H-0267 (H0901-016) January 19, 2009
Electrical Engineering

IBM Research Report

A Generic Form Processing Approach for Large Variant Templates

Yaakov Navon, Ella Barkan, Boaz Ophir
IBM Research Division
Haifa Research Laboratory
Mt. Carmel 31905
Haifa, Israel

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

A Generic Form Processing Approach for Large Variant Templates

Yaakov Navon, Ella Barkan, Boaz Ophir
IBM Haifa Research Lab
Mount Carmel
Haifa 31905, ISRAEL
{navony, ella, boazo}@il.ibm.com

Abstract

In today's world, form processing systems must be able to recognize mutant forms that appear to be based on differing templates but are actually only a variation of the original. A single definition of a representative template actually covers large varieties of the same logical templates. We developed a method and system, similar to the human visual system, which differentiates between templates via features such as logos, dominant words, and geometrical shapes, while ignoring minor details and variations. When the system finds an appropriate template, it then decodes the content of the form. Our approach has been applied in several scenarios with encouraging results.

1. Introduction

With the prevalence and ease-of-use of today's printers, templates for filled-in forms can be printed anywhere, to the extent that there is no longer a need for "printing centers," where template replicas are normally of exact size, identical character fonts and types, matching topologies, and so on. This reality, alongside the existing complexities of identifying scanning artifacts, renders the template matching process even more difficult in today's systems.

The Internet facilitates downloading templates to be filled in (thereby becoming a form), enabling people to print them on any nearby printer [1]. Companies generally print their forms in-house. All these printed forms are subject to printer settings, user preferences, and other parameters. The result is that form processing systems must handle a rich set of forms based upon the one original *logical* template.

Template recognition processes based on local or global image matching methods are not sufficiently tolerant to handle the differences between the templates. For example, when comparing a form to its original template, where part of the text is printed using a different typeface (e.g., template text in regular type and form text in italic), we expect very poor matching results. However, when image matching is feasible, the use of drop out methods [2] is applicable. Template parts are eliminated from the filled form image making the readout of form fields robust. Moreover, it is sufficient to store only the filled in parts of the form; the form can be restored when necessary by mounting the filled in area on the appropriate template.

This paper presents a form handling approach that has a high tolerance for differences in template layouts. We recognize templates in a similar manner to the human visual system, where recognition is based on the content of dominant features while ignoring small differences of the same logical form. The recognition is based on logo similarity, text content, and geometrical shapes such as lines and boxes. The combination of different features eliminates the need for precise matching at the pixel level, while maintaining a high template recognition rate. Figure 1 depicts two forms both used for a health insurance claim. The templates may look the same to the human eye (at the global level), but actually differ greatly in the details (locally); thus, they cannot be handled at the image pixel level.

Various methods for locating information fields on forms have been proposed. Some of them are based on horizontal and vertical lines [3, 4]. Others are based on the location of the field with respect to instruction fields (keywords) [5]. Methods based on lexical information only, such as in [6], first require performing optical character recognition (OCR) on the entire image (the runtime issue) and expect to receive fixed templates. The open question is how generic and robust are those methods when handling forms of large variant templates?

In our approach, we syntactically define the template content such as text content and features, boxes, and fields. Some of the geometrical features are derived automatically from a representative template image. Our approach has been implemented successfully in several applications. The settings of the system required the definition of only a minimal number of representative templates; even though the templates had many printing variants.

2. Template variants form processing

In our approach, we use most of the logical features in a typical template (e.g., logo, keywords, instruction fields, geometrical shapes etc.) to match the right template to the form. The feature information is used to register and locate the data fields to be read. The registration is done with no modification to the input form image, since modifications such as scaling, translation, and rotation could degrade image quality, which would be reflected in the recognition results.

This section presents the main stages of our system:

1. Template definition
2. Template recognition
3. Data fields readout
4. Optimizing run of batch of forms

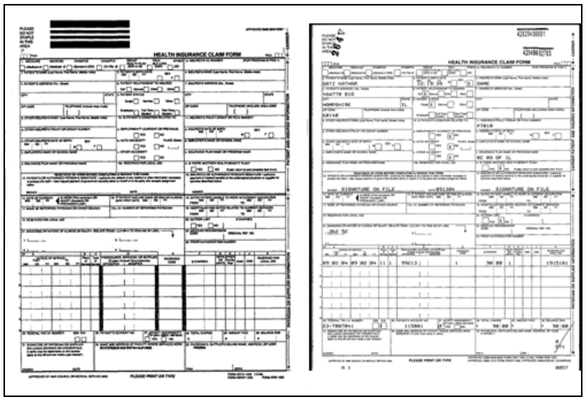


Figure 1: Templates for a claim form, which humans interpret as the same one

2.1. Template definition

The template definition is a logical description of template elements and of the fields to be read from the form. The description is based on human-like interpretation that enables us to handle templates with large variances. The definition includes logos, keywords, checkboxes, lines, and data fields.

Normally, one would expect a single logo per template, located in a fixed position. Keywords are defined by location and content description. This content includes the syntax of the expected string, exact format (spaces, case, font types, etc.) and relation rules between keywords.

Data fields are specified by the filled in area locations, expected syntax, and type of the field context, which may obey some rule constraints (e.g., dates) or items from dictionaries (e.g., names, addresses). It is possible to define geometrical and logical relations between all types of keywords and fields. The logical relations are, for example, "value of a field B exists if field A exists" or "field A is the sum of fields B and C".

Checkboxes are defined by location and shape (box or circle). Figure 2 exemplifies a keyword (red) and data fields (green), and their syntax definitions. Note that a keyword need not be a descriptive name of a data field to be read, e.g., "Policy No." is not defined as a keyword in this example.

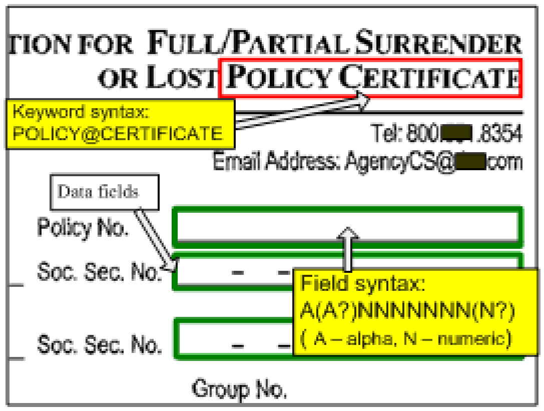


Figure 2: Definition of "Policy Certificate" keyword (red) and data fields (green)

2.2. Template recognition

The processing of an input form starts by associating the form with a template. The template recognition process is based on the recognition and matching of dominant features printed on the input form image and the template image. Template recognition includes the following steps:

1. Preprocessing of the form image
2. Creation of a precedence templates list
3. Recognition by logo
4. Recognition by keywords
5. Recognition by lines

Due to the nature of variant templates it is not expected that each of the above items is found with a high accuracy. Thus, the final decision of template recognition is a weighted function of the recognized items' accuracies.

2.2.1 Preprocessing of form image The input form image can be a binary, gray scale, or color image format. Color images are converted into gray scale and then binarized. Since most of the interesting information is in the form of text and graphics, we use dedicated binarization methods [7,8] to create proper binary images. Figure 3 exemplifies our binarization with the advantages of emphasized text and removal of the noisy background.

The entire form image is globally de-skewed only when large skew angles are encountered. In most cases there is no need to de-skew as forms are mostly rectified when they are scanned. For OCR purposes, local precise de-skewing is performed on field areas.

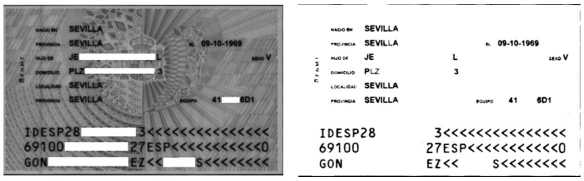


Figure 3: European ID card and its binary image (whitened areas preserve privacy)

2.2.2 Creation of a precedence templates list When we compare several templates to a form, the templates are first sorted according their rough distance from the form image. This distance can be the difference in the number of black runs between the form and the template or the difference in the number of lines. This helps ensure that the correct template is among the first to be checked.

2.2.3 Recognition of template logo Logos are very predominant symbols in templates. Generally, they are large and located at the top of the template. If a logo exists in a template, its location is set in the template definition. At the logo recognition stage, our system searches for a logo in the form image in the vicinity of the expected location.

The matching process takes into account different printing scales and the skew of the logo. To speed up the process, the selection of the search areas is in the proximity of large connected components. Since template recognition is not based only on logos, the matching precision does not have to be exact.

2.2.4 Recognition of template keywords The idea behind the keyword recognition method is to search the form for keywords that were defined for the template. The challenge is to overcome differences between the keywords' location and appearance in the template versus the form images.

Our system includes the following stages:

1. For the first keyword, set a large search window around the defined keyword location. For the next keywords, set precise search windows based on the locations of previously found keywords.
2. In the search window, apply OCR and related technologies (de-skew, binarization, and layout analysis).
3. On the OCR results, apply fuzzy matching and text alignment techniques [9] to locate the precise keyword location.

Figures 4 – 6 exemplify the process.

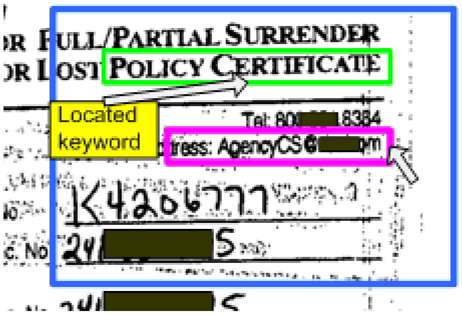


Figure 4: Locating first keyword; search (blue), defined (pink), and located (green)

Our system must recognize a minimal number of keywords; otherwise, the process stops and we proceed to the next template in the list.

The locations of the found keywords serve as a coarse registration between the form and template. This registration is used in the first phase of the line matching process, as described in the next step.

2.2.5 Recognition by lines Any approach to matching lines between the template and the form images needs to be sufficiently robust to handle both global differences in scale and location and local variations, such as excess/missing lines or scale. The variations can arise from the print and scan history of the image and from form mutations. As input, our system takes initial estimates of the global scale and shift from previous stages. These estimates, as well as local scale variations, are continuously estimated and compensated for throughout the matching process.

Our solution is based on the linear programming approach. It sorts the horizontal lines according to their vertical position and attempts to find a first pair to match. A pair of template-form lines is a possible match if they have a large overlap in the tangential direction and are close in the normal direction. For the initial pair, the tolerances for matching are quite large, allowing many potential matches. Using this first pair as a reference point, we try to match another pair at positions relative to the first pair, and so on, such that every matching pair found serves as a reference for the next pair. If a line is not matched, it is skipped. Thus, our approach does a vertical pass on the images, grading the complete matching by the sum of the lengths of the template lines matched. This grading puts an emphasis on correctly matched long lines. Out of several passes with different initial matches, our approach selects the one with the highest grade and compares its grade to the total line length in the template to determine whether the template matches the image.

The vertical lines are matched in a similar manner, reversing the roles of the coordinates.

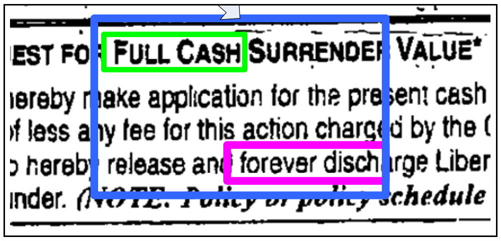
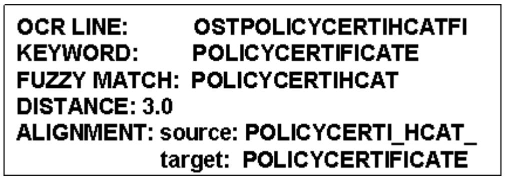

EST FO FULL CASH SURRENDE VALUE*
ıereby ı ake application for the pre ent cash
ıf less a y fee for this action charg d by the (
ɔ hereb release an forever disch rge Liber
ınder. (NOTE: Policy or policy schedule

Figure 5: Locating subsequent keywords: precise search (blue), defined (pink), and located (green)



```
OCR LINE:        OSTPOLICYCERTIHCATFI
KEYWORD:         POLICYCERTIFICATE
FUZZY MATCH:  POLICYCERTIHCAT
DISTANCE: 3.0
ALIGNMENT: source: POLICYCERTI_HCAT_
                      target:  POLICYCERTIFICATE
```

Figure 6: Fuzzy matching of OCR results to defined keyword "Policy Certificate" string

2.3. Read of data fields

The objective of a form processing system is to read data fields in a form. The location of the fields must be very precise since the actual reading is done by OCR, which is very sensitive to noise. For example, when a snippet image of a field includes parts of a frame, the system recognizes vertical bars as "1/l/I".

2.3.1 Data fields location Our approach combines keywords and geometric information to find approximate locations for fields and then uses layout techniques to refine the locations.

The calculation of a field location by keywords uses the relative distances of a field from the closest keyword. The approach derives the relative distances from the template definition and corrects them using the scale factors calculated at the registration stage. The refined location is found by layout analysis of the rough location area. Figure 7 depicts the rough location (red) and the finer location (blue) of the "Policy No." field.

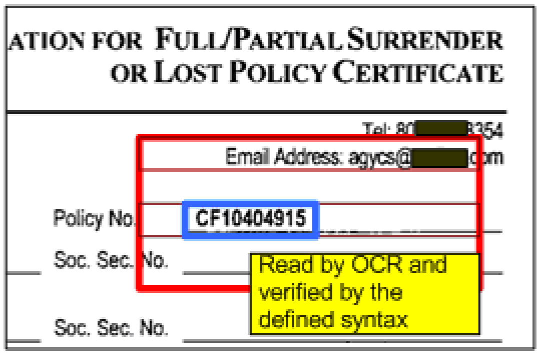

ATION FOR FULL/PARTIAL SURRENDER
OR LOST POLICY CERTIFICATE
Tel: 80[redacted]354
Email Address: agycs@[redacted]com
Policy No.
CF10404915
Soc. Sec. No.
Read by OCR and verified by the defined syntax
Soc. Sec. No.

Figure 7: Field location by "Policy Certificate" keyword

The results of the template recognition at the lines stage is a list of pairs of lines; a line in the form and a line in its corresponding template. The intersection of a horizontal line with a vertical line is a corner. Similar to the line pairs, the system creates a list of corner pairs; a corner in the form has its pair in the template. We define four corner orientations as depicted in bold in Figure 8 and three types of line intersections on the right.

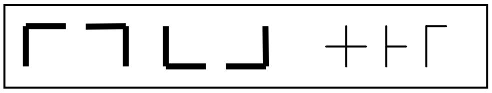
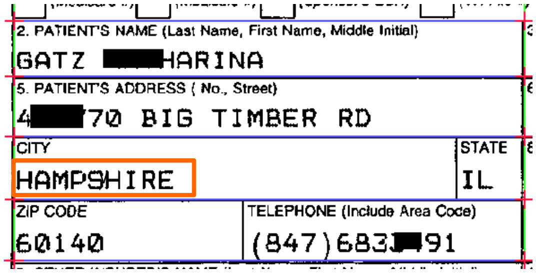


Figure 8: Corner orientations and line intersection types

A regular box is composed of four different corners. To locate a field in the form, the system first finds the corners in the template that are closest to its box, as defined in the definition of the field. The relevant pair corners in the form are derived from the corner pairs list. Our algorithm tries to build a valid box from those corners, according to their orientations, their relative locations, and the field box size. For example, a box cannot be composed of two corners of the same orientation. On the other hand, it can be composed, for example, from two opposite diagonal corners. Figure 9 depicts the location of some fields from corners. The exact field snippet to send to OCR (orange box in Figure 9) is found by layout analysis on the rough field location.

2.3.2 Data fields decoding The system does field decoding by applying OCR to an image snippet of only the field area. The OCR parameters are set according to the field specifications; i.e., numeral/alphabet, font type and size, syntax, etc. The OCR results are enhanced by post OCR logic process, where results are checked against relevant rules which field results must obey, similar to the process for keywords, above.

In some applications it is enough to know whether the field was filled or not. This is done by checking whether a significant portion of its area contains text.



2. PATIENT'S NAME (Last Name, First Name, Middle Initial)

GATZ [redacted]HARINA

5. PATIENT'S ADDRESS (No., Street)

4[redacted]70 BIG TIMBER RD

CITY HAMPSHIRE	STATE IL
ZIP CODE 60140	TELEPHONE (Include Area Code) (847)683[redacted]91

Figure 9: Location of field from corners (red). Exact location (orange) found by layout analysis

2.3.3 Checkbox decoding Checkboxes are an important feature in many forms but it is tricky to determine whether a box is ticked. Checkboxes are defined according to their location, size, and shape. Our algorithm needs to be able to locate the box even under extremely noisy conditions, typically caused by the tick itself. Our system employs a two-phased approach for square checkboxes. In the first pass, morphological operators detect the box corners. Equivalence classes are constructed from connected corners that are correctly aligned and distanced. If the first pass fails, a second pass employing the Hough transform (for squares) is applied. A box is deemed as ticked if a significant percentage of its interior is black. For circular checkboxes, only the Hough transform (for circles) method is used.

3. Experimental results

Two applications currently use the technology described above. The first is an automated form recognition and routing system used by a backoffice services company in the insurance industry. Forms are routed according to type, and—for some forms—the fields filled in by customers. The system is configured to handle several dozen different insurance and health forms. Many of the forms have between two and five recognized mutations, with one insurance form having as many as 50 mutations (one for every state in the U.S.). Recognition by keyword is between 95 to 97% accurate and by line is 93% accurate. Failures are mostly caused by severely degraded images, such as forms that have been faxed several times or printed at half their original size; this makes OCR and line extraction very difficult. Correct recognition of filled/empty fields is around 98%, with misses typically caused by severe degradation of checkbox images and (rarely) by misregistration of text fields.

The second application for this technology is a system for extracting information from scanned European identification cards (used in hotels counters). In this application, the challenge is to recognize and register forms based only on keywords and to read all the card fields correctly. The results were very good: based on an extended benchmark of 173 cards, the system reached a recognition rate of 98.5 % for correctly read fields. Most of the errors were in street addresses that could easily be fixed by verifying them against a database of local addresses.

4. Conclusions

The approach proposed in this paper properly handles a large variety of forms, as exemplified in the results section. Using a combination of many features to recognize the template and the comprehensive path to locate fields precisely makes our method robust. The ability to use a single definition to handle a template and its many variants encourages the use of our approach.

References

[1] AnyForm, http://www.smartform.com/e/.

[2] Bin Yu and Anil K. Jain, "A Generic System for Form Dropout", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 11, Nov. 1996, pp. 1127-1134.

[3] Y. Belaid, et al., “Item Searching in Forms: Application to French Tax Form”, *Int. Conf. on Document Analysis and Recognition*, Aug. 1995, pp. 744-747.

[4] C.D. Yan, Y.Y. Tang and C.Y. Suen, “Form Understanding System Based on Form Description Language”, *Int. Conf. on Document Analysis and Recognition*, Oct. 1991, pp. 283-293.

[5] H. Fujisawa, Y. Nakano and K. Kurino, “Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis”, *Proc. of the IEEE*, Vol. 80, No. 7, 1992, pp. 1079-1092.

[6] R.A. Lorie, “A System for Exploiting Syntactic and Semantic Knowledge in Automatic Recognition”, *IAPR Workshop on Document Analysis Systems*, 1994, pp. 277-294.

[7] Y. Navon, A. Heilper and E. Walach. “Method for OCR Oriented Image Binarization”, *European Patent No. 98480038.3-2201*, Nov. 1998.

[8] Y. Navon, "Layer-based Binarization for Textual Images", Int. Conf. on Pattern Recognition, Dec. 2008.

[9] C. Charras and T. Lecroq, "Sequence Comparison", www-igm.univ-mlv.fr/~lecroq/seqcomp/index.html, 1998.