Automatic Acquisition of Layout Knowledge for Understanding Business Cards

Toyohide Watanabe and Xiaoou Huang
Department of Information Engineering,
Graduate School of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-01, JAPAN
Phone: +81-52-789-4409, Fax: +81-52-789-3808
E-mail: watanabe@nuie.nagoya-u.ac.jp

Abstract

The document knowledge plays very important roles in many document image understanding methods. In these methods the document knowledge is utilized to classify/extract individual item data interpretatively from paper-based sheets as a kind of document model. Today, these knowledge are specified into the document image understanding system in advance. In this paper, we propose an experimental method to acquire the layout knowledge automatically from sample document images: especially, we focus on the acquisition subject for business cards. Our idea is to generate the layout knowledge of business cards from a predefined logical structure. Namely, the logical structure is used as a kind of meta-knowledge to interpretatively generate the layout knowledge of given business cards.

1 Introduction

The document image understanding is today one of most interesting subjects. The objective is to classify and extract meaningful data from paper-based documents automatically. Until today, various kinds of documents have been researched, and also many approaches/methods have been developed. The main characteristic in these approaches/methods is to define the knowledge about documents as document models and then interpret document images analytically, using the document models[1-4]. Namely, the document model is very important in the framework of knowledge-based document image understanding. This document model is ordinarily specified by experters as ready-made knowledge.

If the system could compose the document model by itself, end-users can make use of the document image understanding system as their own tools easily. Until today, the researches about automatic acquisition of document knowledge have not always been reported/published sufficiently[5-7]. This is because this subject is very difficult and also the knowledge representation methods adopted in many document image understanding systems are procedural-form. For example, K.Kise et al. proposed a method to acquire the document knowledge about papers and business cards by means of the generalization/ specialization mechanism of acquired information, using positive and negative samples[5]. Their approach has not used document images directly: their samples were the descriptions for characteristics of document structures, formatted/extracted from individual document images in advance. On the other hand, Q.Luo et al. proposed a method to acquire the layout knowledge about table-form documents directly[6,7]. This method can be evaluated in point of direct manipulation of document images in comparison with the other trials. In Luo's method, the analytic idea that all item fields are always surrounded with horizontal and vertical line segments is an important key to be successful, and the framework may also be adaptable to the other documents.

In this paper, we propose an experimental method to acquire the layout knowledge from business cards. In comparison with table-form documents, individual item data in business cards are not surrounded with any horizontal/vertical line segments. It is not easy to extract individual item data and identify them according to the inherent attributes of items.

2 Framework

The items in business cards are generally allocated into appropriate positions in two dimensional rectangular sheets with the adjacency relationships among meaningfully related items. Namely, the constructive relationships among these items are dependent on the logical structure[3,5-8]. The locations of individual items are relatively given with the physical properties of item data such as data length, character size, etc. The coordinate values of individual item data are different, card by card, though the constructive relationships among individual item data may be determined in advance[8]. Of course, the logical structure should be the same even if the item data are not similar: some items may be abbreviated; and the locations of some data may be exchanged by those of others.

2.1 Characteristics of business cards

The following characteristics could be observed as compsition rules for business cards in Japanese:

1. "holder-name", "title", "company", "mark", "affiliation", "address", "phone-number", "Fax-number" and so on are basic items;
2. Line-based allocation of items is used;
3. Individual item data are separated by spaces explicitly;
4. In many cases, individual item data are composed of character strings with different fonts, sizes and styles, but each item contain characters with the same font, size and style;
5. Generally, the character sizes of "holder-name" are largest in all other item data. Also, the gaps among the composite characters of "holder-name" are relatively larger in comparison with other item data;
6. The character sizes of "address", "phone number" and "fax number" are smallest;

0-8186-7898-4/97 $10.00 © 1997 IEEE

7. "mark" is often observed in some cards, and the location is close to "company";
8. In case that item data are repeated, the character sizes and lengths are the same. Also, the data formats are the same or similar;
9. "affiliation" is located to the upper (lower) side or left (right) side of "holder-name" in horizaontal card (or in vertical card).

Figure 1 is an example of business cards. Though these item data are arranged along horizontal axis, the above characteristics can be applied similarly even if the vertical type cards was analyzed. So, we can compose the logical structure for business cards as illustrated in Figure 2.

2.2 Outline of layout knowledge acquisition

The processing flow is shown in Figure 3. The main idea is to make use of logical structure in order to extract the layout knowledge, in addition to heuristics for composing business cards. Furthermore, the representation form of our layout knowledge is very suitable to specify the spatial and geometrical relationships among item fields/blocks. Our processing procedure is divided into three phases. The first phase is a structure decomposition to separate individual item fields and distinguish the item units to manipulate atomic elements. The second phase is a structure analysis to check up the adjacency relationships among item fields and the hierarchical relationships among item fields/blocks, and then interpret the layout structure in accordance with the logical structure and heuristics for composing business cards. Finally, the third phase is a structure establishment to identify the integrated blocks individually owing to the logical characteristics and physical properties of item fields/blocks.

3 Knowledge Representation

Our representation form of layout knowledge is a multi-ways tree. For example, consider blocks in Figure 4: blocks x, x_1, x_2 and x_3 are individual item fields/blocks. x_1, x_2 and x_3 are children, which are divided by horizontal cutters, when x is looked upon as a parent. Under such a consideration, we can represent the relationships between an outer block and inner blocks as a multi-ways tree. Similarly, blocks which are vertically allocated as shown in Figure 5 also are specified as a multi-ways tree. This cutting process is repeated until all inner blocks cannot be separated furthermore. In order to avoid the ambiguity of cutting directions, the horizontal cutting is superior to the vertical cutting. As seen in Figure 4, the top block in the horizontal cutting is represented as the most left child node and the bottom block becomes the most right child node. Similarly, the most left block in the vertical cutting is located to the most left child node and the most right block is corresponded to the most right child node.

The cutting information is set to the parent node with three cells: [MOD, LEV, CO]. MOD indicates the cutting direction such as vertical cutting (V) or horizontal cutting (H). LEV indicates the cutting level: it presents which levels should be selected when the numbers are assigned to individual gaps from "1" to "n" according to the gap width. And, CO accommodates the coordinate values of individual item fields/blocks. Of course, CO is not set in this acquisition step of layout knowledge, but is used when this knowledge is applied to recognize the layout structure practically. The cutting information is attached to non-terminal nodes, but is not assigned to terminal nodes. The terminal node is attached with only one cell NAME. NAME is the name of item, and is assigned to each node in the structure establishment phase.

4 Structure Decomposition

The processing flow is shown in Figure 3. The main idea is to make use of logical structure in order to extract step, characters/symbols are separated along the line-orientation with respect to the characteristic 2) in Section 2.1. In the second step, the item fields are derived by spaces furthermore according to the characteristic 3). From such steps, the partitioned item fields and individually extracted items are shown in Figure 6. Figure 7 illustrates a multi-ways tree which was composed hierarchically according to these steps.

5 Structure Analysis

This procedure is composed of vertical integration step and horizontal integration step. First, the horizontal integration step is performed, using horizontal integration rule, and then the vertical integration step is performed with the vertical integration rule. The item block of horizontally integrated item fields is represented as a child, attached to the parent of attribute "MOD:V". If the child connected to the parent of "MOD:V" is only one after the horizontal integration step, this child is merged into the parent and this merged node is replaced as a terminal node. While, in the vertical integration step terminal nodes connected to the parent of "MOD:H" are checked up.

5.1 Horizontal integration step

It is impossible to integrate individual item fields/blocks with various relationships by only one rule because the elements are composed of characters with different fonts, sizes and styles with respect to the characteristic 4) in Section 2.1.

[Horizontal integration rule]

We define the following parameters in order to specify three integration rules:

- T_{vi}, T_{vj}: horizontally related, and mutually neighboring item fields/blocks
- S_{vi}, S_{vj}: sizes of T_{vi} and T_{vj}
- Interval(T_{vi}, T_{vj}): gap between T_{vi} and T_{vj}
- s_1, s_2, i_1, i_2: threshold values, which are computed through experiments

a) size-integration rule
if $|S_{vi} - S_{vj}| < s_1$
then integrate T_{vi} and T_{vj}.

b) interval-integration rule
if Interval(T_{vi}, T_{vj}) $< i_1$
then integrate T_{vi} and T_{vj}.

c) size/interval-integration rule
if $|S_{vi} - S_{vj}| < s_2$

and Interval(T_{vi}, T_{vj}) < i_2
then integrate T_{vi} and T_{vj}.

[Judgement rule]

This rule takes a role to judge which above rule is applied to integrate item fields/blocks horizontally.

a) for size-integration rule
if (gap between T_{vi} and T_{vj} is less than j_1,
or gap between T_{vi} and T_{vj} is greater than maximum size of T_{vi} and T_{vj}),
and sizes of T_{vi} and T_{vj} are greater than j_2,
then integrate T_{vi} and T_{vj} by size-integration rule.

b) for interval-integration rule
if sizes of T_{vi} and T_{vj} are less than j_1,
then integrate T_{vi} and T_{vj} by interval-integration rule.

c) for size/interval-integration rule
if gap between T_{vi} and T_{vj} is greater than j_1,
and sizes of T_{vi} and T_{vj} are greater than j_1, or less than j_1 ,
then integrate T_{vi} and T_{vj} by size/interval-integration rule.

Here, j_1 and j_2 are threshold values, derived experimentally from the constraint information for item fields/blocks. Figure 8 is the result generated by the horizontal integration rule, and Figure 9 is a representation by the multi-ways tree.

5.2 Vertical integration step

This integration is performed with respect to the vertically neighboring item fields/blocks.

[Vertical integration rule]

if ((Interval(T_{vi}, T_{vj}) < K*Interval(T_{vi-1}, T_{vi}),
or Interval(T_{vi}, T_{vj}) < K*Interval(T_{vj}, T_{vj+1})),
or (Interval(T_{vi}, T_{vj}) < j_1 ,
and $|S_{vi} - S_{vj}| > j_1$)),
and vertical decomposition level is 2 or more,
then integrate T_{vi} and T_{vj}.

Here, "vertical decomposition level is 2 or more" is imposed on the basis of the characteristic 8) in Section 2.1. Interval(T_{vi}, T_{vj}) is the minimum gap between item fields/blocks, included in T_{vi} and T_{vj}, in the vertical direction. K is 1 or less. Figuer 10 is the result. Also, Figure 11 is the representation of multi-ways tree for Figure 10.

6 Structure Establishment

This procedure identifies individual items by interpreting the multi-ways tree with respect to the characteristics 1), 5), and 7)-9), and generates layout knowledge. The structure classification rule is used in this phase, and is applied to judge individual items. Here, the variable G indicates a node distinguished in the structure analysis phase.

a) checking mark item
if G is only single item block of maximum size,
or (G is located to most right node,
and G is composed of 3 or less item blocks,
and gap between item blocks is less than g_1),
or G includes item block whose block size is less than g_2,
then G is Mark-item.

b) checking holder-name item
if G is maximum block size except Mark-item,
and gap between item blocks is largest,
then G is Name-item.

c) checking address item
if G is minimum block size,
and number of blocks is greater than g_3,
then G is Address-item.

d) checking company item
if G is neither Mark-item, Holder-name-item nor Address-item,
and maximum block size is smaller than Holder-name-item,
and size is larger than other distinguished item blocks,
then G is Company-item.

e) checking affiliation item
if G is located near to Holder-nmae-item,
and difference among block sizes is less than g_4,
then G is Affiliation-item.

Here, g_1, g_2, g_3 and g_4 are threshold values. These rules are clearly applied in the order from rule a) to rule e). Figure 12 is the result generated from Figure 11 by these rules.

7 Experiment

The prototype system based on the previous framework was implemented on SUN SPARCStation-10, using the programming language C. In this experiment, 30 business cards were used. These cards were binalized after scanned in 200 dpi and 256 grey levels. In Figure 13 we show 4 typical examples used in our experiments.

Figure 14 shows the results distinguished from Figure 13, respectively. And, Figure 15 illustrates the multi-ways trees, corresponding to Figure 14. Only one business card was not identified in our experiment. This is because the composition structure of "address" is different from characteristics 4), 6) and 8).

8 Conclusion

In this paper, we proposed an automatic acquisition method of layout knowledge in business cards. The framework is to compose the layout knowledge by interpreting business card images with a logical structure as meta knowledge and constraint information concerning the characteristics for composing business cards. Also, we reported that our method was successful through some experiments. This successful result depends on the fact that the representation form of layout knowledge is very powerful. So, the representation ability is more abstracted than those proposed in other methods. Of course, the interpretative ability has already been made clear in the paper[1-3].

In our future work, we must investigate the following issues: optimization of layout knowledge, and expansion of interpretation mechanism.

Acknowledgements

We are very grateful to Prof.T.Fukumura of Chukyo University, and Prof.Y.Inagaki and Prof.J.Toriwaki of Nagoya University for their perspective remarks, and also wish to thank our research members for their many discussions and cooperations.

References

[1] T.Watanabe, Q.Luo and N.Sugie: "A Cooperative Document Understanding Method among Multiple Recognition Procedures", *Proc.of 11th ICPR*, pp.689-692(1992).

[2] Q.Luo, T.Watanabe, Y.Yoshida and Y.Inagaki: "Recognition of Document Structure on the Basis of Spatial and Geometric Relationships between Document Items", *Proc.of MVA'90*, pp.461-464(1990).

[3] T.Watanabe, Q.Luo and N.Sugie: "Structure Recognition Methods for Various Types of Documents", *Int'l J.of Machine Vision and Applications*, Vol.6, pp.163-176(1993).

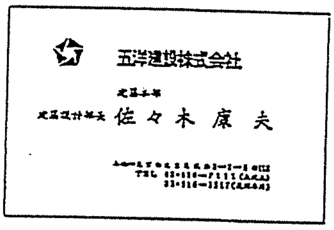
[4] D.Niyogi and S.Srihari: "A Rule-based System for Document Understanding", *Proc.of AAAI'86*, pp.789-793(1986).

[5] K. Kise, N. Yajima, N. Babaguchi and K. Fukunaga: "Incremental Acquisition of Knowledge about Layout Structures from Examples of Documents", *Proc.of ICDAR'93*, pp.668-671(1993).

[6] Q.Luo, T.Watanabe and N.Sugie: "Structure Recognition of Table-form Documents on the Basis of the Automatic Acquisition of Layout Knowledge", *Proc.of MVA'92*, pp.79-82(1992).

[7] T.Watanabe, Q.Luo and N.Sugie: "Layout Recognition of Multi-kinds of Table-form Documents", *IEEE trans.on PAMI*, Vol.17, No.4, pp.432-445(1995).

[8] F.-H.Yao, A.Tamaki and K.Kato: "Layout Recognition of Japanese Business Card", *Proc.of ICSPAT'95*, pp.1112-1116(1995).



五洋建設株式会社

佐 々 木 康 夫

Figure 1 Example of business card

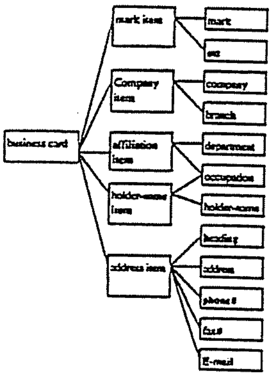


Figure 2 Logical structure

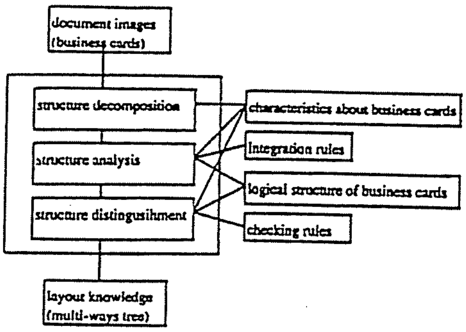
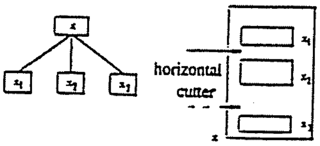


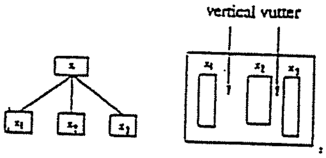
Figure 3 Processing flow



x: [MOD:H, LEV:2, CO:[(x1)(x2)(x3)]]

note: (xi) is coordinate values of surrounding item fields/blocks for xi, separated by cutters

Figure 4 Horizontal cutter



x: [MOD:V, LEV:2, CO:[(x1)(x2)(x3)]]

Figure 5 Vertical cutter

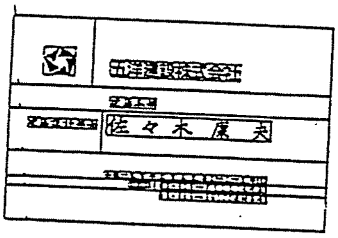
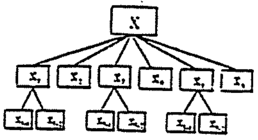


Figure 6 Partitioned item fields and extracted items



x: [MOD:H, LEV:5, CO:[(x1)(x2)(x3)(x4)(x5)(x6)]]
x1: [MOD:V, LEV:1, CO:[(x11)(x12)]]
x2: [NAME:]
x3: [MOD:V, LEV:1, CO:[(x31)(x32)]]
x4: [NAME:]
x5: [MOD:V, LEV:1, CO:[(x51)(x52)]]
x6: [NAME:]
x11: [NAME:]
x12: [NAME:]
x31: [NAME:]
x32: [NAME:]
x51: [NAME:]
x52: [NAME:]

Figure 7 Multi-ways tree for Figure 6

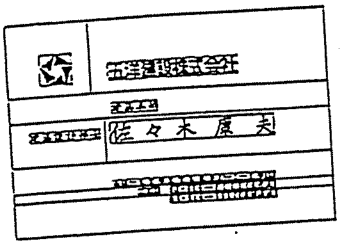


Figure 8 Horizontally integrated blocks

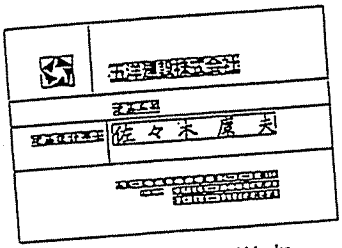
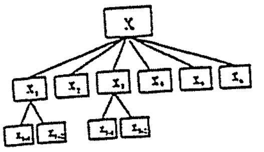
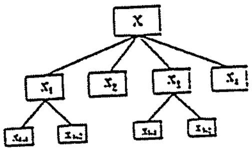


Figure 10 Vertically integrated blocks



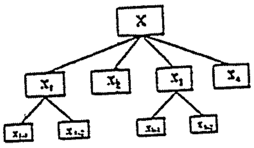
x: [MOD:H, LEV:5, CO:[(x1)(x2)(x3)(x4)(x5)(x6)]]
x1: [MOD:V, LEV:1, CO:[(x11)(x12)]]
x2: [NAME:]
x3: [MOD:V, LEV:1, CO:[(x31)(x32)]]
x4: [NAME:]
x5: [NAME:]
x6: [NAME:]
x11: [NAME:]
x12: [NAME:]
x31: [NAME:]
x32: [NAME:]

Figure 9 Multi-ways tree for Figure 8



x: [MOD:H, LEV:3, CO:[(x1)(x2)(x3)(x4)]]
x1: [MOD:V, LEV:1, CO:[(x11)(x12)]]
x2: [NAME:]
x3: [MOD:V, LEV:1, CO:[(x31)(x32)]]
x4: [NAME:]
x11:[NAME:]
x12:[NAME:]
x31:[NAME:]
x32:[NAME:]

Figure 11 Multi-ways tree for Figure 10



x: [MOD:H, LEV:3, CO:[(x1)(x2)(x3)(x4)]]
x1:[MOD:V, LEV:1, CO:[(x11)(x12)]]
x2:[NAME: affiliation]
x3:[MOD:V, LEV:1, CO:[(x31)(x32)]]
x4:[NAME: address]
x11:[NAME: mark]
x12:[NAME: company]
x31:[NAME: affiliation]
x32:[NAME: holder-name]

Figure 12 Layout knowledge

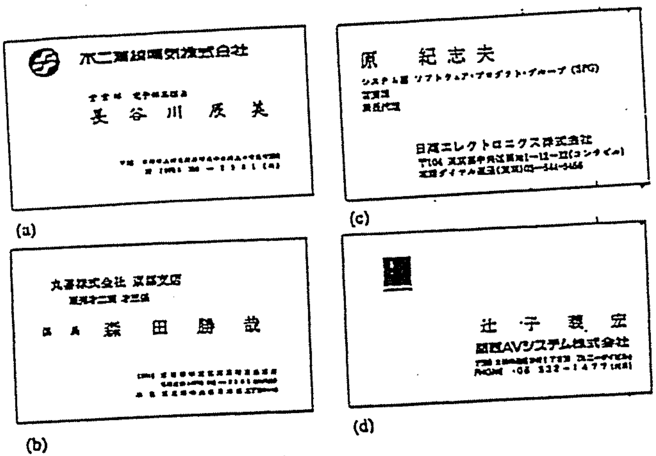


Figure 13 Typical examples in experiment

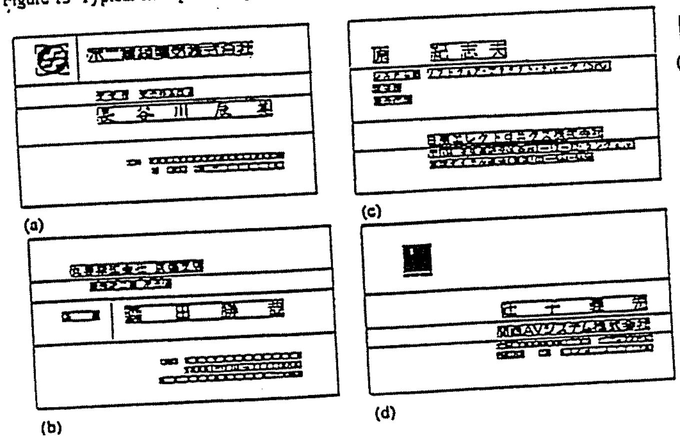
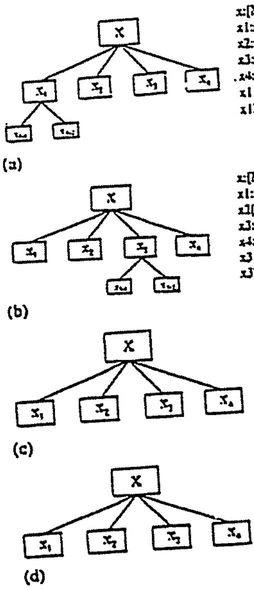


Figure 14 Parttitioned cards in experiment



(a)
x:[MOD:H, LEV:3, CO:[(x1)(x2)(x3)(x4)]]
x1:[MOD:V, LEV:1, CO:[(x11)(x12)]]
x2:[NAME: affiliation]
x3:[NAME: holder-name]
x4:[NAME: address]
x11:[NAME: mark]
x12:[NAME: company]

(b)
x:[MOD:H, LEV:3, CO:[(x1)(x2)(x3)(x4)]]
x1:[NAME: company]
x2[NAME: affiliation]
x3:[MOD:V, LEV:1, CO:[(x31)(x32)]]
x4:[NAME: address]
x31:[NAME: affiliation]
x32:[NAME: holder-name]

(c)
x:[MOD:H, LEV:3, CO:[(x1)(x2)(x3)(x4)]]
x1:[NAME: holder-name]
x2:[NAME: affiliation]
x3:[NAME: company]
x4:[NAME: address]

(d)
x:[MOD:H, LEV:3, CO:[(x1)(x2)(x3)(x4)]]
x1:[NAME: mark]
x2:[NAME: holder-name]
x3:[NAME: company]
x4:[NAME: address]

Figure 15 Representations of layout knowledge in experiment