Reliable OCR Solution for Digital Content Re-mastering

Xiaofan Lin
Publishing Systems and Solutions Laboratory
HP Laboratories Palo Alto
HPL-2001-232
September 28th , 2001*

E-mail: xiaofan_lin@hp.com

digital content re-mastering, OCR, multiple classifier combination, reliability, printing-on-demand, majority voting

This paper addresses the system's aspects of OCR solutions in the context of digital content re-mastering. It analyzes the unique requirements and challenges to implement a reliable OCR system in a high-volume and unattended environment. A new reliability metric is proposed and a practical solution based on the combination of multiple commercial OCR engines is introduced. Experimental results show that the combination system is both much more accurate and more reliable when compared with individual engines, thus it can fully satisfy the need of digital content re-mastering applications.

* Internal Accession Date Only Approved for External Publication?
To be presented at and published in the SPIE Conference on Document Recognition and Retrieval IX, San Jose, CA 20-25 January 2002

Reliable OCR solution for digital content re-mastering

Xiaofan Lin
Hewlett-Packard Laboratories, 1501 Page Mill Road, MS 1L-15, Palo Alto, CA 94304
E-mail: xiaofan_lin@hp.com

ABSTRACT

This paper addresses the system's aspects of OCR solutions in the context of digital content re-mastering. It analyzes the unique requirements and challenges to implement a reliable OCR system in a high-volume and unattended environment. A new reliability metric is proposed and a practical solution based on the combination of multiple commercial OCR engines is introduced. Experimental results show that the combination system is both much more accurate and more reliable when compared with individual engines, thus it can fully satisfy the need of digital content re-mastering applications.

Keywords: digital content re-mastering, OCR, multiple classifier combination, reliability, printing-on-demand, majority voting

1. INTRODUCTION

With decades of research in both academia and industry, Optical Character Recognition (OCR) has already become a mature technology that is very cost-effective in digitizing existing printed materials. Modern OCR software packages from major vendors can achieve nearly perfect recognition on standard laser-quality documents. Their performance on low quality images such as fax and multi-generation copy has also improved a lot. Fruitful work has been done in all the related fields: binarization (converting the gray-scale/color text image to bi-level), zoning (dividing the whole image into homogeneous regions) and the actual character recognition (ACR, including character/word segmentation, feature extraction, classification, post processing based on statistics or linguistics and multiple classifier combination) [1]-[3].

However, even if we decide to leverage the existing OCR technologies rather than to develop from scratch, there is still a lot of work remaining on the system level, which has seldom been discussed. This paper tries to fill the gap between generic commercial OCR engines and a reliable solution in digital content re-mastering (DCRM). We first analyze the unique challenges posed for an OCR solution in DCRM. Then a new reliability metric is proposed and a practical system based on engine combination is introduced with experimental results. Instead of focusing on the "inner pieces" of OCR we are more concerned about building a satisfactory solution with commercial off-the-shelf technologies.

2. OCR FOR DCRM

By DCRM, we refer to the process of digitization and enrichment of non-electronic content. A good example is the printing-on-demand (POD) of books and journals whose original digital versions are not available. The basic steps involve scanning the printed materials into digital images and then generating compact, high quality electronic versions by image processing and document analysis algorithms. One desirable and often indispensable step is to OCR the images and to store the ASCII text in the electronic documents. This will greatly enrich the contents and enable many value-added services beyond simple viewing and printing: indexing, search, summarization and other Natural Language Processing (NLP) features. In this context, there are a few unique system requirements on the OCR part compared with other applications (see Table 1):

- It is not necessary to be error-free.

In most applications, it is still the images that will be either visually shown or printed out. The text behind the image is used only for additional functionalities, which can work well even if there are a few errors. One such implementation is a variant of Adobe's PDF format in which the image is shown on top of "invisible" text.

- Speed is not a big concern.

Generally, the system will be running in batch mode without human interference, so we can distribute the jobs across a number of servers to achieve desired throughput.

- It should be able to keep consistent processing quality for a variety of printed materials in an unattended environment. This poses a major challenge.

Because large amounts of content are going to be digitized, the cost and time of human proofing will be prohibitive. Traditionally, character recognition rate, rejection rate and substitution rate are used to measure OCR performance. However, they are not directly suitable for re-mastering applications. For one thing, since no manual proofing is planned in the production stage, we should accept whatever results OCR gives us and the rejection rate will actually be zero. More importantly, these metrics cannot quantify the consistency requirement. For example, one system makes one error per page on 100 pages and the other system makes 100 errors on a single page and no errors at all on the other 99 pages. As far as re-mastering is concerned, the first system should be fine because the one misrecognized character is unlikely to have any substantial impact on the services. For the second system, because a lot of errors happen on one page, it is quite possible that the quality of service will be too bad to be accepted. Based on this consideration, we can use the standard deviation of the error rates on the test set to quantify the system's performance consistency:

$$\sigma = \sqrt{\sum_{i=1}^{n} (R(X_i) - \overline{R})^2 / (n-1)} \tag{1}$$

where R(X) is the character recognition rate on Page X, and $\overline{R}$ is the average recognition rate on the n pages. In the above example, the average recognition rates are the same for the two systems, but the first system has a much lower standard deviation, which means higher reliability than the second system.

Table 1. Requirements on OCR in various applications

Applications	Volume	Human Intervention	Acceptable Residual Error Rates	Speed	Major Challenges
Daily Office Document Re-entry	Low	Can assist in binarization, zoning and error-check on every page.	Decided by individual organizations	The faster, the better	Recognition rate, features, GUI
Bank Check Processing /Mail Piece Sorting	High	Rejected pages	Near zero	Real-time	Near zero error rate on passed pages with rejections as few as possible
DCRM	High	Almost none	Low	Not strict	Consistent quality without human interference

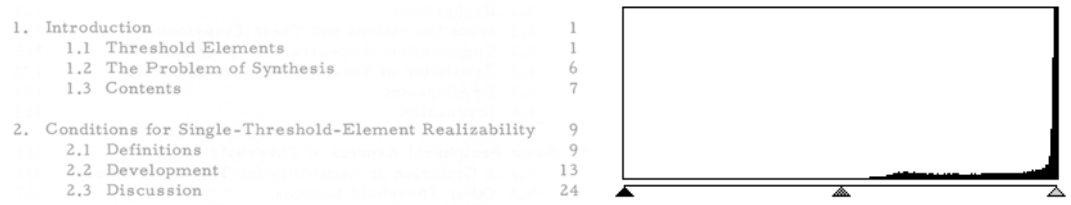
3. RELIABILITYANALYSIS OF COMMERCIAL OCR ENGINES

It will take huge amount of effort to develop a high performance OCR engine. On the other hand, quite a few high-performance OCR products are available on the market and they all provide SDKs for developers. So an efficient strategy is to build the solution on top of commercial OCR engines. Along this direction our first question is: Is there any single "super" engine that directly satisfies our requirements? If the answer is yes, we can simply use it. We have evaluated three major OCR engines (called Engine A, B and C respectively) on the market. Because the system is expected to run in automatic mode, we can feed the original gray-scale or color images into each engine and let it finish the whole job including binarization, zoning and the actual character recognition. The testing results show that although OCR has come a long way in the past decade, every engine still has some sort of weakness, making it not reliable enough on its own.

For example, Engine A's accuracy on the image in Fig. 1(a) is very low. At first glance, the image looks perfect and it is difficult to understand why Engine A cannot perform "decently" on it. In fact, the problem is attributed to the gray-scale

histogram in Fig. 1(b). The gray-scales completely concentrate in the higher end (128-255) and the thresholding algorithm inside the engine must be "fooled" into a bad thresholding decision, which leads to poor recognition.

Engine B sometimes has problems with zoning and will drop off small text regions as noise or even classify a text region as graphics.



(a). Image (b). Histogram

Fig. 1. Sample image and the gray-scale histogram

This problem can also be analyzed from system reliability's perspective. Fig. 2 shows the typical configuration of an OCR system. Because it is a cascading system, any error in one step will spread to later steps. In addition, zoning or binarization errors can easily lead to "burst" errors —a whole block of text missing, unrecognizable or with very poor accuracy. Research [12] shows how bad thresholding can seriously affect recognition rate. Worse still, zoning and binarization turn out to be the weak link for most OCR engines compared with the ACR part for a couple of reasons:

- Most commercial OCR products are designed primarily for the first application in Table 1 and the operator can modify the computer-generated zoning and thresholding. So these steps usually only take secondary place in R&D.
- Improving the zoning and binarization by explicitly combining several methods is still rare, while it is already a common practice to do the combination in the ACR step. There are two factors behind this unbalance. First, combination in zoning and binarization is relatively new and there are still many open problems on this topic [4]. Second, if multiple zoning/binarization results are kept, all the following steps will be repeated and the total processing time will increase dramatically. In the ACR, usually the characters with high confidence from the primary classifier will go directly through without invoking additional classifiers and the overall speed will not decrease too much.

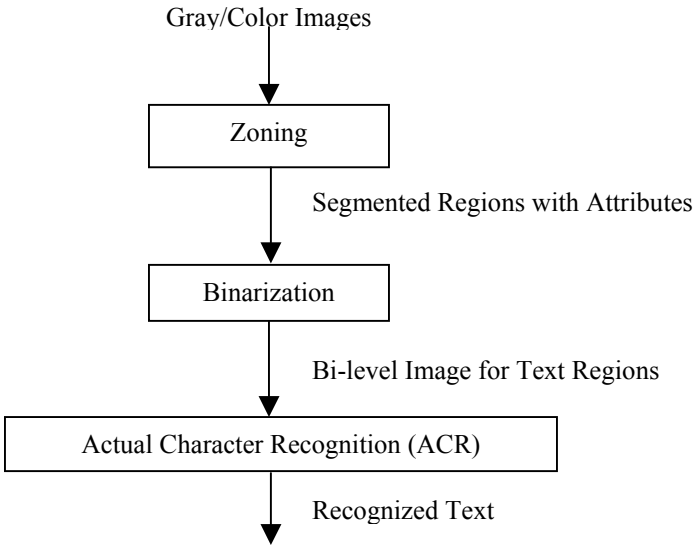


Fig. 2. Workflow of a typical OCR system

4. CONFIGURATION OF A MULTI-STAGE COMBINATION SYSTEM

Since any single engine proves insufficient, it is natural to combine them in some way so that several algorithms can complement each other. Although a lot of research on multiple classifier combination has been published, most are concerned about the character/word level combination, in which the segmented bi-level character/word image is sent to different classifiers [8]-[10]. As mentioned earlier, one bottleneck is in the zoning and binarization. So a sound solution should incorporate all the stages rather than only the ACR. When more than one stages cascade in the system, various topologies exist to build a combined system. Let's consider the following scenario:

The whole process consists of three steps and in each step there are three alternatives. Fig. 3(a) shows the parallel-cascading configuration, in which different methods are first combined at every stage. Fig. 3(b) is the cascading-parallel configuration, in which individual complete systems are first built with only one method in each stage.

If the following two conditions are satisfied:

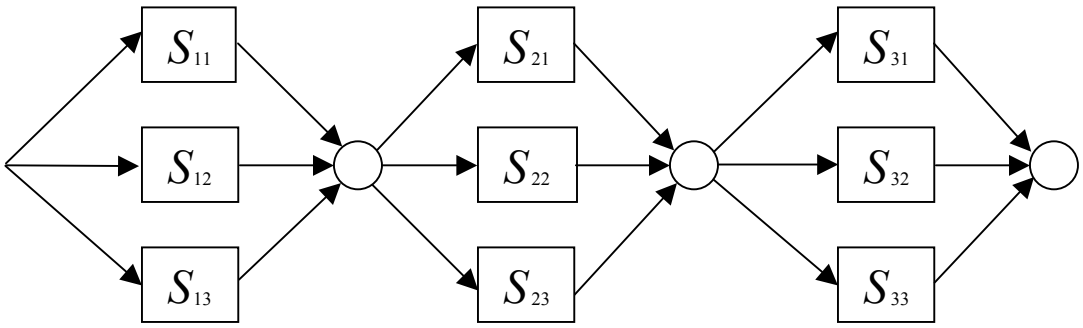
- Independence holds among different steps and different methods for the same step.
- The combination is perfect in that it can already find the best choice.

We can calculate the reliability for two configurations using P_{ji}, the reliability of the Method i of Stage j (S_{ji}):

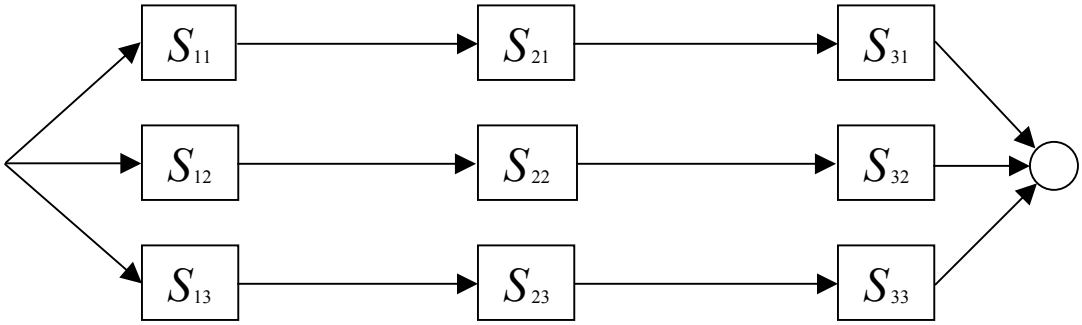
$$P_a = \prod_{j=1}^{3}(1-\prod_{i=1}^{3}(1-P_{ji})) \tag{2}$$

$$P_b = 1-\prod_{i=1}^{3}(1-\prod_{j=1}^{3}P_{ji})) \tag{3}$$

It can be proved that P_a is always larger than P_b. For example, when $P_{ji} = 0.9(j = 1,2,3 \text{ and } i = 1,2,3)$, P_a =0.997 and P_b =0.980. Without any combination, the three-stage system's reliability is merely 0.9^3 =0.729. Obviously, both combination schemes can boost the reliability significantly. On the other hand, most commercial OCR SDKs only expose high-level APIs (for example, they do not provide an interface to retrieve the bi-level image from the thresholding) and it is difficult to implement the first configuration. So we make a trade-off here and choose the second configuration.



(a) Parallel-cascading configuration



(b) Cascading- parallel configuration

Fig. 3. Different configurations for the combination of multiple-stage system

5. OCR SOLUTION BASED ON MULTIPLE COMMERCIAL OCR ENGINES

Fig. 4 shows the proposed OCR solution. The original gray-scale/color image is fed into each commercial OCR engine, which goes through all the steps in Fig. 2 and outputs the text and the bounding box (BBox) for every word.

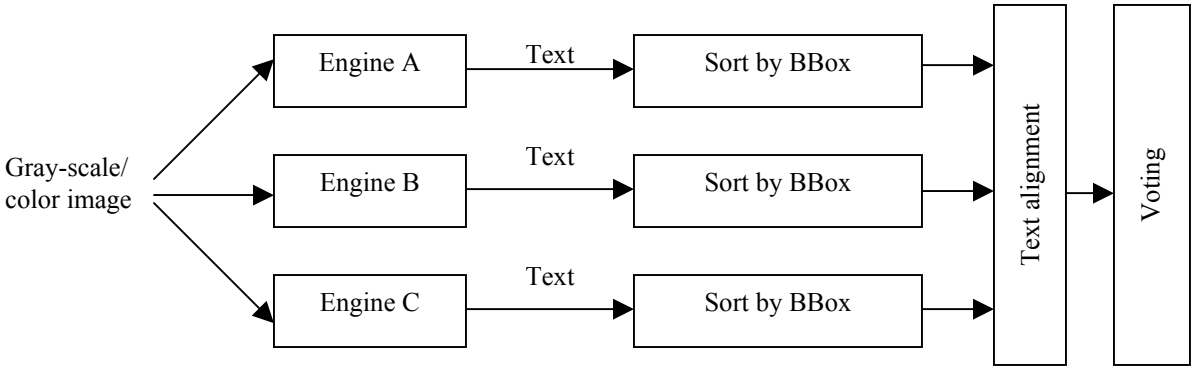


Fig. 4. Integrated OCR solution

5.1 Text stream alignment

Before the combination, the text streams have to be aligned by finding the correspondence among characters from different engines. Besides insertions, deletions and substitutions, displacement problems are also to be handled, because different engines can block the text in different orders. Two kinds of information can be useful in the alignment:

- Bounding box

Ideally, different engines will find each character in the same position of the image. But a couple of factors make this far from reality. First, each engine will do a sort of preprocessing on the image, such as deskewing. Some engines report the coordinates after preprocessing while some report the original ones. Accordingly each character can be reported with different coordinates by the engines. Second, the character segmentation in each engine will generate extra difference in the bounding box. These two factors make alignment by BBox alone quite complicated, if possible.

- Text context

Another source is the text stream itself. By finding the best text match between two text streams, the character correspondence can be identified. Since text match does not use bounding box information at all, it has the widest applicability and is used a lot. Many algorithms have been designed to match up multiple strings of text [5]-[7]. However, there is a dilemma on the displacement issue. If it is not allowed in the match, we cannot correctly synchronize two OCR results when the texts are output in different orders. On the other hand, if displacement is considered and a

subsequence appears more than once in the text stream, the match algorithm can be confused. Our alignment algorithm incorporates both kinds of information. First, the words from each engine are sorted according to their geometric locations (from top to bottom and left to right). Then the sorted text streams are aligned using string match, in which the displacement is not considered because it should have already been corrected during the sorting step.

This is illustrated by Table 2. After the reordering, the text streams in Engine A and B are converted to:

A: 1. Introduction 1 1 1 Thrsho1d E1er ients 1 2 The roblem of Synthesis 6 1 3 Contents 7 2 Conditions br Single-Threshold-Element Realizability 9 2.1 Definitions 9 2.2 Development 3 2 3 Discus sior

B: Introduction 1 1 .1 Thre shold Elements 1 1.2 The Problem of Synthesis 6 1.3 Contents 7 Conditions for Single-Threshold-Element Realizability 9 2.1 Definitions 9 2.2 Development 13 2.3 Discussion 24

The final alignment result is:

```
A: □ONIEN' S 1. Introduction 1 1 □1 Thr□ sho1d E1er ients 1 □□2 The □roblem of
B: CONTENT□S□□□ Introduction 1 1 .1 Thre shold Elem□□ents 1 1.2 The Problem of
A: Synthesis 6 1 3 Contents 7 2 Conditions b□r Single-Threshold-Element Realiz
B: Synthesis 6 1.3 Contents 7 □□Conditions for Single-Threshold-Element Realiz
A: ability 9 2.1 Definitions 9 2.2 Development □3 2 3 Discus sio□□□
B: ability 9 2.1 Definitions 9 2.2 Development 13 2.3 Discus□sio 24
```

5.2 Majority voting

Following text alignment is the combination step. In theory, the combination can be done on three levels [8]: symbol level (only the identity of the best candidate is used), rank order level (all the candidates from each classifier are considered) and measurement level (the confidence score of each candidate is taken into account). We can get the most information out of measurement level and achieve the best result. However, from the system's perspective we choose a simple method---majority voting due to the following considerations:

- Insufficient information for advanced combination. We can only take whatever level of information that is output from each commercial OCR engine. In fact, of the three engines we are using, two give the score of the best candidate but do not provide any alternatives. The remaining one gives the alternatives with confidence scores but no score for the top candidate.
- High extra cost associated with getting confidence information. Most commercial OCR engines divide the functionalities into several tiers with different pricing structures. For example, in some engines the confidence score information belongs to the higher tier, whose license fee is much higher than the basic tier. In the production stage, many machines will be running the program and the difference in the license cost can be huge among different tiers.
- Effectiveness by using voting. One big advantage of combining different commercial engines is that they are developed by unrelated organizations and thus are relatively independent of each other. As analyzed in [11], when several algorithms are unrelated, even majority voting can greatly improve the accuracy.

In cases where the three engines disagree with each other, we simply select the result from the reference engine, which has the highest average recognition rate.

5.3 Correcting deletion and insertion errors through combination

In Section 3, we pointed out that zoning is an important error source and deletion/insertion errors can arise from incorrect zoning. In fact, majority voting can also be used to handle deletion/insertion errors. When two engines find the same sequence that cannot be aligned with the third engine, the sequence will be kept. On the other hand, if a sequence from one engine cannot be aligned with both of the other two engines, it will be dropped.

6 EXPERIMENTAL RESULTS

Twenty 400-dpi gray-scale/color images scanned from various academic journals or books have been used for testing.

First, we show through an example how the proposed solution can correct the variety of errors mentioned in Section 3. For the region shown in Fig. 1, the recognized results and the combination result are shown in Table 2:

Table 2. Results on image in Fig. 1

Engine A	Engine B	Engine C	Combination
1. Introduction 1 1 1 Thrsho1d E1er ients 1 2 The roblem of Synthesis 6 1 3 Contents 7 2 Conditions br Single-Threshold-Element Realizability 9 2.1 Definitions 9 2.2 Development 3 2 3 Discus sior	Introduction 1 .1 Thre shold Elements 1.2 The Problem of Synthesis 1.3 Contents Conditions for Single-Threshold-Element Realizability 2.1 Definitions 2.2 Development 2.3 Discussion 1 1 6 7 9 9 13 24	1. Introduction 1.1 Threshold Elements 1 1.2 The Problem of Synthesis 6 1.3 Contents 7 2. Conditions for Single-Threshold-Element Realizability 9 2.1 Definitions 9 2.2 Development 13 2.3 Discussion 24	1. Introduction 1 1 .1 Threshold Elements 1 1.2 The Problem of Synthesis 6 1.3 Contents 7 2. Conditions for Single-Threshold-Element Realizability 9 2.1 Definitions 9 2.2 Development 13 2.3 Discussion 24

It can be seen that all the three engines have made some errors. For Engine A, there are several mistakes due to possibly poor thresholding. Engine B does not have substitution errors, but it misses the two chapter numbers, gets a wrong word space, and outputs the text in a weird order because its zoning identifies the text as two-column. Engine C is the best, except that one page number disappears. After the combination, all the errors can be corrected.

Next, we present the statistics on the testing pages in Table 3. If m characters in ground truth are recognized as n characters (m=0: insertion, n=0: deletion, m=n=1: simple substitution, otherwise: compound error), we will count max (m, n) errors in the statistics. It can be seen that each engine's error rates vary a lot on different images, from zero to several percent. That is why we introduce the standard deviation in addition to the average error rate as a measurement of reliability. After combination, the average error rate drops by over 40% and the standard deviation decreases by over 30% compared with the best individual engine. This means that the combination system is both much more accurate and more reliable.

Table 3. Test results on the twenty pages

No	Number of Characters	Engine A Errors	Engine B Errors	Engine C Errors	Combination Errors
1	1998	4	4	24	6
2	3140	18	7	7	3
3	3351	6	6	4	2
4	2526	60	0	1	1
5	2531	16	0	0	0
6	2320	0	5	0	0
7	1637	10	25	14	7
8	2480	60	0	5	5
9	2578	150	5	1	0
10	2242	120	1	0	0
11	4268	8	16	8	5
12	1490	26	20	27	22
13	4230	12	11	13	11
14	4071	19	30	17	8
15	1934	12	13	3	0
16	3027	4	13	2	1
17	2042	4	8	2	0
18	3574	10	9	2	2
19	2310	10	32	2	0

20	2960	2	17	1	2
Error rates: $\overline{E}$ =1- $\overline{R}$ (10^{-3})		11.29	4.50	2.86	1.67
Standard Deviation: σ (10^{-3})		16.4	4.56	4.62	3.22

7. CONCLUSIONS

Using several complementary methods to boost recognition performance is not new either in theory or in practice. However, this work makes contributions in the following areas:

- Address the requirements imposed on OCR in different applications, especially in the DCRM.
- Use the standard deviation of recognition rates to measure the reliability of OCR system.
- The reliability bottleneck has been identified, and a practical solution has been introduced with several novel features:
  - The input to the individual engines is the whole gray-scale/color images. So the zoning and binarization results from different engines are also taken into account.
  - Before the standard string-match based text alignment, sorting is performed using geometric position information in order to improve the match speed and accuracy.
  - All the three types of errors (substitution, insertion, and deletion) are properly handled in the combination.

Most important of all, experimental results show that this systematic approach leads to a robust and reliable OCR solution satisfactory for re-mastering applications.

ACKNOWLEDGEMENTS

The text string alignment program was developed by Phil Cheatle. The author would like to thank Steven Simske for his valuable advice on this research and Kathleen Gust for her careful review of this paper. Thanks also go to Burns John, who initiated this research and has supported it from the start.

REFERENCES

1. M. Boker, "Omnidocument Technologies", *Proceedings of IEEE*, pp. 1066-1078, July 1992.
2. S. Mori, C.Y. Suen, and K. Yamamoto, "Historical Review of OCR Research and Development", *Proceedings of IEEE*, pp. 1029-1058, July 1992.
3. J. Schürmann et al, "Document Analysis-From Pixels to Contents", *Proceedings of IEEE*, pp. 1101-1119, July 1992.
4. S. Klink and T. Jäger, "MergeLayouts - Overcoming Faulty Segmentations by a Comprehensive Voting of Commercial OCR Devices", *Proceedings of 5th ICDAR*, Bangalore, India, Aug 1999.
5. R. A. Wagner and M. J. Fisher, "The String-to-String Correction Problem", *Journal of the ACM*, **21(l)**, pp. 168-173, 1974.
6. P. A. V. Hall and G. R. Dowling, "Approximate String Matching", *ACM Comput. Survey*, **12**, pp. 381-402, 1980.
7. W. Miller and E.W. Myers, "A File Comparison Program", *Software, Practice and Experience*, **15(11)**, pp. 1025-1040.
8. L. Xu, A. Krzyzak and C.Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwritten Recognition*", IEEE Trans. Syst. Man Cybernet*. **22 (3)**, pp. 418-435, 1992.
9. T.K. Ho, J. J. Hull, and S. N. Srihari, "Decision Combination in Multiple Classifier Systems", *IEEE Trans. on PAMI*, **16(1)**, pp.66-75, 1994.
10. L. Lam and C.Y.Suen, "Optimal Combinations of Pattern Classifiers", *Pattern Recognition Letters*, **16**, pp.945-954, 1995.

11. L. Lam and C.Y. Suen, “A Theoretical Analysis of the Application of Majority Voting to Pattern Recognition”, *Proceedings of IEEE*, pp. 418-420, 1994.
12. R. Smith, C. Newton and P. Cheatle, “Adaptive Thresholding for OCR: A Significant Test”, *HP Technical Report*, www.hpl.hp.com, 1993.