A Layout-Free Method for Extracting Elements from Document Images

Tsukasa Kochi and Takashi Saitoh

Information and Communication Research and Development Center
32 Research Group, RICOH COMPANY,LTD.
3-2-3 Shinyokohama, Kouhoku-ku, Yokohama-shi, Kanagawa 222-8530, Japan
{kochi,saitoh}@ic.rdc.ricoh.co.jp

Abstract. SGML is a language for defining the layout structure of a document. Various attempts at generating SGML from a document image have not been successful. We focus on extracting some of the important layout elements by using flexible matching strategy and easy model generation. Our proposed approach treats each extracted element as it were independent. Some segmented areas like "title" or "author" are defined locally making the system robust, able to withstand shifting and noise. The system is also easy to operate. Since the system is not full automatic, we need to supply typical models of each component. Our GUI presents the attributes of each segmented area as well as the original bit map images. The color-coded attributes help us to easily edit the extracted component. In experiments with 288 pages of test images, the proposed method is shown to be 95.6% correct for a wide range of documents. By using 145 pages of documents as a learning set, the system recognized 99.2% of feature sets from 148 various types of unknown documents.

1 Introduction

1.1 Background

The increasing use of the Internet and the Intranet has created the need to convert paper documents into digital data so that they can easily be accessed with computer networks. Recently many commercial document management systems have been implemented in offices and libraries. The effective use of systems in office workflow has become very important. When we input documents from a scanner in digital form into such document management systems, we usually give them a file name like `‘file00xx‘` and store them in folders. Although it is usual to store original images with OCR text, documents from OCR consist of simple strings of characters and therefore it is difficult to automatically extract important elements from the document, such as title, author, or date of the document.

The most successful system in this field is the business form processing system. Standard business form processing systems first identify the type of input

S.-W. Lee and Y. Nakano (Eds.): DAS'98, LNCS 1655, pp. 215–224, 1999.

forms and then extract elements from them based on simple comparison of location of their frame lines and elements with a registered model [1] [2]. Although Aoki and Okada [3] presented a form processing system that does not require line properties, it assumes that the extracted elements are positioned in frames. The limitations of business form processing systems are as follows.

1. They can only be used for business forms with distinct frame lines,
2. A limited amount of variation is allowed for the locations of elements.

There have been various attempts to change document images into SGML documents. For example the contents of a document are recognized by referring to models that are usually described by a script language that includes a rule for the relative element arrangement and knowledge of its layout features. Using this knowledge and the OCR results, they tried to structure the whole document, for example Lin [8] presented the logical structure analysis of book document images that generates an SGML document using contents information. The present systems for document image analysis also have some fundamental problems that cannot be avoided.

3. The task of creating models from sample documents requires users to take a lot of time. Walischewski [7] provided an automatic method for creating models but it requires a large number of learning images to improve the model accuracy.
4. One error causes other errors in understanding the document structure because models have been defined by complex structures of document elements depending on each other. Walischewski [7] and Tang [9] had developed a model with attributed directed graph of the document layout.

These problems have made it difficult to use document structure analysis systems for a wide range of documents.

1.2 Summary of the Work

This paper addresses automatic element extraction for scanned documents by using layout feature matching. The design goals of this work are to solve the problems of the present systems for document image analysis:

1. The proposed method is robust against shifting or noise.
2. The proposed system can be easily used for various types of document.

To realize our design goals, this work provides a user-friendly interface and a simple method for creating models for flexible matching to extract documentary elements. Since the proposed method can extract important elements, from a practical point of view, our results can directly support the functions of document maintenance system such as keyword retrieval and title registration. As a result, the proposed method can be used for any type of document by changing the models for every type of document but the basic algorithm for extracting elements does not have to be changed.

Figure 1 shows an overview of the system operation. The system consists of three modules, the details of which are described in the following and in Figure 1.

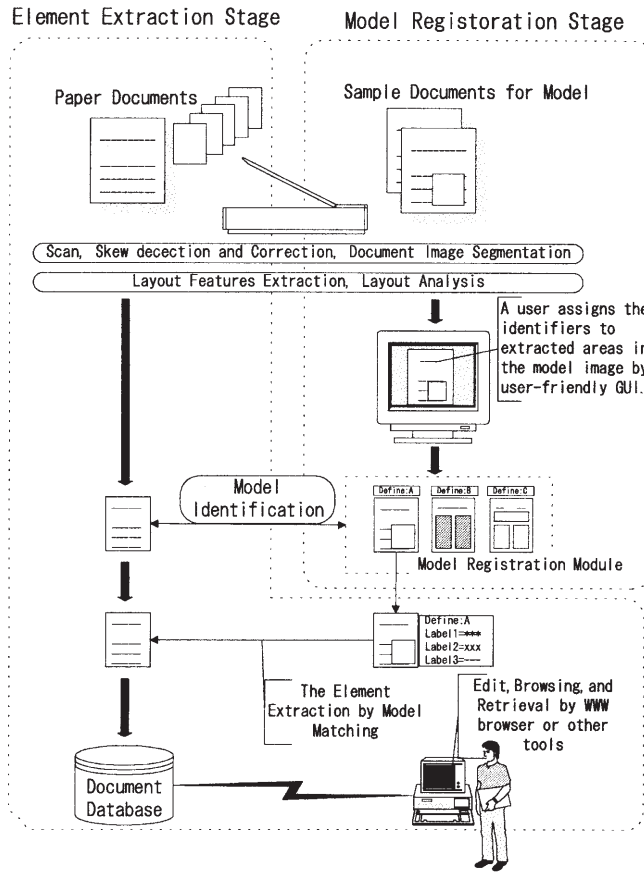

Fig. 1. Overview of system operation

1. **Feature Extraction** The system segments the digitized image of the document before model construction and the element extraction stages.
2. **Model Construction and Registration** A user must assign identifiers to the extracted areas with a GUI and then store the model registration module with them. The model construction procedure concludes with these steps.
3. **Element Extraction** The input data for this module are the segmented images and models. If the model registration has more than one model, the system identifies the input image with the most suitable model in the

model registration module. Then, the system tries to extract elements from the image of the document referring to the model defined by the layout features of each element. Finally, the system outputs the candidate lines of character strings, resulting from the element extraction from the image to the document filing system database. If necessary, the system can improve its knowledge of documents by dynamically modifying the features and their relations.

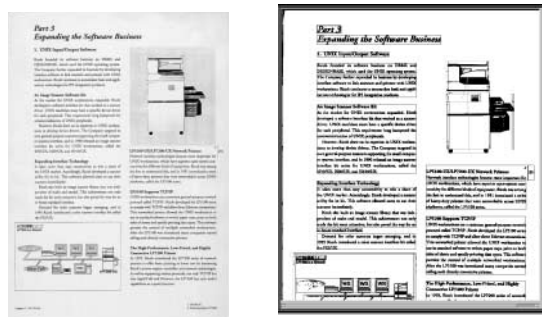

Fig. 2. Sample image

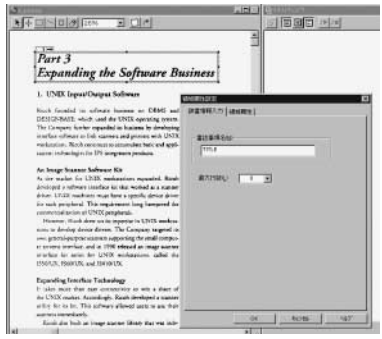

Figure 3: GUI for creating a model

Figure 4: Result of extracting

Since the system has too many components to be shown in detail in this paper, we focus on feature extraction, model construction, and element extraction. This paper is organized as follows. Section 2 presents the layout features of document images. Section 3 defines the schema for constructing models and their specifications. Section 4 gives the strategies used in element extraction. Section 5

describes the outline of the learning system. Section 6 shows the experimental results. Section 7 concludes the paper.

2 Feature Extraction

The feature extraction module follows the document segmentation process. The document image is segmented into area blocks containing lines and character blocks. Saitoh's procedure [10] is used for this purpose. After segmentation, the font type is determined for each line of characters. Hull's procedure [11] is used for the font recognition.

The layout features of the images are obtained by using these procedures. The features have three components, (i) page feature, (ii) area feature, and (iii) line feature. Examples of page features are printed page size, number of areas, and line direction. We use well-known layout features as area and line features such as coordinates, character size, font, and indent. The details of the method for extracting these layout features are skipped because these features are extracted easily by any other well known page segmentation method.

3 Model Construction

In order to provide a user-friendly method for creating models for extracting elements, the models should be structured to be as simple as possible. Our model mainly consists of two parts: the first part is the page data for the whole page image, and the second is the list structure of the segmented area of the document. Each area includes lines, and also each line includes characters.

If we make a model for extracting the title of the page shown in Figure 2, we need only assign title area to the identifier `'TITLE'` by using the GUI shown in Figure 3. Table 1 presents the elements definitions in this case.

Table 1. Elements definition

Name	coordinates	Size	Font	Max	···
`‘TITLE‘`	(415, 303, 1896, 416)	112	1	-1	···
`‘Footer‘`	(135, 4103, 484, 4114)	40	0	1	···
`‘caption‘`	(2491, 4059, 2904, 4148)	40	0	1	···

3.1 Example of a Model

Figure 5 shows an example of a model whose element definition is shown in Table 1. The two lines from the top of Figure 5 define the page data on the model document. The list structure for the areas begins at the third line. The third line gives the area data that define the extracted element named `‘Footer‘`.

The next line gives data of the element. The other extracted elements, ‘caption‘ and ‘TITLE‘ are given in a similar way.

```
<!DOCTYPE page SAMPLE >
<page number=1 image=IMAGE width=3296 height=4677 num_area=36 line_dir=0>
<area id=0 kd=4 label="Footer" max=1 nline=1 xs=135 ys=4103 xe=484 ye=4144 sz=40 ls=0 font=0 column=0 cp=9>
<line num=0 len=13 cp=6 lp=-1 sz=27 xs=139 ys=4106 xe=482 ye=4143>Chapter6 The IPS Era</line>
</area>
<area id=1 kd=16 label="caption" max=1 nline=2 xs=2491 ys=4059 xe=2904 ye=4148 sz=40 ls=0 font=0 column=0 cp=4>
<line num=1 len=7 cp=7 lp=-1 sz=27 xs=2497 ys=4063 xe=2687 ye=4090>1. IS4100/UX</line>
<line num=2 len=19 cp=3 lp=57 sz=28 xs=2492 ys=4109 xe=2900 ye=4147>2. Network printer LP7200</line>
</area>
<area id=2 kd=16 label="" max=-1 nline=1 xs=415 ys=191 xe=700 ye=276 sz=80 ls=0 font=0 column=0 cp=9>
<line num=3 len=5 cp=7 lp=-1 sz=76 xs=419 ys=195 xe=697 ye=273>Part3</line>
</area>
<area id=3 kd=16 label="TITLE" max=-1 nline=1 xs=415 ys=303 xe=1896 ye=416 sz=112 ls=0 font=1 column=22 cp=10>
<line num=4 len=16 cp=7 lp=-1 sz=108 xs=418 ys=305 xe=1894 ye=413>Expanding the software Business</line>
</area>
```

Figure 5: Elements definition

3.2 Simple Model

Our models have only the descriptions of the layout features. They do not contain neither rules nor relations among elements. Other known methods can recognize the logical document structure by using a rule-base. However rule construction requires users to follow a very complex set of procedures to apply the rule to a wide range of documents.

In Figure 5, the extracted elements are defined in the 3rd, 6th, and 13th line. It is clear that our model has a simple structure for expressing document layout information because each definition line has simple options for the layout features of the area, such as `id=N`, `kd=N`, `label='name'`, `max=N` , etc. The simple structure of a model would cause no side effects among the nearby elements. Furthermore, by using computer learning methods and testing a couple numbers of sample images, the system can dynamically improve its performance. This learning system is described in Section 5.

4 Element Extraction

In this section, the main algorithm for extracting elements will be described. The input data for this module are the results from document segmentation module and a suitable model for the input document. The output data of this module are candidates of extracted element par lines. How to use the extracted candidates basically depends on the end-users, so the output candidates are simply stored in HTML form in this module.

4.1 Steps of Extracting Elements

As shown in Figure 6, as mentioned before, a plurality of the models is generated for various types of documents. In step 1, a new document is input, and the input document image is divided into areas, then predetermined layout features are extracted. In step 3, the type of input document is detected and the suitable

model for the input document is selected. In step 4 through step 8, candidates sentences corresponding to each element are extracted according to the model. Figure 7 shows the idea for model matching based on layout features. An extraction error occurring in an element would not influence other element extractions. These steps show that elements are extracted gradually from areas to lines. This strategy has two advantages. One is that it enables effective thresholds, and the other advantage is that the system is robust against fatal errors in the document segmentation module. For example, illegal composition or division of segmented lines. If the system requires higher accuracy, the candidates for lines are sorted again according to the regularity of an element arrangement in a document.

Finally the output are stored in HTML form. Figure 4 shows the extracting results.

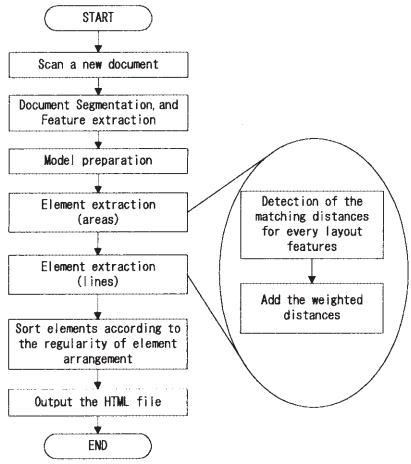

Figure 6: Flow of element extraction

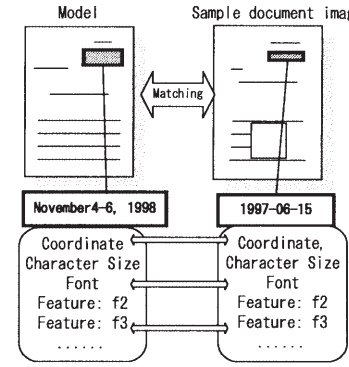

Figure 7: Idea of the model matching

4.2 Concrete Matching Strategy

Here the concrete strategy on the element matching is presented referring to Figure 7. This figure illustrates the retrieval of a segmented area from the input document image, corresponding to the DATE field, "November 4-6, 1998", in the model. The layout features in this figure: location, character size, and font type are automatically extracted with general segmentation methods. The DATE field in the left hand is matched with every segmented sentence in the right hand document in order to determine the distance. The distance s_{e_i} for an element e_i is brought through Mahalanobis distance of the features:

$$d^2_{e_i} = \sum \frac{|f_i|^2}{v_i}$$

where $|f_i|$ denotes the matching distance for each layout feature and v_i denotes the variance to denote a proper weight for each layout feature.

Although this method can recognize logical elements in a document accurately, the system does not use linguistic information from OCR. Thus extraction costs are low, and moreover we can achieve a high accuracy even if we do not know the character direction or what language are using, because there is no need to recognize errors in the OCR. We expect that the system will be improved by using linguistic processing but we do not wish to pursue this issue in this paper.

5 Learning System

In our basic approach to extract elements from document images, a model is made from only a single document. However this model may prevent improving the performance of the system, because the model can not know what layout features the next document might have. In this section, we will look at the learning methods used to make the system capable of improving its knowledge of documents, by dynamically modifying the features in its knowledge base.

Figure 8 shows the outline of the learning system. The input document is compared with the original model stored in model registration module. If a shift in the layout features would be detected more than a constant threshold, the system reacts according to the following choices; (1) retry to extract elements, (2) modify the model, and (3) use a multi-template method. The first two reactions can be performed without any hand-operation so that users do not notice them but the system becomes wiser with every document.

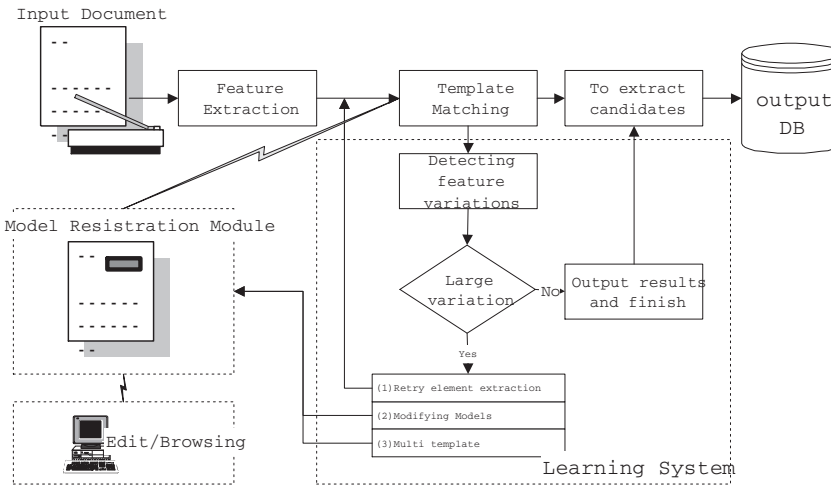

Figure 8: Outline of learning system

6 Experiments

In this section, we present some experiments for extracting elements from the binary images of general documents. We have also compare the performance

of the proposed basic algorithm with the performance of the proposed learning method, in terms of the improvement of extraction accuracy for images whose layout features are variable.

6.1 Data Set

In the experiments described in this section, we used an image set containing 288 pages of business letters, reports, technical papers, magazines, and Japanese articles with character strings aligned vertically. In order to estimate the proposed methods for the various kinds of document sets, we classified the sample images into three sets based on their layout features. Table 2 shows the classification of the image set. Sample images for the learning experiments were divided into two sets, for learning and testing, to compare the performance before and the after learning.

6.2 Basic Engine

The first experiment estimates the basic algorithm for the element extraction. The results are tabulated in Table 3. Table3 shows the correct rate for the element extraction and the effectiveness rate, that means the ratio of the number of correct lines found in the extracted candidates over the number of the whole selected candidate lines. The system solved the task of extracting elements from 145 different registered pages of 17 page types. The correct rate for the element extraction from document images was estimated to be 95.6%, and the effectiveness rate for the element extraction was estimated to be 55.5%.

Table 2: Test images

	pages	elements
Small	100	320–331
Normal	83	425
Large	68	287–304
Vertical	37	64–74
Total	288	1096–2232

Table 3: Result of experiment

	Basic		Leaning	
	correct	effect	correct	effect
Small	98.2	67.1	98.2	72.9
Normal	99.2	51.0	99.2	56.0
Large	85.3	43.6	100	44.8
Vertical	100	47.7	100	65.0
Total	95.6	55.5	99.2	61.4

6.3 Learning System

The second experiment was performed to evaluate the improvement of the basic algorithm. New information is automatically added to our knowledge base. Experimental results for the learning method are shown in the right of Table 3. Compared with the results of the basic algorithm shown in the left of Table 3, the correct rate and effectiveness are both improved, especially the correct rate for the images with a large variation in the layout features. This improvement is mainly due to the use of the multi-template method for the documents with discrete variation in the layout features, such as indent and appearance.

7 Conclusion

A system for extracting elements from document images by feature matching has been described. It has the following features;

1. The proposed method is robust against the feature's variations.
2. Creating models is easy because of the user-friendly interface.
3. The system can be used for many types of documents.

We also described the learning method including model modifying and the multi-template for improving the performance of the basic algorithm. Thus the error rate for element extraction has been reduced by 81.1%. The algorithm is effective for images with large shifts in their layout features. Our result shows that out system can directly support the functions of document maintenance system such as keyword retrieval and title registration.

Acknowledgments

The authors thank Dr. Koichi Ejiri and Dr.Hirobumi Nishida for their valuable comments on earlier drafts of this paper.

References

1. T.Watanabe, et al, "Extraction of data from preprinted forms", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.17, No.4, 1995, pp.432-445. 216
2. J.Yuan, et al, "Form items extraction by model matching", ICPR'96,1996, pp.691-695. 216
3. H.Arai, K.Odaka, "Information Acquisition and Storage of Forms in Document Processing", ICDAR,1997, pp.164-169. 216
4. C.Wenzel, "Supporting Information Extraction from Printed Documents by Lexico-Semantic Pattern Matching", ICDAR, 1997.
5. M.Sharpe, et al, "An Intelligent Document Understanding & Reproduction System", MVA'94,1994, pp.267-271.
6. T.Watanabe, X.Huang, "Automatic Acquisition of Layout Knowledge for Understanding Business Cards", ICDAR, 1997, pp.216-220
7. H.Walischewski, "Automatic Acquisition of Spatial Document Interpretation", ICDAR, 1997, pp.243-247 216
8. C.Lin, et al, "Logical Structure Analysis of Book Document Images Using Contents Information", ICDAR, 1997, pp.1048-1054. 216
9. Y.Tang, et al, "Document Processing for Automatic Knowledge Acquisition", IEEE, Transaction on Knowledge and Data Engineering, Vol. 6, No.1, 1994, pp.3-31. 216
10. T.Saitoh, et al, "Document Image Segmentation and Text Area Ordering" Proceedings of ICDAR, 1993, pp.323-329. 219
11. S.Khoubyari and J.Hull, "Font Function Word Identification in Document Recognition", CVIU,1996, pp.66-74. 219