

## Iterative analysis of document collections enables efficient human-initiated interaction

Joseph Chazalon, Bertrand Couasnon

► **To cite this version:**

Joseph Chazalon, Bertrand Couasnon. Iterative analysis of document collections enables efficient human-initiated interaction. DRR - Document Recognition and Retrieval XIX, Part of the IS

T/SPIE 24th Annual Symposium on Electronic Imaging, Jan 2012, San Francisco, United States. 8297, pp.82970L, 2012, <10.1117/12.911995>. <hal-00686858>

**HAL Id: hal-00686858**

**<https://hal.inria.fr/hal-00686858>**

Submitted on 11 Apr 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Iterative analysis of document collections enables efficient human-initiated interaction

Joseph Chazalon and Bertrand Couïasnon

Université Européenne de Bretagne  
IRISA-INSA (UMR 6074), Rennes, France

## ABSTRACT

Document analysis and recognition systems often fail to produce results with a sufficient quality level when processing old and damaged documents sets, and require manual corrections to improve results. This paper presents how, using the iterative analysis of document pages we recently proposed, we can implement a spontaneous interaction model, suitable for mass document processing. It enables human operators to detect and correct errors made by the automatic system, and reintegrates the corrections they made into subsequent analysis steps of the iterative analysis process. Thus, a page analyzer can reprocess erroneous parts and those which depend on them, avoiding the necessity to manually fix during post-processing all the consequences of errors made by the automatic system. After presenting the global system architecture and a prototype implementation of our proposal, we show that document model can be simply enriched to enable the spontaneous interaction model we propose. We present how to use it in a practical example to correct under-segmentation issues during the localization of numbers in documents from the 18th century. Evaluations we conducted on the example case show, on 50 pages containing 1637 numbers to localize, that the interaction model we propose can reduce human workload (29.8% less elements to provide) for a given target quality level when compared to a manual post-processing.

**Keywords:** document analysis; degraded documents; document sets; iterative analysis; human interaction

## 1. INTRODUCTION

Document analysis and recognition systems often fail to produce results with a sufficient quality level when processing old and damaged document sets, and require manual corrections by human operators to improve results. While most systems tend to rely on human interaction during post-processing stage, it is clear that the ability to interact with such automatic system during the analysis, and not after, can be valuable. For instance, in Fig. 1, our goal was to localize sales numbers, and segment them using separators we detected, then recognize them, optimize recognition results using the fact that, for each line, numbers are in increasing order, and associate the resulting number list to the sale name written on the left (previously recognized) in a database entry. When a separator (a coma here) is missed, a human operator can simply indicate the position of the missing separator at the right time so as to avoid the need for manual correction of subsequent errors in the final document structure, which would require costlier edition actions: localize each number, key its value, etc.



Figure 1. Number and separator detection automatically produced for an excerpt from a page of a document dating from the 18<sup>th</sup> century. On the first line, a separator was missed and caused the under-segmentation of the two last numbers.

---

Further author information:

J. Chazalon: E-mail: joseph.chazalon@irisa.fr

B. Couïasnon: E-mail: bertrand.couiasnon@irisa.fr

This paper presents how, using the iterative analysis of document pages we recently proposed,<sup>1,2</sup> we can implement a human-initiated, or *spontaneous*, interaction model suitable for mass document processing: it enables human operators to detect and correct errors made by the automatic system, and reintegrates the corrections they made into following analysis steps of the iterative analysis process. It triggers a reprocessing of erroneous parts and of those which depend on them, instead of having to fix *during post-processing* all the consequences of each error made by the automatic system.

This iterative approach, which permits to process each document page as many times as needed, exhibits interesting properties regarding the four main challenges we identified for an efficient interaction with a human operator, in the context of analysis and recognition of massive document sets. Those challenge are:

*Challenge 1: Use asynchronous (non-blocking) interaction*

As sets of documents we process are often made of tens of thousands of pages, interacting with a human must not make the human operator wait for work nor block the system each time the human or the system need to communicate information about a given page to the other. In this paper, we consider cases where errors cannot be automatically detected. When errors can be detected, delaying interaction also enables to improve error detection by considering more elements, and to group, order and filter errors to show to human operators.

*Challenge 2: Give the automatic system the ability to reconsider images, results and models*

Instead of only local and isolated corrections, the system we aim at should be able to use external information to complete or correct content detection (like in Fig.1 were a coma is missed); to adapt or change the results linked to corrections a human operator made (like when using a language model to improve a sentence recognition: forcing a word value may change the global optimization result); and even to use new examples to improve content models or document structure models (but machine learning issues will not be discussed in this paper).

*Challenge 3: Minimize the global human interaction cost*

As mentioned previously, an automatic processing without interaction will require costly post-processing corrections. However, manual processing must be used as little as possible, even if it provides valuable information. Thus, it seems necessary to optimize human interaction usage to find the right balance between automatic and manual processing. We will show in this paper that interacting at the right time can be a way to achieve this.

*Challenge 4: Keep a simple and unique document model, with as little interaction-related content as possible*

Existing systems use sophisticated document models to extract contents from difficult documents. Then, interaction must not increase model complexity with synchronization considerations, nor duplicate document knowledge, in particular between analysis and human interface processes to model-check answers supplied by a human.

This paper first describe, in Sec. 2, how existing approaches for interactive analysis comply with those requirements. Then, in Sec. 3, we present the global system architecture and the spontaneous interaction model we propose. In Sec. 4 we detail a prototype implementation of our proposal, and illustrate on a practical example how this interaction can be used to correct under-segmentation issues during the recognition of sequences of numbers in documents from the 18<sup>th</sup> century. Finally, we evaluate, in Sec. 5, the interest of this approach by measuring, for this example, the diminution of human work we obtained compared to post-processing.

## 2. RELATED WORK

There are few document analysis system which propose an interactive analysis model, or at least communicate with a human operators at some point of the analysis.

PerfectDoc<sup>3</sup> is a post-processing tool which enables bulk correction of document structure (zone classification, reading order) and character recognition before producing the final structure of digitized magazines. While the human operator does not wait for tasks, and can detect and correct errors by himself, the system does not validate human information nor uses it to correct previous interpretations, as it could be done in an iterative analysis process.

The ability to interact at different times during the analysis is made possible by DocMining,<sup>4</sup> a very general framework for document processing where document processing units are successively invoked by a scheduling module following a predefined scenario. A persistent structure is associated to each document where each processing unit consumes and produces recognition data. DocMining permits a delayed interaction between

components thanks to its central storage module, and human interaction can be easily implemented as new processing units. However, this system does not allow to reprocess a page to reconsider previous results using new information, as previously. Furthermore, document model seems implicit in this system which builds a pipeline of processes, and the specification of what information human operator can provide may be complicated.

The smartFIX system<sup>5</sup> takes another way: it uses a central repository to enable a delayed interaction, and also proposes the iterative recognition, optimization and control of documents using explicit model definitions: when human operators correct results, they are automatically checked against those document models. However, as few details are given about how external information is reintegrated during the analysis processes, it is hard to know whether human interaction could be more elaborated than just labeling suspicious fields: manual corrections seem to be made in the final document structure, and not at intermediary steps of the analysis process.

While existing approaches may permit an asynchronous interaction with a human operator at some point, they generally do not reprocess document pages, and then they do not use external information to improve previous results. Nagy and Veeramachaneni<sup>6</sup> identify this issue as the need for a *durable interaction*, which “*immediately alters some system parameters and therefore affects how the system deals with new data*”, as opposed to an *ephemeral interaction* which “*merely labels new patterns or modify the results of classification.*” Moreover, there is generally no optimization of human interaction usage in the manual post-processing of existing system. Finally, they seldom rely on document models, or do not explain how interaction can be modeled or expressed. As a result, an iterative analysis of document pages seems necessary to face the challenges we listed.

### 3. ARCHITECTURE AND INTERACTION MODEL

The iterative analysis approach we recently proposed<sup>2</sup> is based on an architecture and several properties we will first present shortly, before detailing what the spontaneous interaction model is.

#### 3.1 Iterative Analysis Framework

The iterative analysis of document pages is made possible thanks to the following elements (illustrated in Fig. 2):

**Strategy Module** It is responsible for the invocation of the different processes during the analysis and their communication. This is the abstraction level where the analysis scenario is played, and it could imply to reprocess many times the same page as long as new information is available.

**Visual Memory** It is a persistent data structure associated to each page which is used for information exchange between processes, but also during document analysis in order to reintegrate external information and use it. It is an associative structure that each process can change, and which gives to each data a corresponding shape and position in its image’s referential. We note  $M_I(t)$  the visual memory related to the image  $I$  at time  $t$ .

**Central Database** The central database stores the associated visual memories for each page in the whole collection. It provides a *collection-centric view* of information stored which can progressively grow.

**Page Analyzer** The page analyzer uses a document model to try to extract relevant information from an image. In an iterative analysis scheme, it is invoked with an image reference  $I$  and the associated visual memory  $M_I(t)$ , and returns  $M_I(t+1)$  to the strategy module when done. At each iteration step, it *reprocesses the whole image* (we integrated several optimizations to keep some results) and makes use of external information. This key feature permits to produce new structured results, which are valid according to the document model.

**Human Interface** When solicited by the strategy module, the human interface enables a human operator to change the content of the visual memory of some image, like any other process. In this paper, we consider this interface from a theoretical point of view, as we will focus on the implementation of the page analyzer.

Fig. 2 shows a single step, repeated as many times as needed, of information exchange (for a given page) between a page analyzer and a user interface in a simple strategy which triggers each component one after another for each page. The communication with the central database is omitted for brevity. At time  $t$ , the strategy modules first invokes the page analyzer on the image  $I$ , with the memory  $M_I(t)$ . The strategy module gathers  $M_I(t+1)$  after the analysis, and transfers it to the human interface, which, in turn, produces  $M_I(t+2)$  and sends it back. Page reprocessing may be seen as costly, but it is necessary to be able to change the behavior of the page analyzer on a given page. Moreover, keeping intermediary results, like recognition results, can be easily implemented.

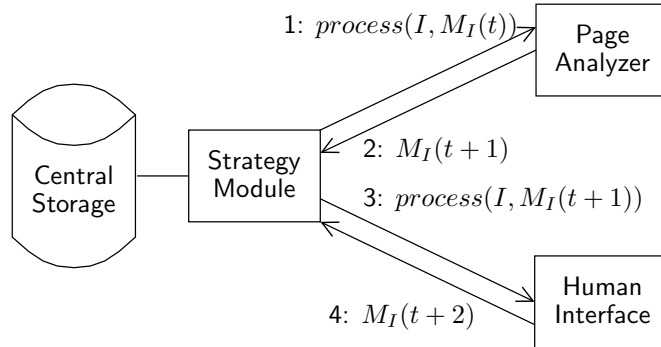


Figure 2. A single step (repeated as many times as needed during the *iterative analysis*) of a simple dialogue strategy between a page analyzer and a human interface, showing the implied components of the architecture and the *visual memory*  $M_I$  associated to the image  $I$ .

### 3.2 Spontaneous Interaction Model

Using the iterative analysis we can build many information flows between the human interface and the page analyzer to propose several forms of asynchronous interaction. In a previous paper,<sup>2</sup> we presented a *directed* interaction model. This approach allowed a page analyzer to ask question the human operator when it detected a problem. The human operator answers back, and model constraints embedded in the question structure ensure that his answer will be valid and properly reintegrated according to the document model.

The current paper focuses on a *spontaneous* interaction model. This new approach deals with cases where the *directed* interaction cannot be used, in particular when errors cannot be automatically detected and localized during the page analysis. At some point in the analysis process, the human operator is allowed to view and change some parts of the content of the visual memory of images. Those changes are used in later automatic page processing, and will hopefully help the page analyzer to improve its response.

While it provides more flexibility, the main drawback of this approach is to make the reintroduction of external information to the analysis process and the final structure more difficult. The reason is that human operators have few constraints on information they provide, and new elements may not comply with the document model: they could be at unexpected positions, or have unexpected values. We will see in the next sections that this *spontaneous* interaction model still permits an early correction of errors, which is valuable.

## 4. IMPLEMENTATION AND EXAMPLE

This section presents how a *spontaneous* interaction model can be implemented and used in a rule-based page analyzer based on the DMOS-P approach,<sup>7</sup> then it shows on an example how this mechanism can be used.

### 4.1 DMOS-P Framework and Language

DMOS-P<sup>7</sup> is a concept-driven grammatical document analysis method. It uses a bi-dimensional extension of Definite Clause Grammars, called *Enhanced Position Formalism* (EPF), to describe pages. The following example illustrates its syntax: to recognize A, we try, from the **top** of the image, to recognize B, and if it fails, we try, inside the **bottomPart**, to recognize C. Derivation is noted “**::=**”, concatenation is noted “**&&**” and alternatives are expressed with multiples derivations for a same left-hand part. Operators are: “**AT**” for the positioning; and “**IN..DO**” for spacial restriction (to limit the part of the image where a rule can be applied).

```

A ::= AT(top) && B. % clause 1, tried first
A ::= IN(bottomPart) DO(C). % tried if clause 1 fails
  
```

Like attribute grammars, rules have input and output parameters indicated by “+” and “-” signs:

```

recognizeNumber(+NumPos, -Value) ::= callClassifier(+NumPos, -Value).
  
```

## 4.2 Information Representation

Information stored in the visual memory associated to an image is made of elements which can be represented by the following construct, where: *Zone* represents the localization of the element in the current image's coordinate system; *Marker* is a tag, like *m\_separator* or *m\_number*, which enables to distinguish different information types; and *Data* is the useful content according to the information tag, like coma for separators, or [15, 115, 16] for numbers (hypothesis list). In the spontaneous interaction model, those contents are provided by the human operators to guide or correct the behavior of the automatic system on a given image.

$$M_I(t) = \{Zone_1 \rightarrow (Marker_1, Data_1); Zone_2 \rightarrow (Marker_2, Data_2)\}$$

## 4.3 DMOS-P Extension for Spontaneous Interaction

Spontaneous interaction is made possible by adding the following operator to the document model language, and providing an implementation for this operator which automatically manages the way external information is used to adapt the page analyzer's behavior.

SPONTANEOUS(+Marker, +Rule)

This construct automatically checks for some data in the visual memory of the current image before choosing whether the rule **Rule** should be applied or not. The content of the memorized element will be used instead of applying the rule in the parts of the image indicated by the *Zone* component of the memorized construct, if the **Marker** tag of the element in the visual memory corresponds to the one expected.

This approach allows to annotate specific parts of the document model to identify those which can produce erroneous results that may be manually corrected by human operators during interaction steps. As a result, an interactive detection and correction of errors through several analysis steps (automatic analysis, asynchronous interaction, then automatic analysis again) is made possible. The key advantage is that it enables to substitute human processing to automatic processing for some parts of the document model in a systematic and simple way: as soon as a rule is "decorated" with the previous extension, this behavior is available with not extra cost, as all information fusion and decision issues are handled automatically. Some improvement of the human interface may be needed to handle new data types.

## 4.4 Example

This example shows a very simple grammar (inspired by a one we use) which processes images like the one in Fig 1: starting at the top of the page, it locates a text line and uses its position to extract sales numbers in this restricted area, and goes on until no more text line is found. Number extraction is done by first detecting all the separators of the line (coma, "à", etc.) and second locating the numbers using the position of those separators. The **detectAllSeparators** rule is a while-like construct which build a list of all the results got from the **separator** rule. In order to enable a spontaneous interaction for the localization of separators, the only thing to do is to "decorate" the **separator(-Sep)** line in the **detectAllSeparators** rule definition.

### Grammar without interaction capability

```
start() ::=
  AT(topPage) &&
  detectTextLinePosition(-Line) &&
  IN(lineArea +Line)
  DO(extractSalesNumbers()) &&
  AT(under +Line) &&
  % read other lines...

extractSalesNumbers() ::=
  detectAllSeparators(-SepLst) &&
  extractNumberBetweenSep(+SepLst).

detectAllSeparators(-[Sep|OtherSep]) ::=
  separator(-Sep) &&
  detectAllSeparators(-OtherSep).
detectAllSeparators(-[]).
```

### Grammar allowing spontaneous interaction

```
start() ::=
  AT(topPage) &&
  detectTextLinePosition(-Line) &&
  IN(lineArea +Line)
  DO(extractSalesNumbers()) &&
  AT(under +Line) &&
  % read other lines...

extractSalesNumbers() ::=
  detectAllSeparators(-SepLst) &&
  extractNumberBetweenSep(+SepLst).

detectAllSeparators(-[Sep|OtherSep]) ::=
  SPONTANEOUS(+m_separator, % Only change
  separator(-Sep) ) &&
  detectAllSeparators(-OtherSep).
detectAllSeparators(-[]).
```

Let us now suppose that the image we want to process contains only 3 numbers; “11”, “22” and “33”, and that those number are separated by comas. If, for any reason, the last coma was not detected, the two last number may not be properly segmented and later recognition will fail. In the case of a collection with a single page, with a simple analysis strategy which ask successively the page analyzer and the human interface to modify the memory content, the following table illustrates a possible behavior. We consider here that the content of the visual memory is the observable system’s behavior, and that extracted numbers and separators are stored for human analysis (the way it was done is omitted in the grammar description for simplicity).

Time $t_i$	Description	Visual memory content $M_I(t_i)$
$t_0$	The analysis starts.	$\emptyset$
$t_1$	The page analyzer tried to locate the separators and the numbers, but it missed one. The memory contains 2 numbers and 1 separator elements. The first number ("11") at zone $Z_{N1}$ is correct, and the second ("2233") at zone $Z_E$ is a fusion of the 2 last numbers.	$\{Z_{N1} \rightarrow (\text{m\_number}, "11");$ $Z_{S1} \rightarrow (\text{m\_separator}, \text{coma});$ $Z_E \rightarrow (\text{m\_number}, "2233")\}$
$t_2$	The human operator has controlled the content localization and adds the missed separator at position $Z_{S2}$ , while removing the erroneous number ("2233").	$\{Z_{N1} \rightarrow (\text{m\_number}, "11");$ $Z_{S1} \rightarrow (\text{m\_separator}, \text{coma});$ $Z_{S2} \rightarrow (\text{m\_separator}, \text{coma})\}$
$t_3$	The system reprocessed the image and used new information to properly segment the two numbers ("22" and "33"). As the number “11” was already in memory, the system did not reprocess this number, saving CPU time. This simple optimization is used for real analysis tasks.	$\{Z_{N1} \rightarrow (\text{m\_number}, "11");$ $Z_{S1} \rightarrow (\text{m\_separator}, \text{coma});$ $Z_{S2} \rightarrow (\text{m\_separator}, \text{coma});$ $Z_{N2} \rightarrow (\text{m\_number}, "22");$ $Z_{N3} \rightarrow (\text{m\_number}, "33")\}$

As we can see, the spontaneous interaction model can be very easily used to enrich the document model and specify which parts of the analysis may be altered by human operators. To integrate this approach in an existing document analysis system, the only required elements are: the simple architecture we presented, which allows an iterative analysis of document pages; a visual memory to exchange and reuse information; and a page analyzer based on a document model language containing an operator similar to the one we presented.

## 5. EVALUATION ON A PRACTICAL CASE

The goal of this last section is to show that spontaneous interaction during the analysis of documents permits to reduce human workload while being easy to integrate in some page analyzer. We isolated a very narrow part (number zones localization) of a real document analysis we performed on French documents dating from the 18<sup>th</sup> century, in order to be able to evaluate the influence of the interaction model we propose, and to avoid mixing human actions with different costs. Real strategies mix various interaction schemes but are hard to evaluate.

### 5.1 Dataset

We used 50 images of old documents dating from the French Revolution (18<sup>th</sup> century), which contain text lines indicating a sale category followed by a sequence of numbers indicating sales references. Those numbers (1637 in the 50 pages) are separated by a symbol (coma or “à” which indicates a range) or, sometimes, a blank space. Fig. 1 shows an excerpt of a page, and contains 8 numbers: 6 for the first sale type (or line), and 2 for the second. For each page, we know the real position of each number (its bounding box) thanks to a manual annotation.

### 5.2 Protocol

We used a page analyzer similar to the one presented in the example in Sec. 4.4: using the image and its associated memory, it tries to localize the separators and the numbers the image contains, and stores them in the visual memory. It permits a spontaneous interaction, as presented in the previous example in Sec. 4.4.

We used the following analysis strategy. First, all images are processed with the page analyzer (first iterative step, first automatic analysis). We kept a snapshot  $S_1$  of the content of all the visual memory structures *before human interaction* for later analysis. Then, a human operator was able to change visual memory contents of all

images (second iterative step, interaction). Finally, all images are *reprocessed* with the *same* page analyzer (third iterative step, second automatic analysis). We kept another snapshot  $S_2$  after the second automatic analysis.

As explained in introduction, we limited the actions performed by the human operator to avoid mixing different actions costs. He was only allowed to correct under-segmented numbers in the following way: he reviewed analysis results and located under-segmented numbers (as in Fig.1), then he removed erroneous elements (separator and numbers) in this area and added one (or more) separators (of fixed size) at the right positions. He did not locate nor recognize numbers, at it should be done automatically during the reprocessing of the page.

### 5.3 Analysis

Our goal was to measure the improvement with spontaneous interaction for the number localization task while evaluating the human work cost it implied, and compare it with the human work cost which would have been necessary to reach the same quality by localizing the numbers directly (post-processing ephemeral interaction).

*Localization quality* was estimated by counting the zones matching the ground truth as follows:

$$\text{if } \frac{\text{surface}(I)}{\text{surface}(E)} > T \text{ and } \frac{\text{surface}(I)}{\text{surface}(R)} > T \text{ then } R \text{ matches } E \text{ else } R \text{ does not match } E$$

where  $\text{surface}(\text{Zone})$  is the number of pixels in  $\text{Zone}$ ;  $E$  is the expected sale number zone (ground truth);  $R$  is the zone actually localized (by the automatic system);  $I = E \cap R$  is the intersection of expected and real zones; and  $T \in [0; 1]$  is a threshold. A threshold of 0.99 (perfect match) allowed us to know how many sale number zones were removed and added between  $S_1$  and  $S_2$ , and we used a threshold of 0.80 (acceptable match) when comparing  $S_1$  and  $S_2$  to the ground truth. This value was not finely tuned but presented a good compromise between too permissive and too restrictive: as the old-fashioned writing style in the document we processed tend to produce very long figures for the numbers “9” and “7” (as in Fig. 1), and missing the very lower part of them has little impact on number recognition.

*Spontaneous interaction cost* was estimated by counting the number of separators added, as it represents the worst case estimation of the count of edition actions (locate an under-segmentation zone + clear it + add a separator). The *post-processing interaction cost* was estimated using the count of zone numbers it would have been necessary to add to reach the same quality level.

### 5.4 Results

**Quality improvement** Our ground truth reference contains 1637 number zones. Before the spontaneous interaction stage (in  $S_1$ ), 1339 numbers zones are properly localized, 233 (14.8% of detected zones) are erroneous (under-segmented, noise, etc.) and 298 are missing (not detected). In  $S_2$ , after the spontaneous interaction stage, which corrects *only* under-segmentation, and the reprocessing of pages, 1460 number zones are well localized, 169 (10.4% of detected zones) are erroneous and 177 are still missing. The proposed approach permits an *improvement* of 7.5% (+121 zones) of the *localization rate* (−40.6% of missing zones) while also permitting to *reduce the relative amount of erroneous zones* (−30.0% of erroneous zones). Fig.3 illustrates this gain.

**Cost reduction** Between  $S_1$  and  $S_2$ , the human operator created 85 new separators, which means that 85 edition actions were performed. Manual post-processing would have required to perform the deletion of the elements detected in 85 erroneous zones, and the careful creation of 121 numbers (and their manually annotation, as the recognition stage is finished). If we neglect the preliminary deletion of erroneous element which could be implicit in both scenarios, we can conclude that, *to reach the same upper quality level for this task*, the spontaneous interaction scenario permits to *reduce human workload by 29.8%*, without even considering that manual number localization is costlier than separator localization.

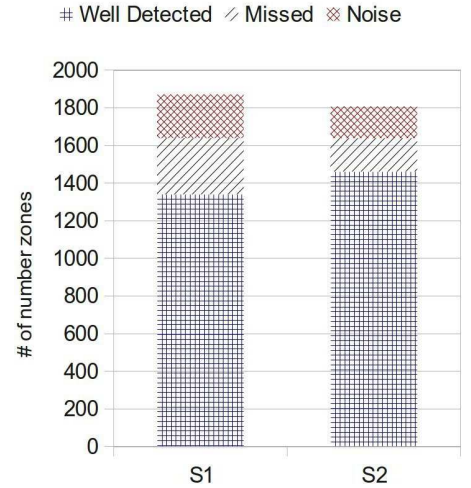


Figure 3. Comparison of the localization of number zones before interaction ( $S_1$ ) and after interaction and reprocessing ( $S_2$ ), showing a reduction of both missing and erroneous (noise) zones.



## 5.5 Discussion

To build the ground truth, we actually used the results of the automatic analysis after spontaneous interaction and manually corrected the remaining errors. The spontaneous interaction stage roughly lasted 45 minutes (121 corrections,  $\sim 22.3$  sec/err), and the post-processing was more than 1 hour 30 minutes long (177 corrections,  $\sim 30.5$  sec/err). This estimation depends on a lot on the human interface, but illustrates that detecting and correcting analysis errors at the right time can reduce the human workload (cost and number of corrections required).

## 6. CONCLUSION

We presented how to enable asynchronous interactions with human operators *during the analysis stage* of document processing, using an iterative analysis approach we recently proposed: information associated to an image can be stored and reused later, allowing a *non-blocking interaction* over sets of pages suitable for mass document processing (it addresses the challenge 1 of the introduction).

We introduced a *spontaneous interaction model*, explained how it can be implemented in a rule-based page analyzer, and showed that this model permits to make use of external information provided by a human operator to guide or to correct the system behavior (or response) for a given image. This contribution permits to detect and correct errors asynchronously with a human, and leads to a *reconsideration of image content or of previous results during later reprocessing*: it becomes possible, for example, to add a separator which does not exist in the image to improve the segmentation of numbers (a step toward challenge 2).

We showed on an example, which was simple enough to permit an easy evaluation, that an appropriate use of *spontaneous interaction could reduce human interaction cost* for a given task: we asked a human operator to correct cases where number zones were under-segmented by providing the missing separators. It permitted to avoid the precise localization of number zones, but also their manual annotation as their automatic recognition remains possible during later reprocessing (a contribution to challenge 3).

Finally, we illustrated how a *document model can be kept simple* by annotating the parts of the automatic analysis where the human may interfere, or substitute himself to the automatic process, with only one new language operator. A document model can therefore include many annotations enabling a spontaneous interaction for various information types, while the approach we proposed hides all the communication issues using a visual memory and a systematic approach to interact with a human operator (an answer to challenge 4).

## ACKNOWLEDGMENTS

This work has been done in cooperation with the *Archives départementales des Yvelines* in France, with the support of the *Conseil Général des Yvelines*.

## REFERENCES

- [1] Guichard, L., Chazalon, J., and Coüasnon, B., “Exploiting Collection Level for Improving Assisted Handwritten Words Transcription of Historical Documents,” in [*Proc. of ICDAR*], (2011).
- [2] Chazalon, J., Coüasnon, B., and Lemaitre, A., “Iterative Analysis of Pages in Document Collections for Efficient User Interaction,” in [*Proc. of ICDAR*], (2011).
- [3] Yacoub, S., Saxena, V., and Sami, S., “PerfectDoc: a ground truthing environment for complex documents,” in [*Proc. of ICDAR*], 452–456 (2005).
- [4] Clavier, E., Masini, G., Delalandre, M., Rigamonti, M., Tombre, K., and Gardes, J., “DocMining: A cooperative platform for heterogeneous document interpretation according to user-defined scenarios,” in [*Graphics Recognition*], Lladós and Kwon, eds., **3088 of LNCS**, Springer (2004).
- [5] Klein, B., Dengel, A., and Fordan, A., “smartFIX: An adaptive system for document analysis and understanding,” in [*Reading and Learning*], Dengel, Junker, and Weisbecker, eds., **2956 of LNCS**, Springer (2004).
- [6] Nagy, G. and Veeramachaneni, S., “Adaptive and interactive approaches to document analysis,” in [*Machine Learning in Document Analysis and Recognition*], Marinai, S. and Fujisawa, H., eds., *Studies in Computational Intelligence* **90**, 221–257, Springer (2008).
- [7] Coüasnon, B., “Dealing with noise in DMOS, a generic method for structured document recognition: An example on a complete grammar,” in [*Graphics Recognition*], Lladós and Kwon, eds., **3088 of LNCS**, Springer (2004).