Document image matching using a maximal grid approach

Article *in* Proceedings of SPIE - The International Society for Optical Engineering · December 2001

DOI: 10.1117/12.450721

CITATIONS
11

READS
38

3 authors, including:

Angelina Tzacheva
University of North Carolina at Charlotte
44 PUBLICATIONS **298** CITATIONS
SEE PROFILE

Yasser El-sonbaty
Arab Academy for Science, Technology & Maritime Transport
55 PUBLICATIONS **474** CITATIONS
SEE PROFILE

Document Image Matching Using a Maximal Grid Approach

Angelina Tzacheva[a], Yasser El-Sonbaty[b], *and* Essam A. El-Kwae[a]

[a]University of North Carolina at Charlotte, Charlotte, NC 28223
[b]Arab Academy for Science & Technology, Alexandria, Egypt

ABSTRACT

A new approach for form document representation using the maximal grid of its frameset is presented. Using image processing techniques, a scanned form is transformed into a frameset composed of a number of cells. The maximal grid is the grid that encompasses all the horizontal and vertical lines in the form and can be easily generated from the cell coordinates. The number of cells from the original frameset, included in each of the cells created by the maximal grid, is then calculated. Those numbers are added for each row and column generating an array representation for the frameset. A novel algorithm for similarity matching of document framesets based on their maximal grid representations is introduced. The algorithm is robust to image noise and to line breaks, which makes it applicable to poor quality scanned documents. The matching algorithm renders the similarity between two forms as a value between 0 and 1. Thus, it may be used to rank the forms in a database according to their similarity to a query form. Several experiments were performed in order to demonstrate the accuracy and the efficiency of the proposed approach.

Keywords: Document Imaging, Document Image Matching, Document Retrieval, Document Image Databases, Document Image Representation.

1. INTRODUCTION

Paper is still the most widely used media for transferring information in an office environment. Generally, data has to be extracted from paper forms and manipulated manually by typing into computers. Among document images, forms are structured documents used for information gathering, storage, retrieval, approval and distribution. A form is defined as a structured document composed of the following elements: horizontal and vertical layout lines, both straight and continuous, preprinted data such as machine printed characters, symbols, and pictures, and user filled-in data such as machine-typed, hand-printed, or handwritten characters [1]. Examples of widely used standard forms include tax forms and health insurance claim forms such as the healthcare financing administration documentation (HCFA) [2]. In an office environment, an electronic form database system minimizes distribution costs, facilitates forms standardization and administrative control, promotes information sharing by ensuring that users have access to the same version of forms and fill them out in a uniform manner, minimizes errors in information capture, and eases retrieval and use of captured information. Traditional manual key entry of such forms is a tedious, time consuming, and an error prone process. Thus, automating the modeling and representation of forms is highly desirable.

A document image is a visual representation of a paper document. Document image understanding is a research endeavor that consists of developing processes for taking a document through various representations starting from a scanned image all the way into semantic representation [3]. A similar definition of document image analysis is the subfield of digital image processing that aims at converting document images to symbolic form for modification, storage, reuse, and transmission [4].

A novel approach for the modeling and representation of forms in document databases is introduced. The model is based on the concept of a maximal grid. The maximal grid is the grid that encompasses all the horizontal and vertical lines in the form and can be easily generated from the cell coordinates. A schematic of this approach is shown in Figure 1. Image processing techniques are used to segment a form into lines. A document frameset is extracted using a set of steps that include:

- Form capturing: A form may be scanned or provided by the user in digital format.
- Form Preprocessing: Skew detection using techniques such as [5, 6, 7].

Document Recognition and Retrieval IX, Paul B. Kantor, Tapas Kanungo, Jiangying Zhou,
Editors, Proceedings of SPIE Vol. 4670 (2002) © 2002 SPIE · 0277-786X/02/$15.00

- Line detection: Line detection is performed using techniques such as Hough transform [8, 9], the Canny Edge Detector [10, 11], or the Block Adjacency Graph [6].

Document representations are then saved into a logical database against which queries are resolved. Such database may be queried using a query form. Queries based on matching document framesets utilize matching techniques such as [1, 12].

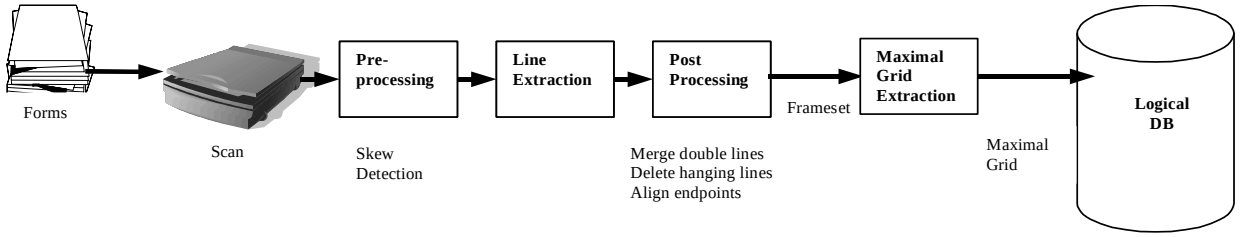


Figure 1. Schematic of the proposed approach for form representation

The contributions of this paper include a model form modeling the frameset of a form document using a maximal grid approach. A matching algorithm is developed to match a query document represented by its maximal grid against a database of documents. The matching algorithm ranks the database images according to their similarity to the query document. Thus, this model can be used for document recognition, an important step in document image processing. The assumption made in this paper is that the initial steps of the block diagram in fig.1. were already performed and the input to the model is a text representation of each document that described the cells within the document forms.

The rest of this paper is organized as follows: In section 2, related work to the problem of form document representation and matching is discussed. The proposed model is introduced in section 3, together with the matching technique. Finally, conclusions and future work are given in section 5.

2. RELATED WORK

A survey of document image analysis may be found in [4, 13, 14]. In [15], a system was developed for retrieving the data stored in HTML tables. The proposed approach constructs the content tree of an HTML table, which captures the intended hierarchy of the data content of the table, without requiring the internal structure of the table to be known beforehand. The approach can be employed by (i) a query language written for retrieving hierarchically structured data, extracted from either the contents of HTML tables or other sources, (ii) a processor for converting HTML tables to XML documents, and (iii) a data warehousing repository for collecting hierarchical data from HTML tables and storing materialized views of the tables.

In [6], the useful data to be extracted from the forms was defined to be contained in the filled-in text. The paper focused on distinguishing the filled-in data from the preprinted entities. The goal was to extract filled-in data from a form in the presence of filed overlap. The problem of character segmentation involves two issues: Separating characters from form frames and reconstruction of broken strokes introduced during separation. To extract the filled-in data, the system should have the form structure knowledge in advance. This knowledge can be acquired by one of the following methods: A form structure file created by a word processing software, Interactive learning, or Scanning and automatic recognition of a blank form.

The Global-Local-Global (GLG) method [16] aims at extracting the logical structure of a form document by analyzing the layout of the fields in the document. First, it divides the whole form globally into several blocks as temporary sub forms by the analysis of data-field rectangles. Second, the logical dependent relationships between the fields are

identified inside every block locally. Third, it re-divides the whole form again into several canonical forms globally. Then a tree called logical structure tree is constructed.
One approach for document modeling and matching in form image databases was introduced in [17]. In that representation, the relationship between the different fields in the preprinted text is modeled as a spatial orientation graph (SOG). Matching preprinted text is then performed using a robust spatial similarity algorithm [18]. The algorithm uses the concept of a rotation correction angle (RCA) to achieve rotation invariance against document skewness. The problem of form matching and retrieval based on framesets has been addressed by several researchers.

A hierarchical representation of form documents for identification and retrieval was suggested in [1]. This approach required no domain knowledge such as the pre-printed data or filled-in data and it could handle geometrical modifications and slight variations. In [19], a form document was represented by the physical features of its components such as length, width, and position of vertical and horizontal lines. However, this representation cannot handle variations on the physical structure of logically the same form. In [20], line crossings were classified into one of various types and the form was represented by the set of type counts. However, slight variations may cause those counts to change considerably. A system for automatically reading Japanese or English documents with complex layout structures was introduced in [21]. The system was based on document image and character segmentation using a set of features and knowledge of the document layout.

A method to recognize the layout structure of multi-kinds of table forms from document images was introduced in [22]. A form was represented using three binary trees constructed for global structures, local structures, and the block divisions. A classification tree was used to manage the relationship among different classes of layout structures. However, such technique is only applicable to table form documents. On a form document, a block is defined to be a rectangular area that is surrounded by horizontal and vertical lines. In this sense, it is a higher level feature than the lines. For form representation, the position and size of blocks or their relationships can be used [22, 23, 24]. However, multi-kinds of forms cannot be handled by such a scheme.

Matching forms based on the grids of their framesets was introduced in [12]. A form document similarity measure was proposed that is insensitive to translation, scaling, moderate skew (<5°) and variations in the geometrical proportion if the form layout. A significant enhancement was that the similarity measure had a good tolerance to line detection errors. However, for forms that are different in terms of the number of rows and columns, the algorithm was computationally expensive. In such case, the minimum number of rows and columns was used as the grid size and all possible combinations were matched. The one that gave the maximum similarity was selected. However, the calculation of the values inside the grid cells needed to be repeated for every possible combination of rows and columns.

3. THE PROPOSED DOCUMENT REPRESENTATION AND MATCHING ALGORITHMS

As defined in [12], a *frame line* is a horizontal line with its endpoints on vertical lines or a vertical line with its endpoints on horizontal lines. A frame line is represented by the coordinates of its starting point and terminal point. Thus, a *horizontal frame line* h_i is denoted h_i and *a vertical frame line* is denoted by v_i. The horizontal frame lines are sorted in an ascending y-order and the vertical frame lines are sorted according to ascending x-order to define the sequences H and V.

$$H = \{h_0, h_1, ..., h_i, ..., h_{n_H-1}\}, \qquad y_i \le y_j \quad if \quad i < j$$

$$V = \{v_0, v_1, ..., v_i, ..., v_{n_V-1}\}, \qquad x_i \le x_j \quad if \quad i < j$$

A *frameset* is the frame structure set {H,V} of a form. A *cell* is rectangle in a form defined by two horizontal frame lines and two vertical frame lines and within which there is no other frame lines. Let C denote the set of cells in a form $C = \{c_0, c_1, ..., c_i, ..., c_{n_C-1}\}$ where n_C is the number of cells in the form. Each cell can be described by a 4-tuple (top, left, bottom, right). Figure 2 demonstrates the above definitions.

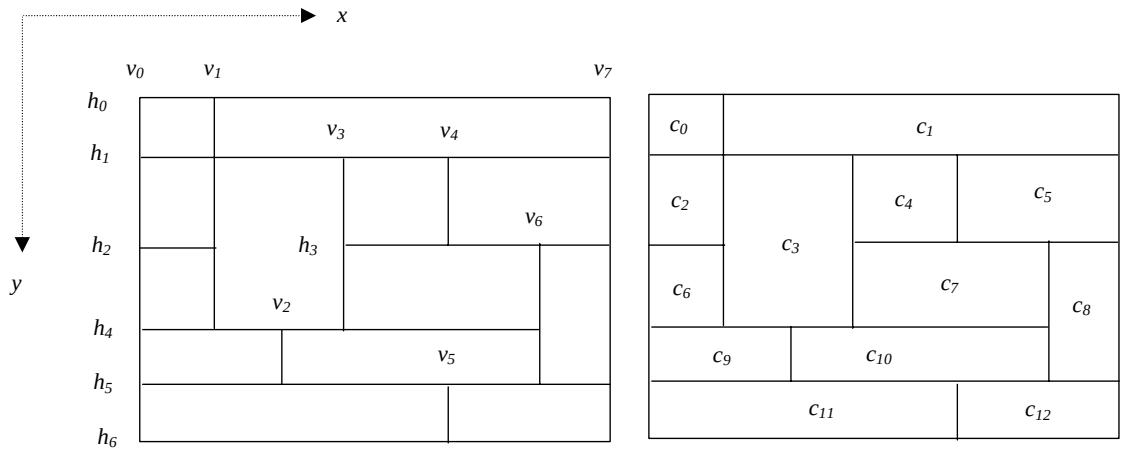
In a frameset, collinear horizontal lines form an H-Group that is represented by the y-coordinates of those horizontal lines, collinear vertical lines form a V-Group that is represented by the x-coordinates of those vertical lines. If there is n_{HG} H-Groups and n_{VG} V-groups in a form frameset, then the following two sequences may be obtained:

$$H_G = \{p_0, p_1, ..., p_i, ..., p_{n_{HG}-1}\}, \qquad p_i \le p_j \quad if \quad i < j$$

$$V_G = \{q_0, q_1, ..., q_i, ..., q_{n_{VG}-1}\}, \qquad q_i \le q_j \quad if \quad i < j$$

$|H_G| \le |H|$, where |X| is the number of elements in the set X, and

$$|V_G| \le |V|$$



(a) Form frame set (b) Cells

Figure 2. A form with its frameset and cells

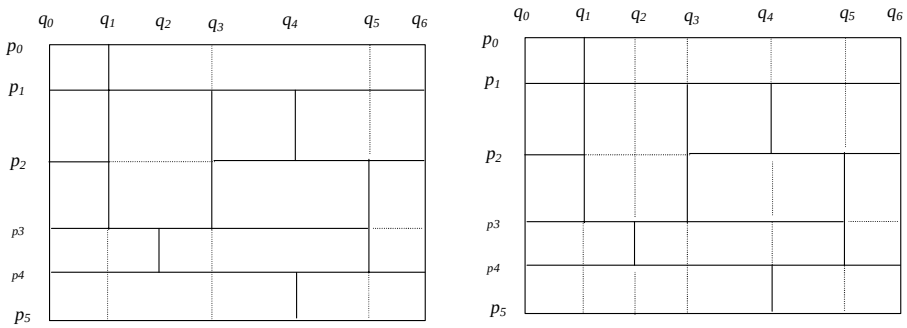
A *grid line* is defined as a set of one or more collinear horizontal lines or a set of collinear vertical lines extended to touch the boundaries of the form frameset. If the sets A is a subsequence of H_G and B is a subsequence of V_G, then A and B form a *grid G(A,B)* on the form's plane. The grid has (|A|-1)x(|B|-1) cells. A *maximal grid* G_{max} is defined as $G_{max}=G(H_G, V_G)$, i.e. the grid where all possible collinear vertical or horizontal lines are included. The maximal grid has ($|H_G|$-1)x($|V_G|$-1) cells. An example for a non-maximal and a maximal grid is shown in Figure.3. The maximal grid completes all the lines that might have been broken due to scanning noise or poor quality of the input form. Thus, including the maximal grid lines for form template matching will make the matching algorithm robust to noise and line breaks. The number of cells N(i,j) within each maximal grid cell (i,j) is defined as the total number of cells in the form template that intersect the maximal grid cell. Let M(i,j) be a maximal grid cell, then:

$$N(i,j) = \sum_{c_k \in C_{Grid(A,B)}} \left(Intersect(c_k, M(i,j)) \times \frac{1}{\sum_{i=0}^{|A|-2} \sum_{j=0}^{|B|-2} Intersect(c_k, M(i,j))} \right), \text{ and}$$

$$Intersect(c_k, M(i,j)) = \begin{cases} 1 & if \ c_k \cap M(i,j) \neq \Phi \\ 0 & if \ c_k \cap M(i,j) = \Phi \end{cases}$$

The way the above equations work is that for each cell in the original form template, if this cell intersects the maximal grid cell under consideration, a number is added to the content of that maximal grid cell. This number is equal to the 1 / the number of maximal grid cells spanned by the template cell. An example is shown in Figure 4. Note that a maximal grid cell can either include one complete template cell or a partial cell. Thus, the value of N(i,j) falls within the range (0,1], where a value of 1 occurs when a maximal grid cell is exactly overlapping an original template cell.

Suppose the set R is defined as the set that has the total numbers in each row of the maximal grid. For example, R for the grid in figure 4 is equal to: {2, 3.5, 3, 2.5, 2}. In the same fashion, the set C may be defined as the set including total number in each column of the grid. For example, C for the grid in figure 4 is equal to: {15/4, 29/20, 77/60, 137/60, 61/30, 11/5}. Note that the sum of all elements in R or in C must be equal to n_C, the total number of cells in the original form. A form template is then represented by the set $\{R, C\}$.



(a) Non Maximal Grid **(b) Maximal Grid**

Figure 3. An example of maximal and non-maximal grids

	q_0	q_1	q_2	q_3	q_4	q_5	q_6	***Total***
p_0	*1*	*1/5*	*1/5*	*1/5*	*1/5*	*1/5*		***2***
p_1	*1*	*1/4*	*1/4*	*1*	*1/2*	*1/2*		***3.5***
p_2	*1*	*1/4*	*1/4*	*1/2*	*1/2*	*1/2*		***3***
p_3	*1/2*	*1/2*	*1/3*	*1/3*	*1/3*	*1/2*		***2.5***
p_4	*1/4*	*1/4*	*1/4*	*1/4*	*1/2*	*1/2*		***2***
p_5								
Total	***15/4***	***29/20***	***77/60***	***137/60***	***61/30***	***11/5***		

Figure 4. Example for the number calculation within a grid cell

In order to match two form templates represented by the sets R and C, assume for now that the maximal grids of the query and database forms have the same number or rows and columns, matching of the two framesets is thus based on matching their maximal grid representations, based on the total difference between the number of cells within each row and column. Assume that the query form $Q = \{R_Q, C_Q\}$ and the reference form $D = \{R_D, C_D\}$, and assume that $|R_Q|=|R_D|$ and $|C_Q|=|C_D|$, then the similarity S between the two forms is calculated as follows.

$$S = 1 - \frac{\sum_{i=0}^{|R|-1} \left| R_{iQ} - R_{iD} \right| + \sum_{i=0}^{|C|-1} \left| C_{iQ} - C_{iD} \right|}{2\left(| n_Q | + | n_Q | \right)}$$

In case the number of rows and columns do not match, similarity matching is performed based on the smaller of the number of rows or columns. In this case, the grid with the larger number needs to merge some of its rows or columns to match the smaller number. If $R_{min} = min(R_Q,R_D)$ and $C_{min} = min(C_Q,C_D)$. All possible combinations that make the grid size in the query and the reference forms equal to $\{R_{min}, C_{min}\}$ is selected and matched as above (Figure.5). The similarity S is calculated as the maximum similarity obtained from this calculation. Note that for each combination, the calculations are minimal, as opposed to matching the grids in [12]. The totals for the merged rows or columns are simply added and the sum is used to represent the row or the column in the similarity matching.

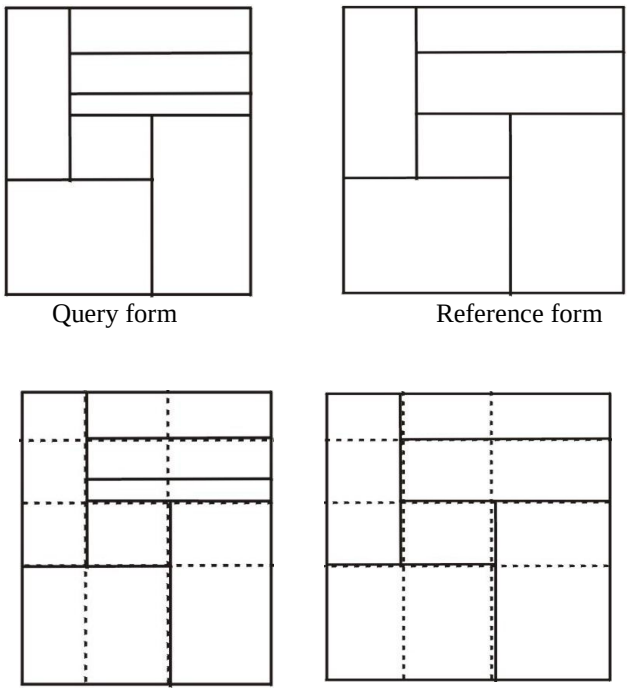


Figure 5. Example for the matching between a query and a reference form of different sizes

It can be seen that both forms are almost similar except for a spurious line added to the query form. The maximal grid values for the query and reference forms of figure 5 are shown in figure 6. The maximal grid for the query form is of size 5x3 while that of the reference form is of size 4x3. The totals for the rows in the query form are: {5/4, 5/4, 5/4, 7/4, 6/4}. The total for the rows is 7, which is equal to the number of cells in the query form. Similarly, the totals for the columns are: {3/2, 3, 5/2}. The totals for the rows in the reference form are: {8/6, 8/6, 11/6, 9/6}. The total for the rows is 6, which is equal to the number of cells in the reference form. Similarly, the totals for the columns are: {3/2, 5/2, 2}.In order to match those forms, their sizes need to be equalized. This can be achieved by combining two rows from the query form into one row. This will give 4 different combinations that may be represented in binary form as follows: {11110, 11101, 11011, 10111, 01111}. The combination for the binary string 11011 is shown in figure 7. It is clear that the effect of the spurious line is local to the area where the spurious line has been added.

¼	½	½
¼	½	½
¼	½	½
¼	**1**	½
½	½	½

1/3	½	½
1/3	½	½
1/3	**1**	½
½	½	½

Figure 6. Example for the matching between a query and a reference form of different sizes

¼	½	½
1/2	**1**	**1**
¼	**1**	½
½	½	½

Figure 7. Merging two rows from the query form to satisfy the binary combination 11011

The performance of the maximal grid approach was compared to the grid approach in [12]. The time required for matching for the maximal grid approach is significantly lower than that in [12] while the similarity results are quite similar. The performance gap even increases rapidly as the size of the forms grow. This is due to the fact that the amount of work needed to match a specific combination in the maximal grid approach I significantly lower than that needed in [12].

4. CONCLUSIONS

A new approach for form document representation and matching using the maximal grid of a document frameset is presented. The maximal grid can be easily generated from the cell coordinates. The algorithm is robust to image noise and to line breaks, which makes it applicable to poor quality scanned documents. Future work includes testing the new representation on a large form database and combine the form representation and matching with image processing routines to extract the document frameset. Another direction is to combine the frameset representation with the spatial layout of text components for improved form identification.

REFERENCES

[1] Pinar Duygulu and Volkan Atalay, “A Hierarchical Representation of Form Identification and Retrieval,” SPIE Journal of Electronic Imaging, 2000.

[2] "Healthcare Financing Administration Documentation Guidelines," May 1997 (http://www.hcfa.gov/forms).

[3] Sargur N. Srihai, Stephen W. Lam, Venu Govindaraju, Rohini K. Srihari, and Jonathan J. Hull, "Document Image Understanding: Research Directions," Technical Report CEDR_TR_92_1, State University of NY at Buffalo, May 1992.
[4] George Nagy, “Twenty Years of Document Image Analysis in PAMI,” IEEE Transactions on Document Analysis and Machine Intelligence,” Vol. 22, No.1, January 2000.

[5] D. X. Le and G. Thoma, “Document Skew Angle Detection Algorithm,” Proc. 1993 SPIE Symposium on Aerospace and Remote Sensing – Visual Information Processing II, Orlando, Fl, April 14-16 1993, Vol. 1961, pp. 251-262.

[6] Bin Yu and Anil K. Jain, “A Generic System for Form Dropout,” IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol. 18, No. 11, November 1996, pp. 1127-1134.

[7] Bin Yu and Anil K. Jain, “A Robust and Fast Skew Detection Algorithm for Generic Documents,” Pattern Recognition, Vol. 29, No. 10, pp. 1599-1629, 1996.

[8] Milan Sonka, Vaclav Hlavac, and Roger Boyle, “Image Processing. Analysis, and Machine Vision,” PWS Publishing, 1999.

[9] Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing,” Addison Wesley, 1992.

[10] J F Canny, “Finding Edges and Lines in Images,” Technical Report AI-TR-720, MIT Artificial Intelligence Laboratory, Cambridge, MA, 1983.

[11] J F Canny, “A Computational Approach to Edges Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, 1986, pp. 679-698.

[12] Jinhui Liu and A.K. Jain, “Image-Based Form Document Retrieval,” Pattern Recognition, Vol. 33, 2000, pp. 503-513.

[13] Kristen Summers, “Toward a Taxonomy of Logical Document,”DAGS95: Electronic Publishing and the Information Superhighway, May 30-June 2 1995,

[14] Sargur N. Srihari, Stephen W. Lam, Venu Govindaraju, Rohini K. Srihari and Jonathan J. Hull, “Document Understanding: Research Directions,” Tech Report CEDAR- TR-92-1, May 1992.

[15] Seung-Jin Lim and Yiu-Kai Ng, An Automated Approach for Retrieving Hierarchical Data from HTML Tables. In *Proceedings of the Eighth International Conference on Information and Knowledge Management (ACM CIKM'99)*, pp. 466-474, Kansas City, MO, USA, November 1999.

[16] Hong Zhao, Shenyang Liaoning, Bing Liu, Shenyang Liaoning, Zao Jiang, Shenyang Liaoning, Tobias Ostgathe “Global-local-global method for logical structure extraction of form document imaging”, Journal of Electronic Imaging, July 2000, Vol. 09(03), pp.296-304

[17] Essam A. El-Kwae, "Spatial Modeling and Representation of Forms in Large Document Databases,” The 5 th World Multi-Conference on Systemics, Cybernetics and Informatics, SCI 2001, July 22-25, 2001, Orlando, Florida USA, Sheraton World Resort, (Accepted for publication).

[18] Essam A. El-Kwae and Mansur R. Kabuka, “A Robust Framework for Content-Based Retrieval by Spatial Similarity in Image Databases”, ACM Transactions on Information Systems, Vol. 17, No. 2, April 1999, pp.174-198.

[19] J. Mao, M. Abayan, K. Mohiuddin, and E. Wallach, "A Model-Based Form Processing Subsystem," Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96), Vienna, Austria, Aug. 1996, pp.691-695.

[20] S. Taylor, R. Fritzson, J. Rastor, "Extraction of Data from Preprinted Forms," Machine Vision and Applications, Vol. 5, 1992, pp. 211-222.
[21] Teruo Akiyama, and Norihiro Hagita, "Automated Entry System for Printed Documents," Pattern Recognition, Vol. 23, No. 11, 1990, pp. 1141-1154.

[22] Toyohide Watanabe, Qin Luo, and Noboru Sugie, "Layout Recognition of Multi-Kinds of Table Form Documents," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 17. No. 4, April 1995, pp. 432-445.

[23] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” ACM Computing Surveys, Vol. 31, No. 3, September 1999, pp.264-323.

[24] Joh Z. Li, M. Tamer Ozsu, and Duane Szafron, “Spatial Reasoning Rules in Multimedia Management Systems,” *University of Alberta, Technical Report TR96-05*, 1996.