Methods and Techniques in Handwritten Form Recognition

DEAN CURTIS, E.A. YFANTIS, JAE ADAMS, TRENTON PACK
Digital Image Processing Laboratory
Department of Computer Science
University of Nevada, Las Vegas
4505 Maryland Parkway, Las Vegas, NV 89154
UNITED STATES OF AMERICA
http://web.cs.unlv.edu/digitalimageprocessinglab

Abstract – The medical records for employees contain vital information that an organization must store, sometimes indefinitely. This becomes a burdensome task as the number of paper records to maintain increases. The solution is to create an autonomous system that can confidently process, recognize and classify scans of paper records, and also facilitate the future addition of new entries. This process started with examining as many forms as possible for consistent form features. Features that were discovered included the form identification number, the form logo, the number of check boxes in a form, the number of structural lines in a form, and the number of words in a form. Once these features have been determined, the recognition process can begin. Several methods have been developed, leading to the current implementation; a standard error weighted translation. This method draws from the development of a Syntax Directed Translation. The initial results of the Standard Error Weighted Translation yield an acceptable recognition efficiency of the random set of sample forms.

Key-Words: Form Recognition, ID Recognition, Syntax Directed Form Translation, Standard Error Weighted Translation

1 Introduction

It is essential for many organizations to maintain medical records of each employee. If the quantity of medical forms grow to a large number for an organization, storage and retrieval for both the short term and long term becomes a challenge. Thus, it has been proposed that the information on these paper forms be stored into a digital format.

The recognition process developed relies on the extraction of several form features. These features will be briefly described in this paper.

Once the features have been extracted, the form recognition procedures can begin. Several procedures had been developed, each of which will be discussed in detail. The first method attempted to recognize the form identification number (form ID) through optical character recognition methods (OCR). This procedure provided unreliable results, so a second procedure was developed. This procedure performed a translation of the form based on concepts in compiler theory. This method is called the Syntax Directed Form Translation. The third method captured the essence of the Syntax Directed Translation, but eliminated the single point of failure issues that stemmed from it. The third method is a Standard Error Weighted Translation.

This report highlights the feature extraction methods and provides detailed descriptions of the algorithms and implementations of the form recognition procedures.

2 Feature Extraction Methods

To facilitate the form recognition algorithms, five feature extraction routines were developed that extract each feature individually. Figure 1 contains a sample form in which the five features are present.

The first two features are the form identification (ID) and the form logo. The third feature is the structural lines that exist within the form. This feature distinguishes between horizontal and vertical lines. The fourth feature is a count of the checkboxes. The last feature is the word count. Each of the last three features not only provides the counts, but also the organization of the features within the form.

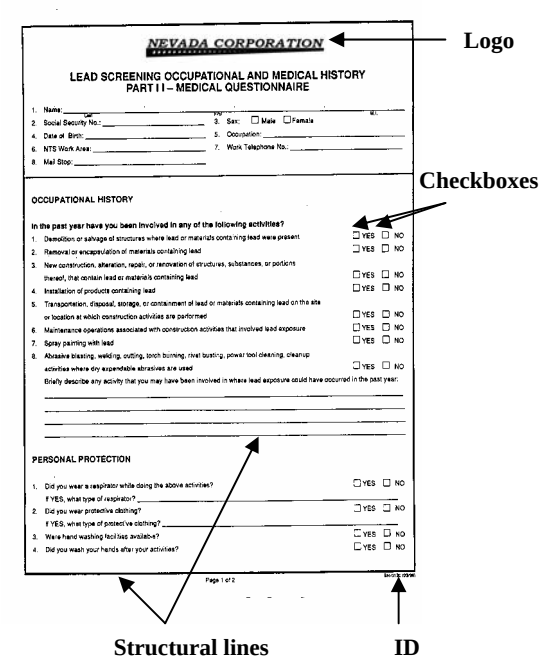

Fig. 1 Form features

3 Direct ID Recognition

The first method developed in form recognition was a method that used the ID detection routine to find the ID, and then identified the letters and numbers of the ID using optical character recognition (OCR) [3].

There were two problems posed with this method. The first problem is that not all forms have a form ID. Many forms consist of several pages, and in this case, the only form that contains the ID is the first page. Subsequent pages do not contain the ID. So, this problem eliminates the possibility of complete form recognition from the ID, as forms that do not have an ID are not identifiable by this routine.

The second problem is that the ID letters and numbers often become connected. This causes problems with the OCR method, as the connection changes the state of the letter or number being identified. Several enhancements have been made to the OCR technique that give the ability to separate the letters or numbers that are connected, but these methods do not maintain the integrity of the letters or numbers. Future work includes developing OCR techniques to separates letters and numbers, while maintaining the integrity of the letters and numbers. Figure 2 provides an example of an ID image that is intact, and another that suffers from this connection problem. It was decided that the ID recognition would best fit as part of the process as opposed to being the only step in form recognition.

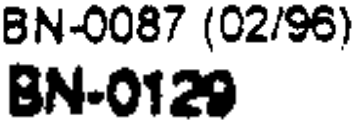

Fig. 2 Two different form identifications

4 Syntax Directed Form Translation

The idea of a Syntax Directed Form Translation is derived from compiler theory [2], and employs the fundamental nature of a syntax directed compiler. The main motivation to develop the syntax directed compiler was to make it possible to perform a parsing in one pass. The idea taken from this concept to be used in form recognition is an implementation of a rule-based form classifier. This provides the ability to only make one pass through a form, and have all the capabilities necessary to recognize the form in this single pass.

The second motivation to employ the syntax directed algorithm is the concrete nature of the translation. Much like the syntax directed compiler technique relies on a context-free grammar to guide a translation, the syntax directed form translation relies on a set of rules that guide the translation. The rules in the case of a syntax directed compiler are based on the language which the context free grammar describes, and in form translation, the rules are based on the features that define the form. For example, the form in figure 1 can be described as a form with 26 checkboxes, 16 horizontal lines, 2 vertical lines, 261 words, a logo and a form identification tag. The features also have an additional characteristic beyond the quantity: the positions and organization of these features.

4.1 Syntax Directed Algorithm

As mentioned above, the syntax directed form translation implements the rule-based nature of the syntax directed compiler. Figure 3 demonstrates the steps of the syntax directed form translation.

The two sources of input for the algorithm are the form to be recognized (in the center of figure 3) and the complete set of forms. It is important to note that the data in the complete form set is not the actual form images, but rather the features that represent each of the forms (both the quantity and organization). The process of recognition in this algorithm follows a very precise canon of steps. Each step is defined by each feature.

The algorithm starts by taking the first form from the complete form set and the form to be recognized, and performs the ID matching routine. If the matching routine returns a "pass", then those two forms match in terms of their identifications, and the form from the complete form set is added to the current matching set. The ID matching routine processes the two forms inputs, and follows the logic of figure 4. This comparison is repeated with each of the remaining forms in the complete form set. So, after this step, all the forms in the current match set match the input image in terms of their identifications.

```
if(f_id exist && t_id !exist)
  return fail;
else {
  if(f_id exist && t_id exist)
    if(f_id location != t_id location)
      return fail;
}
return pass;
```

Fig. 4 ID Matching routine logic. f_id is the form to be recognized and t_id is the form from the matching set.

After the ID matching step, the next step is logo matching. This represents the next rule in the rule-based classifying algorithm. This step, unlike the previous step only works with the set of forms that passed the ID rule (figure 4). So, the logo matching routine works only with the form to be recognized and each form from the current form set, as opposed to working with the complete form set. This step follows the same procedure as the first step in

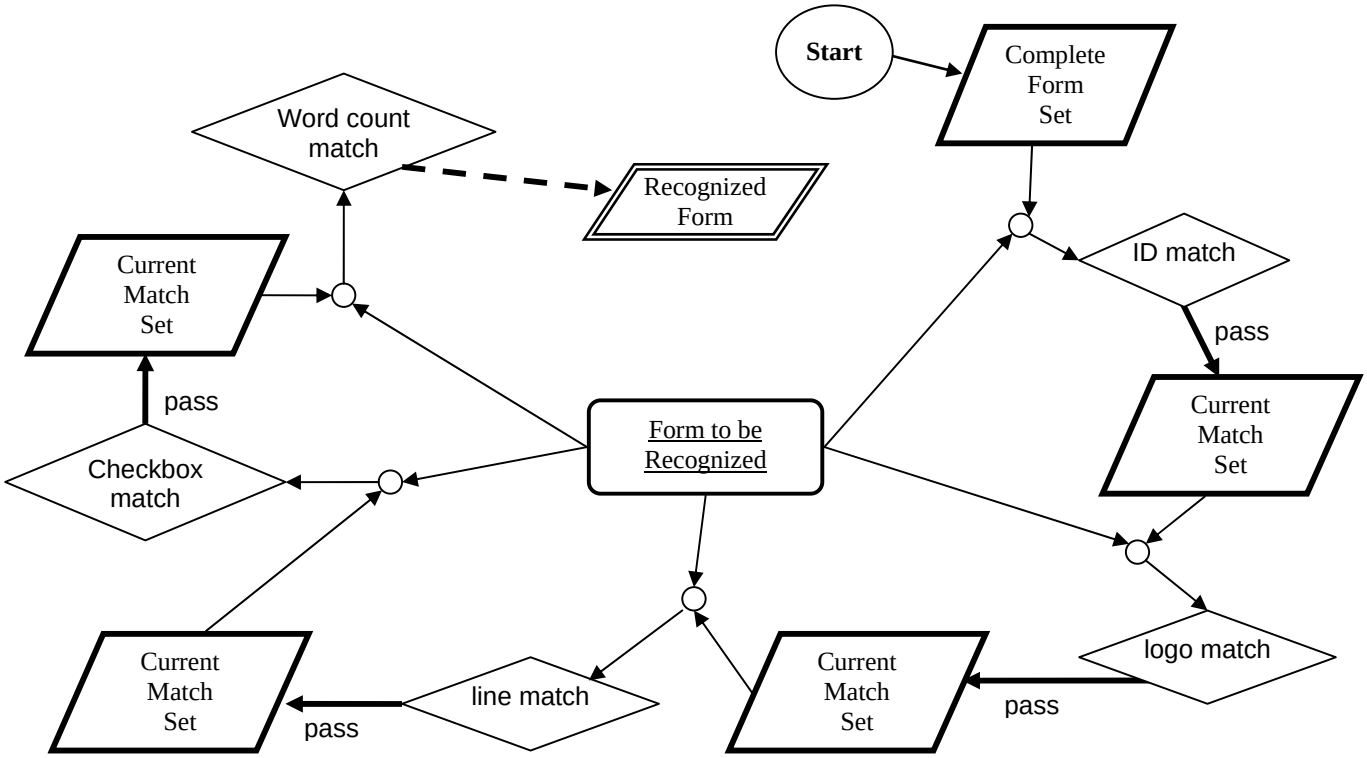

Fig. 3 The algorithm for the syntax directed form translation.

that the form to be recognized and the first form from the current form set are processed through the logo matching routine. If the logo matching routine returns a "pass", then that form from the current form set is placed into the next current form set. This continues until all forms in the current form set have been processed. After all of the forms have been processed, there is a new current form set which contains those forms that passed the logo matching routine and inherently passed the ID matching routine as well.

This process continues for the remaining rules: structural lines, checkboxes and word count. Each rule has its own boolean matching routine. The line matching routine not only checks for a match in the total number of lines, but also checks the positions of the lines. The position checking also applies for the checkbox and word count matching routines. They check for a match in the total number, and check for similarity in the organization.

After the word count matching routine processes the final form set, then the form in the last state (in figure 3 called the "recognized form") is the correct form. The essence of the Syntax Directed Form Translation is that the form in the final state has passed all rules, and is the correct form.

4.2 Issues with Syntax Directed Form Translation

There are three possible fail conditions that the last state could be in after the rule-based classifier finishes: there are no forms in the last state, there are form(s) in the last state but the correct form is not in the last state, and there is more than one form in the last state. These three possibilities are derived from the main problem with this approach: a single point of failure.

The single point of failure in this algorithm does not cause the entire process to crash before completing, but is a problem based on the idea that a form could be a match with the form to be recognized, and all but one feature is recognized correctly. For example, the matching routines for the word count, the checkbox count, the line count and the logo all pass, but the ID matching routine fails. This problem can occur if the forms are scanned, and scanning noise occurs, or a variable was acted upon the form before scanning that affected the state of the ID, thus preventing the ID detection from finding the ID. This is the fundamental essence of the single point of failure in the algorithm. This problem alone was ample motivation to seek out a more confident algorithm.

5 Standard Error Weighted Translation

To eliminate the single point of failure, two fundamental modifications must be made to the Syntax Directed Form Translation.

The first modification is to, instead of processing only a subset at each step as they pass each matching routine, process each possible form with the matching routine. This modification poses negligible implications in terms of runtime, as the worst-case for the Syntax Directed Form Translation is O(n), and the worst-case for this modification is also O(n). The only difference is that it is possible (but rare) for the Syntax Directed Form Translation to be O(1) in the best case, whereas the best case in this modification is the same as the worst-case. So, the modification of having each form be processed by each matching routine does not significantly affect the computational runtime. This modification also eliminates the single point of failure problem, as each form is processed by all matching routines.

The second modification eliminates the Boolean pass/fail nature of the matching routines in the Syntax Directed Form Translation. In the Standard Error Weighted Translation, there is great difficulty in characterizing a form based on the simple pass/fail results of the five main features. So, instead of each matching routine returning simply pass or fail, they return an index (out of 100%) representing the distance that particular feature is from the current form to be recognized (0% meaning no match and 100% meaning a perfect match).

These two modifications provide the foundation for a more confident recognition algorithm. These modifications create a more powerful recognition method called the Standard Error Weighted Translation (figure 5).

5.1 Standard Error Weighted Translation Algorithm

This algorithm captures the essence of the syntax directed form translation, but the elimination of the step-based procedures deviates from the pure Syntax Directed Form Translation. Instead of using "pass or fail" routines, the standard error weighted translation uses what are called qualifiers. Essentially, there is a

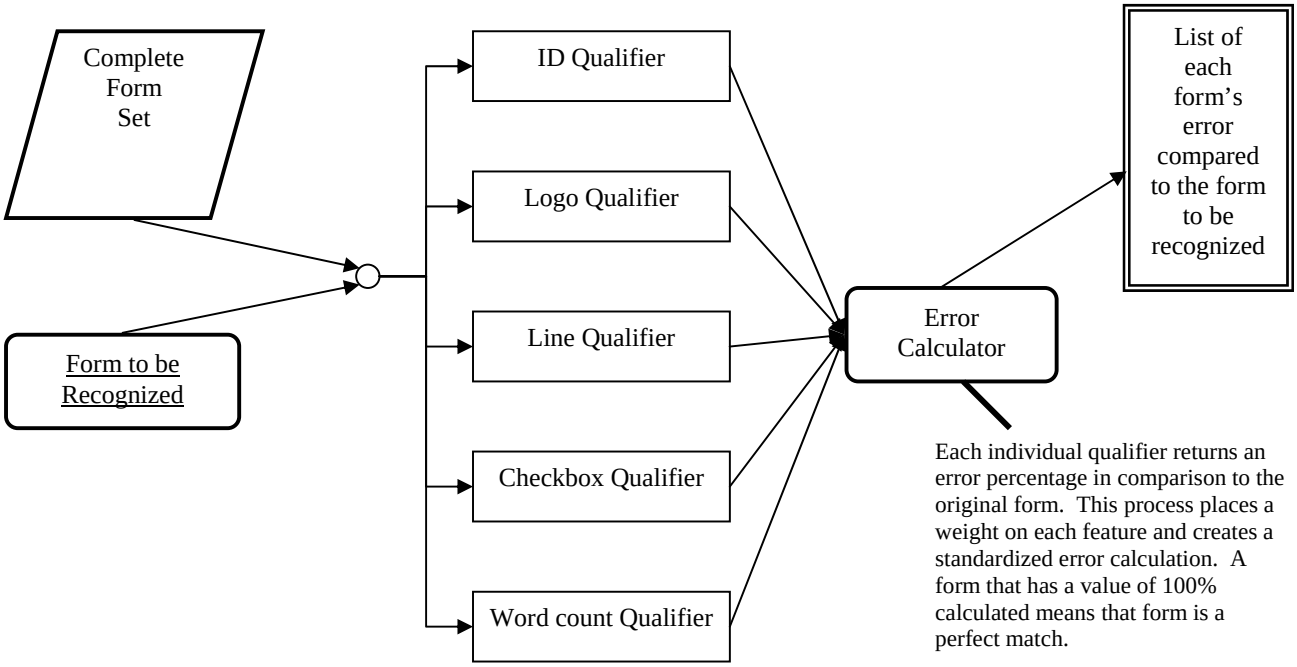

Fig. 5 The Standard Error Weighted Translation

qualifier for each feature, and the qualifier returns an index representing, through standard error calculations, how far the current form is from the form to be recognized in terms of that particular feature (0% means no match and 100% means perfect match). Also, each feature is given a normalization weight. These weights are based on feature priorities. The ID and logo get the lowest weights, and the word count, line matches and check boxes have the higher weights.

The algorithm begins with the complete form set and the form to be recognized. The first form from the set of forms along with the form to be recognized is processed by each qualifier (ID, logo, structural lines, checkbox and word count). Each qualifier returns an index that is weighted and summed up by the error calculator (figure 5). The standard error for this form is stored in a list. Then, the next form from the complete form set follows the same process as the first form, and the error for that form is placed in the same list. This continues until an error has been calculated for all forms from the complete form set. At the end of the process, each form in the system has been processed by each qualifying test, and a list of each form's error is stored. The classification decision is made based on this list of errors.

5.2 Results

Several outcomes were observed upon processing the test sample of forms. One outcome was that the list of accuracies contained an entry that had an index of 100%. In most cases, the next closest index ranged in between 70-85%. In these cases, this gap was enough to name the form with the index of 100% as the correct form. In other cases, there was an entry close to 100% in addition to the entry that was 100%. For example, one entry is 100%, and the next highest is 97%. In these cases, the form with 100% was the correct form, but additional testing is necessary to verify such as the ID verifying method discussed in section 3.

Another outcome was when the highest index was not 100%, but was greater than 90%. In each case, the highest index represented the correct form. There were several instances where the next highest index was within 10% of the highest index. In each

case, the highest index represented the correct form. However, to ensure confidence in the autonomous nature of the system, additional verifying methods are being developed to accommodate these scenarios, such as the ID verifying method discussed in section 3.

Table 1 shows the results of the standard error weighted translation. It is important to note that the highest index in each sample was the correct form. In several cases, there were several indices close to the highest, and in these cases there will need to be verifying methods other than human verification.

Range of highest index (%)	Percentage indexed within corresponding range
100	74%
90-99	10%
60-89	9%

Table 1 Results from the form samples.

6 Conclusion

Three methods of form classification have been presented in this paper. The ID recognition method and the Syntax Directed Form Translation method do not accommodate the autonomous nature of the recognition process. The Standard Error Weighted Translation, on the other hand, accommodates the autonomous recognition, as well as providing a confident result.

The ID recognition method lacks the necessary confidence, and the ID which it relies upon is not included in all forms. The Syntax Directed Form Translation has its own flaws in the single point of failure. The effectiveness is lost, because it is possible for a form to be a match while defective in a feature. So, the Standard Error Weighted Translation has been developed to be used in form recognition.

Future work includes developing verification methods such as the ID recognition discussed in section 3. These verifying methods are necessary to facilitate the autonomous nature of the recognition process. It is not feasible to expect human interaction for verifying purposes. So, methods are being developed that can contribute to both the stand alone nature of the system and the confidence of the system.

Acknowledgements

"Medical Records" (E-records) is research being conducted with the collaboration with Quest Technologies Inc. and the support of the Department of Energy. Special thanks extended to Dr. Stephen Rice, Vice President of Research at UNLV.

References:

[1] Bunke, H. and Wang, P.S.P. "Handbook of Character Recognition and Document Image Analysis" World Scientific (1997)

[2] Aho, A., Sethi, R., Ullman, J. "Compilers: Principles, Techniques and Tools." Addison-Wesley (1986) pp. 33-39

[3] Mori, S., Nishida, H., Yamada, H. "Optical Character Recognition" Wiley-Interscience (1999)