Layout analysis of handwritten letters based on textural and spatial information and a 2D Markovian approach

Mélanie Lemaître1 Emmanuèle Grosicki2 Edouard Geoffrois2 Françoise Prêteux3
mel.lemaitre@wanadoo.fr Emmanuele.Grosicki@etca.fr Edouard.Geoffrois@etca.fr Francoise.Preteux@it-sudparis.eu

1 ULCO/LIL 50, rue Ferdinand Buisson 93056 Paris cedex 19 France
2 DGA/Centre d’Expertise Parisien 16 bis avenue Prieur de la Côte d’Or 94114 Arcueil cedex - France
3 TELECOM Sud Paris, Dpt ARTEMIS 9, rue Charles Fourier 91011 Evry cedex - France

Abstract

This paper addresses the problem of layout analysis of handwritten letters using textural and spatial information with a bidimensional Markovian approach. In this framework, the layout extraction is viewed as a labeling problem which aims to find the optimal configuration of the Markov Random Field performed by the 2D Dynamic Programming method [5]. Experiments have been carried out on the training set of the RIMES database [1] and a rigorous error analysis leads to several improvements. An originality of this work lies in the participation to the first RIMES evaluation campaign: an error rate of 11% has been obtained on the test set.

Keywords: layout analysis, handwritten letters, Markov Random Fields, textural and spatial information, RIMES.

1. Introduction

To perform an automatic processing of unconstrained documents a preliminary step of physical layout extraction is necessary to extract relevant information. This task is still a challenging issue especially when considering handwritten data. Many methods have been proposed to deal with a wide variety of machine-printed documents [8, 12] whereas few methods are dedicated to handwritten documents because of the great variability of handwriting. Existing solutions focus mainly on specific type of documents where the layout is well-constrained such as bank checks, forms or postal addresses. Nevertheless, some unconstrained handwritten document systems have been recently proposed [9, 3].

This article focuses on handwritten letters similar to those sent by individuals to companies or administrations. The final goal is to extract information like the sender identity which are useful for an automatic processing of these letters (see figure 1).

In order to extract the layout of handwritten letters, we have proposed [6] to consider only information about the texture and the spatial position of labels. To that end, we use an algorithm based on a bidimensional Markovian approach. In this framework, the layout extraction is considered as a labeling problem which aims to find the optimal configuration of the Markov Random Field performed by the 2D Dynamic Programming method [5]. First encouraging results have been obtained with 80 letters (an error rate of 15% was reached [6]).

The system presented in this article is tested on part of the RIMES database made of 1150 letters. The paper presents a rigorous error analysis which has allowed us to propose several improvements and gives the results obtained during the first RIMES evaluation campaign which happened in June 2007.

This paper is organized as follows. Section 2 introduces the task, defines the goals and presents the RIMES database and the metric used to compare the layout extraction results obtained by the proposed algorithm to ground-truths. The Section 3 is dedicated to a brief presentation of the algorithm where the chosen approach and model are detailed. The fourth Section gives a rigorous analysis of errors with some improvements. The Section 5 shows the obtained results during the first RIMES evaluation campaign after having described the organization of the evaluation. Finally the last Section draws a conclusion of this work.

2. Layout extraction of handwritten letters

2.1. Goals

In the framework of handwritten letters layout extraction, seven labels have been defined: sender, destination, subject, date and place, opening, body of the letter and signature (see figure 1). The layout extraction task consists in delimiting handwriting blocks in each letter corresponding to these different labels by using bounding boxes, which amounts to labeling each pixels of the image.

The letters considered in the article belong to the RIMES
2.2. The RIMES database

Our method uses the RIMES database, first huge handwritten letters database publicly available. The RIMES project (recognition and indexing of handwritten documents and faxes) intends to evaluate systems dedicated to the recognition and the indexing of handwritten letters sent by postal mail or fax by individuals to companies or administrations [1, 10].

A main goal of the RIMES project is to collect a huge database of handwritten mails. For legal and confidentiality reasons it was not possible to use existing materials. Therefore those letters were collected among volunteer writers thanks to the SCRIBEO’s collect [11]. Volunteers were given a fictional identity and a scenario to write their letter. The other constraint was to write this letter on a white sheet with black or blue ink.

The first version of the RIMES database used for the first evaluation campaign and for our experiments is described in Table 1. The ground-truths of the training set were available whereas those of the testing set were not available before the end of the evaluation. We can notice that a larger database (5605 images) will be soon available.

Table 1. Description of the first version of the RIMES database

<table>
<thead>
<tr>
<th>Training set</th>
<th>Testing set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development set</td>
<td>Validation test</td>
</tr>
<tr>
<td>950</td>
<td>100</td>
</tr>
</tbody>
</table>

2.3. Metric

To compare the layout extraction results obtained by our algorithm to ground-truths, a metric based on grey-level is used. The idea is that errors made on pixels that contain no writing are meaningless for the task. The metric compares labels of each pixels in both hypothesis and ground-truth, and evaluate the grey-level weighted pixel error rate $\hat{Err}$ defined by:

$$
\hat{Err} = \frac{\sum \text{misclassified pixels} \ (255 - I(i,j))}{\sum \text{all the pixels} \ (255 - I(i,j))},
$$

where $I(i,j)$ is the grey-level at the pixel $(i,j)$ of the image $I$.

3. A bidimensional Markovian approach

Layout analysis of unconstrained handwriting documents is still a challenging issue and few articles on the subject are available in the literature. Most of them are based on Markov Random Fields [9]. Indeed they are known to successfully model contextual information in image labeling problems [7] whose goal consists in classifying regions of pixels into a certain number of labels from their observations (scalar or vector).

3.1. Markovian model

According to the Markovian formalism, an $x \times y$ image can be associated to an $n \times m$ grid of sites $s_{i,j}$ whose observations $o_{i,j}$ are produced by hidden states $\omega_{i,j}$ with $1 \leq i \leq m$, $1 \leq j \leq n$ and $\omega_{i,j} \in \{1,...,L\}$ where $L$ is the number of states. The main idea of the approach is then to create for the handwritten letter layout a bidimensional Markovian model linking hidden states $\omega_{i,j}$ to features $o_{i,j}$ extracted from images.

Figure 2. Markov Random Fields with a first order neighborhood

In this Markovian context, the labeling problem of an image aims to find the optimal configuration $\hat{\omega}$ of the states grid among all the possible configurations $\Omega$ knowing the observations $O$. A Bayesian approach is then used:
\[
\hat{\omega} = \arg \max_{\omega \in \Omega} P(\omega|O) \\
= \arg \max_{\omega \in \Omega} P(O|\omega)P(\omega),
\]
where \(O = \{ o_{i,j} \} \) and \( \omega = \{ \omega_{i,j} \} \).

To the classical contextual model introduced by Markov Random Fields, we propose to add a model of position based on some spatial considerations. For example, the label “signature” is more likely to appear at the bottom of the image than at the top. As a consequence, \( P(\omega) \) can be split up into two terms: \( P^T(\omega) \) the probability of transition and \( P^P(\omega) \) the probability of position, \( i.e. \):
\[
P(\omega) = P^T(\omega)P^P(\omega).
\]

The parameters of the model are those of \( P(O|\omega) \), \( P^T(\omega) \) and \( P^P(\omega) \), they are trained on the ground-truths of the training set. The way they are computed is detailed in next sections.

### 3.1.1. Density of observations

Observations are classically modeled by Gaussian mixtures:
\[
P(o|\omega) = \sum_{i=1}^{M} k_i G(o, \mu_i, \Sigma_i, \omega),
\]
where \( M \) is the maximum number of Gaussians, \( G(o, \mu, \Sigma) \) is the value in \( o \) of a Gaussian function of mean \( \mu \) and covariance matrix \( \Sigma \) (in practice, a diagonal matrix), and where \( \sum_{i=1}^{M} k_i = 1 \).

### 3.1.2. Position model

The probability of spatial position \( P^P(\omega) \) is defined by:
\[
P^P(\omega) = \prod_{i,j} P(\omega_{i,j} \mid (i,j)),
\]
where \( P(\omega_{i,j} \mid (i,j)) \) are modeled by Gaussian mixtures of dimension 2: one dimension for the horizontal information and one dimension for the vertical information.

### 3.1.3. Transition model

According to the Markovian assumption of local dependency between states, a first order neighborhood \( N_{i,j} = \{ \omega_{i-1,j}, \omega_{i+1,j}, \omega_{i,j-1}, \omega_{i,j+1} \} \) is used, so \( P^T(\omega) \) can be expressed by:
\[
P^T(\omega) = \prod_{i,j} P(\omega_{i,j} \mid \omega_{k,l}, (k,l) \in N_{i,j}).
\]

With this formalism it is possible to use the Gibbs distributions which are equivalent to a Markov Random Fields [2]:
\[
P^T(\omega) = \frac{1}{Z} \exp(-\sum_{c \in C} V_c(\omega)),
\]
where \( C \) is the set of cliques associated to the first order neighborhood, \( V_c \) is a potential function associated to cliques \( c \) and \( Z \) is a normalisation constant so that \( \sum_{\omega} P^T(\omega) = 1 \).

As we work on a first order neighborhood the associated cliques are defined by:
\[
C_0 = \{(i, j), 1 \leq i \leq m \text{ et } 1 \leq j \leq n \}
\]
\[
C_1 = \{(i, j), (i + 1, j), 1 \leq i \leq m - 1 \text{ et } 1 \leq j \leq n \}
\]
\[
C_2 = \{(i, j), (i, j + 1), 1 \leq i \leq m \text{ et } 1 \leq j \leq n - 1 \}.
\]

The clique potential functions defined in [4] are used:
\[
V_c(\omega) = \begin{cases} -\log(P(\omega_k)) & \text{if } c \in C_0 \\ -\log(I_v(\omega_k, \omega_l)) & \text{if } c \in C_1 \\ -\log(I_h(\omega_k, \omega_l)) & \text{if } c \in C_2 \end{cases}
\]
where \( I_v \) and \( I_h \) are respectively vertical and horizontal interaction terms:
\[
I_v(\omega_k, \omega_l) = \frac{P\left(\frac{\omega_k}{\omega_l}\right)}{P(\omega_k)P(\omega_l)},
\]
\[
I_h(\omega_k, \omega_l) = \frac{P\left(\frac{\omega_l}{\omega_k}\right)}{P(\omega_k)P(\omega_l)}.
\]

### 3.2. Labeling

The decoding of the optimal configuration \( \hat{\omega} \), \( i.e. \) the maximization of \( P(O|\omega)P(\omega) \), is performed by the 2D Dynamic Programming [5].

Knowing the labels of all the sites, the labeled image will be easily inferred by copying the label of the nearest site to each pixel in the image.

### 3.3. Extraction of Features

The feature extraction procedure is an important step as it is directly linked to the observations associated to each site. The proposed features are computed from a sequence of overlapping windows centered on the grid of sites and correspond to the average grey-level of pixels of each window.

The feature parameters are: the size of the window, the sampling step and the number of sites. Classically the sampling step is chosen equal to one third of the size of...
the window. As images have different size according to their resolution, in order to extract the same amount of information the number of sites \( N \) is set equal for all the images. As a result the size of the window is adapted to the size of the analysed image.

In future work, a multi-resolution feature could be used to improve the modeling of observations as proposed in [9].

3.4. First results

An error rate of 15\% has been achieved on the training set of the RIMES database [6]. The number of sites has been chosen equal to 60 × 85. Indeed we have shown [6] that it was the best trade-off between time of computation and performance. An example of results, at this level, is given figure 3.

\[ \text{Figure 3. Example of an image of handwritten letter labeled with our algorithm.} \]

An error analysis is then considered in order to propose different possible improvements.

4. Error analysis

4.1. First class of error

By analysing the different errors generated by our algorithm, we can first notice that some labels are not as well-recognized as others. This point is illustrated figure 3 where one can see that almost all the pixels are well-labeled except those corresponding to the label “opening”. This can be explained by the fact that this label does not appear often in the training set (see figure 4) and as a consequence generates errors in the test phase. Furthermore, most of the errors are done on spatial configurations which don’t appear in the training set.

To cover these problems two solutions exist:

- A bigger database could increase the variability of data.
- To improve the detection of under-represented labels, we can imagine to add a supplementary step of word recognition in the labeling procedure. This idea could be limited to an extraction of key-words like “Dear, Madam, Sir...” which will be used to refine the labels attribution.

4.2. Second class of error

A second kind of error is made on the label “date, place” which is often confused with the label “sender”. This is the case figure 5. We can be explained it by the fact that these two labels are spatially very close.

\[ \text{Figure 5. Example of confusion between the two labels “date, place” and “sender”.} \]

One way to cope with this problem would be to consider, as for the first class of error, a preliminary step of key-word recognition: for example the detection of a zip-code could improve the probability of the label “sender”.

An other simpler way to avoid the confusion would
be to use more global information such as the area of the blocks. Indeed the area of the block “sender” is bigger than the area of the block “date, place”. This information could be introduced as a rule inside a post-processing stage.

4.3. Third class of error

The third kind of error observed corresponds to the case of figure 6. One can notice that different labels like “subject” and “opening” appear in a same handwriting region which should be theoretically impossible. This can be explained by the presence of a block of white pixels none-labeled (corresponding to the background state) between two blocks of handwriting.

![Figure 6](image.png)

**Figure 6.** Example of segmentation error due to the presence of blocks of white pixels

To cope with this problem, we have proposed in a previous work [6] to introduce new states in our model in order to distinguish white pixels of the background from white pixels belonging to a labeled zone. We have shown that this method solved most of problems of segmentation explained before but that errors of labels were more easily propagated leading to an increase of the error rate.

An other way to cope with this kind of error would be a post-processing stage consisting in clustering close small square-bounded regions into larger ones.

4.4. Post-processing step

The conclusion of the error analysis is that several strategies could be used to improve the performances of our algorithm of layout extraction:

- increase of the database,
- modification of the Markovian model (ex: introduction of new states),
- introduction of a preliminary stage of key-word recognition,
- use of a post-processing stage.

As a first study we have chosen, in this paper, to deal with a post-processing stage which could solve the two last classes of errors described before.

To cope with the second class of errors, we can use information about the area of the blocks “sender” and “date, place”. Indeed when these two blocks are present in a letter of the development set, the area of the block “sender” is always greater than the block “date, place”.

If the algorithm detects two different blocks labeled “sender” and no block labeled “date, place”, the area of these two blocks are computed. The block with the smallest area is then labeled “date, place”.

This strategy reduces the error rate of 7% which is now equal to 14%. An example of result is given figure 7.

![Figure 7](image.png)

**Figure 7.** Example of error solved by the first step of the post-processing stage

The second step of the post-processing stage has to solve the third class of error. It consists in clustering close small square-bounded regions into a larger one. The final region is then labeled with the label (different of the label “background”) the most represented in the initial regions.

This strategy allows a reduction of 23% of the error rate which reaches 10.7%. An example of result is given figure 8.

In conclusion, all the improvements introduced the post-processing stage reduces the error rate of 28%. In the future, we could imagine to include directly the post-processing in the model. Nevertheless with a post-processing stage the Markovian model is not modified and then remains always generic.

5. Results obtained at the RIMES evaluation campaign

The first evaluation campaign of the RIMES project [10] took place in June 2007 and was organized as follow:

- Day D-60: the organizers sent the development and
validation sets with the ground truth to the participants;

- Day D-7: the organizers sent the test set to the participants;
- Day D: deadline for sending the hypothesis obtained by automatic systems;
- Day D+7: each participant receives the results obtained at the evaluation campaign;
- Day D+30: final workshop with the presentation of all the systems and their results.

For the layout extraction task there were two participants. Our system achieved an error rate of 11.26% which is very close of the results reached by the other participant.

6. Conclusion

We have presented a system of physical layout extraction of handwritten letters based on some textural and spatial considerations. Experiments have been conducted on a thousand images of the RIMES database.

A rigorous analysis of error allowed us to propose several possible improvements. Among them, a post-processing stage has been presented in this paper. A grey-level weighted pixel error rate of 10.7% has thus been reached on the training set of the RIMES database. It represents a reduction of 28% of the initial error rate obtained without the proposed post-processing stage. Moreover an error rate of 11.26% has been obtained during the first RIMES evaluation campaign in June 2007 which was very close to the results obtained by the other participant.

A future promising improvements of this task would be the introduction of a word recognition stage embedded in the proposed algorithm. The probability that a word belongs to a key-word vocabulary defined for each label could be then multiplied by the probability of this same word to be in a particular label according to spatial and local contextual information.

References


