Pattern Recognition Letters 35 (2014) 34–45

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Model-based ruling line detection in noisy handwritten documents

Jin Chen *, Daniel Lopresti

Computer Science & Engineering, Lehigh University, Bethlehem, PA 18015, United States

ARTICLE INFO

Article history:
Available online 15 September 2012

Keywords:
Ruling line modeling
Ruling line detection
Performance evaluation
Document image analysis

ABSTRACT

Ruling lines are commonly used to help people write neatly on paper. In document analysis, however, they raise hurdles for the tasks of handwriting recognition or writer identification. In this paper, we model ruling line detection as a multi-line linear regression problem and then derive a globally optimal solution under the Least Squares Error. For performance evaluation, we compute the error statistics on the model attributes and also employ human correction of algorithmic results for performance evaluation, instead of using pixel-level performance measures. We demonstrate the effectiveness of our method on three datasets, including modern and historic document images. Specifically, we obtained 95% accuracy in detecting ruling lines in a modern handwriting dataset with 100 documents. Under an interactive evaluation framework, the new algorithm showed performance gains over one existing approach.

1. Introduction

Line processing is needed in various document analysis applications, e.g., forms/invoice processing (Liu et al., 1995; Chhabra et al., 1995; Yu and Jain, 1996; Cesarini et al., 1998; Hori and Doermann, 1995; Chen and Lee, 1998; Tseng and Chen, 1998; Ting and Leung, 1999; Liu and Jain, 2000; Fan et al., 1998; Zheng et al., 2001), engineering drawing processing (Dori et al., 1993; Arias et al., 1995, 1997; Dori and Liu, 1999), music score analysis (Roach and Tatem, 1988; Carter and Bacon, 1992), and off-line handwriting analysis (Zheng and Doermann, 2003; Arvind et al., 2007; Abd-Almageed et al., 2009; Cao and Govindaraju, 2007; Cao et al., 2007). Many techniques work well on relatively clean images of good quality (Tseng and Chen, 1997; Chen and Lee, 1998; Fan et al., 1998; Arvind et al., 2007). However, if lines are severely broken due to low image resolution or they are overlapped by other components, the performance can be significantly degraded. Cao and Govindaraju introduced a method of processing low-resolution noisy medical forms (Cao and Govindaraju, 2007).

In a particular application, prior knowledge can be helpful in designing specific algorithms (Zheng et al., 2005). Unlike in the other applications, pre-printed ruling lines on paper sheets exhibit a simple but strongly correlated pattern:

(1) Ruling lines are parallel straight lines.
(2) They have consistent spacing, lengths, and thickness.

On the other hand, since people usually make use of ruling lines when they are present, separating handwriting that overlaps ruling lines can be a significant challenge. Fig. 1 shows two sample documents used in our experiments.

The protocol of performance evaluation uses either pixel-level metrics or object-level ones. Pixel-level metrics including *precision*, *recall*, and *F-Score* are intuitive measurements for performance evaluation. However, ground-truthing at pixel level is difficult because pixel-level judgement is subjective and this situation becomes more severe when lines are degraded. On the other hand, researchers have presented several object-level metrics (Kong et al., 1996; Hori and Doermann, 1996; Liu and Dori, 1997; Phillips and Chhabra, 1999; Zheng and Doermann, 2003). Although these compound metrics are designed to incorporate meaningful components, some can be difficult to show how significantly the performance differs among algorithms. For example, Liu and Dori design one object-level metric for evaluating performance for engineering drawing processing (Liu and Dori, 1997):

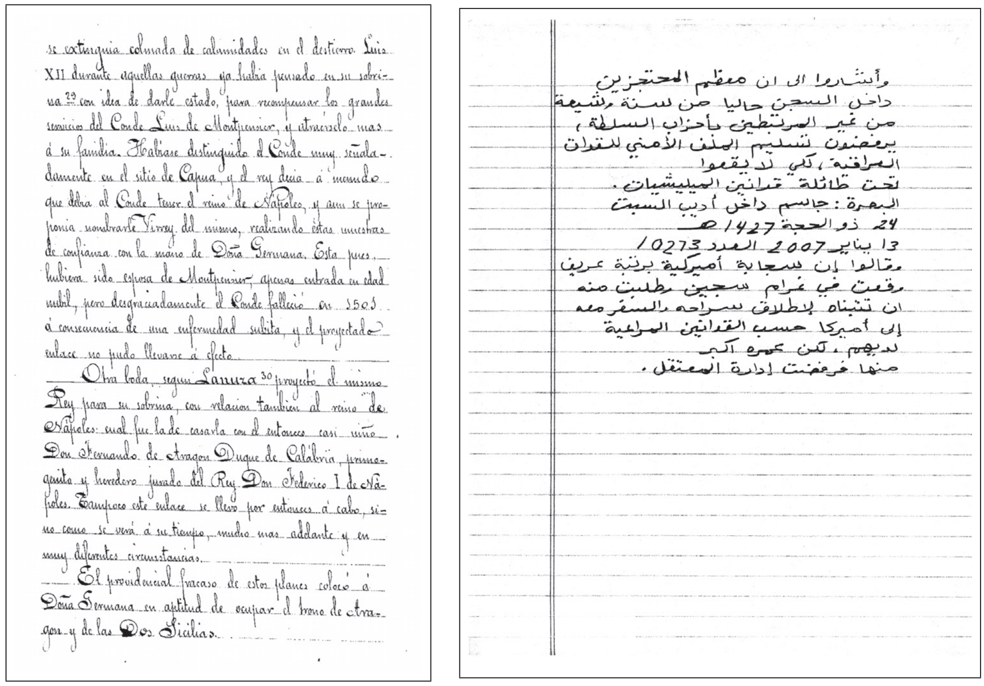
$$Q_v(c) = [Q_{pt}(c)\ Q_{od}(c)\ Q_w(c)\ Q_{st}(c)\ Q_{sh}(c)]^{1/5}, \qquad (1)$$

where the vector detection quality $Q_v(c)$ is a weighted product of five factors: end point quality $Q_{pt}(c)$, overlap distance quality $Q_{od}(c)$, line width quality $Q_w(c)$, line style quality $Q_{st}(c)$, and line shape quality $Q_{sh}(c)$. Suppose two algorithms' Q_v-values differ by 0.1, we still do not know how significantly the differences are. Alternatively, researchers also use the performance of downstream applications for evaluation, such as *word error rates* (WERs) for handwriting recognition (Cao and Govindaraju, 2007; Cao et al., 2007).

In this work, we introduce a model-based ruling line detection algorithm that takes advantages of the model properties of ruling

* Corresponding author.
E-mail addresses: jic207@cse.lehigh.edu (J. Chen), lopresti@cse.lehigh.edu (D. Lopresti).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.08.008



(a) A sample in Germana. (b) A sample in Madcat.

Fig. 1. Sample documents used in our experimental evaluation.

lines. Next, we present the framework of multi-line linear regression and derive a globally optimal solution under the Least Squares Error (LSE). Then we describe an effective Hough transform variant for extracting line segments and the adaptive "Basic Sequential Algorithmic Scheme" (BSAS clustering) to group line segments. The next step makes use of the ruling line properties to detect lines that are missed by the Hough transform. Finally we employ the multi-line linear regression to estimate the model parameters. For performance evaluation, we choose to compute the error statistics of the model attributes individually, rather than defining a single metric. We consider this an effective way of showing how the algorithm performs in different aspects, and indicating what future improvement can be made. In addition, we evaluate performance by measuring the effort needed for a human subject to correct algorithmic errors. To do that, we show a human subject a GUI that enables him/her to interactively correct errors in algorithmic outputs.

The remainder of this paper is organized as follows: in Section 2, we survey approaches for line processing in related applications and existing performance measures in the literature. Next, in Section 3 we provide some background knowledge on linear regression and then introduce the multi-line linear regression model. Then we introduce the ruling line detection algorithm in Section 4 and the experimental evaluation in Section 5. Finally we show experimental results in Section 6 and then conclude in Section 7.

2. Related work

2.1. Line processing

In engineering drawing processing, there exists two methodologies: thinning based (Tamura, 1978; Nagasamy and Langrana, 1990) and medial line extraction based methods (Monagan and Roosli, 1993; Nagasamy and Langrana, 1985; Dori, 1998; Dori and Liu, 1999). Also, there exists work attempting to combine these two methodologies, as in (Hori and Tanigawa, 1993). For music score analysis, *staff lines* are critical for recognizing notes and pitches (d'Andecy et al., 1994). In Roach and Tatem's work, they detected staff lines using a sliding window (Roach and Tatem, 1988). As a run-length based approach, Carter and Bacon presented a Line Adjacency Graph (LAG) method (Carter and Bacon, 1992). Their algorithm was able to handle difficult situations where a symbol tangentially intercepted with the staff lines. d'Andecy et al., attempted to segment music scores into four detectable layers (d'Andecy et al., 1994). To do that, they employed the Kalman filter to separate these layers which was robust to scaling, curvature, and noise in music score images.

Forms/Invoice processing consists of documents without handwriting (Liu et al., 1995; Tseng and Chen, 1997) and those with handwriting (Yu and Jain, 1996; Yoo et al., 1997; Ye et al., 2001; Cao and Govindaraju, 2009). In (Yu and Jain's work, 1996), they presented a block adjacency graph (BAG) method to detect form frame lines. Ye et al., used the morphological *opening* operation with linear shape *structure elements* on foreground pixels to remove frame lines that are longer than a pre-defined length (Ye et al., 2001). Then, to restore information that was removed by the line removal processing, they employed a *closing* operation with a dynamic structure element for different orientations (90°, 45°, and 135°). Given a known form, Cao and Govindaraju applied a template matching method to locate and mask the horizontal ruling lines on low-quality handwritten carbon forms (Cao and Govindaraju, 2009).

Line processing is also necessary for off-line handwriting recognition. Arvind et al., introduced a rule-based method that first detected the ruling lines within segmented handwritten blocks by computing the horizontal projection profiles (Arvind et al., 2007). Zheng et al. (2005) presented a stochastic model-based ruling line detection algorithm that incorporated context to detect ruling lines systematically. Using a vectorization based method called "Directional Single-Connected Chain" (DSCC), the authors separated most handwriting from a set of line segments (Zheng et al., 2001). Rather than treating the peaks on the projection profile as the line positions, they modeled the profile with a Hidden Markov Model