January 18, 1999
RT0297
Human-Computer Interaction 17 pages

Research Report

A Framework for Forms Processing by Using Enhanced-Line-Shared-Adjacent Format

Y. Hirayama

IBM Research, Tokyo Research Laboratory
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Title:

A Framework for Forms Processing by Using Enhanced-Line-Shared-Adjacent format

Abstract

The objective of this paper is to introduce a novel framework for forms processing that provides seamless processing of two different kinds of formats, which are physical formats whose fields have rigidly defined positions and sizes and topological formats in which variations in the positions and sizes of fields are acceptable as long the topological relations between pairs of fields are preserved. An line-shared-adjacent (LSA) cell relation and an LSA format are introduced to define topological formats and then they are enhanced to describe physical information as an enhanced LSA (e-LSA) and an e-LSA format. The e-LSA format has good flexibility to define not only physical and topological formats but also hybrid formats, on which our framework is based. The format has characteristics of both physical and topological formats and enables the framework to handle the two kinds of information seamlessly. The framework consists of four modules: a format generator, a format converter, a format class manager and a form processor and all processes for field detection that our research focuses on are performed by them. In this paper, their collaborative work are illustrated with some example, which supports the effectiveness of our framework.

Keywords: forms processing, physical format, topological format, hybrid format, line-shared-adjacent, LSA, LSA format, LO format, layout analysis, OCR

1 Introduction

Exchange of information written on printed forms, such as tax forms, plays a very important role in private and government business. Even now, information is very often processed on paper rather than electronically. As a result, most offices have to process numerous forms.

These forms have many kinds of formats, which we have classified into two main types. One is the "physical format," whose fields have rigidly determined positions and sizes, and the other is the "non-physical format," whose fields have a variety of positions and sizes. The non-physical format can further be divided into two types: the "topologically specified format" and the "content-specified format." (We call these the topological and content formats.) In the content format, variations in the positions and sizes of fields are acceptable as long the contents are preserved. In the topological format, on the other hand, variations in the positions and sizes of fields are acceptable as long the topological relations between pairs of fields are preserved.

In conventional forms processing in government or company offices, numerous forms with physical format have been used for a long time. In such kind of forms processing, a user gets a prepared blank form from the organization and fills it with a pen or a typewriter. As DTP (DeskTop Publishing) becomes popular, a user can easily use a computer for making not only fill-in data but also a filled-in form itself. Such kind of computerization causes a variety of physical format, that is, filled-in forms that many users have printed with their computer are slightly different from each other, although they should be same except contents that are filled in.

As forms such casual users print are slightly different from each other, it is difficult to apply conventional method simply to processing of the forms. On other hands, as topological relations among cells on a form however are preserved, topological format has an advantage on processing such kind of forms. As computerized forms become more and more popular, it is needed for forms processing system to have capability of processing of not only physical but also non-physical form.

Many attempts to process conventional forms[1] [2] and, on the other hand, at least two attempts to process topological formats [3] [4] are reported. But prior research on forms or table documents [5] [6] don't have enough flexibility to describe and process hybrid formats.

Our research is on physical and topological formats in which all the fields are rectangular and surrounded by vertical and horizontal line segments. The main objective of our research is to propose a framework for forms processing that provides seamless process of both physical and topological formats. The forms processing described in this paper focuses on a field detection on form images and don't include the succeeding processes like a character recognition or other post-processings.

The following items are introduced in our research to support the main objective: (1) Enhance a line-shared-adjacent (LSA) and a line-oriented (LO) format as an e-LSA and an e-LO format respectively. (2) Establish a method for defining a hybrid format with the e-LSA and e-LO format. (3) Define "format class", construct a hierarchical class structure and classify formats into them. (4) Establish a method for processing actual forms with the format.

2 An Overview of Framework

The framework is based on LSA and LO formats whose detail is described in [7]. These formats were originally introduced to deal with topological information, but in this paper, they are enhanced to deal with not only topological but also physical information.

2.1 LSA and LO format

Figure 1 shows a definition of line-shared-adjacent cell relations. An LSA relation is defined as follows: "If two cells are located on the same side of a line, touch the line, and touch each other, the relation between the cells is LSA." In this case, we say "Line 1 is shared by the cells but line 2 is not shared." As shown in the definition, an LSA relation contains no information on the global positions and sizes of cells. When all LSAs in a form is extracted, a set of them is called an LSA format.

While an LSA format based on cells, an LO format that can be equivalently converted from the LSA format focuses on relations between line segments and therefore mainly consists of a set of information on their connection. In the case of Figure 1, we can extract information on line connection as follows: "An edge of line 2 is located on line 1." A set of all such information in a format is called an LO format.

2.2 Enhancement of Format

The main objective of enhancement is to introduce physical information to LSA and LO formats. Physical information can be classified into two categories, one is that on absolute position and the other is on size, which are defined as shown in Figure 2. LSA and LO formats whose elements have capability of having physical information as attributes are called enhanced-LSA (e-LSA) and enhanced-LO (e-LO) formats respectively.

We can define an e-LSA format as a graph as follows: Let $G = (C, L)$ be a graph. C is a set of nodes that correspond to cells, expressed as $C = \{cell_1, cell_2, \ldots, cell_n\}$, and L is a set of edges that correspond to LSA relations, expressed as $L = \{lsa_1$, lsa_2, ..., $lsa_m\}$.

Lsa consists of two cells, a shared line and a shared position, such that $lsa = (cell_p, cell_q, line_r, position)$, where $position \in \{T(top)$, $B(bottom)$, $L(left)$, $R(right)$,

$TB(topandbottom)$, $LR(leftandright)\}$. *Cell* and *line* can have attributes of their position and size, such that $cell_i = (PX_i, PY_i, W_i, H_i)$, $line_j = (PX_j, PY_j, L_j)$, where (PX_i, PY_i) is a position of left-upper point of "*cell*$_i$", (PX_j, PY_j) is that of left or top edge of "*line*$_j$", W_i,H_i are width and height of "*cell*$_i$" respectively, and L_j is length of "*line*$_j$". *Cell* and *line* don't have to have all of the attributes, in other words, each element of cell and line can be "ϕ" that means empty. If a cell has no physical information, for example, it is expressed as (ϕ, ϕ, ϕ, ϕ).

An LO format, which is equivalently converted from an LSA format as described in [7], can have physical information of cells and line segments that can be expressed just the same as those in an e-LSA format.

We call the enhanced format a hybrid format, which can be classified into the following five types: In the following definition, "some" means "not zero and not all".

Level 0 No elements have their physical information.

Level 1 Although no elements have information on their positions, some elements have that on their sizes.

Level 2 In addition to level 1, some elements have information on their positions.

Level 3 Although no elements have information on their positions, all elements have that on their sizes perfectly.

Level 4 In addition to level 3, all elements have information on their positions perfectly.

Level 0, 3 and 4 are special cases of hybrid formats, that is, level 0 corresponds to topological, and level 3 and 4 to conventional physical formats.

2.3 Hybrid Format

A hybrid format provides characteristics of both physical and topological formats, such that some elements have physical information and others not.

We now introduce physical information ratio (PIR), which consists of four elements: PIR=(C_s, C_p, L_s, L_P), which are PIR value related with cell size, cell position, line size, and line position respectively. These values are defined as follows.

A cell can have two pieces of size information, width and height. A format that has n cells therefore can have $2 \times n$ pieces of size information. When the format has $CELL(S)$ pieces of cell size information, C_s is defined:

$$C_s = \frac{CELL(S)}{2 \times n} \times 100\ (\%)$$

When the format has $CELL(P)$ pieces of position information, C_p is defined:

$$C_p = \frac{CELL(P)}{2 \times n} \times 100\ (\%)$$

We can also define L_s and L_p similarly. A line segment can have only single size information, length. A format that has m line segments can have m pieces of size information. When the format has $LINE(S)$ pieces of line size information, L_s is defined:

$$L_s = \frac{LINE(S)}{m} \times 100\ (\%)$$

When the format has $LINE(P)$ pieces of position information, L_p is defined:

$$L_p = \frac{LINE(P)}{m} \times 100\ (\%)$$

While if PIR of a format equals to $(0, 0, 0, 0)$, it is a topological format, in a physical format, values of both C_s and L_s are 100 and C_p and L_p are 0 or 100 that depend on type of the physical format.

2.4 Basic Structure of Framework

Figure 3 shows an overview of the framework. The framework basically consists of four modules, that is, a format generator (FG), a form processor (FP), a form converter (FC) and a format class manager (FCM). The FG and the FP play roles as interface to actual forms, the FC makes conversion between e-LSA and e-LO formats, and the FCM manages a hierarchical format class structure.

3 Modules in Framework

3.1 Format Generator (FG)

The FG is an input interface of the framework. It scans a document, obtains an image, detects line segments and generates an e-LO format. An e-LO format generated directly from an actual format is a level 4 format.

3.2 Format Converter (FC)

The FC is a bi-directional bridge module between interface modules and the FCM. It equivalently converts an e-LSA to an e-LO format and vice versa. Physical information, in other words level of the format, is preserved during the conversion.

3.3 Format Class Manager (FCM)

The main function of the FCM is to construct and manage a hierarchical format class structure in which a format corresponds to a class and a form to an instance.

The class structure is constructed by using an unification operator, which generates an upper class format from two lower class formats. The unification operator has also been enhanced in order to deal with physical information.

We define the enhanced unification operator "$\odot$", which is enhanced from "$\otimes$" defined for topological format, as follows:

Let $G_1 = (C, L_1)$ and $G_2 = (C, L_2)$ be graphs that correspond to two LSA formats.

$$G_3 = G_1 \odot G_2$$

where $G_3 = (C, L_3)$,$L_3 = L_1 \cap L_2$, and the greatest common physical information G_1 and G_2 is set to G_3.

Figure 4 shows an example of class structure construction. The structure is constructed from bottom to up by using the operator.

A and B are formats that are generated from actual forms F_A and F_B respectively. C_A and C_B are format classes that correspond to A and B, in other words, form F_A and F_B are instances of format class C_A and C_B respectively. By applying the operator to A and B, a new format C can be obtained. If C is a connected graph, a new class C_C that corresponds to C is generated as an upper class of C_A and C_B. This relation among three formats and classes compose the minimal unit of the hierarchical structure.

A format whose level is less than 4 is called a "virtual format", while a level 4 format is called a "real format". In this example, if level of C' is not 4, C is a virtual format, while A and B are real formats.

The operator is also used for classification in the case that a new form F_D should be included in the structure. A format D is generated by FG from it. As an example, Figure 5 shows four typical cases of classification.

case (A) When D is just the same as B, F_D is another instance of the class C_B. This result is obtained when a format generated by operation between D and B is just the same as B.

case (B) When a format generated by operation between D and A is just the same as C, the format class D is located as another lower class of C.

case (C) When a format generated by operation between D and A is different from C but is same regardless of physical information, a new format "E", which is generated from D and C, is located at the top of the hierarchy.

case (D) When a format generated by operation between D and A is not a connected format, D and F_D should belong to different class structure from that of C.

The hierarchical structure have an important feature, that is, all instances that are descendants of a upper class format can be processed with it in the FP module.

3.4 Form Processor

This module processes an actual form image by using an e-LO format. It extracts line segments from the image, matches them with the format and detects fields on the image., It determines the format by inquiring of the FCM, which returns an upper class format of all instances. Referring to 5-(C), forms F_A,F_B and F_C can be processed with the virtual format E.

In matching process, the module matches detected line segments and line information in a format[7]. In addition to topological information, physical one in a hybrid format is taken into account to the cost. Matching process by utilizing a hybrid format therefore realizes more robust process than that by only topological format.

4 Processing Cycle in Framework

In this section, processing cycles from generating a format to processing actual forms are described by referring four sample images that are application forms for something shown in Figure 6.

4.1 Case 1: Form (A)

At first, The FG generates an e-LO format from some images of actual forms of (A), which are converted to e-LSA formats whose level is 4 by the FC and sent to the FCM. Next, the FCM constructs a class structure that consists of one virtual format whose level may be 3 or 4 and some of real formats in lower classes. The level of the upper class format depends on actual form images on which absolute positions of elements in the images are approximately same and difference is tolerable or not. Finally, the FP can process actual forms with the single virtual format.

4.2 Case 2: Form (A) and (B)

At first, The FG generates two e-LO formats from actual forms (A) and (B), which are converted to e-LSA formats and sent to the FCM respectively. Next, The FCM applies enhanced unification operator to the two e-LSA formats and obtains an upper class format shown in Figure 7-(A), in which small black rectangles indicates that these T-shape connection don't preserve physical information. If the upper class format is a connected format, the format are registered as an upper class format in the hierarchical class structure.

While the original two formats are level 4 format, the upper class format does not preserve physical information on height of cells in the right half of it and PIR value is $(75(=18/24), C_p, 100(=11/11), L_p)$, where C_p and L_p depends on the same factor as case 1. Figure 8-(A) shows the preserved information. Bi-directional arrows in (A)-1 indicate preserved size information of cells and bold lines in (A)-2 indicate preserved length information of lines. The format is level 1 or 2 format which provides capability to process all of actual forms of both (A) and (B).

4.3 Case 3: Form (A),(B),(C) and (D)

The FG generates four e-LO formats from four types of actual forms, which are converted to e-LSA formats by the FC and sent to the FCM respectively. Note that height of "D" is taller than others and only height of "Name". The FCM applies enhanced unification operator to the four e-LSA formats and obtains one most upper format, which is shown in Figure 7-(B), in which a small circle indicates that the relations around it are not LSA. We call intersections of this type "non-LSA points". This format is also a level 1 or 2 whose PIR value is $(16.7(=4/24), C_p, 26.7(=5/15), L_p)$, where C_p and L_p depends on the same factor as case 1. Preserved information in this case is shown in Figure 8-(B)-1 and (B)-2, which show less information is preserved than case 2.

This format provides capability to process all of actual forms of (A),(B),(C) and (D) and is furthermore an upper class of the format generated in case 2. Comparing the format in case 2 and this one, the former has more physical information (higher PIR value) and enables a system to process form (A) and (B) more correctly than the latter . But it has no capability to process form (C) and (D).

On the other hand, from an another view of processing range, although, that of the latter is wider than the former, because it provides capability to process all of forms. This is a tradeoff problem.

5 Summary

While all forms have the same format in conventional forms processing, each form generated with a user's DTP system may have original format that is different from each other. It is an urgent business to make a new forms processing system that can process such kind forms seamlessly because if a variety of the formats becomes wider, it is very difficult to handle all forms with a conventional forms processing system.

We already introduced LSA and LO formats in the previous paper [7], which could process only topological format. In this paper, we have enhanced them, introduced a hybrid format with the enhanced format, and proposed a new framework for forms processing that provides seamless process of both physical and topological formats.

We have illustrated some processing cycles of typical forms processing that seamlessly handles some forms whose formats are different from each other. These examples shows that this framework is more effective than conventional methods because hybrid formats have both flexibility of topological and robustness of physical formats.

We plan to enhance this framework to process spreadsheet style forms in which number of row or column may change. This enhancement will improve process of multi-entry forms that are generally used in business transactions.

References

[1] R.Casey, D.Ferguson, K.Mohiuddin, and E.Walach: "Intelligent Forms Processing System," Machine Vision and Applications, Vol.5, No.3, pp.143-155(1992).

[2] Hiroyuki Arai, Kazumi Odaka: "Form Processing Based on Background Region Analysis," Proc. of 4th International Conference on Document Analysis and Recognition(ICDAR'97), Vol.1, pp.164-168(1997).

[3] T.Watanabe, Q.Luo and J.A.Paster: "Extraction of data from preprinted forms," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.17, No.4, April, pp.432-445(1995).

[4] Y.Ishitani: "Model Matching Based on Association Graph for Form Image Understanding," Proc. of 3rd ICDAR, pp.287-292(1995).

[5] Juan F.Arias, Atul Chhabra, and Vishal Misra: "Efficient Interpretation of Tabular Documents," Proc. of 13th International Conference on Pattern Recognition(ICPR'96), Vol.3, pp.681-685(1996).

[6] Osamu Hori and David S.Doermann: "Robust Table-form Structure Analysis Based on Box-Driven Reasoning," Proc. of 3rd International Conference on Document Analysis and Recognition(ICDAR'95), pp.218-221(1995)

[7] Y.Hirayama: "Analyzing Form Images by Using Line-Shared-Adjacent Cell Relations," Proc. Of 13th International Conference on Pattern Recognition(ICPR'96), Vol.3, pp.768-772(1996).

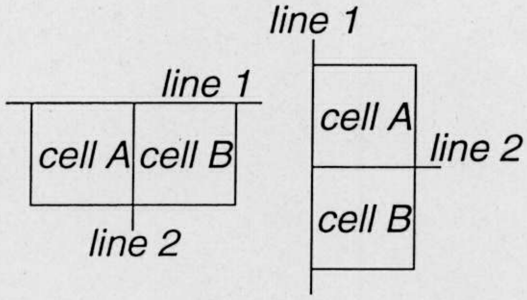

Figure 1: LSA relation

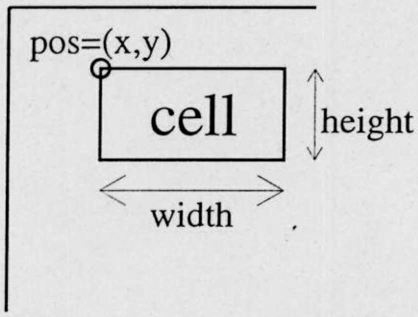

Figure 2: Position and size of a cell

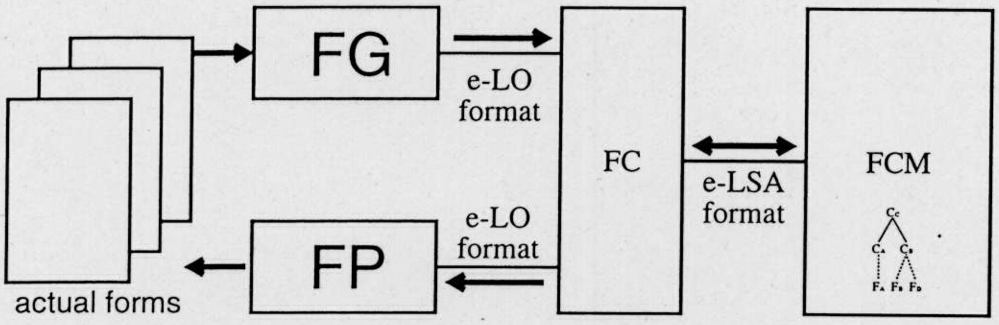

Figure 3: An framework

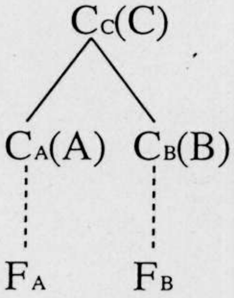

Figure 4: A class construction

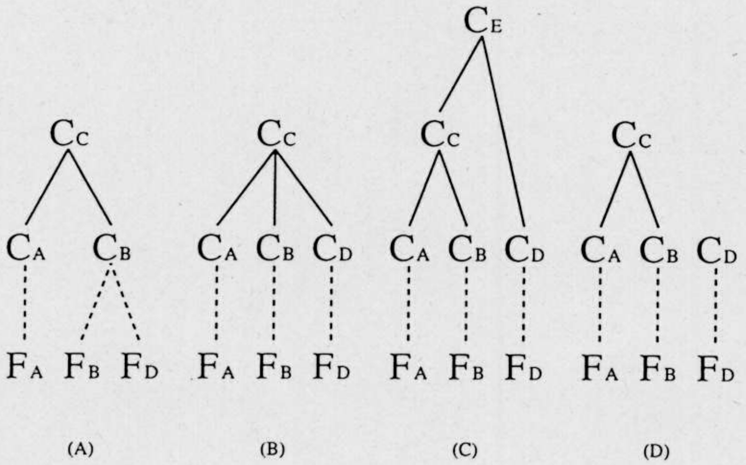

Figure 5: Examples of class construction

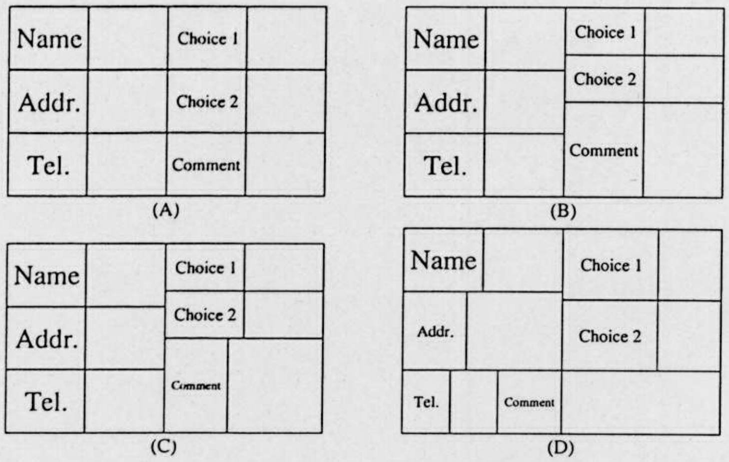

Figure 6: Sample four forms

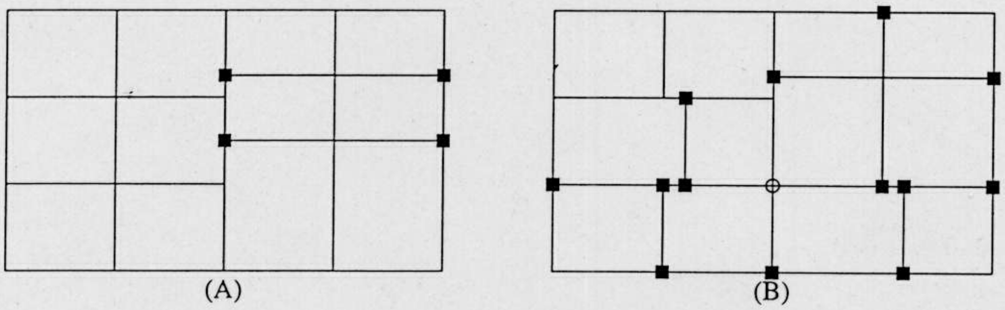

Figure 7: An upper class format

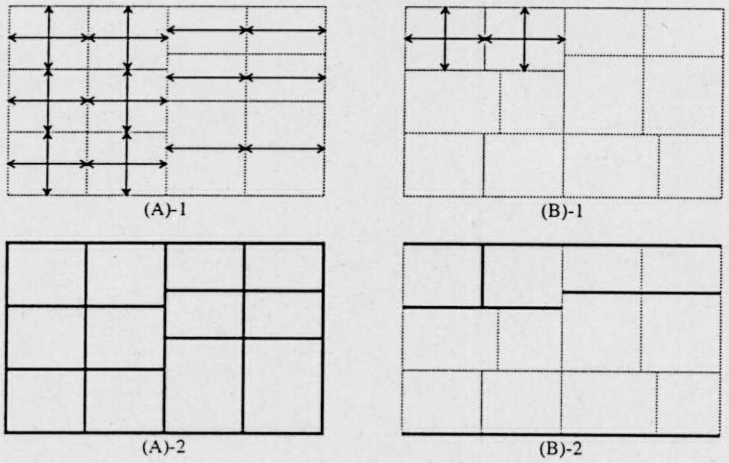

Figure 8: Preserved information