

# Numerical Sequence Extraction in Handwritten Incoming Mail Documents

G. Koch, L. Heutte and T. Paquet  
*Laboratoire PSI, Université de Rouen, FRANCE*  
*Laurent.Heutte@univ-rouen.fr*

## Abstract

*In this communication, we propose a method for the automatic extraction of numerical fields in handwritten documents. The approach exploits the known syntactic structure of the numerical field to extract, combined with a set of contextual morphological features to find the best label to each connected component. Applying an HMM based syntactic analyzer on the overall document allows to localize/extract fields of interest. Reported results on the extraction of zip codes, phone numbers and customer codes from handwritten incoming mail documents demonstrate the interest of the proposed approach.*

## 1. Introduction

Today, firms are faced with the problem of processing incoming mail documents: mail reception, envelope opening, document type recognition (form, invoice, letter, ...), mail object identification (address change, complaint, termination, ...), dispatching towards the competent service and finally mail processing. Whereas part of the overall process can be fully automated (envelope opening with specific equipment, mail scanning for easy dispatching, printed form automatic reading), a large amount of handwritten documents cannot yet be automatically processed. Indeed, no system is currently able to read automatically a whole page of cursive handwriting without any a priori knowledge. This is due to the extreme complexity of the task when dealing with free layout documents, unconstrained cursive handwriting, unknown textual content of the document [4]. Nevertheless, it is now possible to consider restricted applications of handwritten text processing which may correspond to a real industrial need. The extraction of numerical data (file number, customer reference, phone number, zip code in an address, ...) in a handwritten document whose content is expected (incoming mail document) is one particular example of such realistic problem.

In this paper, we propose a method for the automatic extraction of numerical fields in handwritten incoming mail documents. Therefore our primary objective was to design means to extract in a line of text a particular numerical field of interest prior to its recognition. Indeed we postulate that the spatial organization of the connected

components in a numerical field obeys a specific structure and can be exploited in the extraction task.

Although this hypothesis cannot be considered fully realistic, the proposed method is an interesting alternative to the use of a digit recognizer prior to syntactical postprocessing. The proposed method will serve as a syntactical filter prior to recognition. Two components are required for this extraction filter. The first one is dedicated to the labelling of the connected components. Labels are defined as Digit or Irrelevant handwritten information for the task. The second component is the syntactical analyser that finds the best label sequence of each line of text using the known syntax of the numerical field we want to detect.

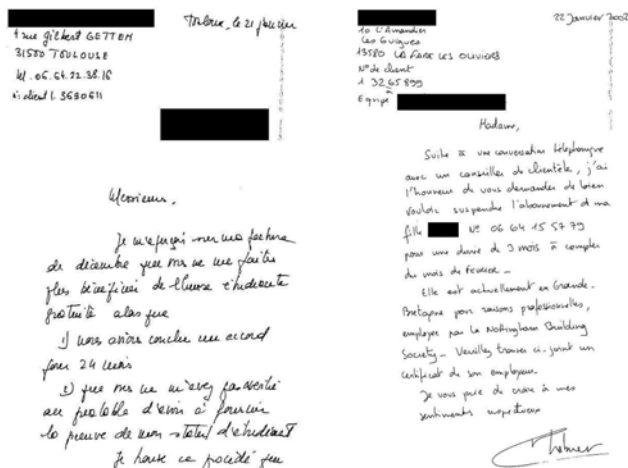
The paper is organized as follows. Section 2 is devoted to the justification and motivation of the proposed method. Section 3 describes the intrinsic and contextual features used to characterize the connected components; results of the connected component labeling module are also given. We present in section 4 the syntactical analysis stage aimed at extracting specified numerical fields. Experimental results on a real database of handwritten incoming mail documents are provided in section 5. Finally, some conclusions and future works are drawn in section 6.

## 2. Overview of the proposed system

The goal of this study is to achieve a system dedicated to the extraction of numerical data, as zip codes, phone numbers or customer codes, in unconstrained handwritten documents. Figure 1 gives two examples of incoming mail documents. One can see that the fields of interest we are looking for can occur anywhere in the document (heading, body of text,...) or they can even sometimes be absent.

At first sight, a naive solution would be to use a common handwriting recognition system in order to recognize all patterns in the document (digits, letters, words) and then to select the only information of interest (digits). However, the use of a recognition system in the case of a full page of handwriting is expensive in computer time and obviously not reliable when dealing with an open vocabulary [5]. Besides, as the information to be extracted represent only a small part of the document, the complete reading of the document is not necessary. A second approach, probably more realistic,

would be to use a single digit recognizer on connected components. When looking forward to the consequences of such an approach it appears that due to the potential presence of connected digits, a segmentation driven recognition strategy would be required. Multiple studies, especially dedicated to numerical amount recognition, have shown the difficulty of such an approach [2].



**Figure 1. Two examples of handwritten incoming mail documents**

As a consequence, we have rather turned towards a fast morphological discrimination between digits and words. This discrimination allows to localize numerical fields of interest within the document without therefore resorting to a digit recognizer but using constraints imposed by the syntactical structure of the numerical sequences.

Our system is divided into the three following stages, once all the connected components have been extracted from the document:

- Preprocessing: text lines are first extracted by grouping together the connected components. As our approach of discrimination is mainly based on spatial features, the precise knowledge of these alignments is essential.
- Connected component labeling: since the method is dedicated to the detection of numerical fields within text lines, we are primarily interested in assigning each connected component to its unknown label which can be either Digit or Reject. A third label has been added since numerical fields can contain digit separators. Each input connected component is characterized in a 7-feature space combining intrinsic and contextual features. Likelihood of each class is then computed.
- Syntactical analysis: this last stage is crucial for the system as it will allow to verify that some sequences of connected components can be kept as candidates. Indeed, the numerical sequences we search for all respect one precise syntax (five digits for a french

zip code, ten digits for a french phone number,...). The syntactical analyzer will therefore be used as a precise numerical field localizer able to keep the only syntactically correct sequences and reject the others.

Although the detection of text lines is the first entry point of the system it is based on rather classical techniques and we will not therefore discuss it in the rest of this paper. We briefly summarize the main steps. The proposed approach is inspired from [3] in which a method for detecting text lines with unknown orientation is presented. Various modifications have been brought to take only into account horizontal lines. The main steps of our approach are the following:

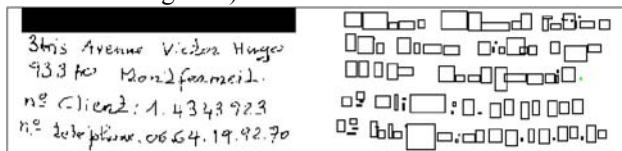
- grouping together connected components whose size is greater than a given threshold. This allows to take only into account connected components corresponding to words or parts of word, whereas the various punctuations and accents are ignored. The grouping is performed according to distance-based criterions.
- fusion of the too near alignment: several alignments can be detected for a same text line. The fusion of these alignments is performed according to distance-based criterions and mean size of between lines of the overall document.
- assignment of isolated components to the nearest lines: this allows to take into account the set of connected components that have been ignored in the processing of the first step.

Now that text lines are extracted, the connected component labeling can be achieved.

### 3. Connected component labeling

During this stage, only the connected components being part of an alignment are processed.

Let us recall that we do not look for recognizing the numerical fields, but we want just to localize them. A handwritten document will therefore be viewed as a collection of lines of connected components, each connected component being just represented by its bounding box (as shown for a part of a handwritten document in figure 2).



**Figure 2. Heading of a handwritten document and its connected component bounding boxes**

As we do not have to know what is the symbolic label of a connected component ("0", "3", "A", "Tel" or something else) but only be able to discriminate digits from the remaining connected components, this implies

that digits must be grouped together into one and only one class. It is therefore necessary to find joint features to characterize them. Besides, it is important to note that a large number of fields (mainly phone numbers and customer codes) contains several separators (point or dash). As these separators are in some cases an important part of the syntax, they have also to be identified. Finally, a large number of fields contains touching digits (figure 3). As our approach does not call on a digit recognizer, it is not possible to consider the explicit segmentation of these connected components (which in all the cases would be a difficult task without resorting to digit recognition [2]). Therefore, connected components that may correspond to "double digits" must be identified.

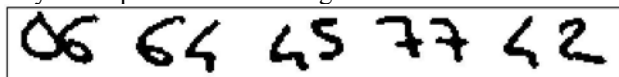


Figure 3. One example of field containing touching digits

Taking into account the above observations, four classes of connected components have to be considered for labeling, namely: digit (class 'D'); touching digits or double digit (class 'DD'); separators such as point and dash (class 'S'); rejection, i.e. all the remaining connected components in the document (class 'R').

Now the problem is to find suitable features that may discriminate as much as possible these four classes. Let us consider for example the numerical fields (phone number and customer code) in figure 4. Obviously there is no conspicuous similarity between digits.

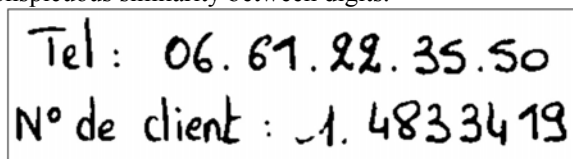


Figure 4. An example of two numerical fields

However, when considering only the bounding box of the connected components (figure 5), we can point out that they are quiet regular both in size and spacing within each numerical field.

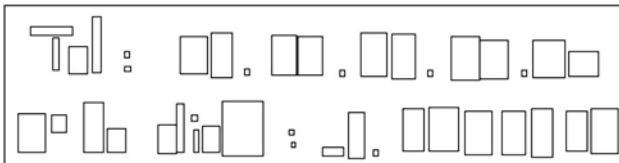


Figure 5. Bounding box of the connected components corresponding to figure 4

Since all the digits have quiet the same aspect, the aspect ratio (height on width ratio) seems to be a relevant feature to characterize them. Obviously, this single feature is not discriminant enough. Other features are therefore added to characterize the regularity of numerical sequences in terms of height, width and spacing of the bounding boxes within the numerical fields.

These regularities are measured in the vicinity of each connected component by taking into account their left and right neighbors on the same line of text as follows. Let  $C$  be the connected component under investigation,  $C-1$  and  $C+1$  its left and right neighbors respectively; let  $H_C$ ,  $W_C$  and  $G_C$  be respectively its height, width and the  $X$ -coordinate of its center of gravity.

By taking into account height and width of  $C-1$  and  $C+1$  and related distances of  $C-1$  and  $C+1$  from  $C$ , the regularity/irregularity in height, width and spacing in the neighborhood of  $C$  can be measured through the following features:

$$f_1 = \frac{H_{C-1}}{H_C}, f_2 = \frac{H_{C+1}}{H_C}, f_3 = \frac{W_{C-1}}{W_C}, f_4 = \frac{W_{C+1}}{W_C},$$

$$f_5 = \frac{|G_C - G_{C-1}|}{W_C}, f_6 = \frac{|G_C - G_{C+1}|}{W_C}, f_7 = \frac{H_C}{W_C}.$$

Finally, each connected component  $C$  being part of an alignment is characterized by a 7-feature vector (the 6 above features explaining the spatial context of  $C$  to which is added the morphological feature "height on width ratio" ( $H_C/W_C$ )). Likelihood of each of the four classes is then estimated using a mixture model. Experiments conducted on the test base of 293 documents have given the Top1 following results, i.e. when retaining only the solution with highest likelihood (Table 1)

Table 1. Top1 results of the connected component labeling on 293 documents

	Reject	Digit	Separator	Double Digit	OK	Confusion
Reject	83 287	8 233	2 395	371	88,33%	11,67%
Digit	3 289	5 290	38	46	61,06%	38,94%
Separator	1 266	28	831	0	39,11%	60,89%
Double Digit	381	135	1	60	10,40%	89,60%

The first remark is that the rejection class is well recognized and that only a few components to reject are confused (11.67%). One can also notice the weak confusions digit/separator and double digit/separator. However there still remain strong confusions for digit, separator and double digit, especially with the rejection class. This comes from the large number of connected components corresponding to words and parts of word which have features close to those of digits, separators and double digits.

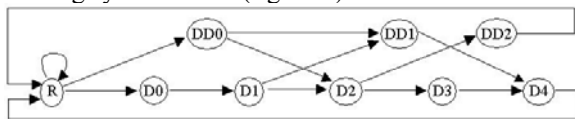
Nevertheless, as the following syntactical analysis stage will take into account the likelihood of each class, the above Top1 labeling rates give just an indication on the feature vector reliability.

#### 4. Field extraction by syntactical analysis

Let us recall that our strong hypothesis is that within a text line, each numerical field exhibits morphological regularities captured by the defined feature vector and obeys a particular syntactical structure that corresponds to a given sequence of digits and separators. The

localization of numerical fields within a text line is therefore achieved through the Viterbi algorithm [1], a commonly used algorithm for sequence alignment. To apply this algorithm, it is necessary to define a hidden Markov model. In our case the alphabet is reduced (only four classes, i.e. digit, double digit, separator and reject) and the numerical field we search for are constrained by a strong syntax (zip codes, phone numbers and customer codes). We have thus chosen to define one model for each syntax, i.e. for each type of numerical field to extract within a line. We now explain how the model is built on a given example (zip code field). The other models are built in the same way

A french zip code is constituted of five digits, each one corresponding to a given state: D0, D1, D2, D3, D4. As a line of text may contain, in addition to the zip code field, words that must be in our case rejected, it is necessary to introduce an additional rejection state, denoted by R. A transition matrix is then built to reflect the probabilities of transition of one state towards the others. For example, if one is in D0 state, the only possible transition for a zip code is D0 towards D1, all others being forbidden. While arguing in the same way on all states, we get the following syntax model (figure 6):



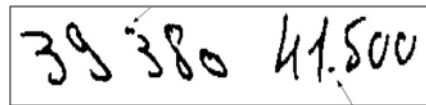
**Figure 6. Syntax model for zip code fields**

Note that for the integration of double digit states, we should have added as many states as there are combinations of two touched digits in a field: in our case 4 states for a sequence of 5 digits. However, the observation of handwritten zip codes makes appear that second and third digits are rarely connected (there is rather a wider spacing): these writer's habits can be explained by the structure of the french zip code since the two first digits stand for the department number and the three last digits for the town number within the department. This observation leads thus to retain only 3 "double digit" states (DD0 to DD2 as shown in figure 6).

Note also that to complete the model, we have moreover to define a matrix of initial states. As the zip codes are most often located at the beginning of a line of text, the likeliest initial states are D0 and DD0. However, to take into account that a line of text may begin with something else than a zip code, an initial probability for state 'R' is also set to a non-zero value. In the same way, the model is completed with a matrix of final states (only R, D4 and DD2 states have a non-zero probability).

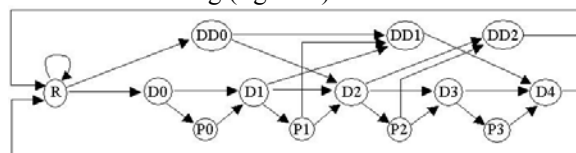
It is worth noticing that this model applies only to perfectly clean zip code fields. However, we cannot disregard a large number of fields containing small connected components which are remaining noise or part of digit as shown in figure 7. In these cases, the model

would reject the field as the zip code syntax is not respected.



**Figure 7. Presence of noise in some zip code fields**

To make the syntax model more tolerant with respect to these noises, we have inserted between two states an optional passage through a parasitic state P. Note that the transition probabilities for these states must remain weak as we do not want to uselessly increase the false alarm. We have therefore set these probabilities to a close to 0 value. Finally, the complete syntax model for zip code fields is the following (figure 8):



**Figure 8. Final model for the zip code syntax**

Phone number and customer code syntaxes are modeled in the same way as zip codes except that the number of retained states is larger (24 for the customer code syntax and 33 for the phone number syntax).

## 5. Experimental results

Experiments have been conducted on two non-overlapping databases of handwritten incoming mail documents provided by the mail reception services of a firm: the first one (292 documents) has been used to build the learning base for the kNN classifier and to find heuristics for setting the transition probabilities of each HMM and to parameterize the system; the second one (293 documents) has been used to test our proposed approach.

The detection of the numerical fields is achieved by analyzing every line of the document. The syntactic analyzer makes its decision in favour of the presence (detection) or absence (reject) of a numerical field on the line under investigation. As the syntactic analyzer may propose several locations when a field is detected, the output of the syntactic analyzer is therefore a list of N solutions corresponding to the N best paths found in the treillis of observations. As a consequence, a field is considered to be well detected if and only if no connected component in the labeled field is rejected and all the connected components in the detected field are included in the labeled field. Table 2 presents the detection results for the 3 kinds of numerical fields to extract when the correctly detected field is proposed as the first solution (Top1), within the two first solutions (Top2) or within the

10 first solutions (Top10) output by the syntactic analyzer.

**Table 2. Detection results of the three syntactical analyzers**

		TOP1		TOP2		TOP10	
ZIP Code	detected	138	41,82%	220	66,67%	292	88,48%
	not detected	192	58,18%	110	33,33%	38	11,52%
Phone number	detected	169	69,55%	195	80,25%	223	91,77%
	not detected	74	30,45%	48	19,75%	20	8,23%
Customer code	detected	91	60,26%	116	76,82%	133	88,08%
	not detected	60	39,74%	35	23,18%	18	11,92%

We can first notice that the best results are obtained for the most coercive syntaxes. Indeed a zip code is only composed of five digits without no separator, whereas the phone numbers and customers codes are longer sequences of digits capable to contain some separators.

A more precise analysis of the system behaviour can be illustrated through table 3 which presents the confusion matrix of each syntactic analyzer for the Top1 detection. It is important to notice that these confusion matrices do not count the well or bad detected fields, but the lines. Thus, a line containing a customer code and detected as a line containing a zip code will appear in the confusion matrix at line "customer code" and column "zip code". Finally, some slots contain two numbers: the first one corresponds to a field detected and well located in the line, whereas the second means that a field of the true type has been detected, but shifted in the line (bad aligned field).

**Table 3. Top1 confusion matrix for each syntactic analyzer**

	ZIP Code	Reject	Phone Number	Reject	Customer Code	Reject	
ZIP Code	137	59	127	45	278	38	285
Phone number	226	14	167	51	22	188	52
Customer code	127	20	55	92	90	21	36
Other numerical fields	219	756	63	912	56	919	
Reject	264	3756	45	3975	44	3976	

From this matrix, one can note that the lines containing no numerical field (line to reject) are effectively well rejected. The other numerical fields are also better rejected by the "customer code" and "phone number" syntactic analyzers than by the "zip code" syntactic analyzer. The same observations can be achieved for all fields. This shows that confusion is weaker as the the syntax is coercive.

## 6. Conclusion and future works

In this paper, we have presented an original method for localizing precisely numerical fields in handwritten incoming mail documents. The originality of the method rests on the localization principle which is performed, through a syntactical analyzer, on the bounding box of connected components. As there is therefore no need of a digit recognizer, the feature extraction process is simplest and faster and the number of classes to discriminate is reduced. Although the overall system has been implemented for extracting zip codes, phone numbers and customer codes on handwritten mails, nothing prevents from applying the same principle to other kinds of numerical fields in unconstrained handwritten documents provided that the field obeys a given syntax.

Future works will bear on two main optimizations of the localization process. The first one concerns the automatic learning of the different syntax models using a Baum-Welch procedure as at present initial state, final state and transition probabilities of each syntax have been roughly estimated on the learning base from human observations. The second one rests on the parallel combination of solutions provided by the three syntactical analyzers in order to decrease the false alarm rate.

## 7. References

- [1] G.D. Forney, "The Viterbi algorithm", Proc. of the IEEE 61(1973), pp. 268-278.
- [2] L. Heutte, P. Pereira, O. Bougeois, J.V. Moreau, B. Plessis, P. Courtellemont, Y. Lecourtier, "Multi-bank check recognition system: consideration on the numeral amount recognition module", IJPRAI 11 (1997), pp. 595-618.
- [3] L. Likforman-Sulem, C. Faure, « Une méthode de résolution des conflits d'alignements pour la segmentation des documents manuscrits », Traitement du Signal 12 (1995), pp. 541-549.
- [4] L. Lorette, "Handwriting recognition or reading ? What is the situation at the dawn of the third millenium", IJDAR (1999), pp. 2-12.
- [5] A. Nosary, T. Paquet, L. Heutte, A. Bensefia, "Handwritten text recognition through writer adaptation", IEEE Proceedings (2002), IWFHR'02, Ontario, pp. 363-368.