved using a formal methodology, as opposed to most of the proposed heuristic approaches found in the literature. Experimental results show the impact of using increasingly constrained "vertical layout language models" in text line detection accuracy.

1 Introduction

Handwritten text transcription is becoming an increasingly important task, in order to provide historians and other researchers new ways of indexing, consulting and querying the huge amounts of historic handwritten documents which are being published in on-line digital libraries.

Transcriptions of such documents are currently obtained with solutions that range from the use of systems that aim at fully automatic handwritten text recognition (Bazzi et al., 1999)

(HTR), to computer assisted transcription (CATTI), were the users participate interactively in the proper transcription process (Toselli et al., 2009).

*Work supported under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018), MITTRAL (TIN2009-14633-C03-01) and also Univ. Politécnica Valencia (PAID-05-11)

The basic input to these systems consists of text line images. Hence, text line detection and extraction from a given document page image becomes a necessary preprocessing step in any kind of transcription systems. Furthermore the quality of line segmentation directly influences the final accuracy achieved by such systems.

Detection of handwritten text lines in an image entails a greater difficulty, in comparison with printed text lines, due to the inherent properties of handwritten text: variable inter-line spacing, overlapping and touching strokes of adjacent handwritten lines, etc.

The difficulty is further increased in the case of ancient documents, due to common problems appearing in them: presence of smear, significant background variations and uneven illumination, spots due to the humidity, and marks resulting from the ink that goes through the paper (generally called "bleed-through").

Among the most popular state-of-the art methods involved in handwritten text line detection we find four main families: based on (vertical) projection profiles (Likforman-Sulem et al., 2007), on the Hough transform (Likforman-Sulem et al., 1995), the repulsive-attractive network approach (Öztop et al., 1999) and finally the so-called stochastic methods (Vinciarelli et al., 2004), which combine probabilistic models such as Hidden Markov Models (HMMs) along with dynamic programming techniques (e.g. Viterbi algorithm) to derive optimal paths between overlapping text lines.

It is worth noting that, most of the mentioned approaches somewhat involve heuristic adjustments of their parameters, which have to be properly tuned according to the characteristics of each

Proceedings of the 6th EACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pages 107–111,
Avignon, France, 24 April 2012.

task in order to obtain adequate results.

In this work, the text line detection problem in legacy handwritten documents is approached by using machine-learning techniques and methods which are widely used in natural language processing (NLP).

It is shown that the text line detection problem can be solved by using a formal methodology, as opposed to most of the currently proposed heuristic based approaches found in the literature.

2 Statistical Framework for Text Line Detection

For the work presented in this paper, we assume that the input image (of a page or selected region) contains one or more paragraphs of single-column parallel text with no images or diagram figures. Additionally, we assume that the input image has been properly preprocessed so as to ensure that their text lines are roughly horizontal. These assumptions are reasonable enough for most legacy handwritten documents.

Similarly to how the statistic framework of automatic speech recognition (ASR) is established, the handwritten text line detection problem can be also formulated as the problem of finding the most likely text lines sequence, $\hat{\mathbf{h}} = \langle h_1, h_2, \ldots, h_n \rangle$, for a given handwritten page image represented by a sequence of observations[1] $\mathbf{o} = \langle o_1, o_2, \ldots, o_m \rangle$, that is:

$$\hat{\mathbf{h}} = \arg\max_{h} P(\mathbf{h} \mid \mathbf{o}) \qquad (1)$$

Using the Bayes' rule we can decompose the probability $P(\mathbf{h} \mid \mathbf{o})$ into two terms:

$$\hat{\mathbf{h}} = \arg\max_{h} P(\mathbf{o} \mid \mathbf{h}) \cdot P(\mathbf{h}) \qquad (2)$$

In the jargon of NLP these probabilities represent the morphological and syntactic knowledge levels, respectively. As it happens in ASR, $P(\mathbf{o} \mid \mathbf{h})$ is typically approximated by HMMs, which model vertical page regions, while $P(\mathbf{h})$ by a "language model" (LM), which restricts how those regions are composed in order to form an actual page. In what follows, a detailed description of this modelling scheme is given.

[1] Henceforward, in the context of this formal framework, each time it is mentioned image of *page or selected text*, we are implicitly referring to the input feature vector sequence "$\mathbf{o}$" describing it.

2.1 Modelling

In our line detection approach four different kinds of vertical regions are defined:

Blank Line-region (BL): Large rectangular region of blank space usually found at the start and the end of a page (top and bottom margins).

Normal text Line-region (NL): Region occupied by the main body of a normal handwritten text line.

Inter Line-region (IL): Defined as the region found within two consecutive normal text lines, characterized by being crossed by the ascenders and descenders belonging to the adjacent text lines.

Non-text Line-region (NT): Stands for everything which does not belong to any of the other regions.

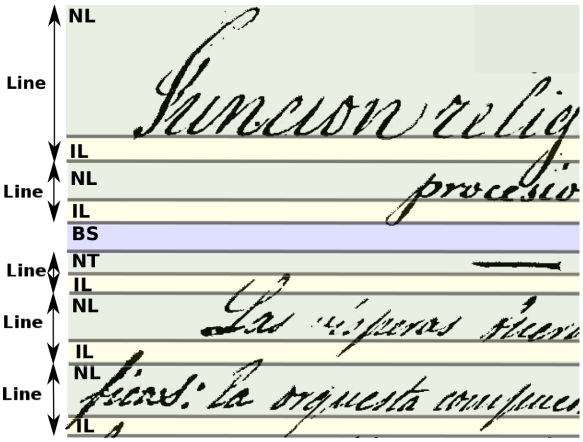

Figure 1: Examples of the different kind of line-regions.

We model each of these regions by an HMM which is trained with instances of such regions. Basically, each line-region HMM is a stochastic finite-state device that models the succession of feature vectors extracted from instances of this line-region image. In turn, each HMM state generates feature vectors following an adequate parametric probabilistic law; typically, a mixture of Gaussian densities. The adequate number of states and Gaussians per state may be conditioned by the available amount of training data.

Once an HMM "topology" (number of states and structure) has been adopted, the model parameters can be easily trained from instances (sequences of features vectors) of full images containing a sequence of line-regions (without any

kind of segmentation) accompanied by the reference labels of these images into the corresponding sequence of line-region classes. This training process is carried out using a well known instance of the EM algorithm called forward-backward or Baum-Welch re-estimation (Jelinek, 1998).

The syntactic modelling level is responsible for the way that the different line regions are composed in order to produce a valid page structure. For example we can force that NL and NT line regions must always be followed by IL inter-line regions: NL+IL and NT+IL. We can also use the LM to impose restrictions about the minimum or maximum number of line-regions to be detected. The LM for our text line detection approach, consists in a stochastic finite state grammar (SFSG) which recognizes valid sequences of elements (line regions): NL+IL, NT+IL and BL.

Both modelling levels, morphological and syntactical, which are represented by finite-state automaton, can be integrated into a single global model on which Eq. (2) is easily solved; that is, given an input sequence of raw feature vectors, an output string of recognized sequence of line-region labels is obtained. In addition the vertical position of each detected line and and line-region is obtained as a by-product.

3 System Architecture

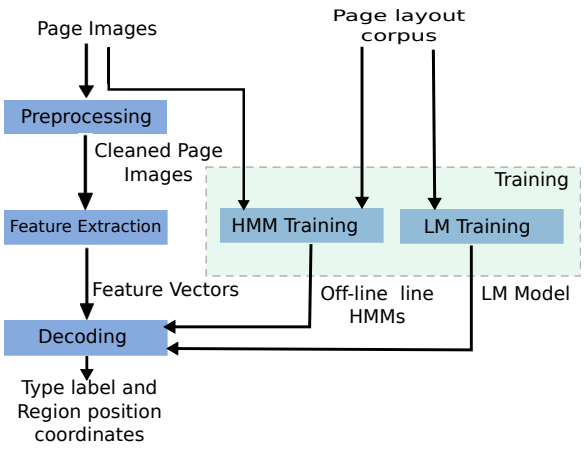

Figure 2: Global scheme of the handwritten text line detection process.

The flow diagram of Fig. 2 displays the overall process of the proposed handwritten text line detection approach. It is composed of four different phases: image preprocessing, feature extraction, HMMs and LM training and decoding. Next we will overview the first two phases, preprocessing and feature extraction, since the rest has already been covered in the preceding section.

3.1 Preprocessing Phase

Initially performing background removal and noise reduction is carried out by applying a bi-dimensional median filter on them. The resulting image skew is corrected by applying vertical projection profile and RLSA (Wong and Wahl, 1982), along with standard techniques to calculate the skew angle.

3.2 Feature Extraction Phase

As our text line detection approach is based on HMMs, each preprocessed image must be represented as a sequence of feature vectors.This is done by dividing the already preprocessed image (from left-to-right) into D non-overlapping rectangular regions with height equal to the image-height (see Fig. 3).

In each of these rectangular regions we calculate the vertical grey level histogram. RLSA is applied to obtain a more emphasized vertical projection profile. Finally, to eliminate local maxima on the obtained vertical projection profiles, they are smoothed with a rolling median filter (Manmatha and Srimal, 1999) (see Fig. 3) . In this way,

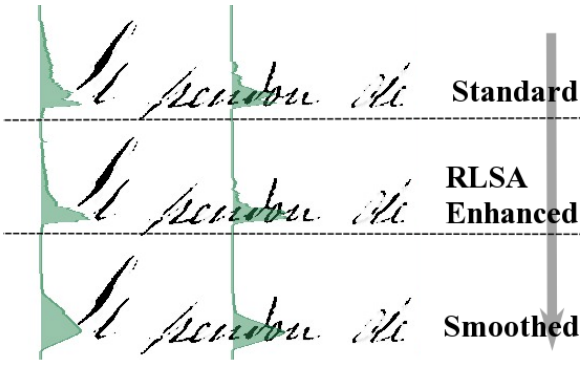

Figure 4: Review of the impact of the RLSA and rolling media filter on the histogram calculation of a sample line.

a D-dimensional feature vector is constructed for each page/block image pixels row, by stacking the D projection profile values corresponding to that row. Hence, at the end of this process, a sequence of L D-dimensional feature vectors is obtained, where L is the image height.

4 Experimental Setup and Results

In order to study the efficacy of the line detection approach proposed in this paper, different experiments were carried out. We are mainly interested in assessing the impact upon final text line detec-

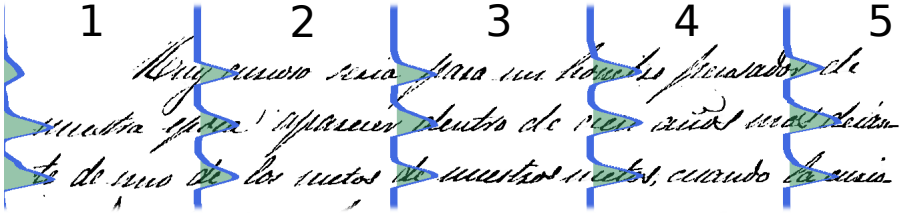

Figure 3: Partial page image visualization of 5 ($D = 5$) rectangular regions across over 3 handwritten text lines. For each region, its vertical projection profile is also plotted.

tion accuracy of employing increasingly restrictive LMs.

4.1 Corpus Description

Experiments are carried out with corpus compiled from a XIX century Spanish manuscript identified as "Cristo-Salvador" (CS), which was kindly provided by the *Biblioteca Valenciana Digital* (BiVaLDi)[2]. This is a rather small document composed of 53 colour images of text pages, scanned at 300 dpi and written by a single writer. Some page images examples are shown in Fig. 5.

Figure 5: Examples of pages images from CS corpus.

In this work we employ the so-called *book* partition, which has been defined for this dataset (Romero et al., 2007). Its test set contains the last 20 page images were as the training set is composed of the 33 remaining pages. Table 1 summarizes the relevant information of this partition.

Table 1: Basic statistics of the Cristo-Salvador corpus partition.

Number of:	Training	Test	Total
Pages	33	20	53
Normal-text lines (NL)	685	497	1 182
Blank Lines (BL)	73	70	143
Non-text Lines (NT)	16	8	24
Inter Lines (IL)	701	505	1 206

Each page was annotated with a succession of reference labels (NL, NT, BL and IL) indicating the kind of line-regions that composed it. Such references were generated by executing standard methods for text line detection based on vertical projection profiles, which were afterwards manually labelled, verified, adjusted and/or rectified by a human operator to ensure correctness.

4.2 Evaluation Measures

We measure the quality of the text line detection by means of the "line error rate" (LER) which is performed by comparing the sequences of automatically obtained region labels with the corresponding reference label sequences. The LER is computed in the same way as the well known WER, with equal costs assigned to deletions, insertions and substitutions (McCowan et al., 2004).

4.3 Experiments and Results

A series of experiments were performed on the CS corpus using a simple hold-out validation as per the CS "book" partition. Initially some parameters were set up: feature extraction dimension D, HMM topology (number of states and Gaussians),number of Baum-Welch iterations, and decoding grammar scale factor (GSF) and word insertion penalty (WIP). After some informal experimentation, adequate values were found for several of them: feature vectors dimension of 2, left-to-right HMMs with 4 states topology, 32 Gaussian mixtures per state trained by running 3 cycles of Baum-Welch re-estimation algorithm. The remaining parameters, all related with the decoding process itself, were tuned to obtain the best figures for each of the two following language models: the *prior* and *conditional* represented by topologically different SFSGs. The *prior* model transition probabilities are estimated from the training set as the fraction of the number of appearances of each vertical region label over the whole count of labels. The conditional model also considers the previous label in order to perform the estimation. These estimates resemble the uni-gram and

[2] http://bv2.gva.es.

bi-gram LMs calculations, except no smoothing strategy is implemented here.

Additionally, it is defined for each test page a *line-number constrained* LM which uses the *conditional* probabilities to populate the model but enforces a total number of possible line-regions to detect as per the number of reference line-region labels of that test page. Table 2 reports the obtained LER results for each of these LMs.

Table 2: Best detection LER(%) obtained for each kind of language model: Prior, Conditional and Line-Number Constrained.

LM	WIP	GSF	LER(%)
Prior	-32	8	0.86
Conditional	-8	16	0.70
LN-Constrained	-128	1	0.34

As can be seen, the more restrictive the LM is, the better accuracy is achieved. Concerning the *line-number constrained*, they are really conceived for its utilization in (parts of) documents or document collections with homogeneous numbers of lines per page.

5 Conclusions

We have presented a new approach for text line detection by using a statistical framework similar to that already employed in many topics of NLP. It avoids the traditional heuristics approaches usually adopted for this task.

The accuracy of this approach is similar to or better than that of current state of the art solutions found in the literature. We find that the detected baselines provided by our approach are of better quality (visually closer to the actual line) than current heuristic methods as can be seen in 6.

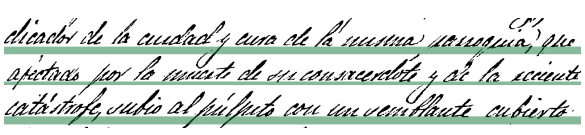

Figure 6: Image shows the difference between our proposed method (upper side of each coloured region) and the histogram projection method (lower side)

In the future we will extend this approach not only to detect, but also to classify line-region types in order to determine for example titles, short lines, beginning and and end of paragraphs, etc. Furthermore, it is envisioned that the proposed stochastic framework serves as a cornerstone to implementing interactive approaches to line detection similar to those used for handwritten text transcription used in (Toselli et al., 2009).

References

Issam Bazzi, Richard Schwartz, and John Makhoul. 1999. An omnifont open-vocabulary OCR system for English and Arabic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):495–504.

Frederick Jelinek. 1998. *Statistical methods for speech recognition*. MIT Press.

Laurence Likforman-Sulem, Anahid Hanimyan, and Claudie Faure. 1995. A hough based algorithm for extracting text lines in handwritten documents. *Document Analysis and Recognition, International Conference on*, 2:774.

Laurence Likforman-Sulem, Abderrazak Zahour, and Bruno Taconet. 2007. Text line segmentation of historical documents: a survey. *International Journal on Document Analysis and Recognition*, 9:123–138, April.

Raghavan Manmatha and Nitin Srimal. 1999. Scale space technique for word segmentation in handwritten documents. In *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, SCALE-SPACE '99, pages 22–33, London, UK. Springer-Verlag.

Iain A. McCowan, Darren Moore, John Dines, Daniel Gatica-Perez, Mike Flynn, Pierre Wellner, and Hervé Bourlard. 2004. On the use of information retrieval measures for speech recognition evaluation. Idiap-RR Idiap-RR-73-2004, IDIAP, Martigny, Switzerland, 0.

Verónica Romero, Alejandro Héctor Toselli, Luis Rodríguez, and Enrique Vidal. 2007. Computer Assisted Transcription for Ancient Text Images. In *International Conference on Image Analysis and Recognition (ICIAR 2007)*, volume 4633 of *LNCS*, pages 1182–1193. Springer-Verlag, Montreal (Canada), August.

Alejandro Héctor Toselli, Verónica Romero, Moisés Pastor, and Enrique Vidal. 2009. Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1824–1825.

Alessandro Vinciarelli, Samy Bengio, and Horst Bunke. 2004. Off-line recognition of unconstrained handwritten texts using hmms and statistical language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):709–720, june.

Kwan Y. Wong and Friedrich M. Wahl. 1982. Document analysis system. *IBM Journal of Research and Development*, 26:647–656.

Erhan Öztop, Adem Y. Mülayim, Volkan Atalay, and Fatos Yarman-Vural. 1999. Repulsive attractive network for baseline extraction on document images. *Signal Processing*, 75(1):1 – 10.