Layout Recognition of Multi-Kinds of Table-Form Documents

Toyohide Watanabe, *Member, IEEE Computer Society*, Qin Luo, and Noboru Sugie

***Abstract*—Many approaches have reported that knowledge-based layout recognition methods are very successful to classify the meaningful data from document images automatically. However, these approaches are applicable to only the same kind of documents because they are based on the paradigm that specifies the structure definition information in advance so as to be able to analyze a particular class of documents intelligently. In this paper, we propose a method to recognize the layout structures of multi-kinds of table-form document images. For this purpose, we introduce a classification tree to manage the relationships among different classes of layout structures. Our recognition system has two modes: layout knowledge acquisition and layout structure recognition. In the layout knowledge acquisition mode, table-form document images are distinguished according to this classification tree and then the structure description trees which specify the logical structures of table-form documents are generated automatically. While, in the layout structure recognition mode, individual item fields in the table-form document images are extracted and classified successfully by searching the classification tree and interpreting the structure description tree.**

***Index Items*—Recognition paradigm for multi-kinds of table-form documents, automatic acquisition of layout knowledge, recognition of document classes, recognition of layout structures, classification tree, structure description tree.**

I. Introduction

THE knowledge-based approach for understanding document images is different from the traditional approaches based on the image-processing techniques, because this paradigm is based on the interpretation of document images with various knowledge as an application of the artificial intelligence technology. This approach is successful with respect to the flexibility, applicability and adaptability in comparison with the traditional image-processing-dependent approaches [1], [2], [3]. In particular, the individually recognized results can be explained explicitly in accordance with the interpretation process of document images. Until today many knowledge-based approaches have been reported, and they focused mainly on the layout analysis issue because the characteristics among various documents are not always independent of application-specific layout structures [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. However, all these approaches paid attention to the layout recognition methods for single-formed or particular application-dependent document images. Namely, these currently proposed methods are not applicable to the recognition/analysis of different kinds of document images. This is because the knowledge, which was extracted from a particular class of documents, is not successful to interpret different classes of documents. In such a framework, it is impossible to deal with a mixed set of different document images at once as the batch processing, even if the knowledge is powerful in the representation ability.

In this paper, we address the layout recognition issue for multi-classes of document images. Of course, the documents to be addressed here are limited though many types/classes of documents are available currently in our real world [14], [15]. Thus, we select table-form documents as our recognition objects. Although the types and kinds of table-form documents are too many, in general the layout structures are well defined by the vertical and horizontal line segments [13]. To attain to our objectives we concentrate on the following subjects:

- Classification of document classes.
- Acquisition of layout knowledge.
- Identification of item fields.

These subjects may have to be investigated independently because each subject itself is important and difficult. However, we attach to these subjects together from a viewpoint of the framework for document image understanding. So, we regard these subjects as important composite factors in case that the document image understanding system could be effectually utilized by end-users. At least, it is not easy for end-users to specify the document knowledge according to the currently investigated methods. The classification subject of document classes is fundamental and requisite to manage such document knowledge individually. Also, the subject for the automatic acquisition of layout knowledge is required strongly with a view to solving such a problem [16].

This paper is organized as follows. In Section II, we describe the basic concept and framework in our approach. Section III summarizes our recognition method. Sections IV, V, and VI report the methods about individual research subjects, respectively. In Section IV, the recognition of document classes is addressed, in Section V, the acquisition of layout knowledge is discussed, and in Section VI, the recognition of layout structures is shown. In Section VII, the results for individual research subjects are arranged through some experiments. Finally, in Section VIII our method is concluded under the characteristics and discussions.

Manuscript received Nov. 1, 1993; revised Aug. 10, 1994.
T. Watanabe is with the Department of Information Engineering, Faculty of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-01, Japan; e-mail watanabe@nuie.nagoya-u.ac.jp.
Q. Luo is with the Department of Electronic Information Engineering at Toyama University.
N. Sugie is with the School of Engineering at Meijyo University.
IEEECS Log Number P95039.

II. Layout Structure of Table-Form Document

Table-form documents are composed of the meaningful item fields, surrounded by vertical and horizontal line segments. Of course, these item fields compose more complicated structures with their neighboring and constructive relationships: hierarchical structures, repeating structures, two dimensional-array structures and so on.

A. Physical Information and Logical Information

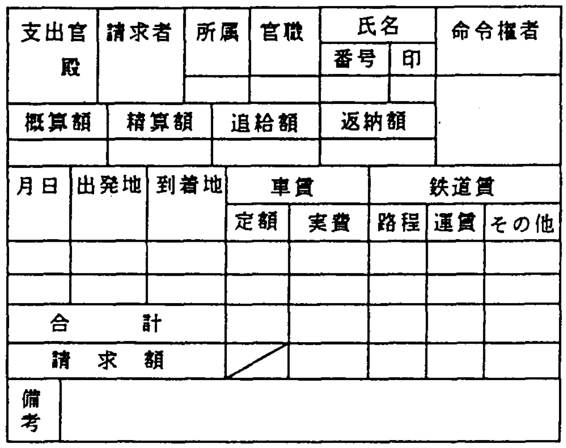
Table-form documents are well formed, and individual item data are allocated into the predefined positions [13], [14], [15]. Therefore, it is convenient to use the physical information such as positions, sizes, lengths and so on in order to extract individual item data. It is not only comprehensive to make use of the physical information with a view to analyzing the layout structure, but also effective in point of the processing efficiency. However, the physical information is not necessarily adaptable to the interpretation of documents whose layout structures are geometrically modified, even if their logical structures are the same: the expanded/reduced documents, the transformed documents and so on. For example, two table-form documents in Fig. 1 are differently interpreted because the positions to be assigned to individual item fields are different at all though they might be interpreted as the same application-specific document conceptually.

While, it is easy to identify these table-form documents as the same structure if we can use the logical information such as the neighboring/constructive relationships among item fields. In Fig. 1, two different table-form documents can be regarded as the same structure on the basis of the logical information. Of course, it is more complicated to compose the logical information as layout knowledge than to collect the physical information because the semantical interpretation for layout structures is required [16].

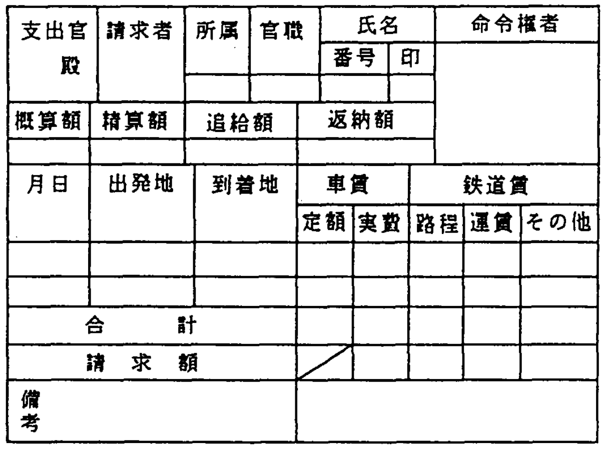
B. Representation of Layout Structure

The logical information is very applicable in comparison with the physical information. However, complicated processing is required to extract the logical information from documents: of course, the abstraction level from physical information to logical information is not always uniquely determined since it is a trade-off problem between the difficulty of information extraction and the efficiency of knowledge interpretation. We adapt a binary tree to represent the layout structures of table-form documents logically. We call this tree the structure description tree [13], [14], [15]. The structure description tree consists of global structure tree and local structure trees, which define the global and local characteristics of table-form documents, respectively.

Global structure tree. A table-form document consists of several blocks, which are meaningful sets of adjacent item fields. The global structure tree describes the neighboring relationships among blocks of structural units. The nodes correspond to these structural blocks and are categorized into three node types, according to the repeating structure and the repeating direction: vertically repeated node "D" (for down); horizontally repeated node "R" (for right); and non-repeated node "S" (for single). Namely, the constructive characteristics of individual blocks are distinguished as "D", "R" or "S" when we scan document images from the left-upper corner to the right-lower corner.



(a) Tab.1



(b) Tab.2

Fig. 1. Examples of table-form documents.

We show three node types in Fig. 2. The edges represent the adjacent relationships among blocks. When a table-form document image is scanned from the left-upper corner, the left and right edges link the blocks which are located respectively at the lower and right sides of the current block, as shown in Fig. 3. This link mechanism interprets the adjacent relationships among blocks on the basis of location relationships among the left-upper corners of individual blocks.

Local structure tree. The local structure trees describe the internal structures of blocks. The nodes indicate individual item fields or blocks. The edges point out that item fields are connected either vertically or horizontally to other item fields or blocks when a block is scanned from the left-upper corner to the right-lower corner. The nodes are divided into the

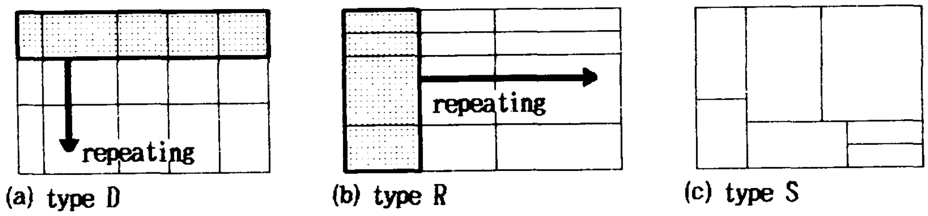


Fig. 2. Adjacent relationships among blocks.

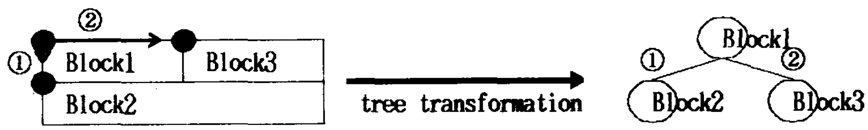


Fig. 3. Node types in global structure tree.

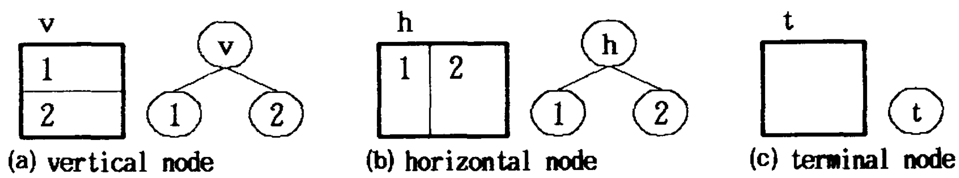


Fig. 4. Node types in local structure tree.

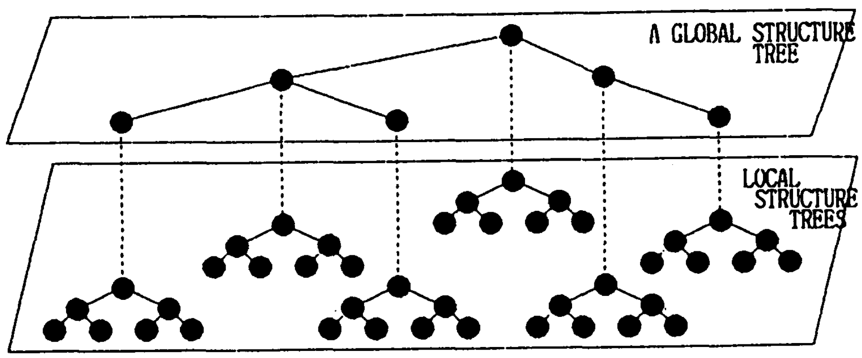


Fig. 5. Structure description tree.

vertical node "v" (for vertical), horizontal node "h" (for horizontal) and terminal node "t" (for terminal). When the parent node is a vertical node, the upper and lower item fields separated by the cutting line are connected through the left and right edges, respectively. When the parent node is a horizontal node, the left and right item fields are connected through the left and right edges, as shown in Fig. 4. The item fields that can not be further divided are terminal nodes.

Fig. 5 shows the relationship between the global structure tree and local structure trees, conceptually. Each local structure tree is attached to the corresponding node in the global structure tree. Fig. 6(a) shows a globally analyzed layout structure for the table-form document in Fig. 1(a). The meshed item fields in Fig. 6(a) indicate that they are name fields, while the blank item fields represent to be data fields. Fig. 6(b) represents the corresponding global structure tree derived from Fig. 6(a). All local structure trees associated with the nodes in Fig. 6(b) are illustrated in Fig. 6(c). Our global structure tree and local structure trees are specified only by the logical information. Therefore, the structure description tree in Fig. 6 is also applicable to the interpretation of the table-form document in Fig. 1(b) though this was meaningfully generated from the document in Fig. 1(a).

C. *Class of Table-Form Document*

The document class is defined as a set of table-form documents whose layout structures can be uniquely identified by the same layout knowledge [3]. The knowledge about document classes must be able to categorize many different table-form documents effectively so that appropriate knowledge of layout structures can be applied to individual documents of the distinguished document classes. This knowledge can be

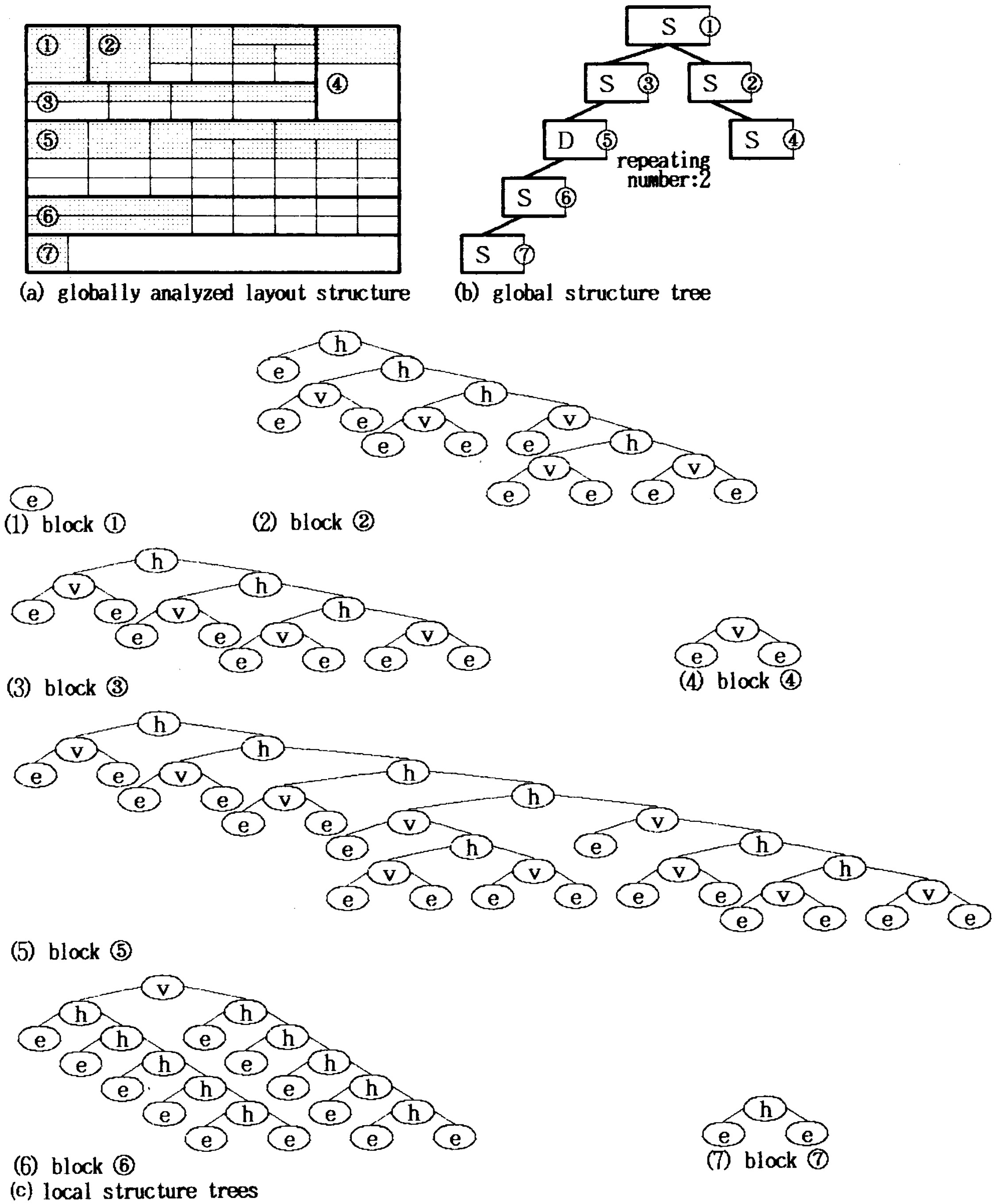


Fig. 6. An example of structure description tree.

represented with a multi-way tree. We call this tree the classification tree. The node corresponds to each document class, and indicates the knowledge of layout structure (such as structure description tree) to be adaptable to the class. While, the edge represents the parent-child relationship among document classes. Namely, child document classes are generated stepwisely from the parent document class. For example, in Fig. 7 we illustrate the classification tree for several table-form documents which are shown in Fig. 1 and Fig. 8. The meshed nodes indicate the document classes for table-form documents in Fig. 1 and Fig. 8.

This tree grows up when a new table-form document, which does not match with the existing document classes consistently, should be registered. The node for the new table-form

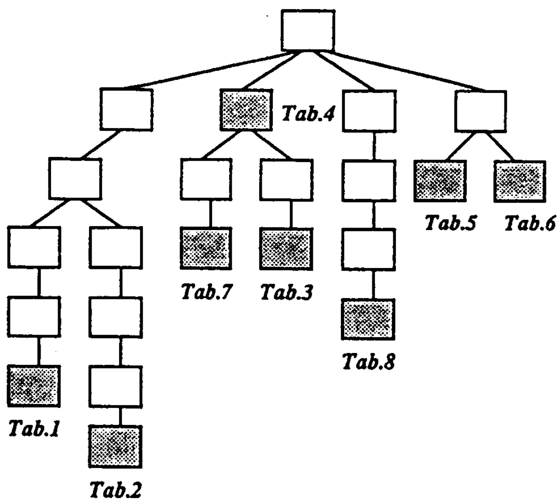


Fig. 7. Classification tree.

document is inserted to the node of the most similar layout structure from the existing document classes. Namely, the new node is attached as a child when some blocks in the existing nodes are moreover partitioned by vertical/horizontal line segments. This block division process is illustrated in Fig. 9. In Fig. 9, the right side is furthermore partitioned in comparison with the left side. Here, the left side is transformed into the upper node, while the right side is done into the lower in our classification tree. Of course, this block division process generates various branches, according to the locations of longest vertical/horizontal line segments.

D. Relationship between Classification Tree and Structure Description Trees

Fig. 10 shows the relationship between classification tree composed physically on the basis of the structural information of table-form documents. On the other hand, the structure description tree represents the adjacent, connective relationship among the item fields/blocks logically and indicates the constructive characteristics such as hierarchical, repeating and two-dimensional array structures. Therefore, the classification tree and structure description tree are not always consistent from a viewpoint of representation.

For example, two table-form documents in Fig. 1 are different with respect to the physical positions of line segments though they are looked upon as the same document in point of their logical structures. Thus, the documents in Figs. 1(a) and 1(b) are registered to different nodes in the classification tree though the structure description trees are the same. Namely, the correspondence is redundant. Fig. 11 shows such a correspondence between nodes in the classification tree and nodes in the structure description tree. The classification tree makes it easy to analyze the physical characteristics of layout structures because it depends on only the number and length of line segments. Additionally, the analysis time is very fast. On the other hand, the structure description tree makes it flexible, adaptable, applicable to interpret various document images because it indicates the logical characteristics of document images. It is not difficult to check up whether or not structure description tree has been already registered. If the distinction between the existing structure description trees and newly input structure description tree is explicit in the global structure tree, the matching step stops in the global structure tree. The checking procedure is approximately as follows:

First, we check up the global structure tree, which was generated newly from input document image, with the existing global structure trees. If the matching between new and existing global structure trees failed, the new structure description tree must be registered as one of different layout knowledge.

Second, when the matching succeeds, the local structure trees must be checked successively. If they mismatched, the input document image is stored newly as one of structure description trees. Otherwise, the structure description tree for the input document image has been already registered.

III. Outline of Recognition

The first phase to recognize the layout structures of individual document images from various classes of table-form documents is to identify the document class, using the classification tree. The second phase is to interpret the document image, using the structure description tree linked from the identified node of the classification tree. If in the first phase an appropriate document class could not be selected from the classification tree, its document image is regarded as a new document class. In this case, it is not only necessary to update the existing classification tree with the new document class, but also to register the structure description tree, which can analyze the new table-form document image, into the layout knowledge-base.

Thus, our recognition system is composed of layout structure recognition module and layout knowledge acquisition module, as illustrated in Fig. 12. Here, the document class recognition process is one sub-module and invoked by both layout structure recognition process and layout knowledge acquisition process. The layout structure recognition process judges first whether the document class, appropriate to the input document image, has been already registered. If it does not exist, the processing stops. Otherwise, the recognition process continues successively to analyze the layout structure with the structure description tree, linked from the node of the classification tree. While, in the layout knowledge acquisition process the input data is an unused table-form document image and the output data is a structure description tree, generated from input document images. This process searches first the classification tree to find out which node in the classification tree the new structure description tree is attached to. Next, the structure description tree is composed automatically, using the generalized composition rule about layout structures of table-form documents. This generalized composition rule is a kind of meta-knowledge. Finally, the structure description tree is registered into the layout knowledge-base and also linked to the node in the classification tree.

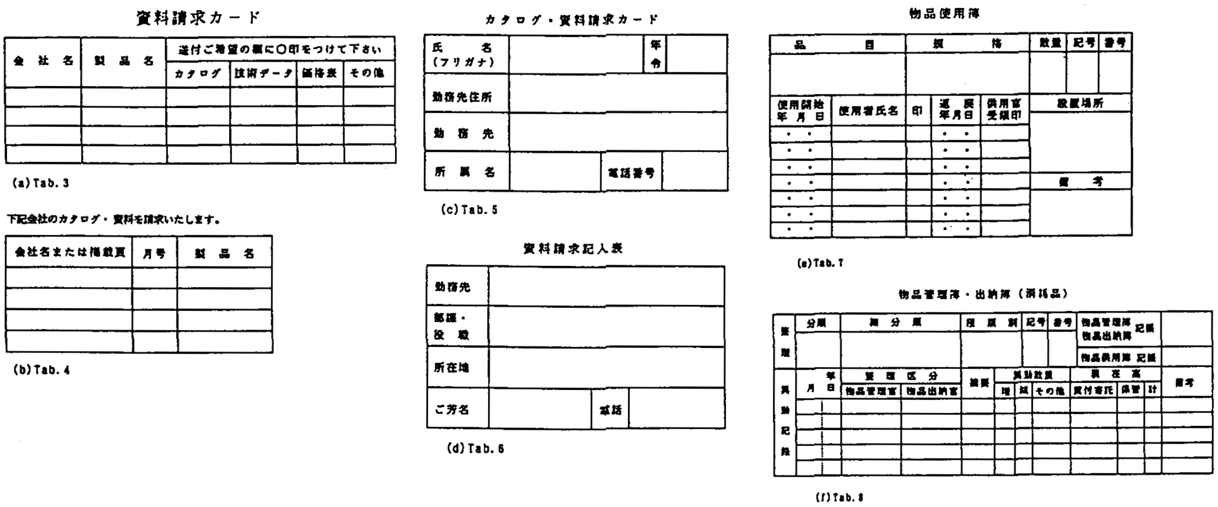


Fig. 8. Other examples of table-form documents.

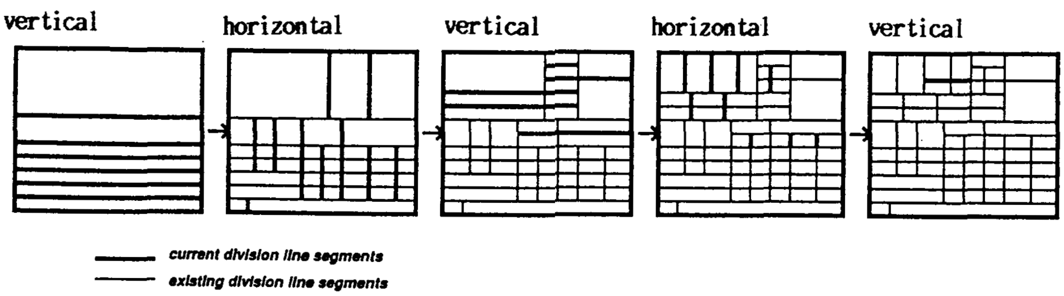


Fig. 9. Block division process.

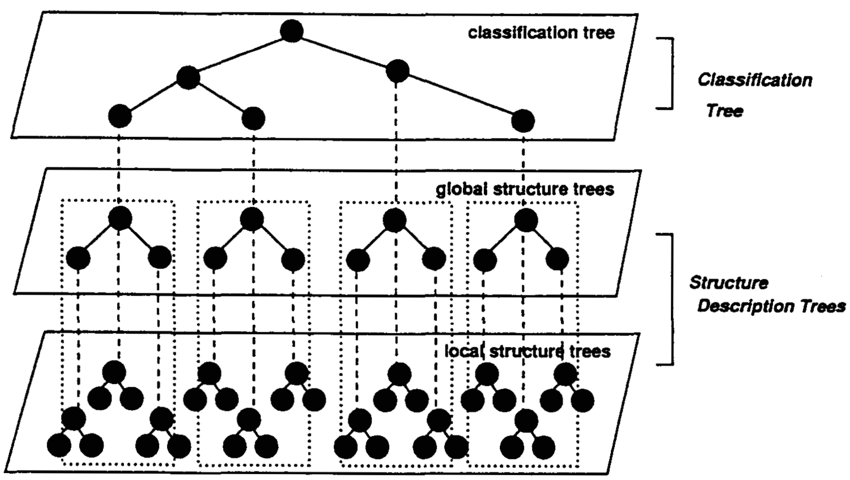


Fig. 10. Relationship between classification tree and structure description trees.

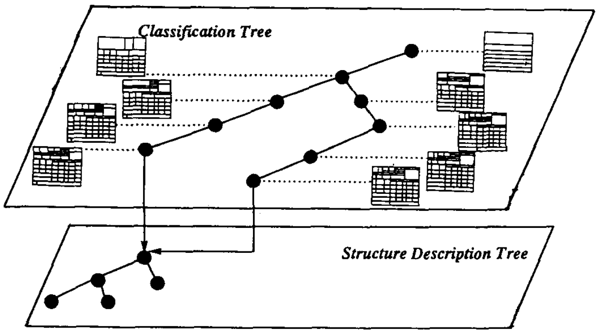


Fig. 11. Structure description tree for classification tree.

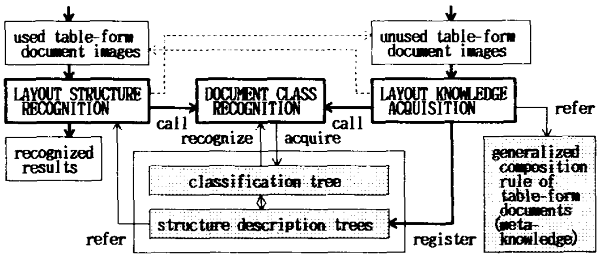


Fig. 12. Framework of layout recognition.

IV. Recognition of Document Class

The classes of table-form documents are distinguished in accordance with the classification tree [3]. In the layout knowledge acquisition process, the procedure takes a role of finding out which node the new document class is linked to. While, in the layout structure recognition process, the procedure finds out the appropriate document class from the existing nodes of the classification tree.

First, we show an algorithm of document class recognition in the layout knowledge acquisition process.

[Algorithm-1]

Initial states: $k = 0$; $Q = ''$; $q = p$.

1) $addqueue(F_0, Q)$.
2) $addqueue(F_1, Q)$.
3) $F := first(Q)$.
4) Compute pixel distributions for F in X- and Y-axes. And then extract candidates from vertical and horizontal line segments.
5) Select candidates, whose terminals connect to most outer surrounded line segments of F. And then set number of horizontal line segments to variable i and number of vertical line segments to variable j.
6) `if` $i \neq 0$ `or` $j \neq 0$, `then` insert (i, j) into $node_k$, and also add regions which were divided by line segments to Q.
7) $F := first(Q)$.
8) `if` $F \neq F_1$, `then goto 4`.
9) If appropriate nodes are not found out after having searched classification tree by following procedure, `then` connect $node_k$ to classification tree.

```
    if p ↓ .ps = NULL then {p ↓ .ps := & (node_k);
          p := & (node_k)}
       else {p := p ↓ .ps;
             while p ↓ .data ≠ node_k and p ≠ NULL
                 do {q := p; p := p ↓ .pt}
             if p = NULL then {q ↓ .pt := & (node_k);
                 p:= & (node_k)}}
```

10) `if` $Q = ''$, `then` Finish.
11) $k := k + 1$; $addqueue(F_1, Q)$; `goto 7`.

Here, Q is a queue to register regions, which are recursively divided by horizontal and vertical line segments until every item field is identified. Additionally, Q has coordinate values to represent the left-upper and right-lower points of divided regions. The coordinate values are denoted by F: in particular, F_0 corresponds to the largest region, surrounded by outer line segments of table-form document image. Moreover, F_1 is an indicator for the region, generated by a division. The node in the classification tree is composed as a record of three elements: $(ps, pt, data)$. ps is an indicator for the most left node; pt is an indicator for the brother nodes; and $data$ is (i, j). p and q are used to search the classification tree, and point out the root node in the classification tree initially. $addqueue$ is a procedure to add an element to Q. $first$ is a procedure to take out the first element from Q. & is a procedure to compute the address of parameter, and $node_k$ represents a k-level node of the classification tree.

Next, we show an algorithm of document class recognition in the layout structure recognition process. The basic algorithm is the same as Algorithm-1, but we must rewrite the step 9.

[Algorithm-2]

1) to 8) the same as Algorithm-1.

9) Search classification tree as follows:

```
  if p ↓ .ps = NULL then Finish
     else {p := p ↓ .ps;
           while p ↓ .data ≠ node_k.data and
                 p ≠ NULL do p := p ↓ .pt}
           if p = NULL then Finish}
```

10) to 11) the same as Algorithm-1.

V. Aquisition of Layout Knowledge

The automatic acquisition process of layout knowledge is shown in Fig. 13. This process is composed of structure decomposition, structure extraction and knowledge composition phases [11], in addition to the identification phase of document class [16]. In the structure decomposition phase, the horizontal and vertical line segments are extracted and then item fields are identified on the basis of these line segments. Next, item fields for printed characters are distinguished as name fields and other item fields are as data fields. In the structure extraction phase, the dependent relationships between name fields and data fields are extracted, using the generalized composition rule. The generalized composition rule is a kind of meta-knowledge, applicable to the interpretation of the layout structures for many classes of table-form documents. Finally, in the phase of knowledge composition, the global structure tree and local structure trees are produced, according to the information obtained from the previous phases.

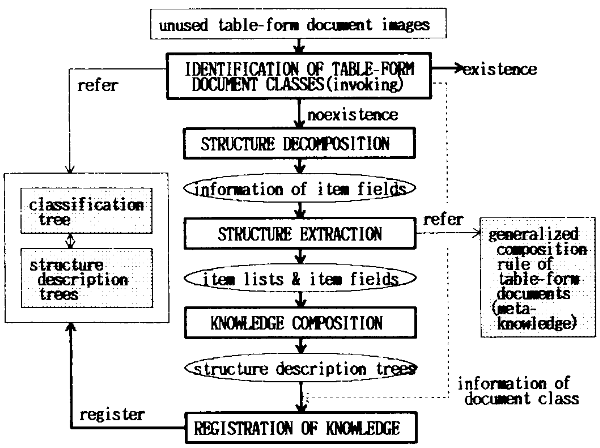


Fig. 13. Layout knowledge acquisition.

A. Dependent Relationship Among Item Fields

There are dependent relationships between name fields and data fields. We define that a data field *a* is dependent on a name field *A* if *A* and *a* have a dependent relationship. The dependent relationship is classified into four classes, as shown in Fig. 14: 1-horizontal dependence, 1-vertical dependence, n-horizontal dependence and n-vertical dependence. The relationships are assigned to the mutually related item fields by arrows.

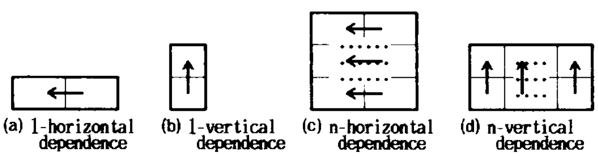


Fig. 14. Dependent relationships.

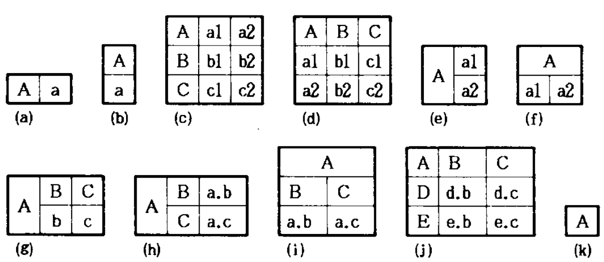


Fig. 15. Structure fragments.

The structure fragment is defined for dependent item fields. Fig. 15 shows some examples of structure fragments. Here, the upper-case characters are name fields and the lower-case characters are data fields. (h) and (i) show hierarchical-form types. (j) is an array-form type and (k) shows a structure fragment of single name field. These structure fragments can be explained, using the dependent relationships. For example, in the array-form type (j), data fields *d.b* and *e.b*, or *d.c* and e.c are vertically dependent on the name field *B* or *C*, respectively; and data fields *d.b* and *d.c*, or *e.b* and *e.c* are horizontally dependent on the name field *D* or *E*, respectively. And also, the name fields *B* and *C*, or *D* and *E* are horizontally or vertically dependent on the name field *A*. Item fields organize structure fragments systematically, and the structure fragments construct blocks with the mutual correspondence. Moreover, the adjacent/connective blocks compose a table-form document.

B. Structure Decomposition

The structure decomposition phase includes three procedures: 1) extraction of line segments; 2) identification of item fields; and 3) classification of item fields. The recursive algorithm for boundary distribution is effective to extract line segments [17]. The most inner areas enclosed by horizontal/vertical line segments are looked upon as item fields and individual item fields are distinguished as name fields or data fields by depending on whether they include printed characters or not. Additionally, the information such as a field attribute(of a name field or a data field), coordinate values(of the left-upper and right-lower corners), etc. is assigned to each item field. This information is useful in order to identify individual document components and generate the structure description tree. Fig. 16 shows a distinguished result for the table-form document image in Fig. 1(a). The black and white areas are distinguished as name fields and data fields, respectively.

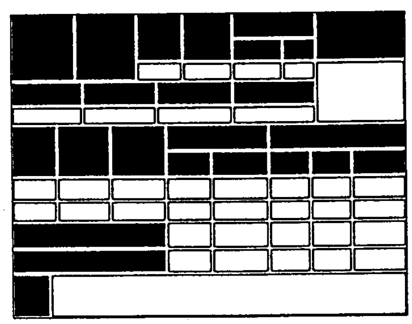


Fig. 16. Discrimination of item fields.

C. Structure Extraction

The structure extraction phase is a procedure which identifies the structure fragments and blocks according to the dependent relationships. The generalized composition rule is applicable in this phase. We use the production system to describe this rule. These production rules are categorized into three groups with respect to the available roles.

Determination of dependent relationship. The following rules are applied to determine the dependent relationships among item fields. The symbols d_i and d_j indicate data fields, and t_j represents a name field.

(a-1) IF d_i is located directly at right side of t_j, THEN d_i is 1-horizontal-dependence of t_j.

(a-2) IF d_i is located directly at right side of d_j, THEN d_i inherits 1-horizontal-dependence property of d_j.

(a-3) IF d_i is located directly at lower side of t_j, THEN d_i is 1-vertical-dependence of t_j.

(a-4) IF d_i is located directly at lower side of d_j,
THEN d_i inherits 1-vertical-dependence property of d_j.

Classification of structure fragments. Item fields are merged meaningfully into structure fragments on the basis of their dependent relationships. The structure fragments are categorized into 11 types, as shown in Fig. 15.

1) Composition of structure fragments

We introduce the item list L_j [t_j, d_{j1}, ..., d_{jn}], which describes a set of related item fields. They are mutually linked in the horizontal or vertical direction, and data fields d_{j1}, ..., d_{jn} have the same horizontal or vertical dependency.

(b-1) IF d_i is 1-horizontal-dependence or 1-vertical pendence against t_j,
THEN compose L_j [t_j, d_i].

(b-2)] IF d_i is located directly at right or lower side of d_{jn} in L_j [t_j, d_{j1}, ..., d_{jn}],
AND d_i is 1-horizontal-dependence or 1-vertical-dependence of t_j,
AND dependent relationship of d_i is consistent to that of d_{jn},
THEN L_j [t_j, d_{j1}, ..., d_{jn}] becomes L_j [t_j, d_{j1}, ..., d_{jn}, d_i].

Here, the condition for dependent relationship of d_i is consistent with that of d_{jn} is set to distinguish 1-dimensional-array-form and 2-dimensional-array-form from structure fragments with repeating property. If their dependent relationships are not consistent, these item fields are not merged.

Next, we compose structure fragments on the basis of these item lists. In this case, it is necessary to combine the mutually related item lists as the same structure fragments when one or more item lists are repeated.

(c-1) IF L_j [t_j, d_{j1}, ..., d_{jn}] exists,
THEN item list constructs structure fragment F_j [L_j].

(c-2) IF L_i and F_j [L_{j1}, ...] exist,
AND L_j is directly linked to L_i from right or lower side,
THEN F_j [L_j, ...] becomes F_j [L_i, L_{j1}, ...].

(c-3) IF L_i and F_j [... , L_{jn}] exist,
AND L_j is directly linked to L_{jn} from right or lower side,
THEN F_j [... , L_{jn}] becomes F_j [... , L_{jn}, L_i].

2) Classification of structure fragments with repeating property

If a structure fragment consists of an item list which has several data fields, it is determined as the repeating structure. Five types (a)-(d) and (j) in Fig. 15 are distinguished, using the following rules.

(d-1) IF a data field d_i in L_i [t_i, d_{i1}, ..., d_{im}] is 1-horizontal-dependence of t_i,
THEN F_i for L_i belongs to type (a).

(d-2) IF d_{i1} in L_i [t_i, d_{i1}, ..., d_{im}] is 1-vertical-dependence of t_i,
THEN F_i for L_i belongs to type (b).

(d-3) IF d_{ix} $(1 \le x \le m,\ m > 1)$ in L_i [t_i, d_{i1}, ..., d_{im}] are 1-horizontal-dependence of t_i,
THEN F_i for L_i belongs to type (c).

(d-4) IF d_{ix} $(1 \le x \le m,\ m > 1)$ in L_i [t_i, d_{i1}, ..., d_{im}] are 1-vertical-dependence of t_i,
THEN F_i for L_i belongs to type (d).

Since the rules (d-1), ..., (d-4) can distinguish one-dimensional array-form structures, we must look for another repeating property in two-dimensional array-form structure.

(e-1) IF F_i belongs to types (c) and (d),
AND the data field of F_i is a data field of F_j and vice versa, at once,
THEN combine F_i and F_j to reconstruct new type for structure fragment of type (j).

3) Classification of structure fragments with hierarchical property

By means of the dependent relationships of n-horizontal-dependence and n-vertical-dependence, we can recognize the structure fragments of types (g), (h) and (i).

(f-1) IF F_i belongs to type (b),
AND its name field is n-horizontal-dependence of t_j,
THEN combine t_j and F_i to reconstruct new type for structure fragment of type (g).

(f-2) IF F_i belongs to type (c),
AND its name field is n-horizontal-dependence of t_j,
THEN combine t_j and F_i to reconstruct new type for structure fragment of type (h).

(f-3) IF F_i belongs to type (d),
AND its name field is n-vertical-dependence of t_j,
THEN combine t_j and F_i to reconstruct new type for structure fragment of type (i).

Applying the above rules, item fields which have not yet been combined to some structure fragment, are organized as structure fragments of types (e), (f) and (k).

(g-1) IF d_{i1}, ..., d_{im} are data fields,
AND d_{i1}, ..., d_{im} are n-horizontal-dependence of t_i,
AND d_{ix+1} is directly linked to d_{ix} $(1 \le x \le m-1)$ from lower side,
THEN combine them as a new structure fragment F_i [t_i, d_{i1}, ..., d_{im}] with type (e).

(g-2) IF d_{i1}, ..., d_{im} are data fields,
AND d_{i1}, ..., d_{im} are n-vertical-dependence of t_i,
AND d_{ix+1} is directly linked to d_{ix} $(1 \le x \le m-1)$ from right side,
THEN combine them as a new structure fragment F_i [t_i, d_{i1}, ..., d_{im}] with type (f).

All item fields which can not be classified using the above rules, are defined as the type (k).

4) Construction of blocks

Some linked structure fragments are meaningful. The following rules transform blocks of types (c) and (h) into new blocks.

(h-1) IF structure fragment F_i exists,
THEN F_i constructs block B_i [F_i].

(h-2) IF F_i belongs to type (c) or (h),
AND F_{j1} in B_j [F_{j1}, ...] belongs to type (c) or (h),
AND F_i is directly linked to F_{j1} at lower side,

THEN B_j [F_{j1}, ...] becomes B_j [F_i, F_{j1}, ...] after combining F_i.

(h-3) IF F_i belongs to type (c) or (h),
AND F_{jn} in B_j [..., F_{jn}] belongs to type (c) or (h),
AND F_i is directly linked to F_{jn} at lower side,
THEN B_j [..., F_{jn}] becomes B_j [..., F_{jn}, F_i] after combining F_i.

(h-4) IF F_i belongs to type (d) or (i),
AND F_{j1} in B_j [F_{j1}, ...] belongs to type (d) or (i),
AND F_i is directly linked to F_{j1} at right side,
THEN B_j [F_{j1}, ...] becomes B_j [F_i, F_{j1}, ...] after combining F_i.

(h-5) IF F_i belongs to type (d) or (i),
AND F_{jn} in B_j [..., F_{jn}] belongs to type (d) or (i),
AND F_i is directly linked to F_{jn} at right side,
THEN B_j [..., F_{jn}] becomes B_j [..., F_{jn}, F_i] after combining F_i.

(h-6) IF F_i belongs to type (g),
AND F_{j1} in B_j [F_{j1}, ...] belongs to type (b),
AND F_i is directly linked to F_{j1} at right side,
THEN B_j [F_{j1}, ...] becomes B_j [F_i, F_{j1}, ...] after combining F_i.

(h-7) IF F_i belongs to type (b),
AND F_{jn} in B_j [..., F_{jn}] belongs to type (g),
AND F_i is directly linked to F_{jn} at right side,
THEN B_j [..., F_{jn}] becomes B_j [..., F_{jn}, F_i] after combining F_i.

D. Knowledge Composition

The constructive relationships between item fields, structure fragments and blocks have been already recognized in the structure extraction phase. We show the hierarchical layer among their document components in Fig. 17. They must be finally represented as a structure description tree so as to be applicable to structure recognition of table-form document images. We describe the procedure for constructing structure description trees briefly. The nodes in a global structure tree correspond to the blocks, while edges link to the other blocks. The first step is to find out the most left-upper corner of block. If it has vertically repeated property, the node "D" is assigned. If it has horizontally repeated property, the node "R" is done. Otherwise, the node "S" is done. Next, the right-side block for the current one is determined as the right child node, and also the lower-side block does so as the left child node. This procedure is repeatedly performed until all blocks are built as nodes.

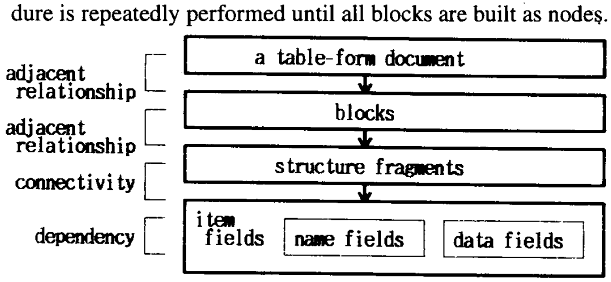


Fig. 17. Structure of table-form document.

The detail structure of every block is expressed by a local structure tree. The nodes in the local structure tree correspond to item fields or structure fragments, and the edges represent the connective relationships among item fields and/or structure fragments. When the most left-upper item field or structure fragment has horizontally dependent relationship with the remaining parts in the block, the node "h" is assigned. Namely, the item field or structure fragment at the left side corresponds to a left child node of the parent node "h", and that at the right side corresponds to a right child node. Similarly, the node "v" is used to link item fields or structure fragments with vertical dependence relationship. This procedure is executed repeatedly until all item fields are replaced by terminal nodes. Especially, as for repeating structure fragments, only item fields located at the first row or column are recorded as the local structure tree. The indicator and number of repetition are assigned to the corresponding node in the global structure tree.

VI. Recognition of Layout Structure

The recognition process of layout structure is illustrated in Fig. 18. This process is composed of the structure analysis and structure recognition phases, in addition to the document class recognition phase [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. In the structure analysis phase, the main functions are the detection of vertical and horizontal line segments, and the extraction of the left-upper corners of rectangular item fields which are surrounded by vertical and horizontal line segments. While, in the structure recognition phase the layout structure is recognized by interpreting the extracted left-upper corners with the global structure tree and local structure trees.

A. Structure Analysis

The structure analysis phase includes three procedures: 1) binarization of document images; 2) extraction of vertical and horizontal line segments; and 3) detection of left-upper corners. These procedures manage document images directly.

Binarization of document images. The procedure transforms table-form document images from multi-values to binary values, using a heuristic threshold value.

Extraction of vertical and horizontal line segments. This procedure extracts vertical and horizontal line segments from binarized document images. Traditionally, the thinning techniques have been utilized for such a processing objective. However, as in our approach it is not a direct objective to extract line segments correctly, we adopt a method to be able to identify the cross-points among vertical and horizontal line segments effectively because item fields are replaced by left-upper corners. Namely, we used the edge extraction filter, as shown in Fig. 19, in order to extract line segments. In Fig. 20, the distinction between our filtering method and traditional thinning method is illustrated drastically. It is difficult to extract cross-shape line segments correctly by thinning techniques. On the other hand, our method can select line segments with strengthened left-upper corners. In document images which are applied by the edge extraction filter, the pixel for "A" becomes black when the following two conditions are satisfied.

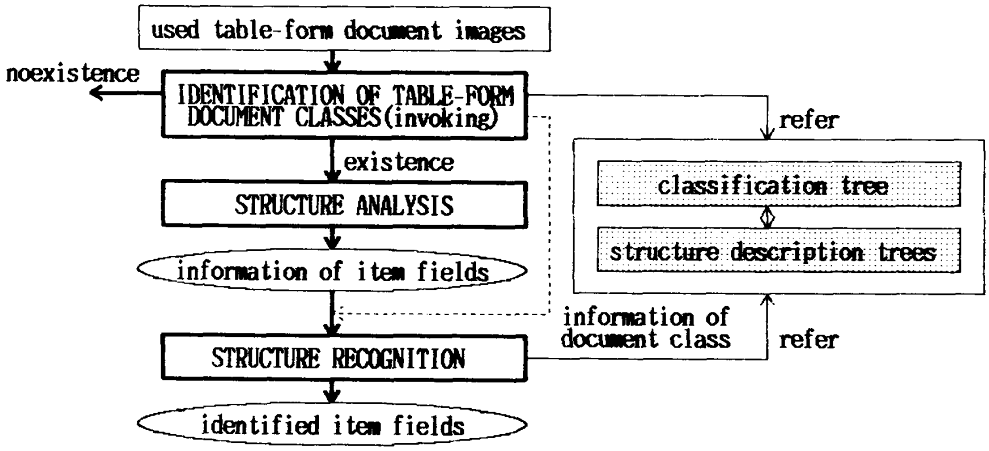


Fig. 18. Layout structure recognition.

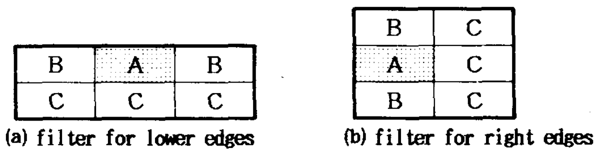


Fig. 19. Filters for edge extraction.

[Condition of edge extraction]

1) All of three pixels for “A” and “B” are black;
2) Three pixels for “C” are at least one or more white ones.

This filtering procedure is scanned for each pixel from left-top to right-bottom line by line.

Detection of left-upper corners. This detection procedure applies the corner detection filter to edge-extracted document images. The corner detection filters are shown in Fig. 21. In document images which are checked up by this corner detection filter, the pixel for “A” becomes black when the following conditions are satisfied.

[Condition of corner detection]

1) In the first filter, two pixels of “B” are both black, and also one of three pixels of “C” or “D” is at least black, respectively;
2) In the second filter,
- “A” and “E” (4 pixels),
- Maximum black pixels of “F” or “G” (3 pixels),
- Maximum black pixels of “H,” “I,” or “J” (10 pixels),
- Maximum black pixels of “K,” “L,” or “M” (10 pixels).
All black pixels in the above cases (27 pixels) exceed over a certain threshold value (this threshold value is assigned heuristically).

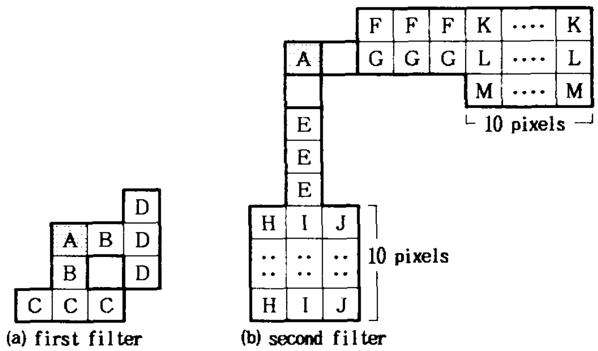


Fig. 21. Corner detection filters.

Here, the reason, which the shapes of horizontal and vertical directions in the second filter are different for “A,” depends on the recovery strategy of scanning errors which may be generated along the horizontal direction when the image-scanner moves vertically. The following algorithm determines the left-upper corners of every item field;

[Detection algorithm of left-upper corners]

1) $i := 0$.
2) Check up coordinate values $(i, x - i)$. $(0 \le x \le i)$
- `If` the cross-points are corners, `goto` 4).
- `Otherwise`, goto 3).
3) $i := i + 1$; `goto` 2).
4) Select the following cross-point as left-upper corner,
- the point near to the center `if` many cross-points are distinguished;
- the point itself `if` only one cross-point is extracted.

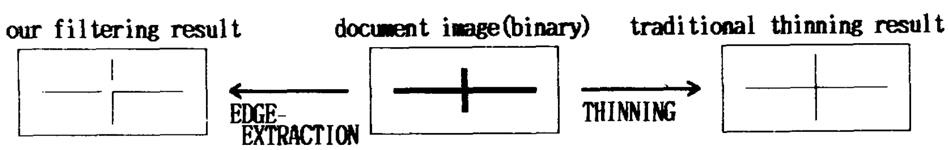


Fig. 20. Effects of edge extraction.

B. Structure Recognition

The structure recognition phase interprets left-upper corners, which correspond to individual item fields, with the structure description tree, and then identifies individual item fields distinguishedly. In this case, the detected left-upper corners do not only correspond to the cross-points between vertical and horizontal line segments correctly, but also may be picked up erroneously from parts of symbols, characters and so on. Since our structure description tree represents logically the neighboring/connective relationships among blocks/item fields, this structure recognition phase can interpret only the left-upper corners. Moreover, reusing detected coordinate values this phase can interpret the geometric relationships among left-upper corners in addition to the neighboring/connective relationship. Namely, erroneously extracted points and undetected left-upper corners are verified on the basis of successfully interpreted points or the other interpretative points are inferred smartly.

The structure recognition phase is composed of two procedures: structure interpretation procedure, which distinguishes meaningful blocks or item fields individually; and item extraction procedure, which extracts the identified item fields.

Structure interpretation. In the recognition phase of table-form document images, the procedure scans first the document images of left-upper corners from the most left-uppest position to the most right-lowest position line by line, and then corresponds left-upper corners with nodes in the structure description tree. The procedure searches another left-upper corner by looking upon the currently detected left-upper corner as the root node in the corresponding local structure tree. The global structure tree distinguishes individual blocks, while the local structure trees identify each item field. In the global structure tree, if the node is the type "R" then the node indicates that the same block is repeated from left to right, and also if the node is the type "D" then the node points out that the same block is repeated from upper to lower. Therefore, in the interpretation of the local structure tree the second or more identification steps of item fields can be inferred on the basis of the firstly detected coordinate values.

If all the left-upper corners can match with the nodes of the global structure tree under such interpretation, the matching procedure transfers the search pointer to the next node of the local structure tree and then interprets the upper-left corner with the local structure tree, linked by the matched node. On the other hand, if the matching failed the procedure suspends temporarily the interpretation and proceeds the next node of the global structure tree. In the global structure tree of binary tree, if the left edge could not be interpreted successfully the right edge is selected alternately. And, if the right edge could not, the left edge is altered. Thus, it is effectual because the matching ranges of left-upper corners for temporarily suspended nodes become narrow. Of course, in the repeating structure the ambiguous correspondence processings are also solvable since the second or more block identifications can be controlled by the result of the first interpretation. The final result is the coordinate values for individual item fields.

Extraction of item fields. The extraction procedure of individual item fields is performed on the basis of coordinate values of item fields, which were pointed out in the structure interpretation step.

VII. Experiments

We implemented a prototype system in order to establish our method for recognizing multi-classes of table-form document images with respect to document class recognition, layout knowledge acquisition and layout structure recognition. Experiments are as follows:

- Classification of document classes: construction of classification tree;
- Acquisition of layout knowledge: composition of structure description tree;
- Recognition of layout structure: identification of individual item fields.

In these experiments, table-form document images were scanned by 240 dpi. Additionally, our prototype system is implemented on Sun-4/SPARCstation+1 by the programming language C.

Construction of classification tree. In this experiment, six kinds of table-form documents in Fig. 1 and Fig. 8 were used. The classification tree has already been shown in Fig. 7. In Fig. 7, we can observe that some document classes are organized as non-terminal nodes, because our classification tree represents a transitive relationship about block division. Additionally, Tab. 1 and Tab. 2 in Fig. 1 are registered to different nodes respectively because our classification tree is constructed on the basis of physical characteristics of line segments. Of course, they must be the same structure description tree.

Composition of structure description tree. In this case, we also used six kinds of table-form documents in Fig. 1 and Fig. 8. All structure description trees were composed successfully. The processing times were 30 or less seconds as arranged in Table I. Also, many parts of the times are exhausted in image-processing-based structure analysis phase. The processing times are very shorter than the pointing method of coordinate values in OCR or the external definition method through the layout description language.

Recognition of item fields. We have already reported that our structure description tree can interpret table-form document images smartly [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Here, we show that the structure description tree which was generated from one table-form document image can be applied interpretatively to another table-form document image successfully, which is different in the geometric structure but is the same in the logical structure. Namely, the objective in our experiment is to make it clear that the structure description tree, generated from Tab. 1 in Fig. 1, can interpret Tab. 2 in Fig. 1 without any troubles. Fig. 22 is a used table-form document on which some characters was written for Tab. 2. And we generated the structure description tree from Tab. 1, as illustrated in Fig. 6. Finally, the item fields, which were identified by the structure description tree (in Fig. 6) from the table-form document image (in Fig. 22), are shown in Fig. 23. All item fields are identified and extracted correctly. In Fig. 23, the first column is a set of extracted data fields, the second is a set of the corresponding name fields, and the third is a set of other name fields in 2-dimensional array-form type of structure fragments.

TABLE I
PROCESSING TIMES IN LAYOUT KNOWLEDGE ACQUISITION

Procedures	Times	
Structure decomposition	25 sec.	28 sec.
Structure extraction	2 sec.	
Knowledge composition	1 sec.	

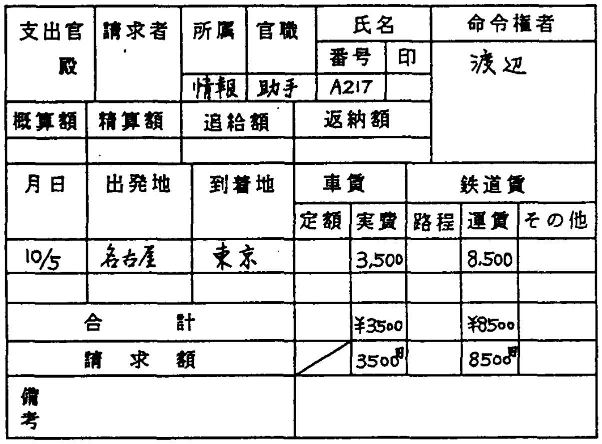


Fig. 22. A used table-form document.

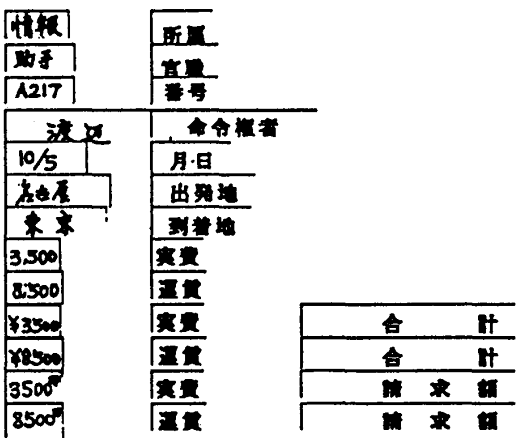


Fig. 23. Identified and extracted item fields.

VIII. CONCLUSION

In this paper, we addressed a recognition paradigm of layout structures for multi-kinds of table-form document images. In particular, we proposed the recognition method of document classes, acquisition method of layout knowledge and recognition method of layout structures. Until today, the researches about the document class recognition and automatic document knowledge acquisition have never been reported. Although Y.Nakano et al. attached to the recognition problem for multi-kinds of table-form document images as an important subject for making the document image understanding system available in practical use, they could not propose any successful method/approach [7]. This is because their knowledge was based on only the physical coordinate values. In our approach, this recognition problem was solved completely with the classification tree based on the physical characteristics of table-form documents and the structure description tree based on the logical characteristics of table-form documents.

Our method based on the assumption that table-form documents are composed systematically of rectangular blocks is not always applicable to table-form documents in which some corners are not roundly or every item field is not surrounded closely by horizontal or vertical line segments. This is because our image-processing as the basic bottom-up way is dependent on the left-upper corners of cross-points between horizontal and vertical line segments or derived from the straight line segments. However, our method can be applied to all table-form documents without any troubles if we can assign pseudo left-upper corners to cross-point locations to be generated by their extensive processings of horizontal and vertical line segments, as one of the preprocessings. We have already discussed this issue in another paper [18].

ACKNOWLEDGMENTS

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

REFERENCES

[1] T. Watanabe, Q. Luo, and N. Sugie, "A cooperative document understanding method among multiple recognition procedures," *Proc. 11th ICPR*, pp. 689-692, 1992.

[2] T. Watanabe, Q. Luo, Y. Yoshida, and Y. Inagaki, "A stepwise recognition method of library cataloging cards on the basis of various kinds of knowledge," *Proc. 10th IPCCC*, pp. 821-827, 1990.

[3] T. Watanabe, Q. Luo, and N. Sugie, "Toward a practical document understanding of table-form documents: Its framework and knowledge representation," *Proc. 2nd ICDAR*, pp. 510-515, 1993.

[4] D. Niyogi and S. Srihari, "A rule-based system for document understanding," *Proc. AAAI 86*, pp. 789-793.

[5] F. Esposito, D. Malerba, G. Semeraro, E. Annese, and G. Scafaro, "An experimental page layout recognition system for office document automatic classification: An integrated approach for inductive generalization," *Proc. 10th ICPR*, pp. 557-562, 1990.

[6] J.L. Fisher, S.C. Hinds and D.P. D'amato, "A rule-based system for document image segmentation," *Proc. 10th ICPR*, pp. 567-572, 1990.

[7] Y. Nakano, H. Fujisawa, O. Kunusaki, K. Okada, and T. Hanano, "Understanding of tabular form documents cooperating with character recognition," *IECE Trans.*, vol. J69-D, no. 3, pp. 400-409, 1986 (in Japanese).

[8] J. Higashino, H. Fujisawa, Y. Nakano, and M. Ejiri, "A knowledge-based segmentation method for document understanding," *Proc. 8th ICPR*, pp. 745-748, 1986.

[9] K. Kise, K. Momota, M. Yanaka, J. Sugiyama, N. Babaguchi, and Y. Tezuka, "Model based understanding of document images," *Proc. MVA 90*, pp. 471-474.

[10] A. Dengel and G. Barth, "High level document analysis guided by geometric aspects," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 2, no. 4, pp. 641-655, 1988.

[11] 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. MVA 90*, pp. 461-464.

[12] Q. Luo, T. Watanabe, and N. Sugie, "A structure recognition method for Japanese newspapers," *Proc. 1st Symp. Document Analysis and Information Retrieval*, pp. 217-234, 1992.

[13] T. Watanabe, H. Naruse, Q. Luo, and N. Sugie, "Structure analysis of table-form documents on the basis of the recognition of vertical and horizontal line segments," *Proc. 1st ICDAR*, pp. 638-646, 1991.

[14] T. Watanabe, Q. Luo, and T. Fukumura, "A framework of layout recognition of document understanding," *Proc. 1st Symp. Document Analysis and Information Retrieval*, pp. 77-95, 1992.

[15] T. Watanabe, Q. Luo, and N. Sugie, "Structure recognition methods for various types of documents," *Int' J. Machine Vision and Applications*, 6, pp. 163-176, 1993.

[16] Q. Luo, T. Watanabe, and N. Sugie, "Structure recognition of table-form documents on the basis of the automatic acquisition of layout knowledge," *Proc. MVA 92*, pp. 79-82.

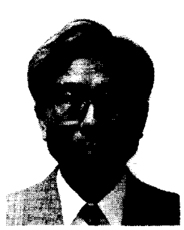
[17] H. Kojima and T. Akiyama, "Table recognition for automated document entry system," *High-Speed Inspection Architectures, Barcoding, and Character Recognition*, SPIE, vol. 1384, pp. 285-292, 1990.

[18] H. Naruse, T. Watanabe, Q. Luo, and N. Sugie, "A structure recognition method of table-form documents on the basis of the information of line segments," *IEICE Trans.*, vol. J75-D-II, no.8, pp. 1372-1385, 1992 (in Japanese).

Toyohide Watanabe received the BS, ME, and PhD degrees from Kyoto University in 1972, 1974, and 1983, respectively. In 1987, he became an associate professor in the Information Engineering Department at Nagoya University. He is currently a full professor.

Dr. Watanabe's research interests include the semantic data model, multi-media database, object-oriented paradigm, parallel and distributed processing, and document image understanding. He is a member of the Information Processing Society of Japan, Institute of Electronics, Information, and Communication Engineers of Japan, Japan Society for Software Science, Japan Society for Artificial Intelligence, ACM, AAAI, and IEEE Computer Society.

Qin Luo received the BE degree from Shanghai Institute of Technology in China in 1984, and the ME and PhD degrees from Nagoya University in Japan in 1990 and 1993, respectively. Currently, she is a research associate in the Department of Electronic Information Engineering at Toyama University. Her field of interest is document image understanding. She is a member of the Information Processing Society of Japan.



Noboru Sugie received the BE degree from Nagoya University and joined the Electro-Technical Laboratories in 1957. From 1979 to 1994, he was a full professor in the School of Engineering at Nagoya University. Currently, Sugie is a full professor in the School of Engineering at Meijyo University in Nagoya. His research activities are in the areas of medical engineering, computer vision, and natural language processing. He is a member of more than 10 academic institutions.