Joost van Beusekom

Optical Document Security in High Volume Office Environments

vom Fachbereich Informatik der Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte

Dissertation

Dekan:
Prof. Dr. Karsten Berns, Technische Universität Kaiserslautern

Berichterstatter:
Prof. Dr. Thomas Breuel, Technische Universität Kaiserslautern
Prof. Dr. Katrin Franke, Gjøvik University College

Vorsitzender der Promotionskommission:
Prof. Dr. Reinhard Gotzhein, Technische Universität Kaiserslautern

Datum der Aussprache:
18. Juni 2010

D 386

Acknowledgements

"Setzt Iech är Zieler net ze niddereg"
(Roland Bauler)

I would like to thank Prof. Thomas Breuel for the trust and the many ideas that lead to this thesis. I am also grateful to Prof. Katrin Franke for the interesting discussions and the positive feedback concerning the content of this thesis.

It is a pleasure to thank all the colleagues and former colleagues who helped in different ways in the development of this thesis: Adrian Ulges for the many interesting discussions over the last years. Faisal Shafait for the rich cooperation on the different topics in document image analysis. Daniel Keysers and Christoph Lampert for their excellent supervision during their stay at the DFKI. Armin Stahl for the guidance and support. Adrian Ulges, Faisal Shafait, Armin Stahl, Matthias Reif, Damian Borth and Jane Bensch for the proofreading. Markus Goldstein and Christian Schulze for the support in technical questions. Ilya Mezhirov for his enlightening answers on whatever programming language questions. Marco Schreyer for the fruitful cooperation and discussions. Ingrid Romani and Gabriele Sakdapolrak for their help in dealing with all the administrative hurdles. Elisabeth Hoffmann and Janis Tweedy for their support in finding papers and citations.

Ech wëll op dëser Plaz och all deene Leit Merci soen déi dofir gesuergt hunn, dat een dat wesentlecht net aus den Aen verléiert: dem Pierrot, dem Pol, dem Liz, dem Fréd, dem Misch, dem Zelito, dem Pierre, dem Georges an all deenen déi ech vergiess hunn. Och e grousse Merci u meng Gesëschter, d'Veerle an de Geert. Meng Dankbarkeet gelt och dem Claudine fir d'Gedold déi hat all déi Zäit opbruecht huet.

Ten laatste wou ik mij nog bij mijn ouders bedanken voor de ondersteuning die zij mij all die jaren toekomen lieten.

Abstract

The widespread use of scanning in the processing of forms in insurance companies, banks and other high volume businesses has created new opportunities for fraud by document tampering and modification. Such alterations can be performed either on the paper itself, or using digital image processing and editing.

This thesis describes several methods that can be integrated into automated mail processing systems to detect potentially fraudulent alterations of documents. Three novel approaches are presented: first, two model-based approaches are presented. Second, an alteration detection approach based on text-line examination is introduced. Important contributions are also made by providing data sets for evaluating the methods.

Using prior information on genuine documents, the model-based approaches use intrinsic document features for authentication of the document source and for detecting alterations. Authentication by identifying the source uses counterfeit protection system (CPS) codes. These permit printer class identification with accuracies of up to 92.5% ($n = 67$, 16 classes). The image-based comparison of CPS patterns provides authentication on a printer level and attains an accuracy of 88.3% ($n = 94$). The second model-based approach models the positional variations of fixed foreground components. These can effectively detect forgeries on the basis of scanning distortions. This method shows an accuracy of 97.0% ($n = 168$) for classification between forged and genuine documents. Finally, an alteration detection approach is presented that works even in absence of prior knowledge. Text-line skew angles and text-line alignment are used to check the document's contents for optical consistency. Evaluation on two pass printed documents and manually pasted text achieves an area under the ROC curve (AUC) score of $AUC = 0.89$ ($n = 191$).

The operation of the alteration detection is supported by additional pre-processing: an integrated approach for orientation and skew detection is presented, showing competitive results of up to 98.8% ($n = 979$) for orientation detection accuracy and up to 98.0% ($n = 979$) for the skew detection accuracy on the UW-I dataset. Border noise removal by explicit detection of the page's content area is presented together with a comprehensive and extensive set of evaluation procedures. Results on public datasets show an accuracy of up to 97.2% on zone-level classification ($n = 1440$). A flexible and accurate case-based reasoning approach is presented for logical labeling, showing accuracy rates of up to 99.6% (n=6770) on the MARG dataset.

The methods described in this thesis suggest that authentication and alteration

detection methods can be used to build an effective filter in high volume document digitization setups. This constitutes a first step into the direction of a fully automated forgery detection system.

Contents

1 Introduction **1**

2 Orientation and Skew Detection **9**
- 2.1 Introduction 10
- 2.2 Related Work 12
  - 2.2.1 Skew Detection Methods 13
  - 2.2.2 Orientation Detection Methods 17
  - 2.2.3 Combined Skew and Orientation Detection Methods 18
- 2.3 Description of the Approach 19
  - 2.3.1 Geometric Text-Line Model 20
  - 2.3.2 Quality Function for Text-Line Extraction 21
  - 2.3.3 Text-Line Extraction 22
  - 2.3.4 One-Step Skew and Orientation Detection 24
  - 2.3.5 Run-Time Improvement 26
- 2.4 Evaluation 27
  - 2.4.1 Ascender to Descender Ratios for Latin Script Languages . 28
  - 2.4.2 Skew Detection Accuracy 28
  - 2.4.3 Orientation Detection Accuracy 30
- 2.5 Conclusion 35

3 Document Cleanup **39**
- 3.1 Introduction 40
- 3.2 Previous Work 41
- 3.3 Border Noise Removal by Detecting the Pageframe 43
  - 3.3.1 Page Frame Model 44
  - 3.3.2 Quality Function 45
  - 3.3.3 Branch-and-Bound Optimization 46
  - 3.3.4 Parameter Refinement 47
- 3.4 Performance Measures 49
  - 3.4.1 Page Frame Detection Accuracy 49

3.4.2 Performance Gain in Practical Applications 51
3.5 Experimental Setup and Results 52
3.5.1 Results on Page Frame Detection Accuracy 53
3.5.2 Performance Gain in Practical Applications 59
3.6 Conclusion 62

4 Logical Labeling for Model Identification 65
4.1 Introduction 66
4.2 Previous Work 67
4.3 Logical Labeling using Case-Based Reasoning 69
4.3.1 Similarity Measure 70
4.3.2 Reuse 71
4.3.3 Block Distance 72
4.4 Evaluation 75
4.4.1 Invoice Dataset 75
4.4.2 MARG Dataset 76
4.5 Results 76
4.5.1 Results on the Invoice Dataset 77
4.5.2 Results on the MARG Dataset 77
4.5.3 Considerations on the Case-Base Size 79
4.5.4 Comparison to Other Labeling Methods 79
4.6 Conclusion 81

5 Model-Based Authentication 83
5.1 Introduction & Related Work 84
5.2 Authentication using CPS Codes 86
5.2.1 Extraction of the Tracking Dots 89
5.2.2 Printer Class Identification 90
5.2.3 CPS Code Comparison 94
5.2.4 Evaluation 97
5.2.5 Results for CPS Code Classification 100
5.2.6 Results for CPS Code Comparison 102
5.2.7 Discussion 104
5.3 Authentication using Positional Variations 105
5.3.1 Alignment 106
5.3.2 Positional Variation Modeling 108
5.3.3 Evaluation 110
5.3.4 Results 111
5.3.5 Discussion 112
5.4 Conclusion and Future Work 113

6 Line Examination for Document Security **117**
6.1 Introduction . . . 118
6.2 Text-Line Features for Document Security . . . 119
6.2.1 Plausibility Check using Skew Angles . . . 120
6.2.2 Plausibility Check using Text-Line Alignment . . . 124
6.2.3 Integration of Skew and Alignment . . . 127
6.3 Evaluation . . . 128
6.4 Results . . . 130
6.4.1 Hypothesis Validation . . . 130
6.4.2 Results on Text-Line Skew Angles . . . 131
6.4.3 Results on Text-Line Alignment . . . 135
6.4.4 Results on the Combination of Both Features . . . 139
6.5 Conclusion & Future Work . . . 139

7 Discussion **143**

A Survey on Forgery Techniques **145**

B List of Comparisons for Evaluation on CPS **151**

C Ground-Truth Generation for Logical Labeling **153**

List of Figures

1.1 Example of forgeries 5
1.2 System overview 7

2.1 Skew variations using a high volume ADF scanner 11
2.2 Histogram of skew angles produced by ADF scanner 12
2.3 Result of page segmentation on skewed and deskewed image 13
2.4 Result of OCR on correctly and wrongly oriented image 14
2.5 Overview of the orientation and skew detection approach 15
2.6 Visualization of common skew detection methods 16
2.7 The anatomy of a line 20
2.8 Text-line model proposed by Breuel [18] 21
2.9 Text-line model with ascender line 21
2.10 Image points used for text-line extraction 22
2.11 Orientation detection using text-lines 24
2.12 Examples of text-line fitting on all four orientations 26
2.13 Histogram of the aspect ratio of connected components on UW-III 27
2.14 Histogram of the std. dev. of skew angles within one page in UW-I 27
2.15 Scatter plot of estimated and supplied skew angles 30
2.16 Processing time vs. number of extracted lines 32
2.17 Accuracy vs. number of extracted lines 32
2.18 Differences in errors 33
2.19 Plot of the accuracy vs. epsilon 34
2.20 Errors in pre-classification 35
2.21 Accuracy vs. number of extracted lines 36
2.22 Processing time vs. number of extracted lines 36
2.23 Examples for the output of the algorithm 37
2.24 Examples of failures 37

3.1 Example of a page frame and different types of noise 40
3.2 Example for demonstrating the parameter refinement on text-lines 48

3.3 Example image demonstrating inclusion of non-text zones 49
3.4 Example of the original and corrected ground-truth page frames . 53
3.5 Histogram of the noise ratio 55
3.6 Run time of the algorithm 56
3.7 Examples of page frame detection results in ICDAR 2007 dataset . 58
3.8 Examples of extracted page frames 59
3.9 Performance of page frame detection relative to the noise level . . 60
3.10 Result of a commercial OCR software 61

4.1 Examples of documents with different labels 67
4.2 Overview of case-based labeling 70
4.3 Illustration of the distance measure 72
4.4 Illustration of the features 74
4.5 Layout types in the invoice dataset 75
4.6 Layout Types 77
4.7 Accuracy versus case base size 80

5.1 Examples of CPS dots 87
5.2 Extraction of CPS dots 90
5.3 Computation of HPS and VPS distances 91
5.4 Examples of different patterns 92
5.5 Histogram of the values of the parameters t_x 94
5.6 Example for the prototype ambiguity 95
5.7 Examples of extracted prototypes 96
5.8 Test sheets from the Electronic Frontier Foundation 98
5.9 Excerpt of the list of printer classes 99
5.10 Example of a diffuse pattern 101
5.11 Example of sparse patterns 103
5.12 Example of differing patterns from the same Xerox printer 103
5.13 Visualization of the approach 106
5.14 Visualization of the Model 109
5.15 Examples of a genuine and a forged document 111
5.16 Example of distortions induced by copying 112
5.17 Examples the alignment and the resulting images 113
5.18 Results of the test on the pseudo-invoices 114
5.19 Results of the test on the copied invoices 114
5.20 Sorted Qualities and Log-likelihoods 115

6.1 Alignment Lines 119
6.2 Text-line skew angle examination 120
6.3 Examples of typographic enhancements 121
6.4 Skew angles of text-lines 122

6.5 Text-line alignment examination 124
6.6 Alignment Measure 126
6.7 Skew angles of text-lines 131
6.8 Histograms of distances to alignment lines 132
6.9 Results for the test on the skew angle 133
6.10 Results for the test on the skew angle 133
6.11 Examples of errors for skew angle-base classification 135
6.12 Results for the test on alignment 136
6.13 Results for the test on the alignment 137
6.14 Examples of errors for alignment-based classification 138
6.15 Results for the test on the combination 140
6.16 Results for the test on the combination 140
6.17 Results for the test on alignment 141
6.18 Results for the test on the alignment 141

A.1 Genuine Invoices 146
A.2 Genuine and Forged Logos 149

C.1 Layout type in invoice dataset 154
C.2 Illustration of the ground truth generation 155

List of Tables

2.1 Measurements of ascender to descender ratios 29
2.2 Comparison of different orientation detection approaches 38
2.3 Comparison of two approaches for different resolutions 38

3.1 Results for the connected component based evaluation 54
3.2 Results for the zone based evaluation 55
3.3 Results on the ICDAR 2007 page segmentation competition dataset 57
3.4 Results for OCR based evaluation approach 60
3.5 Performance improvement on layout-based retrieval 62

4.1 Distribution of layout types 75
4.2 Results on the invoice dataset 78
4.3 Results on the MARG dataset 78
4.4 Distribution of layout types 79

5.1 List of pairs of HPS and VPS distances occurring more than once . 100
5.2 Results on document 06 from the EFF-DFKI dataset 101
5.3 List of the MPF machines used to generate the test set 112

6.1 Measurements of the features for different runs 134

A.1 Distribution of the forgery types 148
A.2 Distortions of the forged invoices 148

C.1 Distribution of layout types 154

Chapter 1

Introduction

There is a German saying “Papier lügt nicht” (“paper doesn’t lie”), which expresses the trust that we commonly show in paper documents. Contracts, wills, doctor’s notes and invoices are just a few examples of typical paper documents which we trust without questioning. If there is any agreement between two parties, both want to have it “black on white”, meaning that if the details are written down on paper, we have solid proof of what has been agreed on.

While this expresses a deep trust in paper documents, what most people tend to forget is that paper documents can be manipulated. This is a serious problem as shown by the following numbers: in the *2008 Report to the Nation on Occupational Fraud and Abuse* [78] the *Association of Certified Fraud Examiners* estimates that $994 billion were lost to fraud in 2008. The fraud schemes of *billing* and *check tampering* that often involve document forgery represent 23.9% and 14.7% of the cases, and are responsible for a median loss of $100,000 and $138,000 per case. In a survey from 1988 among German questioned document examiners [92] Rieß reported that 80% of the cases relate to offenses against property, which is also reflected in the distribution of the type of questioned documents: checks (28.8%), contracts and mandates (11.6%) and bills and receipts (11.6%) are the most frequently analyzed documents.

Despite these numbers, in every day business relations documents remain widely insecure, lowering the threshold for successful fraud by falsification of documents. Insurance fraud e.g. can be easily done by forging invoices as these play a particularly important role when claiming compensation: invoices certify that a person has paid a certain amount of money to buy some product or service. An example could be the receipt that the customer receives when he buys his digital camera, or the invoice that he gets after the accidental damage of his car was repaired. If the customer has car insurance, he will send the invoice of the repairs to the insurance company as a receipt for the costs he had for repairing his car. The insurance company will then reimburse him with the money.

Traditionally, invoices are processed manually. This gives the insurance adjuster the possibility to check the invoice for semantic inconsistencies. He might look for prices that to not fit the item, e.g. having a “Goodyear 195/65R15 Ultragrip 7” tire that costs more than the list prize of 80 EUR. He could also look for obvious optical features of forgery as e.g. correction fluid or other easily observable inconsistencies. For example, the perpetrator could try to remove the original date using correction fluid, or he could try to modify numbers using a fine ball pen. The insurance adjuster, on his side, could try to obtain another invoice from the same repair shop and compare both to see if they share the same layout and same font type. This visual consistency check is important, as it is much harder to produce a visually similar forgery than just to keep the contents of the forged invoice consistent.

A key problem with forgery detection is that, in the effort of continuously increasing the degree of automation in office environments, visual consistency checks become impossible due to data volume and automated processing of invoices: incoming printed invoices are digitized and the insurance adjuster is shown only the extracted information of the invoice, so no visual consistency check is possible anymore. In this scenario, document forgery becomes an easy task. The only two constraints for the perpetrator are that his forgery is machine readable and that it is semantically consistent.

A simple approach to deal with the problem would be to have a manual inspection of each document before the digitization. This is however a tremendous cost factor, which cancels out the cost cutting gained by the digitization.

Another approach would be to increase the effort that a potential forger has to spend to create a forged document, thus increasing the document’s security. There have been several attempts to make documents more counterfeit resistant in the field of optical document security. Most methods base on the idea of adding an extrinsic feature to the document that is hard to forge. *Intrinsic* features are features that are present in the document due to the normal document generation process. In contrast, *extrinsic* security features are features that are added with the sole purpose of making a document more difficult to forge. There are numerous examples of extrinsic features: watermarks, security fibers, holograms, bar codes, intaglio printing, optically variable inks, etc. A good overview can be found in [122].

These approaches are known to be suited for documents with direct or indirect value, according to van Renesse’s classification [122]:

- **direct value:** documents giving access to unconditional and immediate value, e.g. banknotes.
- **indirect value:** documents that certifies a transaction or a right, e.g. diplo-

mas and passports

- **conditional value:** documents that give access to a value after the document has been inspected, e.g. admission tickets or check
- **informative value:** documents that have no immediate value apart from the informations that it contains, e.g. confidential reports
- **fictitious value:** documents that have no immediate value and that do not contain any valuable information, e.g. institution or company stationary

In the scenario of invoices and contracts, however, this approach is not feasible, as the invoice generating parties would have to add extrinsic security features, which would increase the production costs of the invoices dramatically. An approach using intrinsic document features is more appropriate.

Therefore, in this thesis the following research questions will be treated:

1. Is it possible to develop an automatic or semi-automatic system to filter forged documents in an automated document digitization environment?
2. What intrinsic features can be used to detect forged documents and how can these be used in an automatic or semi-automatic setup?
3. What are the requirements for the document image pre-processing to successfully apply the document security approaches?

The contribution of this thesis is a system for authentication of documents. The proposed approach is to introduce new barriers into the automated processing of the documents, which are able to filter the incoming stream of documents based on intrinsic features and give an alarm if a document looks suspicious. An operator can then decide if further steps have to be taken, based on the outcome of the different analysis steps.

But how can forgeries be detected? Or how do people forge documents in the first place? An extensive search in the literature and inquiries to questioned document examiners lead to the conclusion, that, to the author's best knowledge, there is so far no public quantitative study on the methods of forgery for the proposed scenario.

The results of an experiment[1] conducted by the author show that three forgery methods are being used:

- **Print, Paste & Copy (PPC) Forgeries:** out of 25 forgeries, three were generated by replacing a part of the invoice. This was done by printing the new text (presenting higher values or more items) on an empty sheet and

---

[1] Further details about the outcome of this test are given in Appendix A.

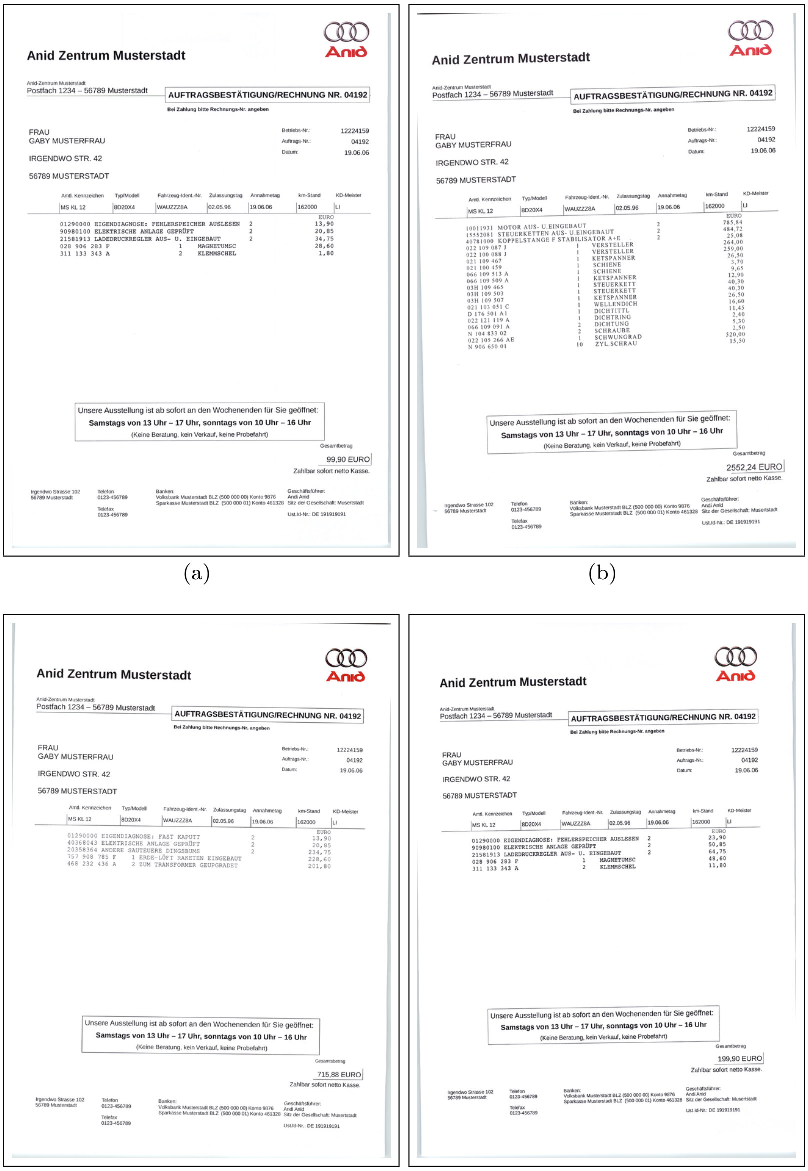
pasting this part onto the genuine document. This was then copied using a color copier. This technique was mostly used by people without a computer science background.

- **Reverse Engineered Imitations (REI) Forgeries:** three candidates generated an editable document by imitating the genuine invoice. Five samples were delivered. They scanned the genuine invoice and used it as a template to generate a new document by retyping all the text, putting the logos into place, etc.
- **Scan, Edit & Print (SEP) Forgeries:** the remaining 17 forgeries were generated by digitizing the invoice and manipulating the digital image. Here, mostly only numbers were modified, e.g. increasing the price of a tire or increasing the number of tires from two to four.

Looking at the forgeries (Examples are given in Figure 1.1), the following observations can be made:

- **visual consistency:** detecting forgeries by visual appearance is a tough problem. The visual appearance of the forgeries was consistent in most cases. Only in one PPC forgery the forged areas could be easily detected due to a slightly different font, which a trained person could eventually detect.
- **skew and alignment of text:** especially in the PPC forgeries a small dis-alignment of the text-lines in comparison to the remaining parts of the invoice could be observed. This concerns both the skew angle of the forged lines relative to the genuine areas as well as the left or right alignment. This effect was almost not observable for the SEP forgeries.
- **comparison to a genuine invoice:** in either scenario, if a genuine document was directly compared to a forgery it was easily visible that the forged document had been distorted. If e.g. the header lines were aligned, the footer lines did not fit any more.

These observations lead to the development of the features and methods that will be presented in this thesis. The presented approach has several advantages: first and most importantly, by the use of intrinsic features only the digitization process has to be extended and the document generating process remains unchanged. Modifying the document generation process in this scenario is very costly and practically impossible. Another tremendous advantage of the proposed methods over previously published intrinsic features is the relatively low resolution of 300 to 600 dpi at which the analysis is done. Other intrinsic features [50] require resolutions of up to 2400 dpi, which is not feasible in a large through-put setup: on the one hand, most high-volume automatic document feeding (ADF) scanners support



Anid Zentrum Musterstadt

Anid-Zentrum Musterstadt
Postfach 1234 – 56789 Musterstadt

AUFTRAGSBESTÄTIGUNG/RECHNUNG NR. 04192

Bei Zahlung bitte Rechnungs-Nr. angeben

FRAU
GABY MUSTERFRAU

IRGENDWO STR. 42

56789 MUSTERSTADT

Betriebs-Nr.:	12224159
Auftrags-Nr.:	04192
Datum:	19.06.06

Amtl. Kennzeichen	Typ/Modell	Fahrzeug-Ident.-Nr.	Zulassungstag	Annahmetag	km-Stand	KD-Meister
MS KL 12	8D20X4	WAUZZZ8A	02.05.96	19.06.06	162000	LI

				EURO
01290000 EIGENDIAGNOSE: FEHLERSPEICHER AUSLESEN			2	13,90
90980100 ELEKTRISCHE ANLAGE GEPRÜFT			2	20,85
21581913 LADEDRUCKREGLER AUS- U. EINGEBAUT			2	34,75
028 906 283 F	1	MAGNETUMSC		28,60
311 133 343 A	2	KLEMMSCHEL		1,80

Unsere Ausstellung ist ab sofort an den Wochenenden für Sie geöffnet:
Samstags von 13 Uhr – 17 Uhr, sonntags von 10 Uhr – 16 Uhr
(Keine Beratung, kein Verkauf, keine Probefahrt)

Gesamtbetrag
99,90 EURO
Zahlbar sofort netto Kasse.

Irgendwo Strasse 102
56789 Musterstadt

Telefon
0123-456789

Telefax
0123-456789

Banken:
Volksbank Musterstadt BLZ (500 000 00) Konto 9876
Sparkasse Musterstadt BLZ (500 000 01) Konto 461328

Geschäftsführer:
Andi Anid
Sitz der Gesellschaft: Musertstadt

Ust.Id-Nr.: DE 191919191

(a)

Anid Zentrum Musterstadt

Anid-Zentrum Musterstadt
Postfach 1234 – 56789 Musterstadt

AUFTRAGSBESTÄTIGUNG/RECHNUNG NR. 04192

Bei Zahlung bitte Rechnungs-Nr. angeben

FRAU
GABY MUSTERFRAU

IRGENDWO STR. 42

56789 MUSTERSTADT

Betriebs-Nr.:	12224159
Auftrags-Nr.:	04192
Datum:	19.06.06

Amtl. Kennzeichen	Typ/Modell	Fahrzeug-Ident.-Nr.	Zulassungstag	Annahmetag	km-Stand	KD-Meister
MS KL 12	8D20X4	WAUZZZ8A	02.05.96	19.06.06	162000	LI

				EURO
10011931 MOTOR AUS- U.EINGEBAUT			2	785,84
15552081 STEUERKETTEN AUS- U.EINGEBAUT			2	484,72
40781000 KOPPELSTANGE F STABILISATOR A+E			2	25,08
022 109 087 J	1	VERSTELLER		264,00
022 100 088 J	1	VERSTELLER		259,00
021 109 467	1	KETSPANNER		26,50
021 100 459	1	SCHIENE		3,70
066 109 513 A	1	SCHIENE		9,65
066 109 509 A	1	KETSPANNER		12,90
03H 109 465	1	STEUERKETT		40,30
03H 109 503	1	STEUERKETT		40,30
03H 109 507	1	KETSPANNER		26,50
021 103 051 C	1	WELLENDICH		16,60
D 176 501 A1	1	DICHTITTL		11,45
022 121 119 A	1	DICHTRING		2,40
066 109 091 A	2	DICHTUNG		5,30
N 104 833 02	2	SCHRAUBE		2,50
022 105 266 AE	1	SCHWUNGRAD		520,00
N 906 650 01	10	ZYL.SCHRAU		15,50

Unsere Ausstellung ist ab sofort an den Wochenenden für Sie geöffnet:
Samstags von 13 Uhr – 17 Uhr, sonntags von 10 Uhr – 16 Uhr
(Keine Beratung, kein Verkauf, keine Probefahrt)

Gesamtbetrag
2552,24 EURO
Zahlbar sofort netto Kasse.

Irgendwo Strasse 102
56789 Musterstadt

Telefon
0123-456789

Telefax
0123-456789

Banken:
Volksbank Musterstadt BLZ (500 000 00) Konto 9876
Sparkasse Musterstadt BLZ (500 000 01) Konto 461328

Geschäftsführer:
Andi Anid
Sitz der Gesellschaft: Musertstadt

Ust.Id-Nr.: DE 191919191

(b)

Anid Zentrum Musterstadt

Anid-Zentrum Musterstadt
Postfach 1234 – 56789 Musterstadt

AUFTRAGSBESTÄTIGUNG/RECHNUNG NR. 04192

Bei Zahlung bitte Rechnungs-Nr. angeben

FRAU
GABY MUSTERFRAU

IRGENDWO STR. 42

56789 MUSTERSTADT

Betriebs-Nr.:	12224159
Auftrags-Nr.:	04192
Datum:	19.06.06

Amtl. Kennzeichen	Typ/Modell	Fahrzeug-Ident.-Nr.	Zulassungstag	Annahmetag	km-Stand	KD-Meister
MS KL 12	8D20X4	WAUZZZ8A	02.05.96	19.06.06	162000	LI

				EURO
01290000 EIGENDIAGNOSE: FAST KAPUTT			2	13,90
40368043 ELEKTRISCHE ANLAGE GEPRÜFT			2	20,85
20358364 ANDERE SAUTEUERE DINGSBUMS			2	234,75
757 908 785 F	1	ERDE-LÜFT RAKETEN EINGEBAUT		228,60
468 232 436 A	2	ZUM TRANSFORMER GEUPGRADET		201,80

Unsere Ausstellung ist ab sofort an den Wochenenden für Sie geöffnet:
Samstags von 13 Uhr – 17 Uhr, sonntags von 10 Uhr – 16 Uhr
(Keine Beratung, kein Verkauf, keine Probefahrt)

Gesamtsbetrag
715,88 EURO
Zahlbar sofort netto Kasse.

Irgendwo Strasse 102
56789 Musterstadt

Telefon
0123-456789

Telefax
0123-456789

Banken:
Volksbank Musterstadt BLZ (500 000 00) Konto 9876
Sparkasse Musterstadt BLZ (500 000 01) Konto 461328

Geschäftsführer:
Andi Anid
Sitz der Gesellschaft: Musertstadt

Ust.Id-Nr.: DE 191919191

(c)

Anid Zentrum Musterstadt

Anid-Zentrum Musterstadt
Postfach 1234 – 56789 Musterstadt

AUFTRAGSBESTÄTIGUNG/RECHNUNG NR. 04192

Bei Zahlung bitte Rechnungs-Nr. angeben

FRAU
GABY MUSTERFRAU

IRGENDWO STR. 42

56789 MUSTERSTADT

Betriebs-Nr.:	12224159
Auftrags-Nr.:	04192
Datum:	19.06.06

Amtl. Kennzeichen	Typ/Modell	Fahrzeug-Ident.-Nr.	Zulassungstag	Annahmetag	km-Stand	KD-Meister
MS KL 12	8D20X4	WAUZZZ8A	02.05.96	19.06.06	162000	LI

				EURO
01290000 EIGENDIAGNOSE: FEHLERSPEICHER AUSLESEN			2	23,90
90980100 ELEKTRISCHE ANLAGE GEPRÜFT			2	50,85
21581913 LADEDRUCKREGLER AUS- U. EINGEBAUT			2	64,75
028 906 283 F	1	MAGNETUMSC		48,60
311 133 343 A	2	KLEMMSCHEL		11,80

Unsere Ausstellung ist ab sofort an den Wochenenden für Sie geöffnet:
Samstags von 13 Uhr – 17 Uhr, sonntags von 10 Uhr – 16 Uhr
(Keine Beratung, kein Verkauf, keine Probefahrt)

Gesamtbetrag
199,90 EURO
Zahlbar sofort netto Kasse.

Irgendwo Strasse 102
56789 Musterstadt

Telefon
0123-456789

Telefax
0123-456789

Banken:
Volksbank Musterstadt BLZ (500 000 00) Konto 9876
Sparkasse Musterstadt BLZ (500 000 01) Konto 461328

Geschäftsführer:
Andi Anid
Sitz der Gesellschaft: Musertstadt

Ust.Id-Nr.: DE 191919191

(d)

Figure 1.1: Examples of forged documents. Image (a) shows the genuine invoice that was given to the candidates. The next image (Image (b)) shows a PPC forgery, Image (c) a REI forgery and (d) shows an SEP forgery.

resolutions only up to 600 dpi or sometimes even less. On the other hand, higher resolutions dramatically increase the hardware requirements for processing these images. Finally, another advantage of most methods that will be presented in this thesis is that they are independent of the printing technique used to generate the document.

The single steps are presented in processing order in this thesis. An overview is presented in Figure 1.2: after scanning, a document is usually not perfectly aligned nor does it necessarily have the correct orientation. Therefore, in Chapter 2, a novel approach for an integrated skew and orientation detection approach is presented.

During the scanning process, border noise appears frequently. Chapter 3 proposes a novel approach for removing border noise by explicitly detecting the main page content area, the so-called page frame.

For automated processing of invoices and also for the later use of model based document authentication (authentication of a document compared to a model learnt from previously seen genuine documents from the same source), the incoming document has to be matched to its correct model. In Chapter 4 a case-based reasoning approach is developed that shows how the logical meta data of different text-parts in an invoice is extracted. This extracted meta information can be used for identifying the insurance adjuster in charge, or to detect the correct invoice source for document security applications.

Having a previously processed model of the invoice allows the insurance adjuster to verify the genuineness of the invoice in question. In Chapter 5 two automated ways of authentication of a document compared to its model are presented, using intrinsic document features. The first approach tries to identify if the same class of printers was used to generate two different invoices by analyzing the so-called counterfeit protection system (CPS) codes from color laser printers and color copiers. This approach is extended to identify if the same printer was used to generate two different invoices by comparing the CPS pattern of both documents. The second approach measured the distortions introduced during scanning and printing of forged documents by modelling the positional variability of fixed components in an invoice (e.g. characters in headers and footers). This model is used to verify an incoming invoice if all the fixed components are exactly where they should be.

If no model information is available, the insurance adjuster might also check for general consistency of the document. Two features that are used for this plausibility check are the text-line skew and the text-line alignment: it is reasonable to assume that for a genuine invoice, the text-lines will have all more or less the same skew angle. Lines differing from the expected skew angle could hint at a possibly tampered document. Also the alignment of text-lines is analyzed in the same way. In Chapter 6 automated methods are presented that analyze these features for document forgery detection purposes.

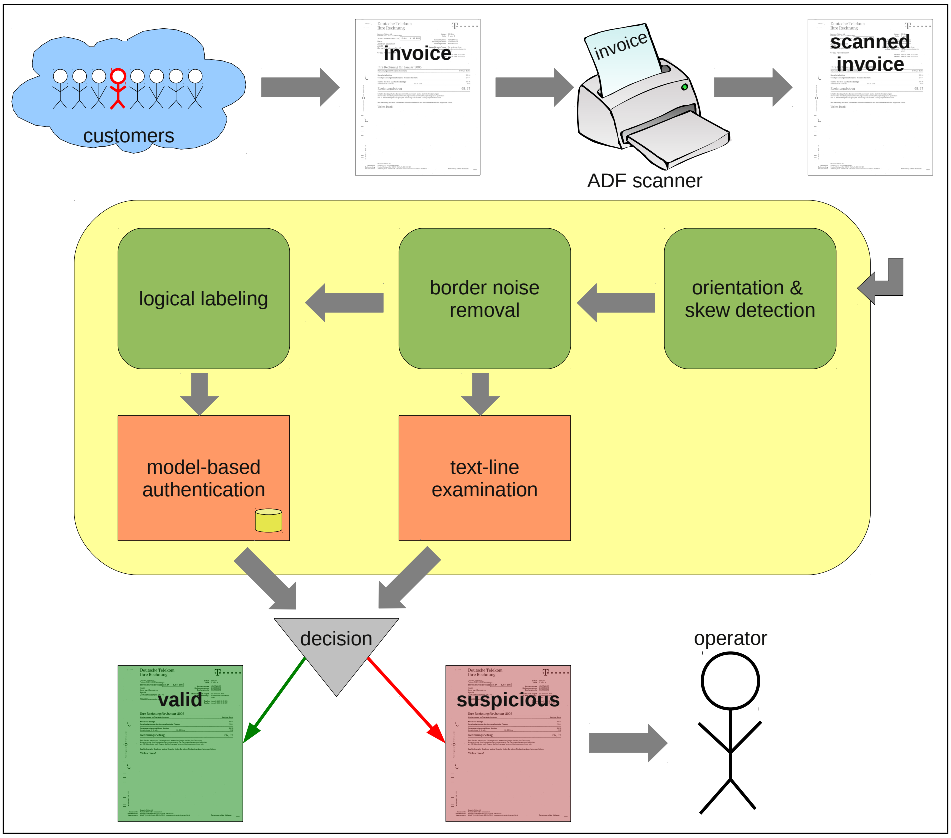


Figure 1.2: System overview: the yellow box represents the contributions of this thesis: the first three steps are general document image understanding methods with the goal of improving and cleaning the digital image as well as extracting meta data from the scanned invoices (Chapters 2,3 and 4. The orange boxes represent the document security related methods (Chapters 5 and 6).

Chapter 2

Orientation and Skew Detection

In large scale document digitization, orientation detection plays an important role, especially when digitizing incoming mail. The heavy use of automatic document feeding (ADF) scanners and, even more, automatic processing of facsimiles results in many documents being scanned in the wrong orientation.

In document security applications, however, knowing the correct orientation of the page is indispensable: accurately measuring text-line rotation angles is difficult. Also, applying model-based techniques on wrongly oriented document images will result in false alarms. Therefore, misoriented scans have to be corrected, as most subsequent processing steps assume the document to be scanned in the right orientation.

In this chapter, a new one-step skew and orientation detection method is presented. The advantage of the proposed method is that it combines accurate skew estimation with robust, resolution independent orientation detection. Its effectiveness is demonstrated on the UW-I dataset and a publicly available dataset from OCRopus[1], an open source optical character recognition (OCR) system. In this chapter, the following contributions are presented[2]:

1. An integrated approach for orientation and skew detection using text-line

---

[1] http://code.google.com/p/ocropus/

[2] This chapter is based on the author's work in
[119] J. van Beusekom, F. Shafait, and T. M. Breuel. Combined orientation and skew detection using geometric text-line modeling. *Int. Jour. on Document Analysis and Recognition*, 2010, and
[118] J. van Beusekom, F. Shafait, and T. M. Breuel. Resolution independent skew and orientation detection for document images. In *Proc. of SPIE Document Recognition and Retrieval XVI*, volume 7247, San Jose, CA, USA, January 2009.

finding.

2. An evaluation proving competitive results concerning accuracy for orientation detection (accuracy of 98.8%, $n = 979$) and for skew detection (accuracy of 98.0%, $n = 979$) on the UW-I data set.
3. Practically only little extra computation costs as the obtained line segmentation may be reused in the following OCR step.

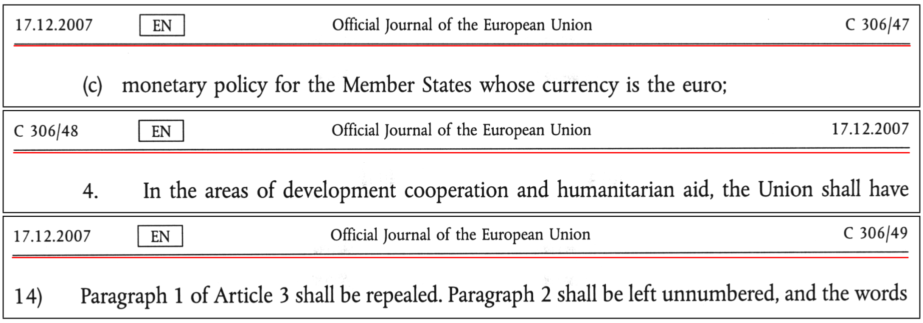
2.1 Introduction

When it comes to the large scale digitization of paper documents, manual interaction is often the bottleneck of the process. Manual interaction is mostly unwanted, mainly because it is expensive. This is the main reason for the use of automatic document feeding scanners. These scanners can easily scan huge amounts of detached sheets without the need of any human intervention. Modern scanners even incorporate techniques for multi feed detection and correction [30], reducing the human interaction to mainly filling and emptying the scanner's paper trays.

The automatic feeding, however, comes along with two drawbacks: first, the orientation of the document is not being manually checked before scanning. Second the resulting document images tend to be slightly rotated, as seen in Figure 2.1. A histogram of document rotation angles produced by scanning documents using an ADF scanner can be found in Figure 2.2, showing the variety of skew angles when using an ADF scanner. From this it is clear, that for accurate measurements as they are needed for document security applications, skew and orientation the page to be analyzed have to be corrected. It is easily comprehensible, that e.g. skew angle measurements for deciding whether the measured skew angles are suspicious or not has to be done on accurately deskewed document images.

Skew and orientation detection are the processes that in combination generate from a rotated document image a horizontally aligned correctly oriented document image. The term *skew* is mostly used for a slight rotation of the page, whereas *orientation* is used to denote if the page is 0°, 90°, 180° or 270° rotated. A more detailed discussion about the definitions of *skew* and *orientation* is given at the end of this section.

Overall, in the domain of mail digitization, orientation detection is an important pre-processing step. The use of ADF scanners is likely to introduce misoriented scans. According to estimates by a company providing document management system solutions around 1% to 5% of the documents are scanned in the wrong orientation. A more important number of misoriented pages originates from electronic facsimile processing: as some fax machines use automatic document feeding that is opposite (from bottom to top) to the intuitive direction of paper insertion



17.12.2007 EN Official Journal of the European Union C 306/47

(c) monetary policy for the Member States whose currency is the euro;

C 306/48 EN Official Journal of the European Union 17.12.2007

4. In the areas of development cooperation and humanitarian aid, the Union shall have

17.12.2007 EN Official Journal of the European Union C 306/49

14) Paragraph 1 of Article 3 shall be repealed. Paragraph 2 shall be left unnumbered, and the words

Figure 2.1: Examples of document images scanned using a high volume ADF scanner. The red line shows a perfectly horizontal line. It should be noted that all three images have different skew angles.

(up-right), it often occurs that misoriented facsimiles are transmitted. The estimated percentage of misoriented facsimiles, according to the information obtained by the same company as above, is about 50%.

As most subsequent processing steps in a document understanding system assume an up-right and deskewed position of the document, subsequent steps may fail when this is not granted. In Figure 2.3 the result of a page segmentation [76] assuming deskewed input on a skewed document image can be seen. It should be noted that for the deskewed image, the page segmentation algorithm gives a good block segmentation (yellow blocks). For the rotated image, however, the page segmentation returns useless results. In Figure 2.4 the result of optical character recognition on a text in two different orientations are shown. The text output on the misoriented document image is not usable anymore. Ocrad[3] was used to generate these results.

In literature, no clear distinction between *orientation* and *skew* of a document image is to be found. In Cattoni's survey [25], skew estimation techniques are presented that cover up to $\pm 90°$. Orientation detection has not been considered. Bose et al. [15] do not make any distinction between orientation and skew detection. Avila et al. [11] define the orientation of a document image to be either in landscape, portrait, up or down orientation. Bloomberg et al. [14] use an approach based on their practical experience, that rotation angles are not greater than $\pm 45°$ in real world setups. Mostly, the skew angles are small $\pm 5°$. The orientation is then defined as one of the angles $0° \pm 5°$, $90° \pm 5°$, $180° \pm 5°$ and $270° \pm 5°$. Skew removal thus consists in removing the small degree of rotation $[-5°, 5°]$. This definition is a reasonable choice for most real-world applications and is thus also used in the following.

---

[3] http://www.gnu.org/software/ocrad/

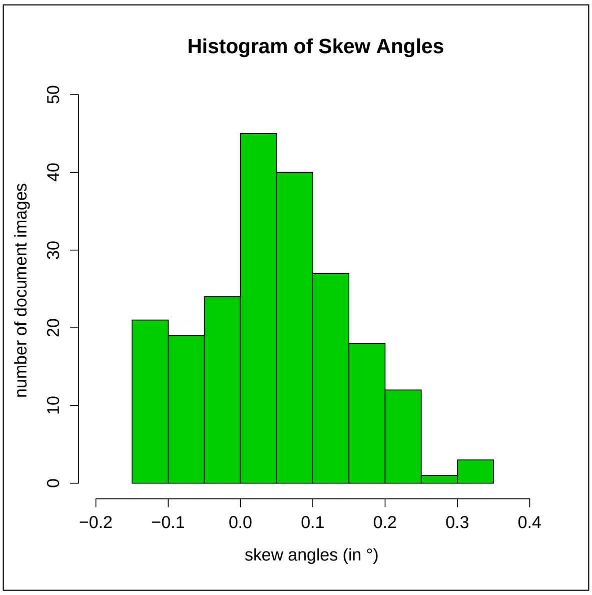


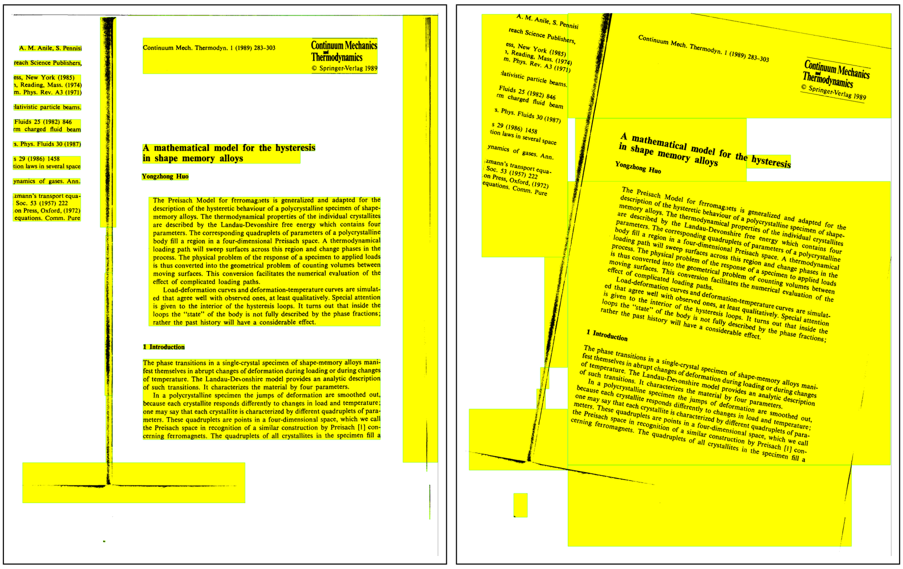
Figure 2.2: A histogram of skew angles of document images when using a high volume ADF scanner is shown. Please note that the skew angles are quite small but still present an important amount of variation.

An illustration of the approach presented in this chapter is given in Figure 2.5. Given an input image with unknown orientation and skew. The orientation that returns the best text-lines according to a model consisting of a base line and a descender line is considered as the correct orientation. The skew angle of the document is obtained by the skew angles of the text-lines.

2.2 Related Work

Previous work on orientation and skew detection can be categorized into three parts:

1. skew-only detection methods: these methods only consider skew detection and are not able to detect upside down or landscape orientations.
2. orientation-only detection methods: these methods are able to detect orientations but do not solve the skew estimation problem.
3. combined skew and orientation detection methods.



A. M. Anile, S. Pennisi
reach Science Publishers,
ess, New York (1985)
ı, Reading, Mass. (1974)
m. Phys. Rev. A3 (1971)
lativistic particle beams.
Fluids 25 (1982) 846
rm charged fluid beam
s. Phys. Fluids 30 (1987)
s 29 (1986) 1458
tion laws in several space
ynamics of gases. Ann.
zmann's transport equa-
Soc. 53 (1957) 222
on Press, Oxford, (1972)
equations. Comm. Pure

Continuum Mech. Thermodyn. 1 (1989) 283–303

Continuum Mechanics and Thermodynamics
© Springer-Verlag 1989

A mathematical model for the hysteresis in shape memory alloys

Yongzhong Huo

The Preisach Model for ferromagnets is generalized and adapted for the description of the hysteretic behaviour of a polycrystalline specimen of shape-memory alloys. The thermodynamical properties of the individual crystallites are described by the Landau-Devonshire free energy which contains four parameters. The corresponding quadruplets of parameters of a polycrystalline body fill a region in a four-dimensional Preisach space. A thermodynamical loading path will sweep surfaces across this region and change phases in the process. The physical problem of the response of a specimen to applied loads is thus converted into the geometrical problem of counting volumes between moving surfaces. This conversion facilitates the numerical evaluation of the effect of complicated loading paths.

Load-deformation curves and deformation-temperature curves are simulated that agree well with observed ones, at least qualitatively. Special attention is given to the interior of the hysteresis loops. It turns out that inside the loops the "state" of the body is not fully described by the phase fractions; rather the past history will have a considerable effect.

1 Introduction

The phase transitions in a single-crystal specimen of shape-memory alloys manifest themselves in abrupt changes of deformation during loading or during changes of temperature. The Landau-Devonshire model provides an analytic description of such transitions. It characterizes the material by four parameters.

In a polycrystalline specimen the jumps of deformation are smoothed out, because each crystallite responds differently to changes in load and temperature; one may say that each crystallite is characterized by different quadruplets of parameters. These quadruplets are points in a four-dimensional space, which we call the Preisach space in recognition of a similar construction by Preisach [1] concerning ferromagnets. The quadruplets of all crystallites in the specimen fill a

Figure 2.3: Result of page segmentation on skewed and deskewed image. Yellow blocks represent a segmented block. It should be noted that the algorithm expects deskewed images. Thus the result in the right image shows considerable errors in page segmentation.

2.2.1 Skew Detection Methods

For skew detection, Cattoni et al. [25] gave a good overview of the state of the art back in 1998. He proposes the following categorization:

- analysis of projection profiles: the basic idea is to compute projection profiles [88] for "all" skew angles and evaluate an objective function that is maximal when the correct skew angle is used. A visualization of the basic idea of the projection profile methods can be seen in Figure 2.6(a). The drawback of this method lies in its computationally expensive search: the coarser the angular parameter space is searched, the more time it will need.
- Hough transform [36]: using the fact that text-lines are parallel and the characters in each text-line are aligned, the Hough transform is applied to estimate the skew angle of the document. A visualization can be found in Figure 2.6(b). Character positions (x, y) are mapped to curves in parameter space (r, θ), where r is the distance between the line and the origin and θ is the angle between the normal vector of the line w.r.t. to the x-axis. The angular resolution depends on the resolution of θ in parameter space. Characters lying on a line will lead to peaks in parameter space.

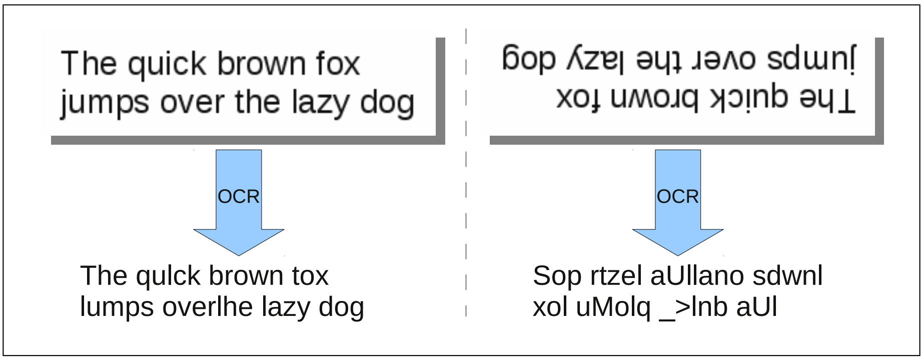


Figure 2.4: Result of optical character recognition using Ocrad on a test image in different orientations. The result of the correctly rotated image is not perfect but readable. The result of the 180° rotated image is of no practical use.

- clustering of components: assuming that characters in one line are aligned and closer to each other than to characters in different lines, their spacial relationship is used to estimate the skew angle. An example can be found in Figure 2.6(c).

- other techniques: other approaches include the analysis of vertical deviation, the analysis of the gradients, the Fourier transform, and morphological operations.

Approaches using Projection Profiles

In 1997, Del Ninno et al. [33] present a method belonging to the family of projection profile methods. A projection profile giving more weight to lines with long runs of black or white pixels is computed. Using this projection profile, areas of interest are identified. For these areas the projection profile along different skew angles is computed to maximize an objective function. Evaluation on a non-public dataset containing 114 images showed results of 97% correctly deskewed document images. However, no clear definition of a correctly detected skew is given. The average angle measurement error has been reported being below 0.2°.

A solution to a more complex problem, the problem of documents containing text lines where the skew angles vary depending on the vertical distance along the page has been proposed by Spitz [108]. Using projection profiles, iteratively, the different skew angles of the text-lines are computed.

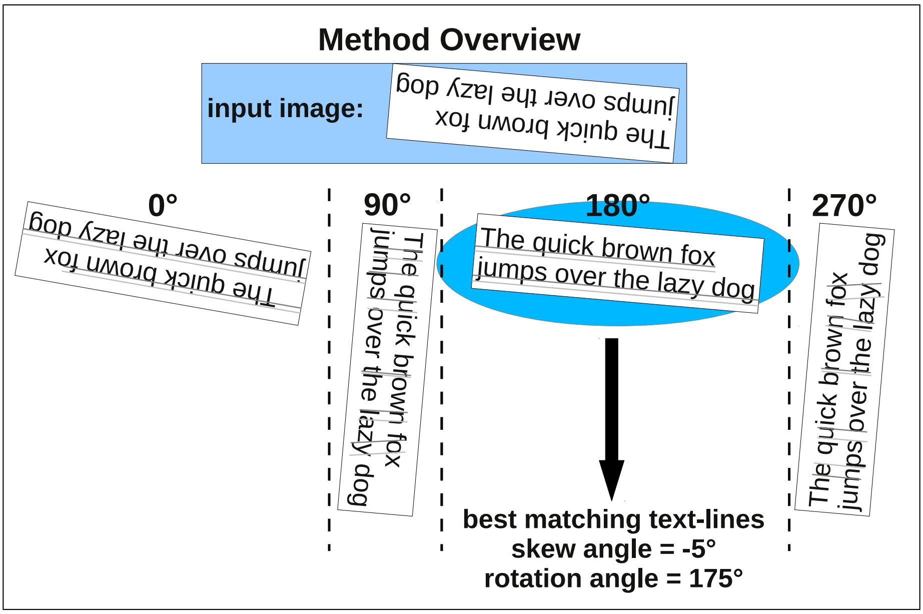
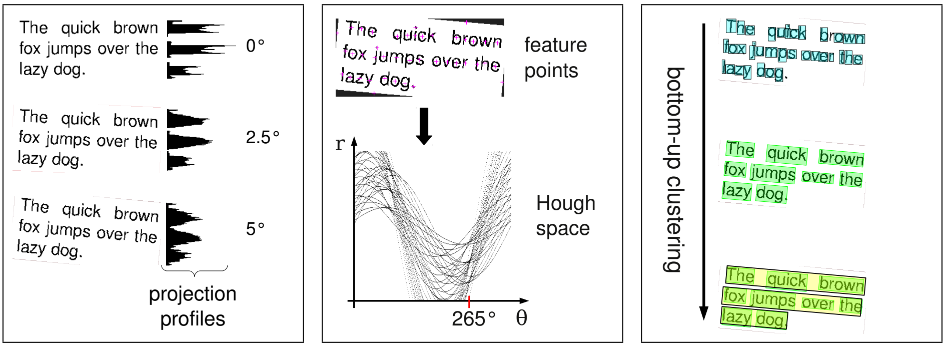


Figure 2.5: Overview of the orientation and skew detection approach. The orientation fitting best to the base line and descender line text-line model is searched. In this example the best fit is for the 180° orientation. Skew measurement from the text-lines returned −5°, thus the overall rotation angle of the input image is 175°.

Approaches using Hough Transform

In 2000, Amin et al. [5] presented a Hough transform based method. Using the Hough transform, an initial estimate of the skew angle is computed. Using the least squares method, a more accurate angle is computed. An average angle measurement error of 0.19° is reported. Considering the detected skew as correct is the difference is below 1°, an accuracy of 97.1% is achieved. A comparison with six other methods has been done. It could be shown that the proposed method works more accurate than the other six approaches.

A classical Hough transform based method was presented by Manjuntah [70] in 2006. After filtering the connected components using morphological operations, the Hough transform is computed for the remaining components. However, the proposed method does not work with text containing images.



(a) Projection profile (b) Hough transform (c) Clustering

Figure 2.6: Visualization of common skew detection methods. Figure 2.6(a) shows a the projection profiles for three different skew angles. Figure 2.6(b) shows the Hough space of the above feature points. Figure 2.6(c) shows an example of how a clustering method could cluster single components into words and lines.

Approaches using Clustering

Okun et al. [81, 82, 80] presented in 1999 a skew detection method for low resolution (50 dpi) document images. Using the first eigenvector of the covariance matrix of each connected component, an approximate skew angle is computed. This is used to group nearby components with similar skew angles to a line. The extracted lines are then used for estimating the skew angle of the entire page. An error rate of 1.2% on a 116′460 images sampled from the UW-I [87] dataset were reported. The authors defined an estimated skew angle as correct if the absolute difference to the ground truth skew angles is less than 1°. Comparison to two Hough transform based methods proved their method to outperform these methods. Considered skew angles in the test were $[0°, 179°]$.

In 2001, Das and Chanda [32] presented a method using morphological operations to group pixels to text-lines. The median of the obtained line angles is returned as the page skew angle. Evaluation on images from the UW-I and UW-II dataset and on privately collected images with different scripts showed reasonable results on small skew angles. Comparison to four methods from other authors has been done.

Lu [69] presented in 2003 an approach where connected components are merged with their nearest neighbor components to chains. The slope is estimated from each of these chains and finally the document skew is estimated using the mean or mode of the slopes. An evaluation on a non-public dataset showed reasonable results with a mean skew angle measurement error of 0.32°.

Methods based on minimum spanning trees [15, 46, 85] also belong to the class of clustering methods for skew detection. Variants of these approaches have also been used for skew detection on different scripts [8].

Other Approaches

In 2000, Safabakhsh et al. [93] proposed an approach using the rotation angle of the minimum area bounding rectangle of the connected components as an estimate for the document skew angle. A limited evaluation on a private dataset of 8 pages containing English, Arabic and Farsi text showed reasonable results. Skew angles in the range of $[-10°, +10°]$ were considered.

Antonacopoulos [6] presented a skew detection approach that is able to detect different skew angles for different text blocks. Page segmentation is first done by analyzing the white space that represents the image background. In short, page segmentation is the process of segmenting the document image into homogeneous areas, e.g. lines and paragraphs. After this step, the text zones are identified. In these text zones, the rectangles covering the background are analyzed to find rectangles forming a white separator line between two text-lines. The median skew angle of a set of lines inside a text block is defined as the skew angle of this block.

In 2004, Lins and Avila [66] presented a method that uses an analysis of the left-most points of a document image to detect the skew. However, the authors implicitly assume that the text is aligned to the left border of the image, which reduces its practical usefulness.

In 2005, Xi et al. [125] used a wavelet decomposition to find the skew angles of scanned forms. The main idea is to use the horizontal and vertical rulers and table separators present in forms for rotation angle estimation. After wavelet decomposition, two images are obtained presenting different responses for lines with different slopes. These images are binarized and the rotation angle of each image point is computed using central moments. A histogram over the obtained angles is used to estimate the skew angle. Evaluation on a non public dataset proved good results. Another approach examining the lines in form documents has been presented by Xie et al. [126].

2.2.2 Orientation Detection Methods

For orientation detection, several approaches have been presented in the past. They can be divided into two categories:

- **ascender to descender ratio methods:** these methods use the ratio of ascender characters like “b,d,f,h,k” to descender characters “g,j,p,q,y” to estimate the orientation of a document. As ascenders are more frequent in Latin script, a statistical analysis of the ratio is used to decide upon the orientation.

- **recognition driven approaches:** these approaches recognize single characters or at least simple features of characters to determine the orientation for which the characters are most likely to be known characters.

A method using the ascender to descender ratio was presented by Caprari [24] in 1999. 0° and 180° orientations are detected using an asymmetry measure computed from projection profiles, which makes it sensible to document skew. An accuracy of 100% is reported on a non-public dataset of 226 images containing mainly text and only little skew.

Aradhye [9] presented in 2005 an up/down orientation detection method for Roman and Non-Roman scripts. Text blobs are extracted and the so called opening to the left and the opening to the right is computed for each blob. The ratio between left and right opening is used to decide whether the page is rotated 0° or 180°. The method was tested on different datasets and reached accuracies from 87% to 100%. On the UW-I dataset an accuracy of 99% was obtained.

In 2006 Lu et al. [67] presented a method for language and orientation detection using the distributions of the number and position of white to black transitions of the components in the line. The performance of their method is reported on a partially non-public dataset and achieves an success rate of 98.2% for documents with at least 12 text-lines.

Rangoni et al. [91] presented a recognition driven approach for orientation detection. Using text-line level character recognition, for each of the four possible orientations the extracted words are compared to a dictionary. The orientation returning the most words that can be found in the dictionary is considered to be the correct one.

A similar approach has also been followed by van Beusekom et al. [113]: instead of doing a full recognition step, however, only the similarity of the characters from the document is compared to characters that are correctly oriented for all four orientations. The orientation for which the similarity is highest is then considered to be the correct orientation.

2.2.3 Combined Skew and Orientation Detection Methods

Methods solving skew and orientation detection together can also be found in the literature. Most often, however, not a single integrated approach is used but two different approaches, one for skew detection and one for orientation detection.

In 1994 Le et al. [58] presented a system capable of detecting of portrait or landscape mode images and subsequently the skew angle. 180° rotated document images are not considered. The approach uses rules based on projection profiles and textual features for orientation detection. The Hough transform is used in the second step to determine the skew of the page. Evaluation has been done on a non-public dataset of pages of medical journal articles. An error rate of 0.1% is reported.

In 1995, Bloomberg et al. [14] presented a method for orientation detection that uses the ratio of the number of ascender characters to the number of descender characters used in the English language. As ascenders are more frequent than descenders, this is used to detect the correct orientation. Ascenders and descenders are extracted using morphological operations. Skew detection is also provided. This is done by finding the maximum of an objective function, the so called "skew function". Using two different search strategies, the best angle is computed. The reported error rate for orientation detection on the UW-I [87] dataset is one wrongly detected orientation versus 938 correctly detected. The orientation of 41 documents could not be extracted but are not considered as errors. Skew detection error histograms are also given for the UW-I dataset. An implementation of this method is available in Leptonica[4], an open source image processing library.

A more recent method using the ascender to descender ratio for orientation detection is presented by Avila et al. [11]. Using the x-height line and the base line of the text-lines, the number of ascenders and descenders is computed. If the number of descenders is higher than the number of ascenders, the page is considered to be rotated by 180°. Skew detection is done by grouping connected components to lines and building an angle histogram of the lines. The reported error rate for orientation detection is below 0.1% on a non-public dataset.

Lu [68] et al. presented in 2007 a method to solve the orientation detection and skew detection problem fast and accurately. Similar to their previous work, characteristic peaks are detected in the white run histograms. This is used to estimate the skew angle. Orientation detection is typically solved using ascender to descender ratio. A comparison between their method and other methods found in the literature is done. Unfortunately, the test set consists of only 52 documents from a non-public dataset.

2.3 Description of the Approach

Considering the previously published methods presented in Section 2.2.3, it should be noted, that none of these methods is really an integrated approach to solve skew and orientation detection simultaneously. Also, previously published skew detection methods either depend on the angular resolution of the detection or on heuristics for grouping components. All these disadvantages can be overcome by the proposed method. First of all it uses a single method to detect skew and orientation. Second, it extracts globally optimal text-lines, not depending on any clustering heuristic. Third, no angular resolution has to be fixed. The

---

[4] http://www.leptonica.com

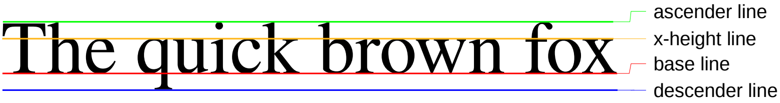


Figure 2.7: The anatomy of a line. It shows the four different lines that are being distinguished: the descender line, the base line, the x-height line and the ascender line.

angular resolution obtained by the method is only limited by the scan resolution. Moreover, the proposed method allows integration into existing systems at virtually no cost, as most of the line information that is needed to detect the skew and orientation is needed anyway in the subsequent page segmentation and optical character recognition processes.

To detect the skew angle of a document image, linear structures are to be found. In document images, the most frequent linear structures are text-lines. Therefore, the proposed approach uses text-lines to detect the skew. Modeling text-lines by just one geometric line does not allow to detect the orientation of a text-line. A more flexible model is the model presented in Section 2.3.1. It explicitly models the line of descenders. According to the observation [14], that for Latin script, characters with ascenders are more likely than characters with descenders, this model can be used to identify the orientation of a text-line.

The next three sections describe the text-line extraction. First, the text-line model is explained in Section 2.3.1. In the subsequent Section 2.3.2 the quality function for a text-line is explained. Section 2.3.3 briefly explains the branch-and-bound algorithm that is used to extract the text-lines using the previously mentioned quality function.

Then, in Section 2.3.4 the proposed approach that uses text-line extraction for orientation and skew detection is described. Finally, in Section 2.3.5 an extension of the original method is proposed that considerably improves the performance in terms of speed.

2.3.1 Geometric Text-Line Model

The text-line model proposed by Breuel [18] uses three parameters (r, θ, d), where r is the y-intercept, θ is the angle of the baseline from the horizontal axis, and d is the distance of the line of descenders from the baseline. A visualization of the ascender, descender, x-height and base line is shown in Figure 2.7. An illustration of this text-line model can be found in Figure 2.8. The advantage of explicitly modeling the line of descenders is that it removes the ambiguities in baseline detection caused by the presence of descenders.

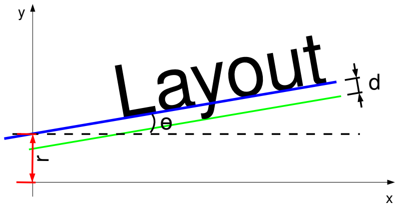


Figure 2.8: An illustration of the Roman script text-line model proposed by Breuel [18]. The baseline is modeled as a straight line with parameters (r, θ), and the descender line is modeled as a line parallel to the baseline at a distance d below the baseline.

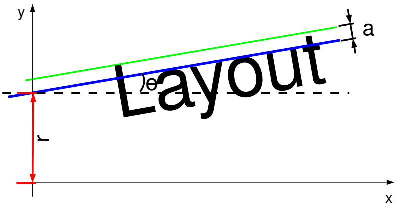


Figure 2.9: An illustration of the text-line ascender model. The x-height line is modeled as a straight line with parameters (r, θ), and the ascender line is modeled as a line parallel to the baseline at a distance a above the x-height line.

2.3.2 Quality Function for Text-Line Extraction

In the following, the quality function for a single text-line is explained. Later, in Section 2.3.3 it is explained how this is used to extract all the text-lines from a document image.

Based on this geometric model of Latin script text-lines, geometric matching is used to extract text-lines from scanned documents as in [18]. A quality function is defined which measures the quality of matching the text-line model to a given set of points. The goal is to find a collection of parameters (r, θ, d) for each text-line in the document image that maximizes the number of bounding boxes matching the model and that minimizes the distance of each reference point from the baseline in a robust least square sense. The RAST algorithm [17, 20] is used to find the parameters of all text-lines in a document image. The algorithm is run in a greedy fashion such that it returns text-lines in decreasing order of quality.

Consider a set of reference points $\{x_1, x_2, \cdots, x_n\}$ obtained by taking the middle of the bottom line of the bounding boxes of the connected components in a document image (see Figure 2.10). The goal of text-line detection is to find the maximizing set of parameters $\vartheta = (r, \theta, d)$ with respect to the reference points $\{x_1, x_2, \cdots, x_n\}$:

$$\hat{\vartheta} := \arg\max_{\vartheta} Q_{x_1^n}(\vartheta) \tag{2.1}$$

$$Q_{x_1^n}(\vartheta) = Q_{x_1^n}(r, \theta, d) = \sum_{i=1}^{n} \max(q_{(r,\theta)}(x_i), \alpha \cdot q_{(r-d,\theta)}(x_i)) \tag{2.2}$$

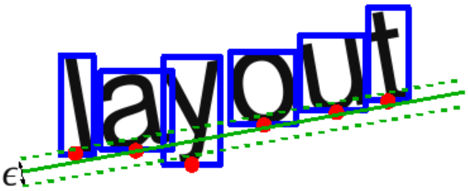


Figure 2.10: An illustration of the image points used for text-line extraction. The center point of the bottom line of the bounding rectangle of the connected component, marked with a red dot, is used as reference point for the line finding. The parameter ϵ, defining a margin within which points are considered to contribute to the line is visualized by the dotted lines: every reference point in between the dotted lines contributes to the line.

$$q_{(r,\theta)}(x) = \max\big(0, 1 - \frac{d^2_{(r,\theta)}(x)}{\epsilon^2}\big) \tag{2.3}$$

The maximum of the sum in Equation 2.2 consists of two terms: the first term computes the contribution of a point x_i to the base line, the second term computes the contribution of a point to the descender line. The maximum of both is taken, as one point may belong to either the base line or the descender line, but not to both. The factor α is used to account for the different priors of a point belonging to the base line or the descender line. Setting $\alpha = 1$ would give the same importance to points on the descender line as on the ascender line. Setting $\alpha = 0$ would ignore the points lying on the descender line.

The values of Equation 2.3 lie in the interval $[0, 1]$. For all references points for which $d_{(r,\theta)}(x) \geq \epsilon$, the contribution $q_{(r,\theta)}(x)$ is zero. These points are considered not belonging to the line (r, θ) as they are too far away from it.

Depending on the application, ϵ may be set to different values. In order to cope with noise and binarization artifacts, $\epsilon = 5$ is a good choice for the task of text-line extraction on images scanned with a resolution of 150 dpi to 400 dpi. Setting $\epsilon = 0$ leads to the effect that only points lying exactly on the line contribute to the quality measure.

2.3.3 Text-Line Extraction

The RAST algorithm is used to extract the text-line with maximum quality as given by Equation 2.1. RAST is a branch-and-bound algorithm that recursively splits the parameter space into smaller subspaces and first analyzes to most promising subspaces. This is done using a priority queue that keeps track of all subspaces that need to be analyzed. When a solution is found, all reference points that contributed with a non-zero quality to the extracted text-line are removed from the list of reference points and the algorithm continues its search. In this way, the algorithm returns text-lines in decreasing order of quality until all text-lines have been extracted from the document image.

As mentioned in Section 2.3.1, the text-lines are defined by three parameters (r, θ, d). The text-line extraction consists in finding parameter triples that maximize Equation 2.2. This is done using a branch-and-bound search. In the first step, the parameter space P is defined:

$$P = [r_{min}, r_{max}] \times [\theta_{min}, \theta_{max}] \times [d_{min}, d_{max}] \tag{2.4}$$

For practical applications, $r_{min} = 0$ and r_{max} are fixed using the image size. As large skew angles $|\theta| > 20°$ are unlikely to occur (in the afore mentioned applications, such skew angles will likely lead to a paper feed error on the scanning device), the range of the rotation angle $[\theta_{min}, \theta_{max}]$ is set to $[-20°, 20°]$. Finally the possible distances of the descender line to the base line has to be set. For the current setup this is fixed to $[0, 30]$. The theoretical size of a 12 pt character scanned with 300 dpi is 50 px. For common fonts the distance between the ascender and the baseline is about 1/4 to 1/3 of the total height, thus the above setting of the parameter d gives enough flexibility.

The branch and bound algorithm works as follows:

0: initialize the search space and insert it into priority queue Q

1: stop if Q is empty, else get the top search space S from Q

2: if S is small enough: save S and discard all image points contributing to the line from further consideration; Stop if enough results are found. Otherwise, continue with Step 1.

3: split S into two subspaces S_1 and S_2

4: compute upper bounds for the quality of S_1 and S_2

5: sort S_1 and S_2 into Q. Continue with Step 1.

The computation of the upper bound for the quality of the text-line defined by the parameter subspace is done using interval arithmetic. The upper bound can be seen as the maximum number of points that are covered by all the lines defined by the subspace. If not all the text-lines need to be extracted, the algorithm is stopped after extraction of n text-lines. A space is considered as a solution if it is small enough to identify the unique line in the image. The parameters of the solution represent the detected line.

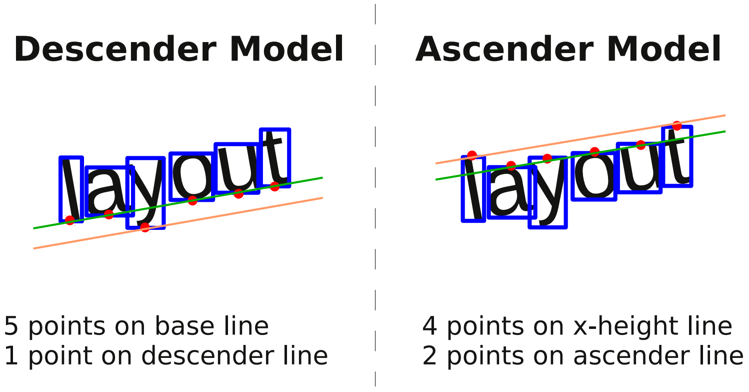

Descender Model
Ascender Model
layout
layout
5 points on base line
1 point on descender line
4 points on x-height line
2 points on ascender line

Figure 2.11: An illustration of the orientation detection using text-lines. Using the descender model, more points lie on the base line compared with the text-line extracted using the ascender model. Thus the quality of the last one will be lower than the quality of the first one.

As mentioned in Section 2.3.3, the set of reference points $\{x_1, x_2, \cdots, x_n\}$ is obtained by taking the middle of the bottom line of the bounding boxes of the connected components in a document image. Instead of taking all connected components, a filtering step is used to discard too small or too big components, in order to increase the robustness of the method. The modes of the width and height of the connected components are computed. Based on these, $min_{area} = 2 \times y_{mode}$, $max_{height} = 10 \times y_{mode}$, $min_{height} = 0.5 \times y_{mode}$, $max_{width} = 10 \times x_{mode}$, $min_{width} = 0.5 \times x_{mode}$ and $max_{aspect} = 10.0$ are defined and used as thresholds for the width, height, aspect-ratio and area of the connected components. To simplify reading, in the following the filtered set of connected components is referred to simply as *the connected components*.

2.3.4 One-Step Skew and Orientation Detection

The key idea of the approach is to use ascender modeling in the same way as modeling descenders. Now, if both text-lines are extracted, the text-line for the ascender as well as for the descender model, the descender model will have a higher quality as more points are lying on the base line, as fewer descenders are expected than ascenders. An intuitive example is given in Figure 2.11.

An illustration of the ascender text-line model is shown in Figure 2.9. The x-height line (the line passing through the top of non-ascending lower case characters like “x”, “a”, “c”, etc.) is modeled as a straight line with parameters (r, θ), and the ascender line is modeled as a line parallel to the x-height line at a distance a above the x-height line.

Consider a set of reference points $\{y_1, y_2, \cdots, y_n\}$ obtained by taking the middle of the top line of the bounding boxes of the connected components in a document image. The goal of text-line detection is to find the maximizing set of parameters $\vartheta = (r, \theta, a)$ with respect to the reference points $\{y_1, y_2, \cdots, y_n\}$:

$$\hat{\vartheta} := \arg\max_{\vartheta} Q_{a,y_1^n}(\vartheta) \tag{2.5}$$

The quality function can then be defined as:

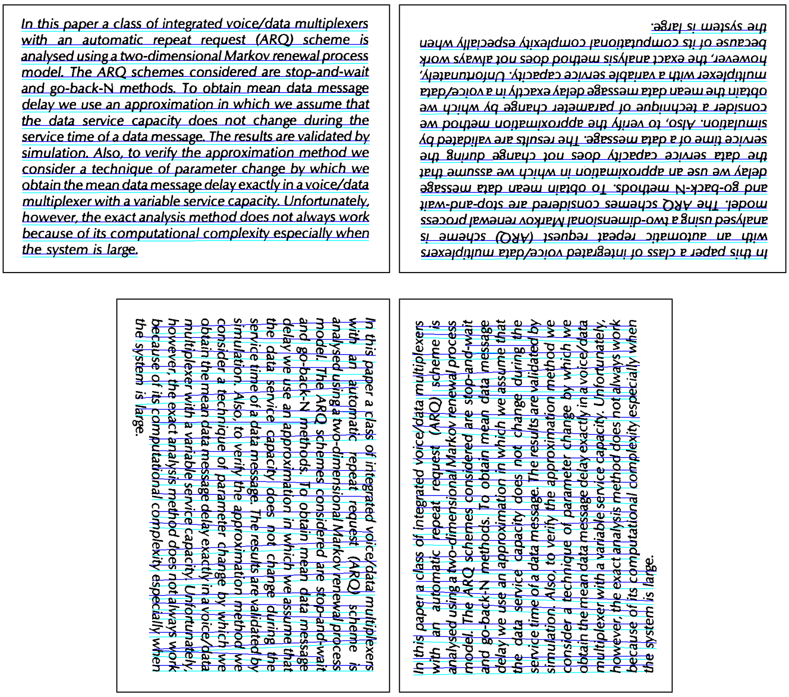
$$\begin{aligned} Q_{a,y_1^n}(\vartheta) &= Q_{a,y_1^n}(r, \theta, a) \\ &= \sum_{i=1}^{n} \max(q_{(r,\theta)}(y_i), \alpha \cdot q_{(r+a,\theta)}(y_i)) \end{aligned} \tag{2.6}$$

where the local quality is computed in the same way as Equation 2.3.

Since in Latin script ascenders are more likely to occur than descenders, more components will match to the ascender line than to the descender line. A component matching to ascender line achieves a lower score (due to the factor α in Equations 2.2 and 2.6) as compared to a component matching to baseline/x-line. Therefore, in general the total quality of the descender line (Equation 2.2) will be higher than the total quality of the ascender line (Equation 2.6). This information is used in this work to find the upside down orientation of the page. The qualities of n best lines returned by RAST first using the descender model and using the ascender model are summed up. If the quality of the ascender model is higher than the descender model, the page is reported as upside down (180° rotated).

Note that computing the ascender model for a given page image in a correct orientation is equivalent to computing the descender model for a 180 degree rotated page. Therefore, for any given image only the descender model for the original image and for 180 degree rotated image is computed. The image that results in better descender quality is reported as the one with the correct orientation. This concept is then easily extended to detected pages with a 90° and 270° orientation. The horizontal text-line model does not fit well on vertical text-lines, so for a right side up portrait page the total quality of n best lines in the vertical direction is much lower than the total quality of n best horizontal lines. Hence by computing the descender quality by rotating the page with all four orientation, one can find out the correct orientation of the page. An illustration is shown in Figure 2.12.

An interesting aspect of the proposed method is that besides an estimate of page orientation, estimates for page skew θ are obtained automatically of all detected text-lines at this stage. The skew angle of the text-line with the highest quality is chosen as the global skew of the page which has already shown to give accurate results for skew detection [18]. Another, even more interesting aspect of the method



In this paper a class of integrated voice/data multiplexers with an automatic repeat request (ARQ) scheme is analysed using a two-dimensional Markov renewal process model. The ARQ schemes considered are stop-and-wait and go-back-N methods. To obtain mean data message delay we use an approximation in which we assume that the data service capacity does not change during the service time of a data message. The results are validated by simulation. Also, to verify the approximation method we consider a technique of parameter change by which we obtain the mean data message delay exactly in a voice/data multiplexer with a variable service capacity. Unfortunately, however, the exact analysis method does not always work because of its computational complexity especially when the system is large.

Figure 2.12: An example figure illustrating the results of fitting descender line model on all four orientations of the image. The blue line represents the baseline, whereas the cyan line represents the descender line. Note that the model fits well on only two of the four orientations. Between these two the one in the correct orientations achieves a higher quality due to a fewer number of descenders.

in the context of a subsequent OCR system, is the line information that is already computed. This information can then be used to extract the lines and feed them to a line recognizer.

2.3.5 Run-Time Improvement

In a high volume digitization setup, computation time is an important aspect. Although the computation time needed by the text-line extraction process is justifiable in the context of page segmentation, running the extraction 4 times might be too slow if one is only interested in the orientation of the page. Furthermore, the line extraction converges only slowly on pages that having only few aligned components, as for 90° and 270° misoriented pages. This computational overhead may render the method impractical.

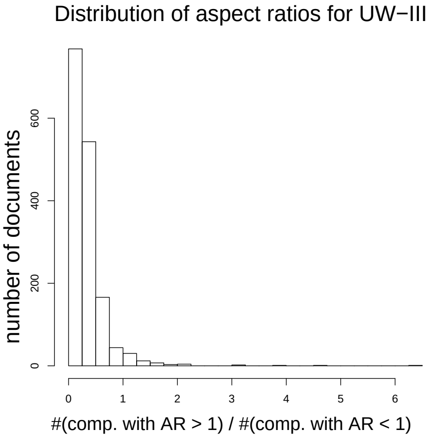


Figure 2.13: Histogram of the aspect ratio of connected components of UW-III documents. For each document the number of connected components with aspect ratio $<$ 1 was divided by the number of component es with aspect ratio $>$ 1.

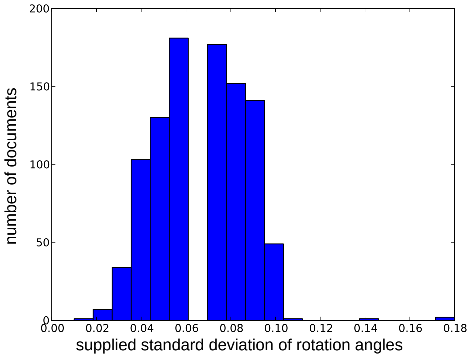


Figure 2.14: Histogram of the distribution of standard deviation of skew angles within one page in UW-I dataset.

Therefore, in order to speed up the proposed approach, a fast pre-classification is done on the basis of the aspect ratio of the bounding boxes of the connected components. Measurements on correctly aligned document images showed that the ratio of the number of bounding boxes with an aspect ratio $ar > 1$ to the number of bounding boxes with an aspect ratio $ar < 1$ is about $3.5 : 1$. This is used to discard two possible orientations from the process. The distribution of this ratio is shown in Figure 2.13. It can be seen that a strict decision at the boundary of $ar = 1$ leads to misclassifications. To avoid this, a margin has been defined for which all four orientations are analyzed. According to the measurements in Figure 2.13 this was set to $0.66 < ar < 1.5$.

2.4 Evaluation

The evaluation of the proposed method is divided into two main parts: First, the ascender to descender ratio is measured for a number of languages using Latin script to support the claim made by Bloomberg [14]. Second, the skew estimation accuracy is evaluated. The last part of the evaluation is dedicated to measure the performance on orientation detection.

The dataset used for the ascender to descender ratios for different languages is the Project Gutenberg E-Book DVD[5]. Licence notices in English language at the end of the text files have been removed to avoid falsification of the results.

The training dataset, used for adapting the method and the parameters, is composed of 159 images of the UW-III [87] dataset. The total size of the UW-III dataset is 1600 images. Every 10th image (in alphabetical order) from the total dataset was included. Hence the training set consists of images A00A, A00K, ..., W1UA. Image W0H4 was manually removed as it contains text in two different orientations. In order to have equal distribution over all orientations, the first image in alphabetical order was left unchanged, the second was rotated by 90°, the third one by 180°, etc.

The test set on which the main evaluation of the system's performance was done on is the UW-I dataset. The UW-I dataset consists of 979 binarized images containing scans of scientific journals in English language.

For skew detection, the accuracy is defined as the difference between the estimated skew angle and the skew angle from ground truth. For the orientation detection evaluation, the accuracy is defined as the number of correctly detected page orientations divided by the total number of analyzed document images. The decision whether an orientation is correct or not is made using the ground truth information.

The accuracy is defined as the number of correctly detected page orientations divided by the total number of analyzed document images. The decision whether an orientation is correct or not is made using the ground truth information.

The accuracy of the skew estimation is analyzed on the UW-I dataset. in Section 2.4.2. Evaluation of the orientation detection is described in Section 2.4.3.

2.4.1 Ascender to Descender Ratios for Latin Script Languages

Bloomberg's [14] measurement on the frequency of ascenders to descenders in English text obtained a ratio of three to one. This is sustained by the measurements for several different languages that can be found in Table 2.4.1. It can thus be concluded that the proposed method will work on most languages using Latin Script. The Project Gutenberg DVD dataset was used for these measurements.

2.4.2 Skew Detection Accuracy

In this experiment, the accuracy of the skew detection capability of the proposed method on the UW-I dataset is evaluated. For this purpose the slope/skew information of the best line (line with the highest quality defined by Equation 2.2)

---

[5] http://www.gutenberg.org/wiki/Gutenberg:The_CD_and_DVD_Project

Language	Total (in million)	Ascenders	Descenders	Ratio
English	2150	19.9%	7.2%	2.8
French	256	12.5%	7.1%	1.7
Finnish	64	14.9%	6.9%	2.1
Dutch	45	17.4%	7.9%	2.2
German	41	20.9%	5.2%	4.0
Spanish	30	14.5%	7.6%	1.9
Italian	11	13.5%	7.0%	1.9
Swedish	8	19.7%	8.4%	2.3
Portuguese	8	12.6%	7.0%	1.8

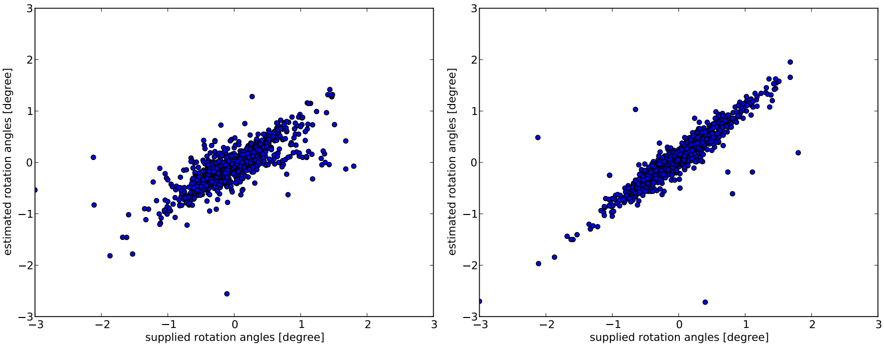
Table 2.1: Measurements of ascender to descender ratios for different languages on the Project Gutenberg dataset. The first column gives the analyzed language, the second the total number of analyzed characters. The third and fourth column contain the percentage of ascender and descender characters, and the last column the ascender to descender ratio.

only is chosen. Other possibilities, like the median, mean, weighted median, or weighted mean, of all the different rotations of lines have been reported to give similar results [18]. Using only one line also makes the skew detection algorithm fast.

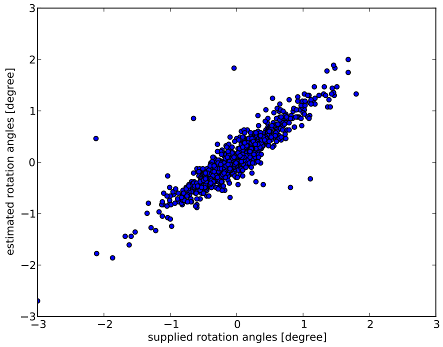
Results

Figure 2.15 shows a scatter plot of the detected rotation angles and the ground-truth rotation angles as supplied with the UW-I dataset. For the sake of comparison, the skew detection accuracy of the Tesseract [107] and the Leptonica [14] document analysis systems was measured as well. It is clear from the plot that the proposed method is able to accurately find the skew angles of most document images. Some outliers are due to pages having only formulas or images in them such that no reliable text-line could be found.

Page rotations in the UW-I dataset were estimated using a triangulation scheme on multiple sets of three points on each document page [26]. Since multiple sets of points were used there is an associated standard deviation for each estimated page rotation angle. Figure 2.14 shows a plot of the histogram of standard deviations of skew angles within one page in the UW-I dataset. Hence there is no single page rotation angle that can be called “correct” for these pages. Also, for many pages, there is a significant difference in the skew of lines that are near the top of the page and the lines that are near the page bottom. This means that the variance between estimated and ground-truth page rotations may simply be due to the intrinsic variability of baseline orientations on these pages. If a difference of less than 0.5 degrees between the estimated and the supplied rotation angle



(a) Tesseract (b) Leptonica



(c) Proposed Method

Figure 2.15: Scatter plot of supplied rotation angles of the UW-I dataset and skew angles estimated by (a) Tesseract open source OCR system, (b) Leptonica open source image processing library, and (c) method presented in this paper.

as correct is considered as correct, rotation angles of 960 out of 979 documents are correctly identified resulting in an accuracy of 98%. Using the same criteria, Leptonica correctly identified skew of 966 (98.7%) documents, whereas Tesseract identified skew of 895 (91.4%) documents correctly.

2.4.3 Orientation Detection Accuracy

Three different tests have been run to evaluate the orientation detecting performance:

1. The overall accuracy and processing speed of the full system is analyzed, thus using four line extraction steps. The main parameter influencing the

processing time is the maximum number of extracted text-lines. The effect of the number of extracted lines on the accuracy and the running time has been analyzed.

2. The accuracy and processing speed of the system using the preliminary classification on the aspect ratios of the connected components is measured.
3. A quantitative performance comparison of the proposed approach to other methods is done.

Orientation Detection Using Four Text-Line Extraction Steps

In this evaluation, the performance of the orientation and skew detection method that analyzes the text-lines for all four orientations is presented. The test was run on the UW-I dataset.

For each image, the output of the orientation detection system was compared with the ground truth. The following test parameter set for the maximum number of extracted lines is used: $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$

The plot in Figure 2.16 shows the computation time needed with respect to the number of extracted text-lines. It should be noted, that the processing time increases with the number of text-lines to be extracted.

In the plot in Figure 2.17 the accuracy of the method in relation with the number of extracted lines is shown. It can be seen, that the accuracy for very few lines is already quite high, that it then reaches a maximum around 32 extracted lines and then decreases slightly to finally stay at a constant level. The absolute difference in correctly detected orientations between extracting 32 and 512 lines is only three documents. A manual inspection of different failures showed, that extracting more lines on documents with only few text-lines can lead to misdetection of the orientation. Examples can be found in Figures 2.18(a) and 2.18(b). It can also be noted that for some documents extracting more lines is advantageous. The orientation of image *A00M* (see Figure 2.18(c)), e.g. is recognized correctly when 512 lines are extracted but not if only 32 lines are extracted. In that case a rotation angle of 90° is detected, due to the long nicely structured tables.

Finally, the choice of $\epsilon = 5$, which was motivated in Section 2.3.1, is validated empirically. In Figure 2.19, the accuracy versus different values of ϵ is shown. The tested values are: $\epsilon \in \{0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30\}$. It can be seen that for values of $\epsilon \in [2, 20]$, the performance stays relatively stable at a high value. For values of $\epsilon > 20$, the performance drops dramatically.

Orientation Detection Using Two Text-Line Extraction Steps

In this experiment, the performance of the orientation and skew detection method using the pre-classification on the basis of the aspect ratio of the connected compo-

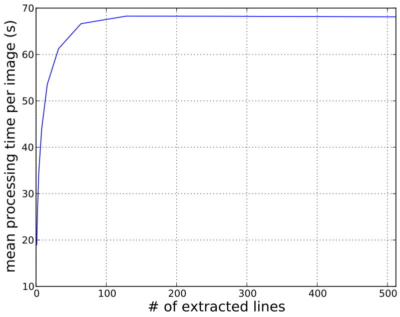


Figure 2.16: Plot of the processing time vs. the number of extracted lines for the method using text-line extraction for all four orientations. The more lines are extracted, the more time is needed. As the number of text-lines on a document page cannot be arbitrarily high, a saturation effect can be noticed.

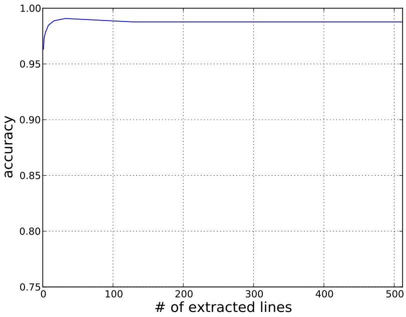


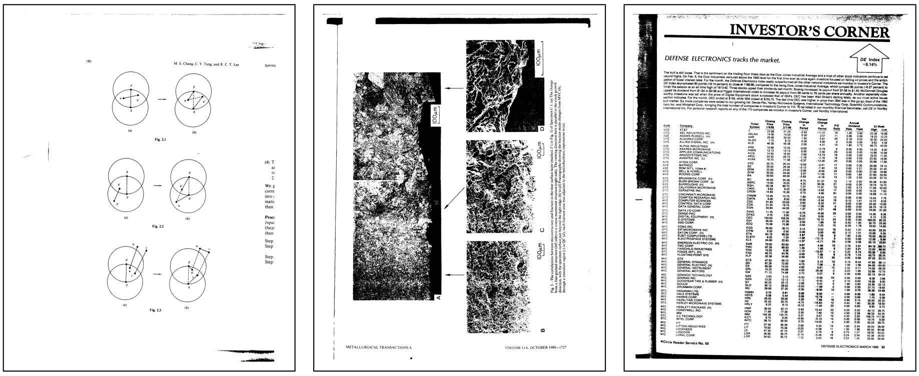
Figure 2.17: Plot of the accuracy vs. the number of extracted lines for the method using text-line extraction for all four orientations. For very few extracted text-lines the accuracy is slightly lower but staying at a high level for four or more extracted text-lines.

nents 2.3.5 is analyzed on the UW-I dataset. The following test parameter set for the maximum number of extracted lines is used: $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$.

The plot in Figure 2.21 shows the accuracy of the method in relation to the number of extracted lines. It can be observed that there is a peak for a moderate number of extracted lines, in this case 16. Then the accuracy drops slightly but stays stable afterwards. A manual inspection was done to find out how many documents were wrongly classified on the aspect ratio of the connected components: it showed that only 7 images were misclassified on the basis of their aspect ratios of the connected components. Examples can be found in Figure 2.20. It should be noticed, that all images show areas where due to binarization many small connected components are present. These are likely to disturb the statistics of the total page.

The plot in Figure 2.22 shows the time consumption of the method in relation with the number of extracted text-lines. As expected, extracting less text-lines speeds up the method. This information, in combination with the above mentioned accuracy considerations allow the user to specify the parameters according to his needs: if only orientation and skew detection are needed, only few lines have to be extracted. If the full text-lines are needed anyway for subsequent processing steps, then all lines can be extracted with nearly the same overall performance.

Comparing the performance of the full method to the method using fast pre-classification shows that the gain in running time is significant, around the factor of seven. The accuracy, however, drops only slightly, approximately about 0.7%.



(a) H00L (b) H04I (c) A00M

Figure 2.18: Examples of images that lead to errors depending on the number of text-lines extracted. Orientations of images *H00L* and *H04I* are correctly recognized when extracting only 32 lines, but not if at maximum 512 lines are extracted. Image *A00M* is recognized correctly for 512 extracted lines but not for 32 extracted lines.

Performance Comparison on UW-I

In this part, the proposed method is compared with a previously published method by Bloomberg [14]. For this evaluation, the test setup used by Bloomberg was followed, which is the same as in the two previous tests: the document images from the UW-I dataset are fed to the orientation detection system and the detected orientation is compared with the ground truth orientation. Examples of resulting images of the one-step skew and orientation detection can be found in Figure 2.23. The proposed method was set to extract at maximum 512 lines, which for normal documents should cover all lines.

Table 2.4.3 shows the results per orientation and a comparison to Bloomberg's method. It shows that although Bloomberg's method works quite well, the proposed approach achieves even better results. From 979 images in the data set only 12 could not be classified correctly.

A manual verification of the errors showed the following problems:

- Upper case letters only: in image *N02J* and *N03G* no lower case letters were present, leading to a false classification as 180° rotated images. So the ascender to descender ratio can not be used. For the same reason *D067* has been classified as 270° instead of 90°.

- Long columns: in image *A00M* the main part of the page consists of long lists of numbers in a table. This lead to the misclassification to 90° orientation although it is a 0° rotated page.

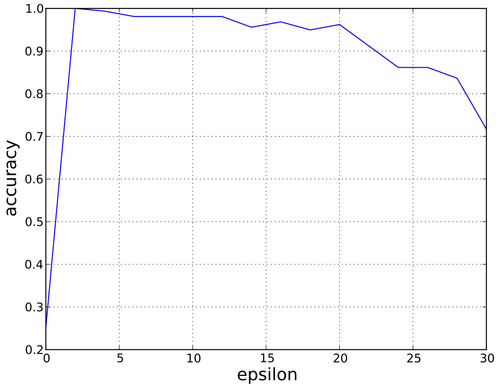


Figure 2.19: Plot of the accuracy vs. the parameter ϵ for the method using text-line extraction for all four orientations. Small values for ϵ give bad accuracy rates as the overall quality of the text-lines will be low. For reasonable values of ϵ the accuracy stay high, while dropping for high values.

- Mathematical symbols: pages consisting mainly of mathematical symbols are also hard to be oriented correctly as line finding will not find many good lines. This holds for *A002* (classified as 270° instead of 90°) and *H00Y* (classified as 180° instead of 0°).

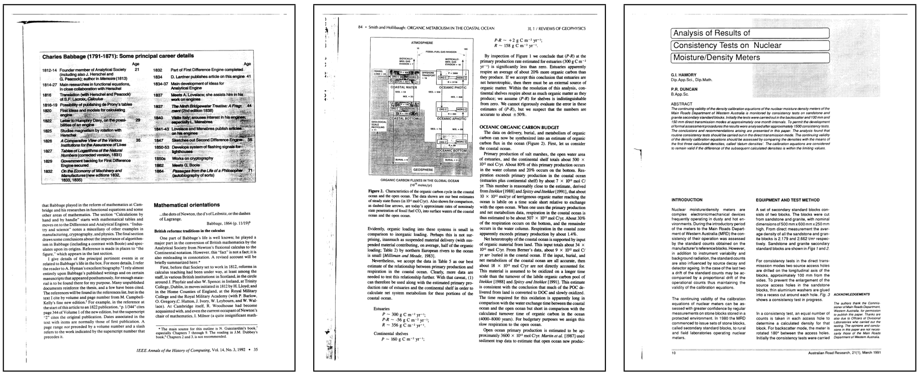
Examples of these failures are shown in Figure 2.24. A performance comparison on single pages showed that Bloomberg's method is about twice as fast as the presented method using pre-classification and only one extracted line. The comparison, however, is not too significant: on the one hand, different code bases are used. Especially for Leptonica, the code base has been well optimized. On the other hand, the nature and also the resulting information (orientation versus orientation, skew angle and line extraction) obtained by both methods widely differ. So if a subsequent OCR step is done, the text-lines have to be extracted anyway and then the presented approach has only little computation overhead.

Resolution Independence

In another setup, the resolution independence was tested. A set of images that is available with OCRopus [23] open source OCR[6] is used. These images are synthesized from electronic documents using the following resolution settings: 150, 200, 300 and 400 dpi. Nine different images are present in four different resolutions.

---

[6] http://ocropus.googlecode.com/files/ocropus-0.2-test-images.tar.gz



(a) C005 (b) E029 (c) V004

Figure 2.20: Example images whose orientations are misclassified on the basis of the aspect ratios of the connected components. It should be noted that all images show areas where many small connected components are present.

For each image four different orientations were tested, leading to a total test set size of 144 images. These images were chosen as they can be freely obtained on the web, so that they can be used for comparison by other researchers.

The results for the second test showed a 100% success rate on the 144 test images. This good result is favored by the test images containing no disturbing elements like images, numbers or mathematical symbols.

The same test was run for Bloomberg's method. The results are given in Table 2.4.3. It shows that on high resolutions (400 dpi) it is not able to return high confidences for the obtained orientation estimates. Inspection of the intermediate images of Bloomberg's method shows problems for the dilation using the horizontal structuring element for pages with 90° and 270° rotation. This is likely due to the too small horizontal structuring element.

2.5 Conclusion

In this chapter an integrated approach for skew and orientation detection using a line extraction algorithm was presented. The effectiveness of the method has been demonstrated on a publicly available dataset. A comparison to a widely used open source orientation detection technique [14] was done. Experiments showed that the proposed method outperformed the analyzed method for both UW-I dataset and a dataset having documents rendered at different resolutions. A particular advantage of this method is that it can compute both orientation and skew estimates in one

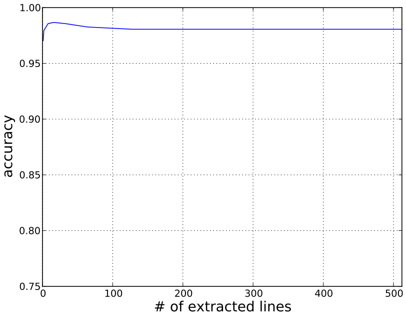


Figure 2.21: Plot of the accuracy vs. the number of extracted lines for the method using pre-classification on the bases of aspect ratio of the connected component. Even with only a few extracted lines a high accuracy is obtained.

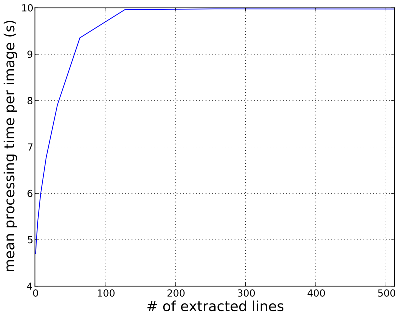
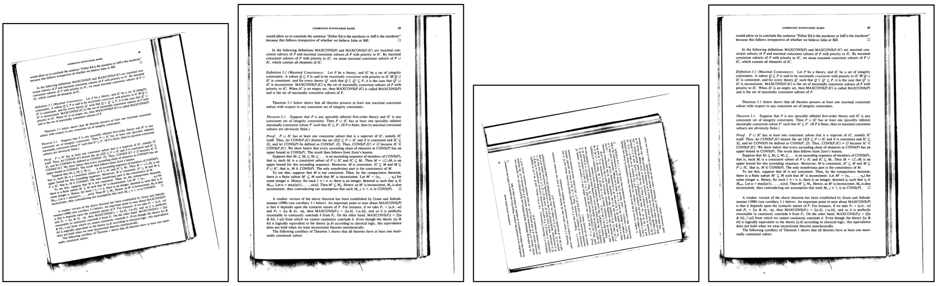
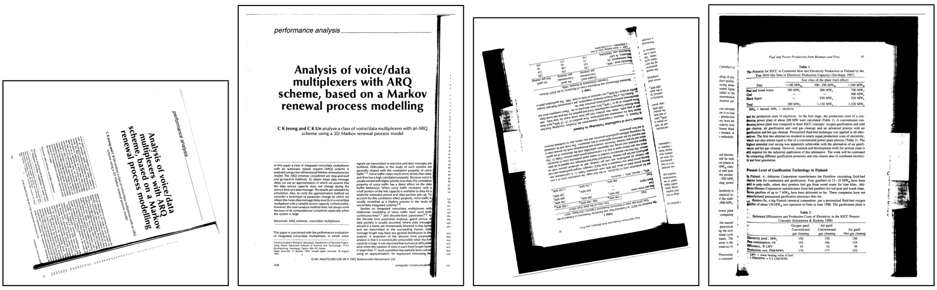


Figure 2.22: Plot of the processing time vs. the number of extracted lines for the method using pre-classification on the bases of aspect ratio of the connected component. Extracting more text-lines increases the processing time.

step. Furthermore, is optical character recognition is a subsequent step, the already computed text-line information can be reused.

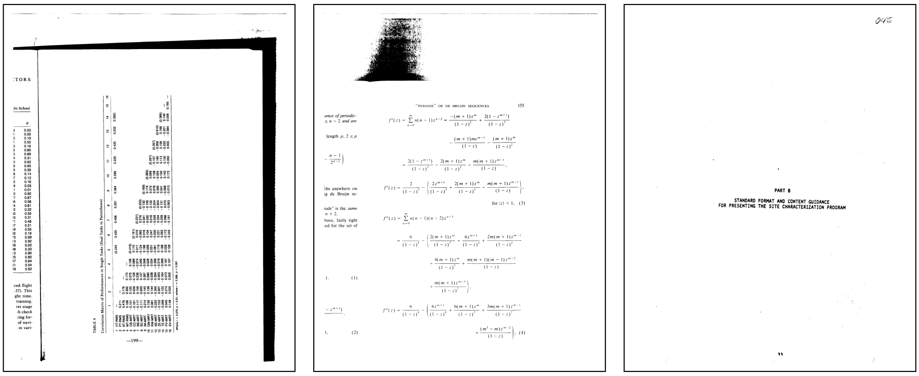


(a) A004 rotated 10° (b) A004 deskewed (c) A004 rotated 100° (d) A004 deskewed



(e) A005 rotated 75° (f) A005 deskewed (g) D04G rotated 190° (h) D04G deskewed

Figure 2.23: Examples for the output of the orientation and skew detection algorithm.



(a) D067 (b) H00Y (c) N02J

Figure 2.24: Examples of failures. Image *D067* shows a page being disoriented by the proposed method due to the many lines consisting only of numbers. Image *H00Y* shows another page where the method failed due to mathematical formulas. Finally image *N02J* shows an image that fails to be oriented correctly due to the capitalized text.

	Results from [14]		Method (Full)		Method (Fast)		GT
Orientation	Found	Correct	Found	Correct	Found	Correct	
0°	936	935	963	962	955	955	970
90°	2	2	5	5	6	5	9
180°	0	0	8	0	8	0	0
270°	0	0	3	0	10	0	0
No result	41	0	0	0	0	0	–
Correct		95.8%		98.8%		98.0%	–

Table 2.2: A comparison of orientation detection results on UW-I dataset. The table shows that the proposed method detects the right orientation for a larger number of documents than does Bloomberg's method. The column showing the ground-truth information is entitled "GT".

Resolution	Total	BB [14] Correct	BB [14] No result	Prop. Method
150 dpi	36	36	0	36
200 dpi	36	36	0	36
300 dpi	36	36	0	36
400 dpi	36	27	9	36
Total Correct		93.8%		100.0%

Table 2.3: Results of Bloomberg's [14](BB) and the proposed method on test images with different resolution. The results show that Bloomberg's method failed to find a correct orientation for several documents rendered at 400 dpi. The proposed method find the correct orientation in all cases yielding a 100% result in this test.

Chapter 3

Document Cleanup

When it comes to document security, even small variances in the documents appearance can be important to detect possible forgeries. Therefore, sensitive measurements have to be done. In the presence of noise, however, these measurements will lead to unreliable results. If e.g. too much noise is present, accurate extraction of the intrinsic document features as positions of the connected components will be difficult. Therefore, it is important to remove the noise beforehand, as much as possible.

In document image understanding, the term *noise* is used to describe the effects of many different processes having all as consequence degradation of the image quality: copy border noise comes from the copying process and is often to be found as thick black borders around the page. Pixel noise can be found all over the page and may come e.g. from bad binarization or from noisy print outs.

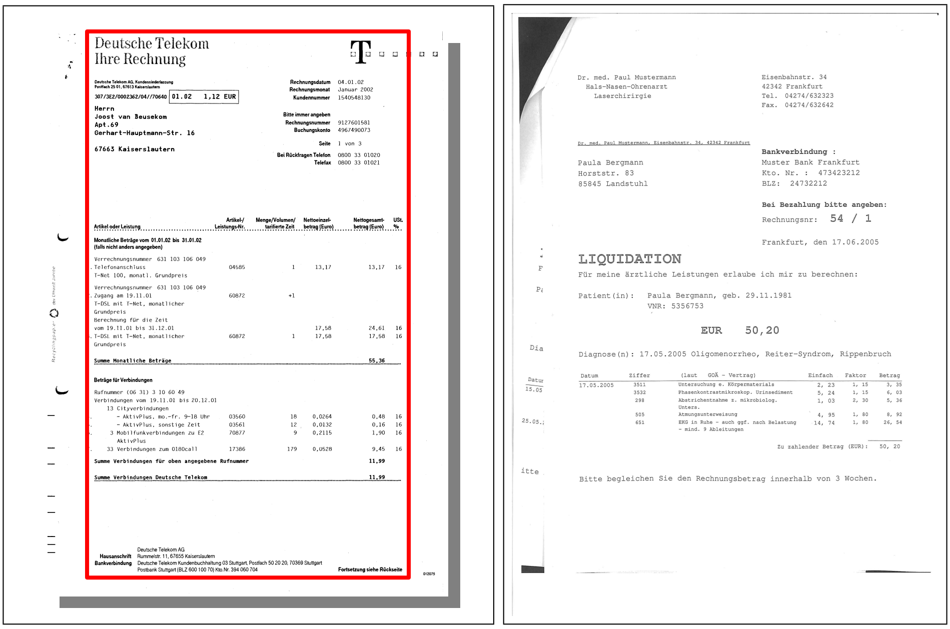
Due to the controlled lightening conditions while scanning using ADF scanners, for every day documents, most noise can be observed around the border of the pages: when the scanning area is bigger than the document page, or when physical copies of documents are scanned, border noise will be visible. Manually removing this noise is not feasible due to the costs this step would engender. Therefore, an automatic step has to be added to solve the problem.

In this chapter a method is presented to remove border noise by detecting the page frame, the area of the page that contains the main page content. The contributions presented in this chapter are the following[1]:

1. The first approach for effective noise removal by explicitly detecting the

---

[1] This chapter is based on the author's work in
[104] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel. Document cleanup using page frame detection. *Int. Jour. on Document Analysis and Recognition*, 11(2):81–96, 2008, and
[103] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel. Page frame detection for marginal noise removal from scanned documents. pages 651–660, Aalborg, Denmark, June 2007.



Deutsche Telekom
Ihre Rechnung

Deutsche Telekom AG, Kundenniederlassung
Postfach 25 01, 67613 Kaiserslautern

307/3E2/0002362/04//70640 | 01.02 1,12 EUR

Herrn
Joost van Beusekom
Apt.69
Gerhart-Hauptmann-Str. 16

67663 Kaiserslautern

Rechnungsdatum 04.01.02
Rechnungsmonat Januar 2002
Kundennummer 1540548130

Bitte immer angeben
Rechnungsnummer 9127601581
Buchungskonto 4967490073

Seite 1 von 3

Bei Rückfragen Telefon 0800 33 01020
Telefax 0800 33 01021

Artikel oder Leistung	Artikel-/ Leistungs-Nr.	Menge/Volumen/ tarifierte Zeit	Nettoeinzel- betrag (Euro)	Nettogesamt- betrag (Euro)	USt. %
Monatliche Beträge vom 01.01.02 bis 31.01.02 (falls nicht anders angegeben)					
Verrechnungsnummer 631 103 106 049					
. Telefonanschluss	04585	1	13,17	13,17	16
T-Net 100, monatl. Grundpreis					
Verrechnungsnummer 631 103 106 049					
. Zugang am 19.11.01	60872	+1			
T-DSL mit T-Net, monatlicher Grundpreis					
Berechnung für die Zeit					
vom 19.11.01 bis 31.12.01			17,58	24,61	16
. T-DSL mit T-Net, monatlicher Grundpreis	60872	1	17,58	17,58	16
Summe Monatliche Beträge				**55,36**	
Beträge für Verbindungen					
Rufnummer (06 31) 3 10 60 49					
Verbindungen vom 19.11.01 bis 20.12.01					
13 Cityverbindungen					
. - AktivPlus, mo.-fr. 9-18 Uhr	03560	18	0,0264	0,48	16
. - AktivPlus, sonstige Zeit	03561	12	0,0132	0,16	16
. 3 Mobilfunkverbindungen zu E2 AktivPlus	70877	9	0,2115	1,90	16
. 33 Verbindungen zum 0180call	17386	179	0,0528	9,45	16
Summe Verbindungen für oben angegebene Rufnummer				**11,99**	
Summe Verbindungen Deutsche Telekom				**11,99**	

Deutsche Telekom AG
Hausanschrift Rummelstr. 11, 67655 Kaiserslautern
Bankverbindung Deutsche Telekom Kundenbuchhaltung 03 Stuttgart, Postfach 50 20 20, 70369 Stuttgart
Postbank Stuttgart (BLZ 600 100 70) Kto.Nr. 394 060 704

Fortsetzung siehe Rückseite

012075

(a) Page frame eexample

Dr. med. Paul Mustermann
Hals-Nasen-Ohrenarzt
Laserchirirgie

Eisenbahnstr. 34
42342 Frankfurt
Tel. 04274/632323
Fax. 04274/632642

Dr. med. Paul Mustermann, Eisenbahnstr. 34, 42342 Frankfurt

Paula Bergmann
Horststr. 83
85845 Landstuhl

Bankverbindung :
Muster Bank Frankfurt
Kto. Nr. : 473423212
BLZ: 24732212

Bei Bezahlung bitte angeben:
Rechnungsnr: 54 / 1

Frankfurt, den 17.06.2005

LIQUIDATION

Für meine ärztliche Leistungen erlaube ich mir zu berechnen:

Patient(in): Paula Bergmann, geb. 29.11.1981
VNR: 5356753

EUR 50,20

Diagnose(n): 17.05.2005 Oligomenorrheo, Reiter-Syndrom, Rippenbruch

Datum	Ziffer	(laut GOÄ - Vertrag)	Einfach	Faktor	Betrag
17.05.2005	3511	Untersuchung e. Körpermaterials	2, 23	1, 15	3, 35
	3532	Phasenkontrastmikroskop. Urinsediment	5, 24	1, 15	6, 03
	298	Abstrichentnahme z. mikrobiolog. Unters.	1, 03	2, 30	5, 36
	505	Atmungsunterweisung	4, 95	1, 80	8, 92
	651	EKG in Ruhe - auch ggf. nach Belastung - mind. 9 Ableitungen	14, 74	1, 80	26, 54

Zu zahlender Betrag (EUR): 50, 20

Bitte begleichen Sie den Rechnungsbetrag innerhalb von 3 Wochen.

(b) Textual noise example

Figure 3.1: In Figure 3.1(a) an example of a page frame (red rectangle) and non-textual border noise are depicted. In Figure 3.1(b) copy borders and textual noise can be seen.

page's content (accuracy of 97.2% on zone level classification on the UW-III data set, $n = 1440$).

2. The first approach for textual and non-textual border noise removal.

3. Comprehensive evaluation scheme for multi-level evaluation of page content detection methods.

3.1 Introduction

In document security applications, high accuracy measurements of e.g. the text-line orientation need to be done. Noise, as in many other applications of document image understanding methods, causes problems that, depending on the amount of noise, decrease the accuracy of the methods considerably. Therefore, it is important to remove noise before the actual analysis of the document.

In ADF scanning environments, the main source of noise comes from the border areas of the document: if the scan area is bigger than the actual sheet, the sheet borders will be visible. Punching holes or the deterioration from staples also produce degradations that one would like to ignore in further processing. Also, many stationaries present various marks near the edges of the sheets. An example of a typical image scanned using an ADF scanner and presenting border noise is shown in Figure 3.1(a).

Concerning border noise, two different kinds of noise are distinguished: textual and non-textual noise. Textual noise are characters and text parts that are visible in the digital image but that are not actually part of the original page. Textual noise can be observed while scanning copies of stapled documents or copies of books. An example image is given in Figure 3.1(b).

In literature, many approaches try to explicitly detect and remove noise, as will be presented in Section 3.2. As in the current application the main source of noise is border noise, a method is presented to remove this type of noise implicitly by detecting the main content area of the document, which is called *page frame.* If the page frame is detected correctly, everything outside the page frame can be discarded, and thus also the border noise. An example of a document and its page frame is shown in Figure 3.1(a).

3.2 Previous Work

Related work in the domain of page frame detection can be divided into three different scopes:

- non-textual border noise removal
- textual border noise removal
- textual and non-textual border noise removal

Most methods to remove **non-textual noise** perform filtering on statistics on the connected components size and aspect ratio [79, 12, 19]. This might work well for non-textual noise, however, there is no chance in removing textual noise using these methods, as the textual noise will have to the same characteristics as the characters inside the page frame.

Other approaches tend to specifically remove artefacts that come from the copying process, as e.g. copy borders. In 2001, Fan et al. [38] presented a method for border noise removal. Using features as the size, the position and the aspect-ratio of connected components, a rule-based decision is made for each non-character connected component if it is border noise or not. Visual evaluation of the results on a non public dataset was done and proved to work well.

In 2004, Avila et al. [10] presented a method for removing noisy borders from documents, with the restriction that the noisy parts have to touch the image's border. Region growing is then performed. The performance of their system is measured by the compression ratio of the cleaned to the original image. Evaluation was done on a non-public dataset.

Several approaches for removing **textual and non-textual noise** have also been presented. Le et al. [57] presented a rule-based system for automatic border detection for textual and non-textual border noise removal. Their main assumption is that the borders are close to the image's edges and that they are separated by white space. A first rough initialization of the page frame is done based on the estimated font size and on the assumption that the distances between neighboring components is smaller for border components than for text components. Several other rule-based steps are used to refine the initial page frame. According to the authors, visual evaluation of the results proved good results.

Peerawit et al. [86] used projection profiles of the extracted edge image, to detect the left and right borders of the page frame. These are represented by peaks at the left and right end of the page. A similar approach was followed by Stamatopoulos et al. [109]. Here projection profiles on the smeared image are used to first remove thick black borders and then the textual noise. Both methods use several rules and assumptions making it difficult to adept the method to a broader class of documents. Both methods are evaluated only by visual inspection.

Cinque et al. [29] presented a method to solve the problem of textual and non-textual noise removal (called "partial extra pages" and "marginal artifacts" respectively) on gray scale images. This is done by counting transitions between pixel sequences of high values and sequences of low values for the image rows and columns. The four borders of the page frame are moved towards the center of the document image until the number of transitions exceeds a predefined threshold. An evaluation by visual inspection was done on a subset of the UW-I dataset.

So far, the previous methods have been trying to identify copy borders, border noise and textual noise and remove it. The proposed approach has a different view point: instead of identifying the noise around the page, the main page content is being identified. The bounding rectangle containing the main page content is defined as the page frame.

After the method presented in this chapter was published in 2008, Zhou et. al [127] published in 2009 a similar method: text-lines are extracted and the left and right border of the page frame are fixed. Then the upper and lower border are computed. The presented evaluation shows good results on the public dataset UW-III. The evaluation scheme originally presented by the author [104] (see Section 3.4) has been adopted by the Zhou et al.

All of the presented methods show considerable drawbacks: Le's method [57] uses many hand-crafted rules prohibiting easy adaption of the method to different document types. Furthermore, just as Fan's [38], Avila's [10], Peerawit's [86] and Stamatopoulos [109] methods, textual border noise cannot be detected.

One shortcoming all the presented methods have is the absence of an objective evaluation: several authors evaluate their method only on a visual basis [29, 86, 109]. Le et al. [57] evaluate on the number of correctly removed borders. Avila [10] presents an indirect evaluation based on the compression ratio of the cleaned image. The assumption is that the cleaned image, e.g. the image where the noise borders have been removed can be compressed more efficiently. This measure, however, does not penalize too small page frames. If e.g. the whole page content would be removed, the best compression ration would be achieved but the resulting image would be just empty.

In this chapter, a new approach for border noise removal is proposed. Instead of explicitly detecting the border noise, textual or non-textual, the page frame, the area containing the main page content, is detected. Removing everything outside the page frame will lead to nicely cleaned document images containing no border noise. Also, a new evaluation scheme for the process of page frame detection is presented. It takes into account different levels of page layout representation, leading to a good set of measurements that allow to identify the strengths and the weaknesses of the method.

In Section 3.3 the proposed method is explained. Section 3.4 explains the proposed evaluation scheme. Results are presented in Section 3.5.

3.3 Border Noise Removal by Detecting the Page-frame

The novelty of the proposed approach is the explicit modeling of the page contents for subsequent border noise removal. Previous methods all tried to remove the border noise explicitly.

This modeling implies a characterization of the page content. Different concepts are found in literature, most of these presenting hierarchical document models [34, 64, 31]. None of these models includes the description of border noise. Therefore, the existing models have to be extended by the concept of the page frame that defines the area of the content of the page. The resulting hierarchical model is defined as follows, from bottom to top:

- P_f and P_b are the foreground and background pixels of a binary document image D.

- P_f is divided into a set of connected components $C = \{c_1, \dots, c_m\}$ such that $c_i \cap c_j = \emptyset, \forall i \neq j$ and $\cup_{i=1}^{m} = P_f$

- C is grouped into a set of text-lines $L = \{l_1, \ldots, l_n\}$ such that $l_i \subseteq C$, $l_i \cap l_j = \emptyset,\ \forall i \neq j$
- $Z = \{z_1, \ldots, z_r\}$ is defined as the set of zones z_i, where $z_i \cap z_j = \emptyset, \forall i \neq j$ and $z_i \subseteq C$.
- the page frame F is defined as the minimum bounding rectangle containing all connected components belonging to main document in the image.

The information for the different levels of the hierarchy is extracted using different algorithms: a fast labeling algorithm is used for the extraction of the connected components [45]. Text-line information is extracted using a method presented by Breuel [18] (details are given in Section 2.3.1). The zone information is extracted using the Voronoi page segmentation algorithm [53]. Other algorithms could also be used, but both algorithms proved to work well on standard document image evaluation datasets [102].

3.3.1 Page Frame Model

The content of most documents lie inside a rectangular structure, due to the observation that most display medias are of rectangular nature (paper sheets, displays, etc.). Therefore it is reasonable to model the page frame of a scanned document images as a rectangle. A rectangle can be modeled by the following five parameters: $\vartheta = (l, t, r, b, \alpha)$. The parameters l and r represent the left and right borders of the page frame, whereas the parameters t and b represent the top and the bottom of the page frame. The parameter α is the skew angle of the document. Skew is removed beforehand using the method presented in Chapter 2. Thus the page frame model for a deskewed document image can be defined by four parameters only $\vartheta = (l, t, r, b)$ representing an axis-aligned rectangle.

Given the page frame model and the document model, the problem of page frame detection is defined as an optimization problem:

$$\hat{\vartheta}(C, L, Z) = \underset{\vartheta \in T}{\arg\max}\, Q(\vartheta, C, L, Z) \tag{3.1}$$

where T is the parameter space and $Q(\vartheta, C, L, Z)$ is the quality of the page frame with parameters ϑ with respect to the set of connected components C, the set of text-lines L and the set of zones Z.

The next step consists of defining a quality function that incorporates the necessary information (Section 3.3.2). This quality function is used for finding the parameter set that has highest quality (Section 3.3.3).

3.3.2 Quality Function

The quality function defines what a good page frame is and what not, based on the information that it incorporates. The quality function is needed by the subsequent parameter search.

Referring to most common documents, the following observations concerning the page content can be made:

- alignment of text: text is mostly left aligned or justified, leading to many characters aligned to the page-frame's left and right borders
- positions of non-text blocks: non-text blocks (images, logos, graphs) occur often on top or on bottom of document pages

The first observation can be used to define the quality function of the left and right parameters $\vartheta_h = \{l, r\}$ of the page frame. Although the four parameters $\vartheta = \{l, r, t, b\}$ are not independent, the decomposition of the problem into two subproblems seems a reasonable step, considering the differences in the nature of the two subproblems. Therefore the focus of the first part is to fix the quality function for the left and the right parameters of the page frame.

Measuring the text-line alignment could be done by counting the number of aligned connected components. For documents containing highly structured text, as e.g. tables, this may lead to problems, as columns of numbers are likely to give good qualities using the proposed criterion. Therefore, text-lines are used to define the quality for ϑ_h. Instead of counting up the number of connected components, the number of text-lines touching the left or right page frame border is considered. Text-lines are more robust against the afore mentioned problem while touching the page frame (for aligned text-lines).

The second observation is used in the parameter refinement step in Section 3.3.4 to fix the parameters $\vartheta_v = \{t, b\}$.

Considering first ϑ_h and the text-line alignment, the optimization problem in Equation 3.1 can be written as follows:

$$\hat{\vartheta_h}(L) := \arg\max_{\vartheta_h \in T} Q(\vartheta_h, L) \tag{3.2}$$

where $Q(\vartheta_h, L)$ is defined as:

$$Q(\vartheta_h, L) := \sum_{j=1}^{N} q(\vartheta_h, L_j) \tag{3.3}$$

where $q(\vartheta_h, L_j)$ is the quality contribution of a text-line L_j to the page frame defined by parameters ϑ_h.

The contribution of a text-line to the page frame is computed by the distance of the text-line to the respective page frame border. Therefore, a text-line is represented by the bounding box rectangle of the text-line with parameters (x_0, y_0, x_1, y_1), where x_0 and y_0 define the lower left corner and x_1 and y_1 the upper right corner of the bounding box of the text-line. The quality contribution of the line L_j to the left and right page frame borders can be written as:

$$\begin{aligned} q_1(\vartheta_h, (x_0, x_1)) = & \max\big(0, 1 - \frac{d^2(l, x_0)}{\epsilon^2}\big) + \\ & \max\big(0, 1 - \frac{d^2(r, x_1)}{\epsilon^2}\big) \end{aligned} \tag{3.4}$$

The parameter ϵ defines the margin in which a text-line contributes to the quality: a text-line that is farther away than ϵ pixels from the corresponding page frame border, will not contribute to the quality of the page frame.

Integrating only q_1 in the quality function will already work for single column documents. For multicolumn documents, however, this is likely to restrict the page frame only to the text column with the highest number of text-lines. To avoid this problem, a penalty term is added for text-lines on the “wrong” side of the page frame: that means, that the x_0 parameter of the line bounding box is left of the left border of the page frame or x_1 is right of the right border of the page frame. This penalty term is defined as follows:

$$\begin{aligned} q_2(\vartheta_h, (x_0, x_1)) = & - \max\big(0, 1 - \frac{d^2(l, x_1)}{(2\epsilon)^2}\big) \\ & - \max\big(0, 1 - \frac{d^2(r, x_0)}{(2\epsilon)^2}\big) \end{aligned} \tag{3.5}$$

The total quality is then defined as:

$$q(\vartheta_h, (x_0, x_1)) = q_1(\vartheta_h, (x_0, x_1)) + q_2(\vartheta_h, (x_0, x_1)) \tag{3.6}$$

It should be noted that the penalty term q_2 has a larger influence boundary (2ϵ) than q_1 (ϵ) in order to weaken its influence in comparison to q_1. Informally speaking, for the right border, it is more important having many text-lines starting at the same x_0 position than to avoid having too many text-lines outside of the page frame.

3.3.3 Branch-and-Bound Optimization

A branch-and-bound approach is used to search for the optimal parameter set ϑ_h. The initial parameter space (search space) is recursively split into smaller subspaces. For each subspace an upper bound is computed that is used to guide the search to the more promising subspaces. Finally, when the a subspace is found that is small enough, given the necessary accuracy, this subspace is the optimal result.

The upper bound is computed using the quality function defined in Equation 3.6. For each parameter subspace, the contribution of each text-line to the overall quality is computed.

The initial search space (parameter space) for the left and right page frame borders is defined as $T = [l_{min}, l_{max}] \times [r_{min}, r_{max}]$. The branch-and-bound algorithm is depicted in the following, similarly as in Section 2.3.1:

```
0: initialize the search space $T$, compute the upper bound for the quality and
   insert it into priority queue $Q$
1: stop if $Q$ is empty, else get the top search space $S$ from $Q$
2: if $S$ is small enough: return $S$
3: split $S$ into two subspaces $S_1$ and $S_2$
4: compute upper bounds for the quality of $S_1$ and $S_2$
5: sort $S_1$ and $S_2$ into $Q$ using the computed upper bounds. Continue with
   Step 1.
```

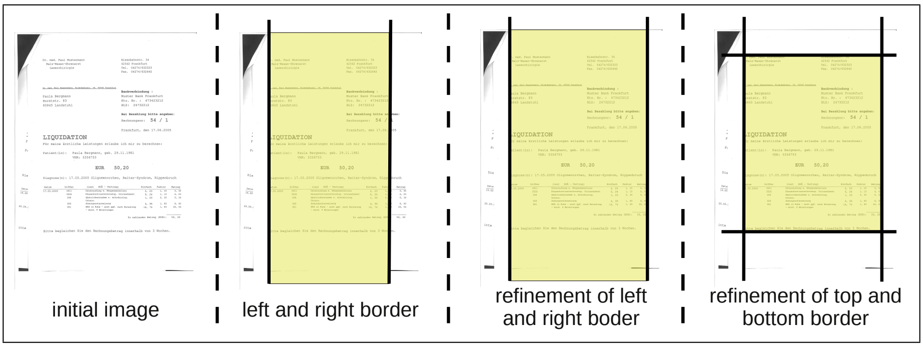
Just as in [17] a match list is used to speed up search by avoiding redundant computations. This match list tracks for each subset which text-lines contribute to the page frame quality. All lines that do not contribute to the page frame are ignored.

The result of the algorithm will be a pair of parameters $\hat{\vartheta} = (l, r)$ that defines the left and right position of the page frame with the highest quality according to the quality function in Equation 3.2.

3.3.4 Parameter Refinement

The optimal branch-and-bound algorithm in Section 3.3.3 returns the best possible left and right border for page frame according to the defined quality function. If the text-lines are not justified or if a high number if indented lines are present, the computed page frame from Section 3.3.3 may cut through text-lines. Also, $\vartheta_v = (t, b)$ has still to be defined to obtain a complete pageframe.

First, the left and right border of the page frame are readjusted to avoid cutting text-lines. This is done by extending the page frame such that all text-lines are included that overlap with a certain amount the area of their bounding box to the page frame.



initial image

left and right border

refinement of left and right boder

refinement of top and bottom border

Figure 3.2: Example image demonstrating the parameter refinement on text-line level. The left image shows the text-lines. The center image shows the optimal page frame according to Equation 3.3 (the left and the right borders are fixed). It should be noticed, that the page frame adapted to the indented text block in the center of the page and that parts of other lines are missed. After applying the refinement for the left and right border, this issue is solved. Finally, the top and the bottom borders are fixes (right image).

Second, the parameters $\vartheta_v = (t, b)$ for the top and the bottom page frame borders have to be fixed. Using the text-lines contributing to the left and right page frame borders, an initial setting for (t, b) is found by using the top bounding box line of the uppermost text-line for t and the bottom bounding box line of the lowest text-line for b. Formally, this is defined as follows: let $C = \{c_1, \ldots, c_n\}$ be the set of contributing text-lines. Then $b = \min_i(c_{i,y_0})$ and $t = \max_i(c_{i,y_1})$, where c_{i,y_0} is the lower coordinate of the bounding box of the contributing text-line c_i and c_{i,y_1} is the upper coordinate of the bounding box of the contributing text-line c_i. An example is depicted in Figure 3.2.

For many documents, the proposed settings will lead to the correct page frame. There are cases, however, that cannot be covered by the proposed parameter refinement method:

- pages starting or ending with non-text zones, e.g. logos, images, drawings or graphics.
- pages having isolated small text-blocks as e.g. page numbers

In both cases, the document parts will be missed, as these parts normally do not contain any text-lines. To cope with this difficulty a different level of representation of the document layout has to be used. Instead of using text-level segmentation, the zone level information is used. This is extracted using the Voronoi page segmentation algorithm proposed by Kise [53]. For each of these zone boxes, it is checked whether it overlaps horizontally with the so far detected page frame. In

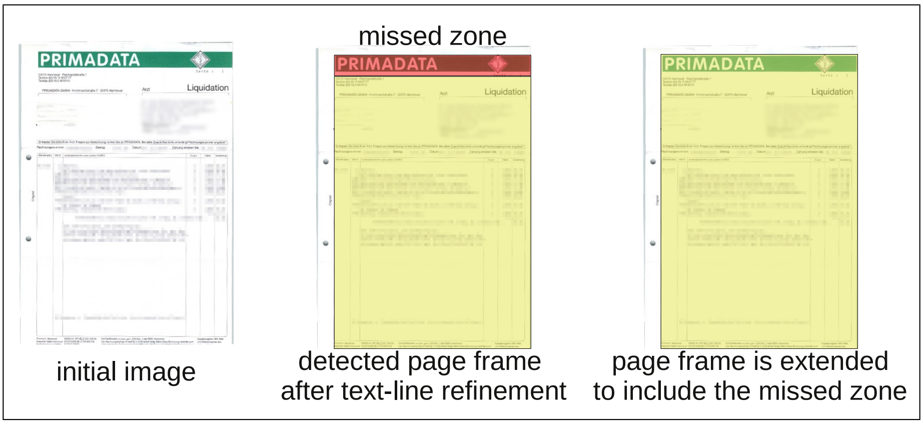

missed zone
initial image
detected page frame after text-line refinement
page frame is extended to include the missed zone

Figure 3.3: Example image showing the inclusion of non-text zones into the page frame. The left image shows the initial document. The image in the middle shows the detected page frame using the text-line level refinement. It also shows the missed zone on top of the page. By including the missed zone, the page frame is extended to include all page content (right image).

that case the page frame is extended by including the detected zone. A visualization of the different refinement steps is given in Figure 3.3. An example of a detected page frame on a typical invoice can be found in Figure 3.8(b).

3.4 Performance Measures

First, in Section 3.4.1, direct performance measures are presented. These performance measures directly measure the goodness of the detected page frame compared with a given ground truth page frame. Different levels of comparison are presented. Second, in Section 3.4.2 the improvement of real-world applications after detecting the page frame is shown. These indirect measurements show, that page frame detection is an important step that should not be neglected.

3.4.1 Page Frame Detection Accuracy

In this section, objective direct performance measures are presented that allow automatic evaluation of page frame detection methods. Previous approaches for marginal noise removal [57, 29, 38, 10] used visual inspection to decide whether noise regions have been completely removed or not. Then, the error rate is defined as the percentage of documents on which the noise was not completely removed. These approaches have to considerable drawbacks: first, evaluation has to be done manually, making large scale evaluation problematic. Second, manual inspection introduces a degree of uncertainty as often no formal definition of visually correct or wrong is given.

As a document layout can be described on different levels, the performance measures to be presented also use different level information. This is necessary to fully describe the strengths and weaknesses of page frame detection methods. First, a pixel level measure computing the area overlap between the ground-truth and the detected page frame is presented. Next, the connected component based measure computing the ratio of correctly classified connected components is explained. Finally, a performance measure on zone level is presented.

Area Overlap

Let F_g be the ground-truth page frame and F_d be the detected page frame. The area overlap between the two page frames is defined as

$$A = \frac{2|F_g \cap F_d|}{|F_g| + |F_d|} \tag{3.7}$$

The area overlap A will vary between zero and one depending on the overlap between ground-truth and detected page frames. If the two page frames do not overlap at all $A = 0$, and if the two page frames are identical, i.e. $|F_g \cap F_d| = |F_g| = |F_d|$, then $A = 1$. This gives a good measure of how closely the two page frames match.

The area overlap A, however, does not give any insight in the type of errors made by the method. Moreover, small errors as including a noise zone near the top or bottom of the page into the page frame or missing a page number may result in a large error in terms of area overlap. To evaluate the page frame detection algorithm in more detail, a performance measure based on connected component classification is defined.

Connected Components Classification

Connected components lying inside the ground-truth page frame are defined as 'positive', whereas connected components outside the ground-truth page frame are considered as 'negative'. Using this classification scheme, the classification results are measured in terms of 'true positives', 'true negatives', 'false positives' and 'false negatives'. The error rate is defined as the ratio of incorrectly classified connected components to the total number of connected components.

The error measure based on classification of connected components gives equal importance to all components, which may not be desired. For instance, if the page number is not included in the detected page frame, the error rate will still be very low because page number comprises a very small fraction (typically about

0.03% to 0.1%) of the total number of connected components in the page frame. However, the page number carries important information for the understanding of the document. To compensate this shortcoming, a performance measure based on detection of ground-truth zones is introduced.

Ground-Truth Zone Detection

This performance measure uses the ground-truth page zone information to measure the accuracy of the method. The idea is to measure if any important zone, as e.g. the zone containing the page number, is missed. For the zone-based performance measure, three different values are determined:

- Totally In: Ground-truth zones lying completely inside the computed page frame (complete overlap)

- Partially In: Ground-truth zones lying partially inside the computed page frame (partial overlap)

- Totally Out: Ground-truth zones lying totally outside the computed page frame (no overlap)

Using this performance measure, the ‘false negative’ detections are analyzed in more detail. Zones that lie ’totally outside’ the page frame are a serious problem and can be detected using this performance measure.

3.4.2 Performance Gain in Practical Applications

In this section, the performance measures on basis of real-world applications are defined. The first measure computed the optical character recognition error rate (OCR). The second application to show the improvement on a real-world problem using page frame detection is a system for layout-based document image retrieval.

OCR Accuracy

The OCR accuracy is often measured using the edit distance (also called Levenshtein distance [59]). The edit distance between the ground-truth text and the recognized text is the minimum number of point mutations (insertion, deletions, and substitutions) required to convert the recognized text string into the ground-truth text string. As textual noise will result in additional characters, this distance is likely to increase in the presence of textual border noise. The goal of this measure is to show that page frame detection can help in improving OCR results.

Layout-Based Document Image Retrieval

In layout-based retrieval, the purpose is to query document image databases by layout, in particular by measuring the similarity of different layouts in comparison to a reference or query layout. Blocks originating from marginal noise result in incorrect matches, thereby increasing the error rates of the retrieval system. Different layout analysis or page segmentation algorithms use different methods to deal with noise in a document image. The goal of this performance measure is to determine the decrease in retrieval error rates when page frame detection is used as a pre-processing step.

3.5 Experimental Setup and Results

The main part of the evaluation has been done on the UW-III dataset [87]. It consists of 1600 binarized and deskewed document images from scientific journals. In order to develop and adapt the quality function system, a training dataset of 160 images out of the 1600 was chosen. In order to make the results replicable, every 10th image (in alphabetical order) from the dataset was included into the training set. Hence the training set consists of images A00A, A00K, ..., W1UA. The parameter refinement steps (Section 3.3.4) were also introduced based on results on the training images to cope with different layout styles and the presence of non-textual content at the top or bottom of a page image.

The second test was run on the ICDAR 2007 page segmentation contest dataset [7]. The dataset contains 23 magazine pages, among which 6 are in the training set and 17 are in the test set. All 23 images are used as test images and the proposed method was used without any parameter tuning on these images. The same parameter settings is thus used in all experiments.

The method remained for all tests unchanged. Also the parameters were fixed once for all tests. The parameter ϵ defining the range in which a text-line contributes to the page frame is set to $\epsilon = 150$. The initial parameter space is fixed in a way that allows the left page frame border to be anywhere left from the middle of the image. The right page frame border can vary between the right border of the image and the center: $T = [l_{min}, l_{max}] \times [r_{min}, r_{max}] = [0, \frac{W}{2}] \times [\frac{W}{2}, W]$. The overlap of a text-line with the page frame in order to be included into the page frame during parameter refinement (Section 3.3.4) is set to 50% of the area of the text-line.

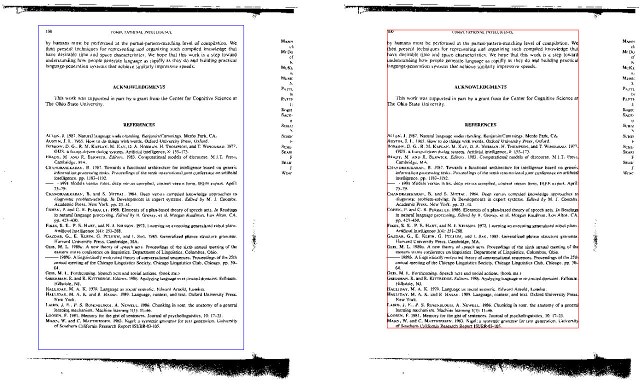


Figure 3.4: The left image shows a document together with its original ground-truth page frame. The right image shows the corrected ground-truth page frame obtained by computing the smallest rectangle including all the ground-truth zones.

3.5.1 Results on Page Frame Detection Accuracy

Area Overlap on UW-III

The evaluation of page frame detection on the basis of overlapping area (Equation (3.7)) showed a page frame detection accuracy of 91%. As this number was was unexpectedly low, a visual inspection of the UW3 ground-truth page frame showed that it does not tightly enclose the page contents area as shown in Figure 3.4. Hence, the correct page frame of documents in the test set was fixed to the bounding box of all ground-truth zones for each document. Testing with the corrected ground-truth page frame gave an overall mean area overlap of 96%. In the following, when mentioning the ground-truth page frame, this corrected ground-truth page frame is meant.

Connected Component Classification on UW-III

The result for the connected component classification measure is given in Table 3.1. The high percentage of true positives shows that the page frame mostly includes all the ground-truth components. The percentage of true negatives is about 73.5%, which means that a large part of noise components are successfully removed.

As the UW-III dataset contains document images generated by different consecutive copying processes, the performance on the different classes of copies is analyzed. The results for the Nth generation photocopies show that the percentage of true negatives goes down to 42.8% which may lead to the conclusion, that the computed page frames for this subset are typically bigger than the ground-truth page frame. The total error rate defined as the ratio of ‘false’ classifications

Table 3.1: Results for the connected component based evaluation. The number in brackets gives the number of documents of that class. Error rates in [%].

Doc. Type	True Positive	False Negative	True Negative	False Positive
Scans (392)	99.84	0.16	76.6	23.4
1Gen (1029)	99.78	0.22	74.0	26.0
NGen (19)	99.93	0.07	42.8	57.2
all (1440)	99.8	0.2	73.5	26.5
total (abs.)	4,399,718	8,753	187,446	67,605

to the total number of connected components is 1.6%. Since the test set contains only 19 images in the NGen category, the total results do not reflect the performance on such severely degraded documents. A detail study of the performance of the proposed page frame detection method on documents with different noise levels is presented later in this section.

In order to quantify the amount of marginal noise in a document image, the noise ratio of a document image is defined as

$$\text{Noise ratio} = \frac{n_{pb}}{n_p} \tag{3.8}$$

Where n_{pb} is the number of foreground pixels outside the ground-truth page frame, and n_p is the total number of foreground pixels in a document image. A histogram of the noise level of the documents in the test set is shown in Figure 3.5. Interestingly, there are many documents with noise levels above 50%. The mean error rate obtained for each of these noise level based document categories is plotted in Figure 3.9. The plot shows that the algorithm works well even on documents with high amount of noise. The error rates on all three performance measures used are below 10% for noise levels up to 80%.

Zone-Based Classification on UW-III

The results for the zone based measure are given in Table 3.2. It can be seen that the percentage of missed zones is slightly higher than the corresponding percentage of false negatives on the connected component level. One conclusion is that the missed zones do not contain a large number of components, which is typically true for page numbers, headers and footers of documents. These zones contain only few connected components and therefore do not contribute much to the mean false negative errors on the connected component level. In some cases, the text-line finding algorithm merges the text-lines consisting of textual noise to those in the

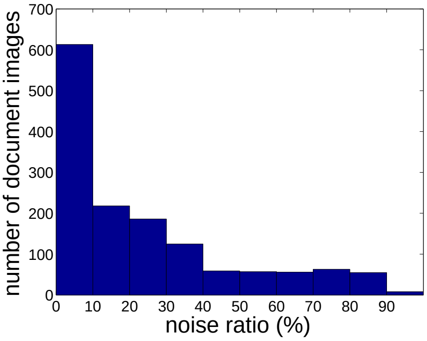


Figure 3.5: Histogram of the noise ratio (Equation 3.8) of the documents in the test set.

Table 3.2: Results for the zone based performance evaluation. Error rates in [%].

Document Type		Totally In	Partially In	Totally Out
Scans	(392)	97.6	0.7	1.7
1Gen	(1029)	97.1	1.0	1.9
NGen	(19)	97.5	0.0	2.5
all	(1440)	97.2	0.9	1.9

page frame. This leads to a large portion of textual noise being included in the page frame.

Processing Time on UW-III

A box plot of the run times of different steps of the proposed method is shown in Figure 3.6. The execution times were computed on an AMD Athlon 1.8 GHz machine running Linux. The worst case running time of the unmodified RAST algorithm is exponential in the problem size [16]. In practice, however, such a case rarely appears. For page frame detection, the RAST algorithm took less than 10 msec per page on the average. Hence if it is integrated with a document analysis system that already computes text-lines and zones, page frame detection will not add a significant increase in the computation time. When page frame detection is used as a monolithic system for document image cleanup, the total running time is of interest, which is 3-4 seconds per page.

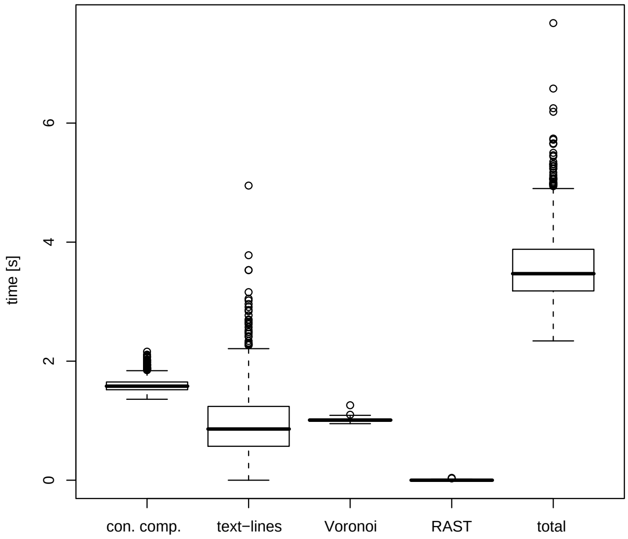


Figure 3.6: Run times of the different steps of the algorithm for the test on UW-III.

Evaluation on Magazine Pages

The UW-III dataset contains scanned pages from journal articles with Manhattan layouts. In order to test the performance of the approach on non-Manhattan layouts that are more representative of layouts in magazines, the ICDAR 2007 page segmentation contest dataset [7] was chosen. Some of the results are shown in Figure 3.7. It can be seen that the page frame detection works correctly for most of these pages. In Figures 3.7(a)-(c) the algorithm found the correct page frame without making any error. In Figure 3.7(d) one vertical text-line was missed by the proposed algorithm since it was not detected by the text-line detection algorithm. Another error is shown in Figure 3.7(e) where the last column consisted only of a few lines and hence was excluded from the detected page frame. Note that in Figure 3.7(d), margin notes were included in the computed page frame because the title lines were spanning across the main text content area and the margin notes area. A similar error is shown in Figure 3.7(f) where the last column consists only of images and hence was not included in the page frame.

The quantitative evaluation of the performance was done using the ground-

Table 3.3: Results for the text-line based error measure on magazine pages from the ICDAR 2007 page segmentation competition [7]. Error rates in [%].

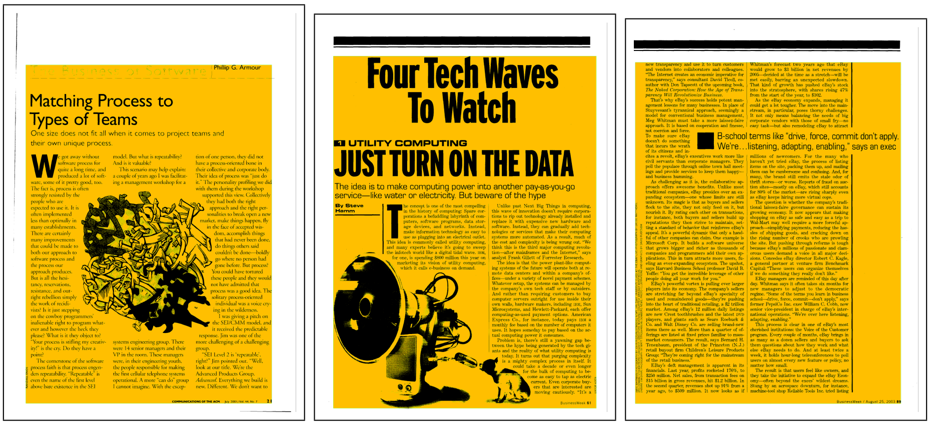
Type of Line	Totally In	Partially In	Totally Out
Horizontal (2393)	98.6	0.0	1.4
Vertical (15)	66.7	0.0	33.3
Inverted (22)	81.8	13.6	4.6
all (2430)	98.23	0.12	1.65

truth zone detection measure (see Section 3.4.1). Since ground-truth information was not available, the number of text-lines in each document was counted. The percentage of totally in, partially in, and totally out text-lines is computed. The results are shown in Table 3.3. The results show that the page frame detection correctly included 98.23% text-lines from the test images. The partially missed lines were those with inverted text and were on the top of the page. The text-line extraction algorithm detected only parts of these text-lines and hence the remaining part was not included in the detected page frame. The missed errors were due to margin notes and isolated vertical lines.

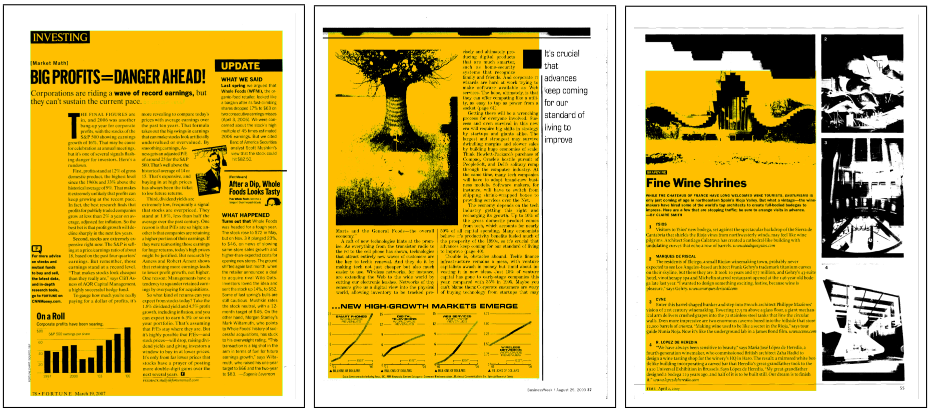
Advantages and Limitations

A particular advantage of the proposed method is that it is robust to skew between the main page frame and the textual noise. An example image is shown in Figure 3.8(a) where the presented method successfully finds the main page content area despite the presence of textual noise with a different skew angle. Also note that there is little non-textual noise in this page. Therefore the assumption by Peerawit et al. [86], that noisy regions can be separated from text regions based on edge density, will not work here.

A few limitations of the presented page frame detection algorithm were also revealed during the course of evaluation. Although the algorithm works very well for most of the layouts even under large amount of noise, yet for a few layouts the algorithm does not give 100% result even for noise-free documents. This happens for documents with few text-lines beside the margin of the document and also, if there is no text-line that spans across the main content area and the page margin. These text-lines lie completely outside the computed left and right page frame border parameters ϑ_h (Equation 3.2). So the parameter refinement step (Section 3.3.4) fails to include these text-lines into the page frame. To deal with such layouts, the quality function can be modified to include an offset between the page frame parameters and the main content area of the page.



(a) mp00004 (b) mp00248 (c) mp00262



(d) mp00307 (e) mp00253 (f) mp00290

Figure 3.7: Example of page frame detection on magazine pages from the ICDAR 2007 page segmentation contest. (a) image with non-Manhattan layout with left aligned text. (b) image with large variety of font sizes and a vertical text-line. (c) image with text-lines extending beyond other lines on the page with some vertical text-lines. (d) image with a mixture of content type (e) image with margin notes (f) image with a column containing no text-lines.

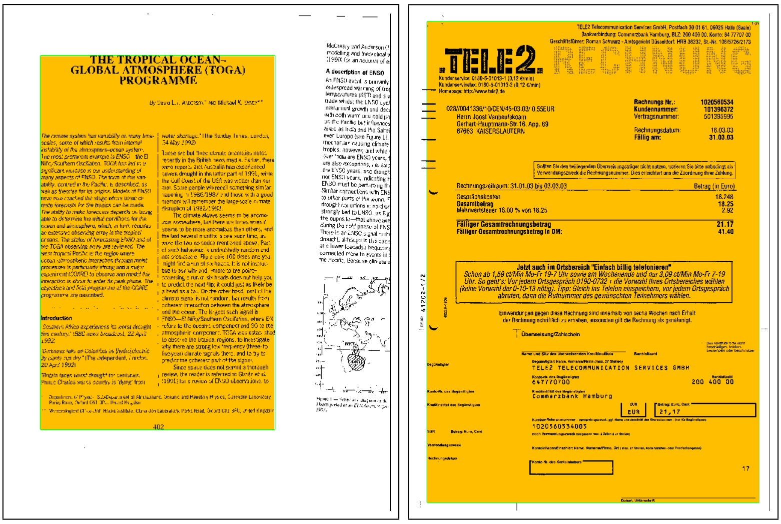


Figure 3.8: In Figure 3.8(a) an example image (S021) showing the page frame detection in case of skew between the main page frame and the textual noise can be found. Figure 3.8(b) shows the detected page frame on a typical invoice.

3.5.2 Performance Gain in Practical Applications

The use of page frame detection in an OCR system showed significant improvement in the OCR results. For this purpose Omnipage 14 - a commercial OCR system - was chosen. The ground-truth text provided with the UW-III dataset has several limitations when used to evaluate an OCR system. First, there is no text given for tables. Secondly, the formating of the documents is coded as latex commands. When an OCR system is tested on this ground-truth using error measures like the Edit distance, the error rate is unjustly too high. Also, the emphasis in this work is on the improvement of OCR errors by using page frame detection, and not on the actual errors made by the OCR system. Hence, the UW-III documents are first cleaned using the ground-truth page frame, and the output of Omnipage on the cleaned images was used as the ground-truth text. This type of ground-truth gives us an upper limit of the performance of a page frame detection algorithm, and if the algorithm works perfectly, it should give 0% error rate, independent of the actual error rate of the OCR engine itself.

First, OCR was performed on the original images and the Edit distance to the estimated ground-truth text was computed. Then, the computed page frame was used to remove marginal noise from the documents, and the experiment was run again. The results in Table 3.4 show that the use of page frame detection

Table 3.4: Results for the OCR based evaluation with page frame detection (PFD) and without page frame detection. The total number of characters is about 4.8 million.

	Del.	Subst.	Ins.	Total Errors	Error Rate
w/o PFD	34966	29756	140700	205422	4.3%
with PFD	19544	9828	53610	82982	1.7%

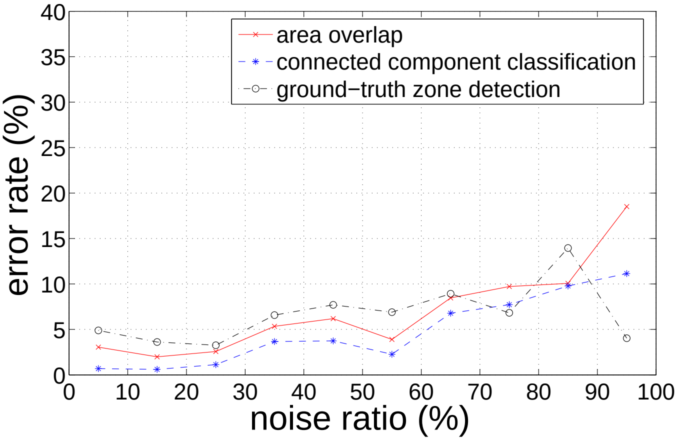
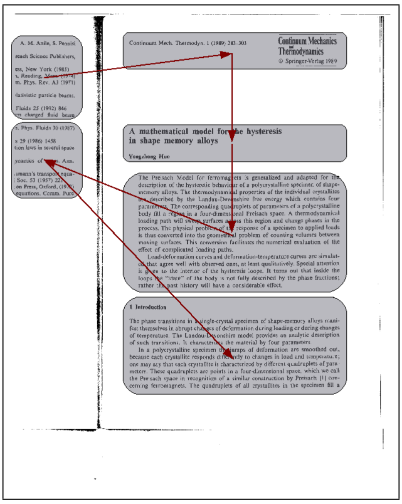


Figure 3.9: Performance of the page frame detection on different documents categorized by their noise level. The three lines show the three different error measures introduced in Section 3.4 with increasing noise level.

for marginal noise removal reduced the OCR error rate from 4.3% to 1.7%. The insertion errors are reduced by a factor of 2.6, which is a clear indication that the page frame detection helped in removing a lot of extra text that were treated previously as part of the document text. There are also some deletion errors, which are a result of the changes in the OCR software's reading order determination. One example is shown in Figure 3.10 for which the reading order changed after document cleaning.

The effect of using page frame detection on the performance of a layout based document image retrieval application showed a significant decrease in retrieval error rates. Van Beusekom et al. [111] introduced several similarity measures for layout based document image retrieval. They evaluated the performance of these similarity measures on the MARG database. The experiments showed that the best distance measure for this task is the overlapping area combined with



A. M. Anile, S. Pennisi
reach Science Publishers,
ess, New York (1985)
1, Reading, Mass. (1974)
m. Phys. Rev. A3 (1971)
lativistic particle beams.
Fluids 25 (1982) 846
rm charged fluid beam

s. Phys. Fluids 30 (1987)
s 29 (1986) 1458
tion laws in several space
ynamics of [illegible]. Am.
zmann's transport equa-
Soc. 53 (1957) 222
on Press, Oxford, (19[illegible])
equations. Comm. Pure

Continuum Mech. Thermodyn. 1 (1989) 283–303

Continuum Mechanics and Thermodynamics
© Springer-Verlag 1989

A mathematical model for the hysteresis in shape memory alloys

Yongzhong Huo

The Preisach Model for ferromagnets is generalized and adapted for the description of the hysteretic behaviour of a polycrystalline specimen of shape-memory alloys. The thermodynamical properties of the individual crystallites are described by the Landau-Devonshire free energy which contains four parameters. The corresponding quadruplets of parameters of a polycrystalline body fill a region in a four-dimensional Preisach space. A thermodynamical loading path will sweep surfaces across this region and change phases in the process. The physical problem of the response of a specimen to applied loads is thus converted into the geometrical problem of counting volumes between moving surfaces. This conversion facilitates the numerical evaluation of the effect of complicated loading paths.

Load-deformation curves and deformation-temperature curves are simulated that agree well with observed ones, at least qualitatively. Special attention is given to the interior of the hysteresis loops. It turns out that inside the loops the "state" of the body is not fully described by the phase fractions; rather the past history will have a considerable effect.

1 Introduction

The phase transitions in a single-crystal specimen of shape-memory alloys manifest themselves in abrupt changes of deformation during loading or during changes of temperature. The Landau-Devonshire model provides an analytic description of such transitions. It characterizes the material by four parameters.

In a polycrystalline specimen the jumps of deformation are smoothed out, because each crystallite responds differently to changes in load and temperature; one may say that each crystallite is characterized by different quadruplets of parameters. These quadruplets are points in a four-dimensional space, which we call the Preisach space in recognition of a similar construction by Preisach [1] concerning ferromagnets. The quadruplets of all crystallites in the specimen fill a

Figure 3.10: Result of Omnipage 14 showing the recognized text blocks in the document and their reading order. Note that Omnipage fails to find a correct reading order and inserts the textual noise zone between the text zones from the page contents area.

the Manhattan distance of the corner points as block distance together with the minimum weight edge cover matching for establishing correspondences between the matched layouts. The documents in the MARG database are categorized with respect to 9 different layout types, 59 publishers, and 161 journals. Given a query document, the target is to retrieve a document of the same class based on layout information only. The error rates are then determined as the percentage of correctly retrieved documents using leave-one-out cross validation.

In this work retrieval experiments were performed both with and without using page frame detection. Since each method for page segmentation has a different way of dealing with noise, four well-known page segmentation algorithms were compared for use in layout based retrieval: the X-Y cut [76], Docstrum [79], whitespace analysis [12], and the Voronoi-diagram based approach [52].

In the first experiment, the document images were used directly for page segmentation without any page frame detection. The blocks extracted from the documents were used for the purpose of layout based retrieval.

In the second experiment, the document images in the database were cleaned by performing page frame detection and removing all the foreground pixels outside the detected page frame. Following the document cleaning, page segments were extracted from the cleaned images and the retrieval experiment was repeated. The decrease in error rates for each of the three subdivisions of the data set (according to type, publisher, and journal) was used as a performance measure.

Table 3.5: Comparison of the error rates [%] for layout-based document image retrieval with and without using page frame detection.

Segmentation algorithm	Page frame detection	MARG database classes		
		journal	type	publisher
Voronoi	no	31.0	7.5	7.0
	yes	29.7	5.3	5.4
X-Y cut	no	36.3	11.7	13.6
	yes	33.5	8.6	8.0
Docstrum	no	40.9	14.0	14.4
	yes	32.1	7.4	7.1
Whitespace	no	48.3	20.3	24.6
	yes	31.2	7.2	6.1

The use of page frame detection for layout-based document image retrieval resulted in lower error rates on all three classes of layouts for each algorithm as shown in Table 3.5. These results show that the Voronoi-diagram based approach performs better than other algorithms both with and without page frame detection. The use of page frame detection with the Voronoi algorithm lowers the retrieval error rates by 4% for the correct journal, 30% for the correct type, and 20% for the correct publisher. These results clearly demonstrate the usefulness of page frame detection in practical applications.

3.6 Conclusion

In this chapter an algorithm for page frame detection. The approach does not assume the existence of whitespace between marginal noise and the page frame, and can detect the page frame even if the noise overlaps regions of the page content area. Also, an objective and reasonable set of performance measures was defined based on area overlap, connected component classification, and ground-truth zone detection accuracy for determining the accuracy of the presented page frame detection algorithm. It was shown that the algorithm performs well on all three performance measures with error rates below 4% in each case. It was also demonstrated that the presented method can handle documents with a large amount of noise with reasonable accuracy. The error rates on all three performance measures used are below 10% for noise levels up to 80%. The major source of errors was missing isolated page numbers. Locating the page numbers as a separate process and including them in the detected page frame may further decrease the error rates. The benefits of the page frame detection in practical applications were highlighted by using it with an OCR system and a layout-based document image retrieval system, where it showed a significant decrease in the error rates in both applications.

Chapter 4

Logical Labeling for Model Identification

In document security applications, model-based approaches are used to detect suspicious abnormal variations of a document relative to its model. Finding the model that corresponds to the document in question can be done by identifying the source of the invoice. This is achieved by extracting the corresponding information from the document by logically labeling the text regions.

In this chapter, a method is presented that can identify specific text blocks, as e.g. the invoice source or the invoice body. Using this information, the text in this block is labeled accordingly and can thus be used to retrieve the correct model from the database, by textual query on e.g. the address of the invoice source.

The process of detecting the *function* of a text block is called *logical labeling.* The approach proposed in this chapter solves the logical labeling using a case-based reasoning approach. It allows for fast and easy adaptation to new label domains and avoids the need of huge labeled datasets for training the system. The following contributions are presented in this chapter[1]:

1. The first[2] Case-Based Reasoning (CBR) approach for logical labeling.

---

[1] This chapter is based on the author's work in
[112] J. van Beusekom, D. Keysers, F. Shafait, and T. M. Breuel. Example-based logical labeling of document title page images. In *Proc. of the 9th Int. Conf. on Document Analysis and Recognition*, pages 919–923, Curitiba, Brazil, September 2007, and
[111] J. van Beusekom, D. Keysers, F. Shafait, and T. M. Breuel. Distance measures for layout-based document image retrieval. In *Proc. of the 2nd Int. Workshop on Document Image Analysis for Libraries*, pages 232–242, Lyon, France, April 2006.

[2] Hamza et al. [43, 44] independently developed a different case-based reasoning approach at about the same time

2. A flexible method that is easily adaptable to other labeling domains, as shown by the evaluation on different datasets.
3. A thorough evaluation of the proposed method proving that the proposed approach outperforms existing methods on standard public datasets (accuracy of 99.6% on the MARG data set, $n = 6770$).

4.1 Introduction

A common approach to authenticate documents is to check if the properties of the document in question coincide with known examples of previous documents of the same type. For an automated implementation of this process, one needs to find out to what type the document belongs to. In a border control scenario, the different types may be passes from different countries. In the context of an insurance scenario, these types can be invoices from different sources.

In this context, one possibility to select the correct type is to identify the invoice source. This can be done by extracting the textual information about the invoice source and use the extracted text as query to find the fitting type.

This text-based approach requires optical character recognition (OCR) and may seem cumbersome at a first glance. But in the proposed scenario, the sole goal of the digitization of incoming mail is not only to save space in the archive but also to automate the document handling process: incoming mail could be sent automatically to the insurance adjuster in charge of the customer. The necessary information for the insurance adjuster could already be extracted, leaving him with nothing to do but to decide if the customer will be reimbursed or not. Thus, the step of extracting the textual information from the document is needed anyway.

It is clear, that different applications of logical labeling come with different label sets. For invoices, labels as e.g. “address”, “invoice body” and “contact information” might be useful. For other types of documents, e.g. business letters, one might only be interested in finding the “destination”, the person that the letter was sent to. In a different application, the labeling of title pages of scientific journal papers, “author”, “abstract”, “affiliation” and “title” might be suitable. Developing a system that solves the logical labeling problem for all type of documents is thus more than an ambitious goal. Example images showing different types of documents with different labels are shown in Figure 4.1.

In order to deal with a wide variety of labels, a flexible method has to be used. The novel approach presented in this chapter bases on case-based reasoning [1]. The general idea of case-based reasoning for solving a new problem is to identify the most similar case in the case-base, and to reuse the solution that was used in the most similar case to solve the current problem.

In the current application, a problem is represented by a document and its

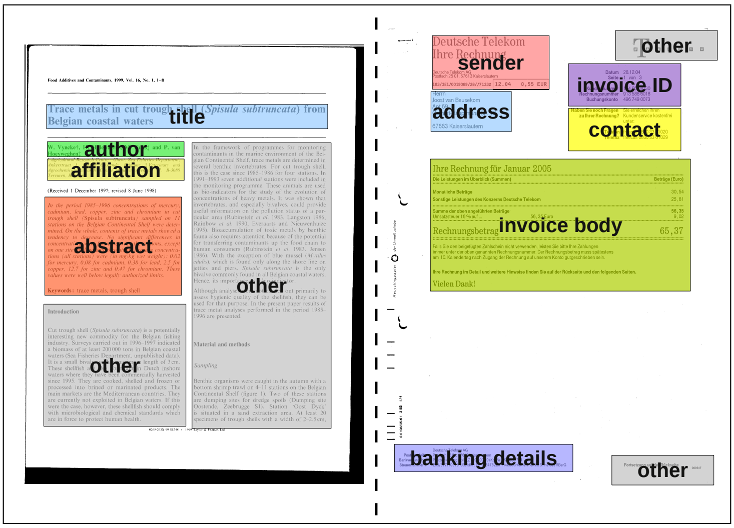


Figure 4.1: Examples of documents having different logical label sets. The left image shows an example of a labeled journal title page, whereas the right image shows an invoice with its respective labels.

zone-level page segmentation, represented by a set of rectangular blocks. The solution of the problem, which also is part of the case, is the assignment of labels to the segmented zones.

Unlike existing approaches, the proposed CBR approach has the advantage that it does not require label specific background knowledge. Furthermore it does not need a large training set to learn how to correctly assign the labels to the blocks. Hence, it is applicable on a wide range of document types. The functionality of the proposed approach is shown for the task of labeling invoices and for the task of labeling title pages of scientific journals (Section 4.4).

Section 4.2 gives a short overview of related methods. Section 4.3 presents the CBR approach for logical labeling. Evaluation setup and performance measures are presented found in Section 4.4 and Section 4.5. Section 4.6 concludes the chapter.

4.2 Previous Work

Due to the lack of a general approach for solving the logical labeling problem for a broad class of documents types, many different approaches have been developed for various classes of documents. A good overview is given in the survey papers by Mao [72] and Cattoni [25]. Due to the wide diversity of the methods proposed in literature and also due to the different focusses of the methods, having a strong influence on the labels and test sets, quantitative comparison of the approaches is not easily possible.

Labeling title pages of scientific papers has been investigated by several authors: Kim et al. [51] propose a rule based system using optical character recognition (OCR) to obtain plain text as well as document image analysis methods for extracting block features. They obtain an overall accuracy of 96.7% for labeling the title pages for three different layout types on the MARG [39] database[3]. Mao et al. [71] extend this rule-based approach to be usable for a broader class of layout types, without the need to define new rules for each new layout type. They evaluate their approach on a small part of the MARG dataset and claim to have better results than the initial system. In [73] Mao et al. present a labeling method based on Hidden Markov Models. These are used to represent the layout using projection profiles. Their approach was tested on 69 pages and compared with other models. The accuracy ranges between 70% for the baseline heuristic model and 91% for their Hidden Semi-Markov model.

Liang et al. [62] present a method capable of adaptively learning the layout type (or layout model). The block-based layouts are represented as graphs which express the relative position of the blocks to each other. The model graph is matched to the unlabeled graph. The result of the graph matching combined with the model labels are used for labeling the new graph. Unfortunately no quantitative evaluation of this method is given.

Aiello et al. [2] present a complete document understanding system also capable of performing logical labeling. Features including aspect-ratio, font style and number of lines are used with a decision tree as classifier. An overall accuracy of 96.8% is reported on the UW-III [87] and the MTDB [47] datasets.

Hamza et al. [43, 44] presented a CBR approach for extraction of invoice data. Using the output of an optical character recognition system, keywords, alphabetic and non-alphabetic structures are extracted. A graph is then constructed representing the spatial relationships between the keywords and the pattern structures. Using graph comparison techniques, similar cases are identified, the solution of which is adapted to serve as solution for the new case. Different evaluation measures have been introduced. An overall system accuracy of 85.2% is reported on a private dataset.

Previous methods have all one considerable disadvantage: they have to be carefully retrained when different label sets are needed. Others rely on high-level OCR information, including text and font information. This is rarely available in an early pre-processing step. Therefore, the method proposed in this chapter uses a low-level feature set and a case-based reasoning approach. The case-based

---

[3] http://marg.nlm.nih.gov/index2.asp

approach can be considered as a nearest neighbor approach that can easily by extended to new types by adding more labeled samples.

4.3 Logical Labeling using Case-Based Reasoning

As the variation of the labels and their respective text blocks for a given invoice type is quite low, a case-based reasoning approach seems well suited for the task of logical labeling. Other advantages are that no huge training datasets are needed. Also, the distance measure for finding similar distances permits to incorporate prior knowledge common to most documents and labels. Finally, the process will be adaptable to new document types, which is easily done by adding new cases to the case base.

The idea of case-based reasoning [1] is that similar problems have similar solutions. A typical example is the case of a mechanic: the engine of a car does not work anymore. This problem is considered as the *new case*. Then the mechanics makes some observations, e.g. that starter does not work. From his knowledge of a previous case, he remembers a previous case where also the starter didn't work. This is the *retrieval* of a similar case. In that case, the mechanic checked the wiring and detected an open fuse. In the current case he also starts checking the wiring, this step is called *reuse*. Finally, he finds out that the electric cable is broken and that therefore, the starter does not work. He thus *revised* the suggested solution and *retains* a new case in his case base.

Adapting this approach to the problem of logical labeling leads to the following problem description: given the segmentation of a new document image, represented by a set of blocks and the image itself (illustrating examples can be found in Figure 4.1): find the most similar case in the case base and reuse its labels. In the proposed application, the case base consists of a number of cases representing labeled and segmented document images. Visually speaking, a case can be seen as one labeled example of a document layout type. The segments for each page are given as block coordinates: $x_{low}, y_{low}, x_{high}, y_{high}$ represent the coordinates of the lower left and the upper right corner of the block. Each block is associated with one string containing the label for this block and a feature vector, which is extracted from each block and is stored together with the coordinates and the label of that block. This feature vector is needed to compute the distance measure.

Reusing the labels after finding the most similar case is done by copying the labels to the new document. A schematic view of the presented approach is depicted in Figure 4.2.

The current section is organized as follows: in Section 4.3.1 the overall similarity measure that is used to find the most similar case is described. The additionally required block distance is described in Section 4.3.3.

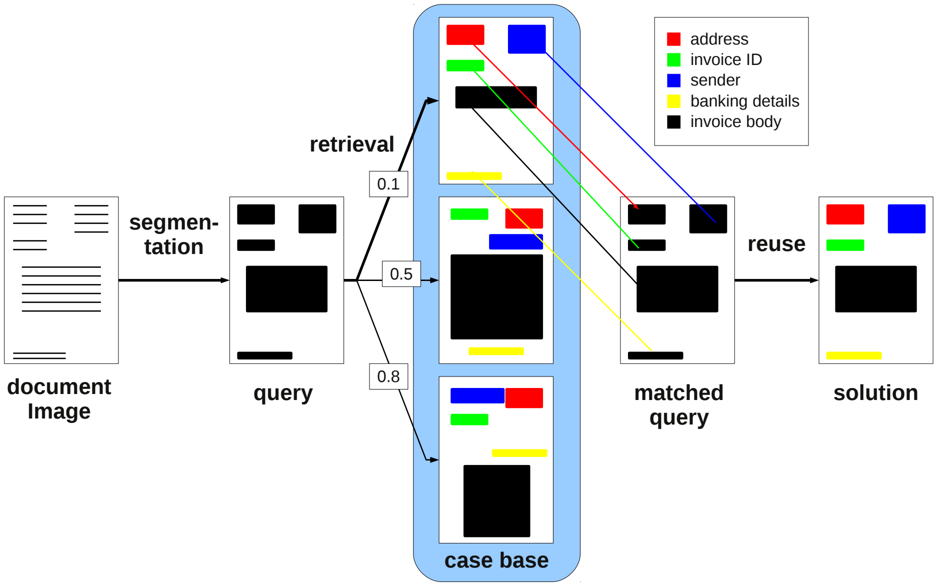


Figure 4.2: Illustration of the method: after segmenting the document image into text blocks, the most similar case is retrieved from the case-base. The most similar case is the one where the distance is smallest (in the current example $d = 0.1$). The distance measure uses positional and textural features. The most similar case is used as a reference and then the labels are transferred using the matching information obtained during the distance computation.

4.3.1 Similarity Measure

In order to find similar cases for the new unlabeled document, a distance measure has to be defined. Given a segmented and unlabeled document image, the most similar segmented and labeled document image in the case base has to be found.

The similarity measure is defined on a text block level, as this is the most frequent abstraction level used for logical labeling [54, 51, 2, 71]. The idea is to build a global measure out of a *block distance measure* that describes how similar text blocks are to each other. The overall distance between two documents is computed by assigning blocks of the query document to the blocks of the case document while minimizing the sum of the assigned block distances.

Considering two document images, I_q being the unlabeled query document image and I_c being the case from the case base for which the similarity has to be computed. Each page segmentation L_q and L_c consists of a set of blocks $B_q = \{\mathbf{b}_{q1}, ..., \mathbf{b}_{qm}\}$ and $B_c = \{\mathbf{b}_{c1}, ..., \mathbf{b}_{cn}\}$. Let $D_{block}(\mathbf{b}_{qi}, \mathbf{b}_{cj})$ be the block distance between blocks b_{qi} and b_{cj}. For the ease of notation, this distance will be written d_{ij}. The overall distance between two document images will be written as $\mathcal{D}(L_q, L_c)$.

First, the block distances for all pair of blocks is computed. This results in the following cost matrix:

$$\mathcal{C} = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{m1} & \cdots & d_{mn} \end{bmatrix} \tag{4.1}$$

The computation of the overall distance $\mathcal{D}(L_q, L_c)$ is done by assigning blocks from L_q to blocks from L_c. Different strategies are possible to assign the blocks:

- Solving the minimum weight edge cover that assigns each block of L_q to at least one block in L_c. No block will remain unassigned.

- Solving the assignment problem that assigns each block of L_q to at most one block in L_c. Unassigned blocks may remain.

- Solving the transportation problem that assigns parts of blocks of L_q to parts of blocks in L_c.

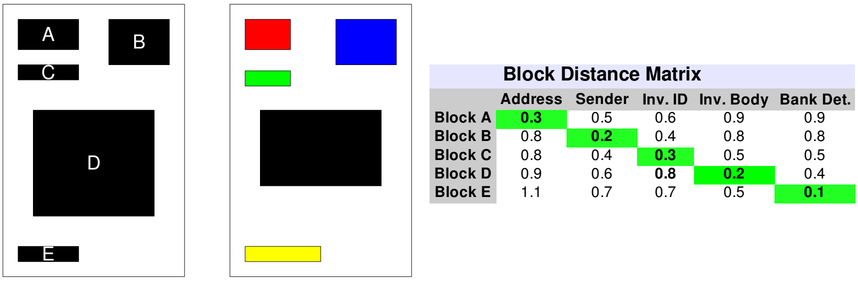
In [111] the authors showed that the minimum weight edge cover solution gives best results for layout-based document image retrieval. Furthermore, it has the advantage that no block remains unlabeled, even if n and m differ. The minimum weight edge cover is obtained using the Hungarian algorithm [55]. The result is not only the total distance between two layouts but also an assignment of each block of L_q to at least one block in L_c. As the segmentation L_c is in the case-base, an assignment between an unlabeled block b_q from L_q and a labeled block b_c from L_c is obtained, which is interpreted as b_c being a block with a same "function" in the page as block b_q.

In Section 4.3.2 the reuse step will be discussed. In Section 4.3.3, the computation of the block distances is explained.

4.3.2 Reuse

After having found the best matching document, reusing the solution of the best case is straightforward: the minimum weight edge cover returns the assignment matrix containing the information about which blocks were matched in order to obtain the minimum distance. This information is used to copy the labels from the blocks of the most similar case, which are already labeled, to the blocks of the document image.

It occurs, that a block are assigned with more than one label. This is the case when the number of blocks differs between the document image from the case base



Block Distance Matrix

	Address	Sender	Inv. ID	Inv. Body	Bank Det.
Block A	**0.3**	0.5	0.6	0.9	0.9
Block B	0.8	**0.2**	0.4	0.8	0.8
Block C	0.8	0.4	**0.3**	0.5	0.5
Block D	0.9	0.6	**0.8**	**0.2**	0.4
Block E	1.1	0.7	0.7	0.5	**0.1**

Figure 4.3: Illustration of the distance measure: “Block A” to “Block E” represent blocks from the first unlabeled document images, “Address” to “Bank. Det.” the labeled blocks of the document image in the case-base. The block distance matrix gives the block distances between all pairs of blocks. The highlighted cells symbolize the assignment of blocks: block *A* is assigned to the “address” block, block *B* to the “sender” block, etc. The total distance for this constructed example is 1.1

and the new document image. In that case a fall-back solution has to be used. One possibility could be to take the label of the block with the smallest distance, as this is more similar to the query block according to the distance measure. Instead of only choring the best case form the case base, one could also choose the n best cases and do a voting on the assigned labels.

As the focus of interest is the distance measure, the performance evaluation in Section 4.4 is done on an assignment level. Further details are given in the corresponding section.

4.3.3 Block Distance

Block similarity may be defined in many ways. In the present application, the definition of similarity (or dissimilarity) bases on two observations: first, positions of the labeled blocks in the page depend on their label and second, blocks with the same label have similar visual appearance. Both assumptions seem reasonable for fixed-layout documents, as e.g. invoices: an “address” block will be mostly found on top of the page and will have a similar visual appearance as other “address” blocks. The same holds for “invoice body” and other blocks. The “banking details” blocks’ position is much more dependant on the layout used by the invoice source: some put this information onto the right top side, some add it after at the end of the page. This variability can be handled by adding examples of each layout type to the case-base.

Both observations are combined to build up one block distance measure, composed of two components, a positional part and a content descriptive part:

$$D_{\text{all}}(\mathbf{b}_{qi}, \mathbf{b}_{cj}) = D_{\text{ov}}(\mathbf{b}_{qi}, \mathbf{b}_{cj}) \times D_{feat}(F_{qi}, F_{cj}) \tag{4.2}$$

where D_{ov} is the positional component and $D_{feat}(F_i, F_j)$ is the distance between the two feature vectors F_{qi} and F_{cj} extracted from the blocks b_{qi} and b_{cj}.

Positional Block Distance

A block distance measure that proved to work well, has been developed in [111]: in this layout based retrieval system, different block distances have been tested. Evaluation showed, that the overlapping distance worked best. Therefore, this is chosen as the positional component in the distance. It is defined as follows:

$$D_{\text{ov}}(\mathbf{b}_{qi}, \mathbf{b}_{cj}) = 1 - \frac{2 \times \text{Ov}(\mathbf{b}_{qi}, \mathbf{b}_{cj})}{\text{area}(\mathbf{b}_{qi}) + \text{area}(\mathbf{b}_{ci})} \tag{4.3}$$

where $\text{Ov}(\mathbf{b}_i, \mathbf{b}_j)$ is the number of overlapping pixels of both blocks and $area(b)$ is the number of pixels of block b.

Appearance-Based Block Distance

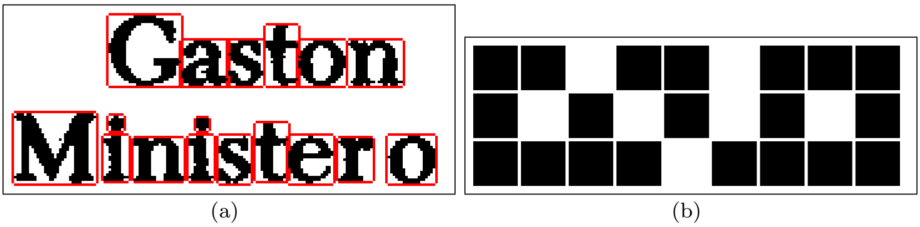
In order to cover the observation that blocks with the same labels are visually similar, content-based features are needed to describe the texture of the blocks. In the domain of image retrieval many features have been developed for textural descriptors. Many of them have found application in block classification systems. Keysers et al. have shown in [49] that the following features perform well on block classification[4].

- Run-length histograms: run-length histograms of black and white pixel sequences in four directions: horizontal, vertical, main diagonal (upper left to lower right corner) and side diagonal (lower left to upper right corner). Each histogram uses 8 bins, with exponentially increasing bin size: $\{1\}$, $\{2, 3\}$, $\{4, ..., 7\}$, $\{8, ..., 15\}$, $\{16, ..., 31\}$, $\{32, ..., 63\}$, $\{64, ..., 127\}$, $\{128, ...\}$. These values are normalized by the total number of run-length counts in all directions to obtain values between 0 and 1. In total 64 run-length features are computed, 4×8 for black and 4×8 for white.

- Connected component sizes histograms: width and height of connected components are used to compute two 8-bin histograms. An 8×8 histogram is

---

[4] Block classification is the task where the blocks are classified according to their physical type of content like images, text and drawings.



(a) (b)

Figure 4.4: Examples of the features used to describe the block content: in Figure 4.4(a) connected components are shown with their bounding box. In Figure 4.4(b) an example for run-lengths is given: the upper row contains two 2-pixel and one 3-pixel black runs, the middle row contains five 1-pixel black runs and the bottom row shows two 4-pixel black runs.

computed for the combination of width and height of the connected components. These values are also normalized by the total number of components in the block. In total 80 connected components features, 8 × 8 for the 2D histogram and 2 × 8 for the two 1D histograms are used.

Thus F is written as: $F = (f_1, \ldots, f_{144})$, where f_i are the entries of the different histograms, all concatenated into one feature vector. As a measure of distance between block content is needed, features performing well for block classification systems are good candidates for this task. Examples for both feature types are depicted in Figure 4.4.

After extracting the feature histograms from the blocks, the distance measure between two feature vectors has to be defined. As the features are represented by histograms, distance measures for histograms are used. Puzicha et al. show in [90] that the Jensen-Shannon divergence [105] (JSD) is a reasonable distance measure for histograms of textural features: Given two probability distributions P and Q of a discrete variable, the JSD is defined as follows:

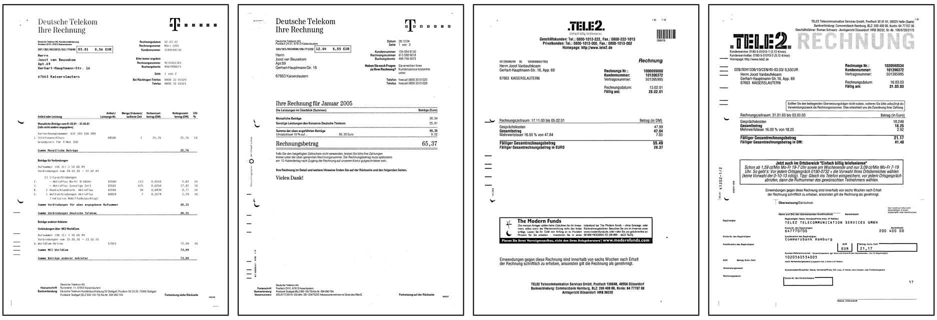
$$D_{JSD}(P||Q) = \frac{1}{2}D_{KLD}(P||M) + \frac{1}{2}D_{KLD}(Q||M) \tag{4.4}$$

where P and Q are two feature histograms, $M = \frac{1}{2}(P+Q)$ and the Kullback-Leibler divergence (KLD) [56] is defined as

$$D_{KLD}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{4.5}$$

Thus, the feature distance D_{feat} between two blocks b_{1j} and b_{2j} is defined as:

$$D_{feat}(F_{qi}, F_{cj}) = D_{JSD}(F_{1i}||F_{2j}) \tag{4.6}$$



(a) Telekom old (b) Telekom new (c) Tele2 old (d) Tele2 new

Figure 4.5: Layout types in the invoice dataset. The two images to the left show the old and the new Telekom invoices. The images to the right show the old and the new Tele2 invoices.

	Telekom old	Telekom new	Tele2 old	Tele2 new
# samples	16	41	15	23
percentage	16.8	43.2	15.8	24.2

Table 4.1: Distribution of layout types in the invoice dataset in absolute values (first row) and in [%] (second row).

4.4 Evaluation

In order to evaluate the performance of the case-based approach, two different dataset were used: first, a private dataset consisting of invoices was used to prove the general validity of the approach (Section 4.4.1). Then, a second dataset, the MARG database containing 1553 document images was chosen for deeper evaluation and also to show the flexibility of the proposed approach (Section 4.4.2). Both setups where run with exactly the same method, only using different case-bases.

4.4.1 Invoice Dataset

The invoice dataset consists of 95 documents from two different invoice sources. The generation of the ground truth was done semi-automatically and is described in Appendix C. As both invoice sources changed the layout of their documents in the considered time interval, two layout types had to be distinguished for each invoice source, leading to four different layout types. One example of each layout type can be found in Figure 4.5. The types have been named according to the invoice sources. Table 4.1 shows how many layouts of which type are in the dataset.

A leave-one-out cross validation test has been run on the invoice dataset. In this evaluation method one document image is used as “unlabeled” document and the remaining document images are used as the case-base. This process is repeated for all documents in the dataset.

For the second test the case base consisted only of 4 documents, one for each layout type (leave-allbutone-out test). The first document in alphabetical order was chosen as a representative for the corresponding layout type.

4.4.2 MARG Dataset

The MARG dataset contains binarized, skew corrected images of title pages of medical journal articles and the corresponding ground-truth. To obtain reproducible results, the ground truth blocks of the datasets were used. In a real world application, a standard page segmentation algorithm may be used.

The robustness of the method against varying layouts is tested on the MARG dataset, as it presents divisions of the data according to differing layout types. These layout types are further subdivided into different journals. To avoid splitting the dataset arbitrarily into a test- and a training-set, the following two tests were carried out. In leave-journal-out cross validation, all document images from the journal were removed from the case base before finding the best match. Analogously, leave-type-out cross validation removes all the document images of the same layout type from the case base before finding the best match. Definitions of the layout-types are given in Figure 4.6.

Finally, tests were conducted to evaluate how accuracy is effected by increasing case base size. The accuracy of the system for case-bases ranging in size from 1 to 100 has been tested. The case-bases for these tests were randomly sampled. For each size of the case-base, 50 runs were carried out.

For evaluation purposes, only the assignment level has been used: if the two blocks representing a two-column abstract are matched against a one-column abstract, two correct matches are counted. If one block is assigned to the abstract and one to the author, one error and one correct assignment are counted. The motivation for this assignment-based evaluation is that the actual matching process is the one to be analyzed, and not the fall-back solution. The accuracy is thus defined as the number of correct label assignments divided by the number of all assignments with that label.

4.5 Results

In the following, results for the different setups are presented. In Section 4.5.1, the results on the invoice dataset are given. Section 4.5.2 presents the results on the MARG dataset. The results of the experiment on the case base size are presented

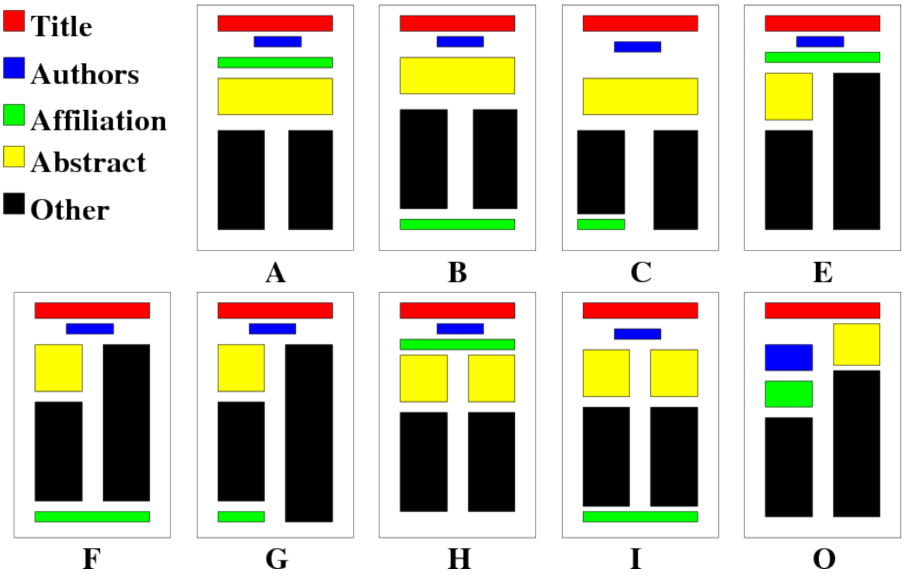


Figure 4.6: Illustration of the different layout types in the MARG dataset.

in Section 4.5.3. In Section 4.5.4 a comparison to two previously published methods is presented.

4.5.1 Results on the Invoice Dataset

The results of the test on the invoice dataset can be found in Table 4.2. For the leave-one-out test, the method works robustly with accuracies of near 100% for most labels. It should be noted, however, that the performance drops when the case base is reduced to its minimal size of one example per layout type. Furthermore, the numbers of assignments differs slightly from those from the leave-one-out test. This is due unlabeled document image being matched to the wrong layout type, containing a different number of blocks. Due to the constraint that any block is matched to at least one block, extra assignments are inserted.

4.5.2 Results on the MARG Dataset

The results for the leave-[one|type|journal]-out cross-validation on the MARG dataset can be found in Table 4.3. For the leave-one-out cross-validation the number of incorrectly labeled blocks is low. The few errors are mostly due to mislabeling of “Affiliation” and “Abstract” blocks. Visual inspection of the mislabeled

Label	leave-one-out		leave-allbutone-out	
Address	95	100%	91	90.1%
Inv. ID	95	100%	108	80.5%
Contact	95	100%	96	88.5%
Inv. Body	95	100%	93	97.8%
Bank Det.	94	97.9%	91	93.4%
Sender	79	98.7%	80	90.0%
Other	258	99.6%	250	97.6%

Table 4.2: Accuracy (in [%]) and number of assignments for the leave-one-out and the leave-all-but-one-out test. As expected, the accuracy drops if the case base is reduced to its minimum size of one example for each layout type.

Label	leave-one-out		leave-jour-out		leave-type-out	
Title	99.6	1565	99.4	1565	99.2	1565
Author	99.9	1558	99.1	1557	95.7	1561
Affiliation	99.2	1578	97.8	1580	91.0	1607
Abstract	99.7	2069	99.0	2094	93.9	2198
Overall	99.6	6770	98.9	6796	94.8	6913

Table 4.3: Accuracy (in [%]) and number of assignments for the leave-one-out, the leave-journal-out and the leave-type-out evaluation. It should be noted, that the accuracy is highest for the biggest possible case-bases as obtained in the leave-one-out test. Accuracy drops when journals or even whole layout types are removed from the case-base.

blocks showed, that these errors result mainly from the fact that a few journals have single title pages in the database where no “Abstract” block is present. In consequence, “Abstract” blocks in the best matching document will be matched to another type of block and this produces errors. Furthermore, “Title” and “Author” are being confused only rarely. This is mainly due to special versions of articles from the same journal where subtitles are present. These subtitles are labeled as “Title” in the ground-truth. Size, position and texture of these subtitle blocks are similar to “Author” blocks, so they are likely to be mislabeled.

The number of assignments in Table 4.3 gives the absolute number of assignments between blocks for the different leave-[one|type|journal]-out tests. The reason for these numbers to vary slightly from test to test is explained by an example: consider a document containing a one-column abstract, represented by one block, that is matched to a document containing a two-column abstract, represented by two blocks. This gives two correct assignments. If in the next test the first document is matched to a document having a one-column abstract (one block), it will return only one correct match. If one block is matched to the abstract, the other

Table 4.4: Distribution of layout types (typeA to typeH and typeO) in the MARG dataset in absolute values (first row) and in [%] (second row).

A	B	C	D	E	F	G	H	O
196	238	223	218	432	19	59	20	148
12.6	15.3	14.3	14.0	27.8	1.2	3.8	1.3	9.5

one to something else, one correct and one wrong assignment will be counted.

For the leave-journal-out test, on average 1547 labeled layouts remain in the dataset when all title pages from the same journal are removed before finding the best match. The main source of error for this test, apart from the problems stated above, results from mislabeling between the two classes “Affiliation” and “Abstract”. Visual inspection of the errors leads to the conclusion that in most of these cases, the best matching document does not fit to the given query layout leading increasing numbers of wrong block assignments.

For the leave-type-out evaluation, all documents of the same type are removed before searching the best match. There are in total 9 different layout types in the MARG database. They are called “typeA” to “typeH” and the ninth type is “typeO”, standing for “other types”. The distribution of the number of documents per type can be found in Table 4.4. There is a clear trend in Table 4.4 that “Affiliation” and “Abstract” mislabeling increase when the quality of the best match decreases. The performance for “Title” remains nearly constant, which is due to the unique position of the title. It is almost always above all the other blocks.

4.5.3 Considerations on the Case-Base Size

In Figure 4.7 the accuracy is plotted versus the size of the case-base. It can be seen that accuracy is reasonably high even for a small case-base. With 12 examples in the case base the accuracy rises over 90%.

An other test consisting of a case base with one example case per journal (case base size of 310) yielded an accuracy rate of 99.4%, which is close to the result obtained by the leave-one-out test.

4.5.4 Comparison to Other Labeling Methods

Comparison to other methods is a difficult task. Many different features and many different labeling domains have been previously presented. Also, some systems have been evaluated as a whole, leading to labeling errors due to problems in preceding steps. Also, most authors test their approach on private datasets, making comparison to new methods hard.

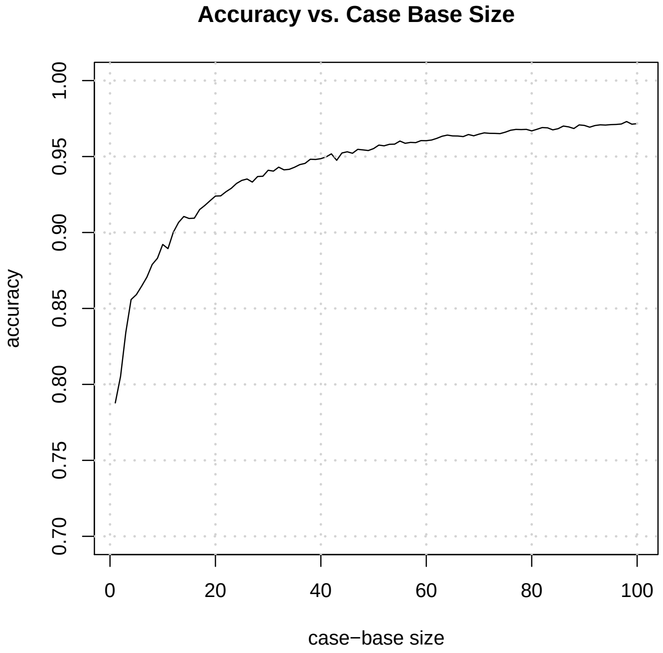


Figure 4.7: Accuracy versus case base size: on the x-axis the case base size is plotted. The y-axis shows the accuracy rate for the given case base size. It should be noted, that the accuracy rises quickly and returns good results for already quite small case bases (≥ 16)

Kim's system [51] was shown to have a mean accuracy of 96.7% on a dataset that is related and similar to the MARG dataset. In total they evaluated on 11,651 documents. The labeling module itself has an accuracy of 99.0% which is comparable to the proposed method, even if only small case-bases are used.

The extension proposed by Mao [71] has been evaluated on 49 title pages from 7 different journals. Training was done on 161 documents. Comparison to Kim's method [51] showed that their approach outperforms Kim's.

The accuracy of the zoning and labeling system presented by Mao [73] ranges from about 99% to 55% depending on the label type and the method used. Unfortunately, no overall accuracy is given.

Hamza's method [43] was evaluated on 950 document images. Their global solving approach, which is similar to the presented view of copying the labels given the closest case from the case base, was tested on 150 document images and obtained an accuracy of 85.3%.

From these results it becomes clear, that although a direct comparison is not possible, the proposed case-based reasoning method using low-level features as block positions and content descriptors works comparably well on the two tested tasks.

4.6 Conclusion

In this chapter a case-based reasoning approach was presented for logical labeling. The information extracted by the logical labeling step is needed in subsequent document security and general document management and processing applications. Using OCR in combination with the results from the logical labeling allows the system to find the respective document type for the later security check.

A case-based reasoning approach has been presented for extracting the logical labels from document images. Distance measures for fixed-layout document types were used as a similarity measure for retrieval of the most similar case in the case-base. The similarity measure uses layout block information that is used to update the labels of the unlabeled document. Evaluation of the proposed approach was done on two different datasets, one containing images of invoices and one containing title pages of journals. In both cases good accuracy rates could be shown. For the MARG dataset, accuracy rates range from 94.8% to 99.6%, depending on the number and the quality of the remaining layouts in the dataset for the leave-[one|type|journal]-out cross validation test. It could be shown that even with small case-bases the accuracy rates are reasonably high.

Another advantage of this approach is its flexibility concerning the labels: the labels are defined by the labeled examples that are presented to the system. So the system can easily be adapted to other labeling tasks, e.g. business letters.

The presented test setup works with ground truth labeling information. In a real world setup, a standard segmentation algorithm e.g. Voronoi [52] can be taken to compute segmentations of the documents. The output of this segmentation can then be used to label the blocks.

Chapter 5

Model-Based Authentication

Authenticity of printed documents is an important aspect in business relations. As many documents present money-worth claims, the receiver of a document has to be sure, that the document is genuine. This is especially true if the document originates from an untrusted source.

Most methods for authentication add extra features to the document. In this chapter, two approaches are presented to secure documents against forgeries generated by using standard computer hard- and software. The first approach uses the counterfeit protection system codes produced by color laser printers and copiers for automatic authentication of the source.

The second printing technique independent approach uses distortions introduced by scanning and printing to detect forged documents. The positional variations of characters in fixed headers and footers are used for this goal.

The following contributions are presented in this chapter[1]:

1. The first research contribution for document authentication using the document intrinsic counterfeit protection system (CPS) codes for printer type classification (accuracy of 92.5%, $n = 67$, 16 different classes).

---

[1] This chapter is based on the author's work in
[114] J. van Beusekom, M. Schreyer, and T. M. Breuel. Automatic counterfeit protection system code classification. In *Proc. of SPIE Media Forensics and Security XII*, San Jose, CA, USA, January 2010,
[116] J. van Beusekom, F. Shafait, and T. M. Breuel. Document signature using intrinsic features for counterfeit detection. In *Proc. of the 2nd Int. Workshop on Computational Forensics*, volume 5158 of *Lecture Notes in Computer Science*, pages 47–57, Washington, DC, USA, August 2008, and
[115] J. van Beusekom, F. Shafait, and T. M. Breuel. Image-matching for revision detection in printed historical documents. In *Proc. of the 29th DAGM Symposium on Pattern Recognition*, pages 507–516, Heidelberg, Germany, 2007.

2. The first approach for image-based authentication of two CPS patterns (accuracy of 88.3%, $n = 94$).
3. The first public data set for evaluation of methods using the CPS codes is generated, in cooperation with the Electronic Frontier Foundation.
4. A novel approach for document authentication using scanner distortions by measuring the document intrinsic positional variation features. Evaluation on a new data set showed the approach's good performance, achieving an accuracy of 97.0% ($n = 168$).
5. Generation of two new data sets for evaluation of document forgery detection methods.

5.1 Introduction & Related Work

J. Hails [41] defines authentication in the context of documents as "*. . . showing that writing is what it is claimed to be*". In the context of invoices, the focus is to make sure that the invoice has been created by the person or company that it claims and that all the contents are genuine.

Authentication is often ensured by signatures: these represent a hard to forge feature that only the creator may have added to the document. It can thus be used to prove the document's originality. Signatures have always been a critical issue, even in ancient times, where the number of paper documents was limited, compared to their tremendously wide-spread use nowadays. In these days, where modern technologies enable a broad mass of people to easily counterfeit documents and invoices, it becomes more and more important to assure the genuineness of a document.

The signet rings from the monarchs that were used to sign the documents in ancient times have nowadays been replaced with all kinds of modern security features. A good overview over signature features for document security is presented by van Renesse [122]. Several categories of features are distinguished, depending on the effort to detect or verify them. Three different levels effort, called *level of inspection* are commonly being distinguished:

- **first line inspection:** this groups all the inspection methods that can done using only the human senses. Examples of signatures that can be used in a first line inspection are watermarks and holograms.
- **second line inspection:** inspections requiring additional tools to verify a feature or a document fall into this category. Examples are ink that is only visible under ultra violet light or bar codes that are only readable using a bar code scanner.

- **third line inspection:** this category covers the more sophisticated analysis that is normally done by experts, e.g. questioned document examiners. Examples are physical and chemical analysis of the ink composition to date a document.

There is a vast variety of different features available to make documents relatively secure even through first line inspection, e.g. using special sort of paper, eventually integrating a watermark, by adding holographic images [106], specialized printing techniques [4] and other physical and chemical signatures [42]. Many other types of features can be found in literature ([121, 124, 123]).

Having holograms, bar codes or other extra security features is certainly a reasonable way to make documents more robust against tampering. They require, however, extra steps before or during the creation of a document, which will dramatically increase the costs of producing these. A more practical solution has to be found for assuring the authenticity of documents of every day life without adding extra features.

Therefore, in this chapter, the focus is on using features “inside” the document as a signature, the so called *intrinsic* features. In contrast to *extrinsic* features that are added only for document security purposes, intrinsic features are byproducts of the normal document generating procedure. This has the advantage that the creators of the documents can continue to use their usual technique to generate the documents and while achieving a certain degree of tampering resistance.

Several uses of intrinsic features have been presented in previous publications: printer identification, the process of assigning a print-out to a unique printer or a printer type, has been intensively be studied by different groups. Mikkilineni et al. [3, 74, 50] present features that can used to determine the model of laser printer that was used to print a document.

Schreyer et al. [99, 98, 100] worked on detecting the printing technique used to print a document. Also classification between printed documents and copied versions of printed documents has been analyzed. Using discrete cosine transform features, good performance could be shown even when scanning with moderate resolutions of 400 dpi.

Another frequent intrinsic document feature is handwriting. Many printed documents contain handwritten parts as notices or signatures. Two related questions can be distinguished: off-line writer identification and off-line signature verification. In this context, off-line means that only the image of the signature or the handwriting is available, in contrast to online data, where stroke information is also used. The first problem consists in identifying the writer of a document in question using a previously trained model of the writers handwriting [96, 97, 13]. In signature verification the question is whether a signature on a document has been generated by the person claimed by the signature or if someone else forged the signature [27, 89].

Considering the scenario of an insurance company processing incoming invoices, it is likely that different invoices form the same invoice source are processed over time. This information can also be used for authentication. If a new invoice from the same source suddenly looks different or was printed on a different printer, it could be that the document has been tampered. The assignment of the invoice to previously encountered invoices can be accomplished by the textual information extracted after logical labeling (Chapter 4).

In this chapter, two different document intrinsic approaches are used for authentication of documents: the first approach uses counterfeit protection system codes[2] and the second uses the positional variability modeling of the page content.

The CPS codes consist of small yellow dots that are invisible with the bare eye. These dots are generated by many color laser printers and color copiers to allow backtracking of possible fraudulent use of these printers. Each printer integrates a unique pattern that can thus be used to identify the printer that was used to generate the document. As information about the interpretation and the decoding of the CPS codes is classified, an image-based approach is presented to use these codes for authentication. The first approach does a printer classification of the pattern based on the separating distances of the base pattern to neighboring repetitions. The second approach tests if two pages contain the same pattern or not. If the patterns differ, this could be a hint that a document has been forged. Details about the CPS based methods are given in Section 5.2.

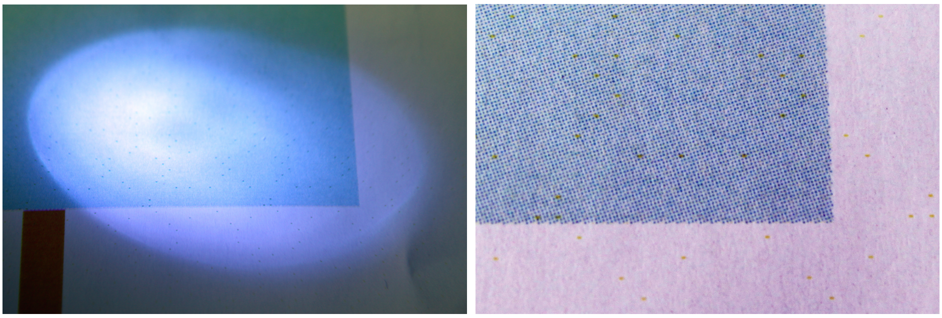
The second approach to authenticate documents uses positional variation modeling of the page's content. This approach bases on two observations: first, most methods involving consumer usable scanning and printing technologies slightly distort the layout of the page. Second, many documents have regions that are the same for all invoices from the same source, as e.g. headers or footers. This is especially true for invoices. In these regions, the distortions can be detected and this is used to identify possibly forged documents. This approach is explained in detail in Section 5.3.

5.2 Authentication using CPS Codes

The advent of new printing technologies, especially the wide spread use of color laser printers and color copiers has simplified the way to create printed documents in high quality. Recent cases reported to the American Society of Questioned Document Examiners (ASQDE) revealed their increasing involvement in the production of counterfeited banknotes [28, 60] and forged documents [84].

---

[2] One might argue that this feature is not intrinsic as these are added "by purpose" on the print-out. The author, however, considers this feature as intrinsic as it is added without any additional intervention by the document creating party.



(a) Using blue LED (b) Using high resolution camera

Figure 5.1: Examples of CPS dots on a print-out. The left picture shows the dots by making them visible using a blue LED light source. The right image shows a detail of a high resolution image taken from a part of the page. From the visual appearance of this pattern, it is likely, that an HP color LaserJet was used [110].

In an effort to stop the use of color laser copiers and printers as a counterfeiting tool, CPS codes are printed on the documents. The code consists of a pattern of yellow dots that is invisible for the unaided human eye. An example of such a pattern can be found in Figure 5.1. The pattern is distributed on the entire surface of a document but may be overprinted by text or other document content.

Sending such document to a licensed laboratory allows the examiners to extract the so called *inspection code*. This is transmitted to the manufacturer of the printer. The manufacturer will then check his customer database to find the owner of that specific printer.

In public only little is known about this technology. Also, it is not clear who has access to the decoding technology and for what purposes it is being used. The Electronic Frontier Foundation attracted the wider public's attention to privacy issues related to these codes, also called tracking dots, yellow dots or machine identification codes. Their work on detecting and decoding tracking dot patterns is presented on their web page [37]. An interesting set of slides [95] from a talk held at the *Chaos Communication Camp 2007* on this subject gives a good overview over the few publicly known details about the tracking dot patterns. Also, the *Computer Counter Culture Group* at the MIT Media Lab started a campaign to stop the use of CPS codes [75] due to the privacy issues related to these codes.

As detailed information about the decoding of the patterns is classified, no direct application of the pattern can be used for authentication. Several observations, however, have been made:

- **base pattern:** although the yellow dots are present all over the page, only a small subset of dots actually contains the printer specific information. This base pattern is then repeated all over the page.

- **stability:** for most printers the base pattern is the same for all print-outs. Only for some Xerox printers also the time and the date is encoded into the base pattern.

- **pattern types:** different manufacturers have different pattern types. Canon patterns, e.g. consist of a small, irregularly looking base pattern that repeats many times. Xerox type patterns however are larger and placed on a regular grid.

From all these observations, it can be concluded that an image based approach can be followed to use the CPS codes for document authentication: visually different patterns on two print-outs indicate that two different printers have been used. If the patterns have the same appearance and the base patterns are the same, it can be concluded that the same printer was used for both print-outs. The approach followed in this section is to use the geometrical pattern itself for authentication of the document, instead of extracting the pattern and decoding it.

In this section, an automated authentication on printer-class level is presented: by detecting the printer class on which a document was printed, it can be verified if the same class of printers was used for the questioned document as for previously seen documents from the same invoice source. This presents an important first barrier to forgers as the forger does not necessarily have the same printer type as the original document generator. The classification follows the methodology of Janis S. Tweedy [110]. The vertical pattern separation (VPS) distances are used to distinguish 13 different classes of CPS codes. Several printer types or manufacturers are assigned to each class. The second barrier is build by detecting if exactly the same printer was used. This can be done by verifying if the pattern on the questioned document is exactly the same as in the previously seen invoices.

Automatic image-based comparison of CPS codes is a challenging task: first of all, the dots have to be extracted, which is explained in Section 5.2.1. As these dots are very small and their color is close to the color of the white background, a specialized binarization method is needed to segment the dots from the background. For computation of the HPS and VPS distances, presented in Section 5.2.2, it is important to have a robust method as it regularly occurs that dots are missing (due to text printed on the dots) or that noise leads to extra dots. Detecting if two base patterns are identical (Section 5.2.4) is not less complicated: it has to deal with all of the above problems and in addition to that it has to make a hard decision if two patterns are identical or not.

5.2.1 Extraction of the Tracking Dots

In a first step, the tracking dots have to be extracted. Tracking dots are small yellow dots printed on the document. Their size is about 0.007 inch (approximately 4 pixels in a 600 dpi scan). Examples are shown in Figure 5.1.

Several ways are possible to extract the tracking dots: MIT Seeing Yellow [75] proposes to extract the blue channel. Conversion to CYMK and using the Y channel is also possible. A subsequent binarization step [83, 94, 101] could be used to decide between foreground (dot candidate) and background. This approach, however, leads to many false candidates.

Therefore, a manually tuned extraction was chosen: examining the color values of the dots on different pages showed that in RGB color space the R and G values are close to 255, while having a slightly lower B value. The following binarization method was deduced:

$$I(x,y) = \begin{cases} 0, & \text{if } min(I_R(x,y), I_G(x,y)) - I_B(x,y) < T_1 \text{ and} \\ & I_R(x,y) > T_2 \text{ and } I_G(x,y) > T_2 \\ 255, & else \end{cases} \tag{5.1}$$

where $I_R(x,y), I_G(x,y)$ and $I_B(x,y)$ are the values of the red, green and blue channel of image I at position (x,y), 0 being black and 255 being white. A good choice for the thresholds are $T_1 = 20$ and $T_2 = 240$. The result of this step is a binary image where the black pixels represent dot candidates.

This task specific binarization has the advantage that it returns a relatively good set of dot candidates with only few isolated noise pixels. Also, the number of missed dots using the proposed extraction is negligibly low.

Problems occur if areas with dithered colors are present in the image. After binarization, these areas tend to show significant amounts of pixel noise dots similar in size to the ones form the CPS code. As these may be very numerous it is preferable to remove these dots to make the system more robust. This filtering is done by morphological closing (dilation followed by erosion) using a quadratic mask. The effect is that dots closer than half of the width of the mask are being connected together, whereas singular dots remain unchanged. A computationally fast implementation using run length encoding for binary morphology was used [22]. The width of the mask is fixed to half of the median distance between the neighboring connected components, which is normally about 12 px.

From the resulting image the connected components are extracted. The filtering is done on a connected component basis: big connected components (width or height bigger than 4 px) are being ignored. This also removes the dithered parts of the image, as these remain connected after the filtering.

Example images in Figure 5.2 show the intermediate results for each step. It should be noted that the resulting set of dots is not perfect, e.g. there may be

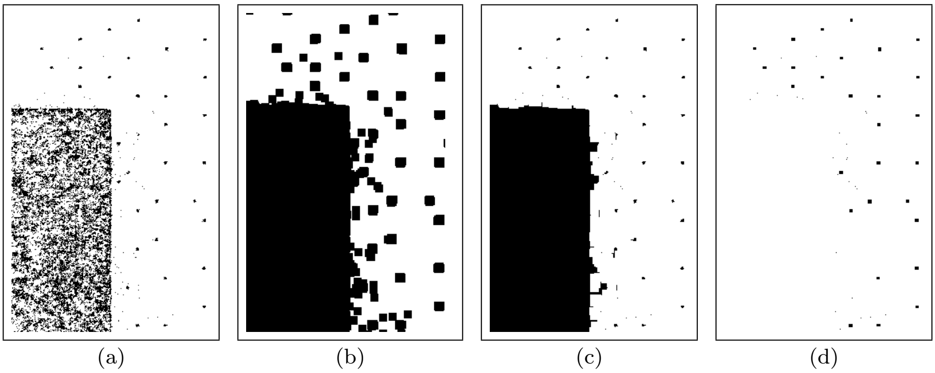

(a) (b) (c) (d)

Figure 5.2: Visualization of the CPS dot extraction: (a) shows binarized image using the task specific binarization. After dilation image (b) is obtained. Erosion of (b) leads to (c). Finally, using connected component based filtering the final image (d) is obtained where the black pixels represent the extracted dots.

false positives (due to noise or printer artefacts) or missed dots (due to overprinting or due to the filtering), which will lead to increased requirements concerning the robustness for the next steps.

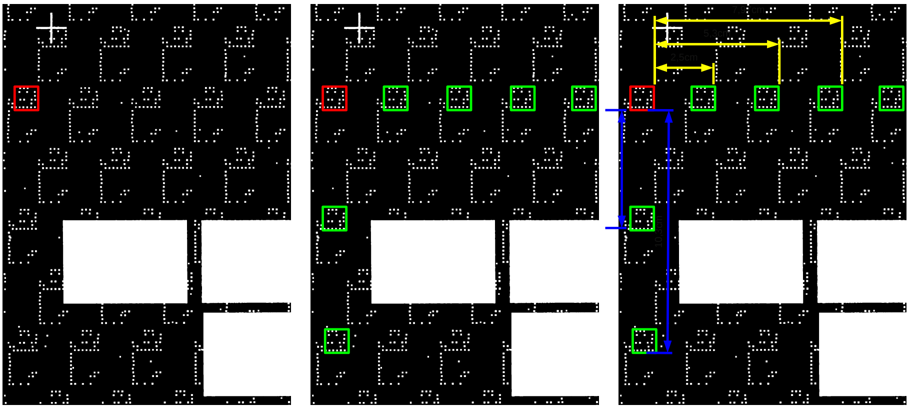
5.2.2 Printer Class Identification

The proposed method for printer class identification uses a feature and a classification scheme proposed by Tweedy [110]. It uses the vertical pattern separating distances. This idea is extended in this chapter to use also the horizontal pattern separation (HPS) distance to cope with the problem that depending on the paper feed inside the printer, the CPS codes may be rotated by 90°. Thus, the proposed method extracts both, the HPS and the VPS distances from a print-out.

The computation of the distances is done by matching a randomly selected sub pattern of dots, the so called *search pattern* on the extracted dots itself. The translations from the search pattern to the matched patterns are measured. The histogram of measurements shows peaks at positions that are multiples of the HPS or VPS distance respectively. In the next section, the details concerning the computation of the HPS and VPS distances are presented. A visualization of the main idea is given in Figure 5.3.

Computation of Horizontal and Vertical Pattern Separation Distances

After extracting the dots, the next step consists of computing the vertical and horizontal pattern separation distances (VPS and HPS respectively). These dis-



1. select sub-pattern 2. match sub-pattern 3. compute HPS/VPS

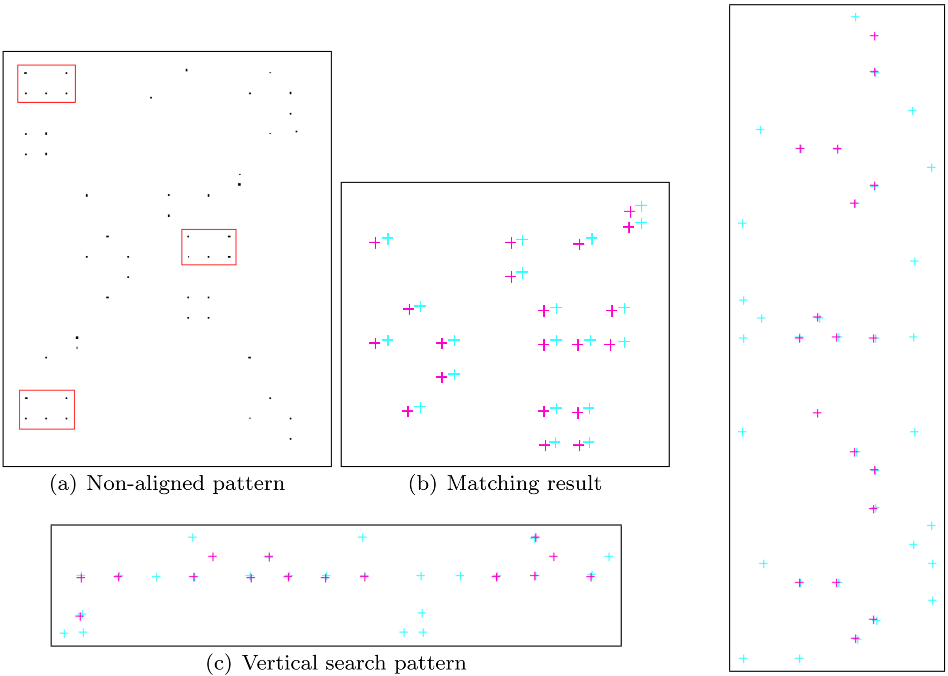
Figure 5.3: Computation of HPS and VPS distances: first, a sub pattern is selected. This is matched at different positions in the same column or row respectively. The computed translation parameters in x and y direction are used to extract the HPS and VPS distance of the pattern.

tances are used for the subsequent classification scheme. For reading simplicity, in the following, only the computation of the HPS distance is discussed. The VPS distance follows an analogous scheme, just switching the directions x (horizontal) and y (vertical).

The approach for computing the HPS takes a random local subset of the tracking dots and matches this subset to the remaining dots at approximately the same y position. For each obtained match the translation parameter t_y in y direction is returned. Statistics on these values are used to estimate the HPS and VPS distance of a pattern.

The reason for only considering matches on the same y position lies in the fact, that not all CPS pattern repetitions are on a regular grid. Some CPS code repeat their *base pattern* shifted in x direction in neighboring columns. A base pattern is defined as the smallest set of dots that can not be split into equal sub patterns and that explains all the tracking dots in the image by just translating it in x and y directions. An example of such a repetitive pattern with offset is shown in Figure 5.4(a). Without this restriction, the matching would return distances that represent only a fraction of the real distance.

The local pattern subset has to be chosen carefully: it must be assured, that its width is not bigger than the smallest known HPS distance. Elsewise, only multiples of the distance will be found. Another constraint is that the pattern should be big enough to allow for robust matches. If only a few (e.g. two or three) dots are to be matched, many matches on random positions will be found. Therefore, for each direction a different search pattern is used for the matching: starting from a randomly selected point, an area around this point is defined such that in the direction of measurement its size is smaller than the smallest known pattern separation distance. In the other direction, it is extended to a wider area to cover more points. This allows the method to find more robust matches. A visualization of such search patterns is given in Figures 5.4(c) and 5.4(d).



(a) Non-aligned pattern

(b) Matching result

(c) Vertical search pattern

(d) Horizontal search pattern

Figure 5.4: Example of different patterns: Image (c) and (d) show an example of a horizontal and a vertical search pattern. Image (a) shows an example of a non-aligned pattern. Image (b) shows the result of the matching: the blue crosses represent the sub pattern to be searched for, the red ones represent the repeating patterns above or below the subset pattern. The alignment has been slightly displaced in order to be able to see the different crosses.

Matching the Search Pattern

After choosing the search patterns, the positions where the dot pattern in the same geometric relation can be found have to be computed. Visually speaking, one wants to find all repetitions of the horizontal search pattern on the right or left side of the search pattern. Similarly the vertical repetitions of the vertical search patterns are computed. This is done using the technique described by Breuel in [17]. It uses an optimal branch-and-bound search algorithm, called RAST (Recognition by Adaptive Subdivision of Transformation Space). This method allows robust and accurate finding of the positions of repetitions of the search patterns.

The RAST algorithm performs a branch-and-bound search on the parameter space. For each direction, a separate search is done. The transformation parameter space can thus be reduced to parameter $t_x \in [-W, W]$ (in the case of the VPS distance: $t_y \in [-H, H]$) where W and H are the page width and height. To be more robust against small distortions of the page, t_x is set to allow for small variations, too. Examples of the matching results can be found in Figure 5.4(b).

Estimating HPS and VPS Distances

The approach to solve this problem is to measure recurring translation values. If a pattern has a certain HPS distance, the translation values returned by the search are most likely multiples of the HPS distance, apart from a few “noise” matches. Using several iterations of the search, a histogram of translation values is generated that shows characteristic peaks having a distance approximately equal to the HPS distance.

As the selection of the search pattern is a random process, it may happen that the selected search pattern does not represent any pattern but e.g. only consits of noise dots. It may also happen that a search pattern is chosen that is repeated inside the base pattern. Therefore, several iterations of search pattern selection and matching are run, each returning a set of matching parameters. All these results are collected in one priority queue $R = \{(t_{x,1}, q_1) \dots (t_{x,n}, q_n)\}$, where q_i is the quality of the match i and n is the total number of results returned by all iterations. The quality q_i is the number of dots from the search pattern that could be matched to the remaining dots. A low quality thus means, that only a few dots from the search pattern could be matched. This quality is used to select the best m matches for generating the statistics, as matches with higher quality lead to a higher degree of robustness.

Using the m best results, a histogram of the translation values t_x is generated by computing all pairwise distances. An example of such a histogram is shown in Figure 5.5. Using this histogram, the HPS distance is computed by histogram comparison: for all possible HPS distances, a reference histogram is generated. It is generated by distributing equally high peaks in the histograms at all different

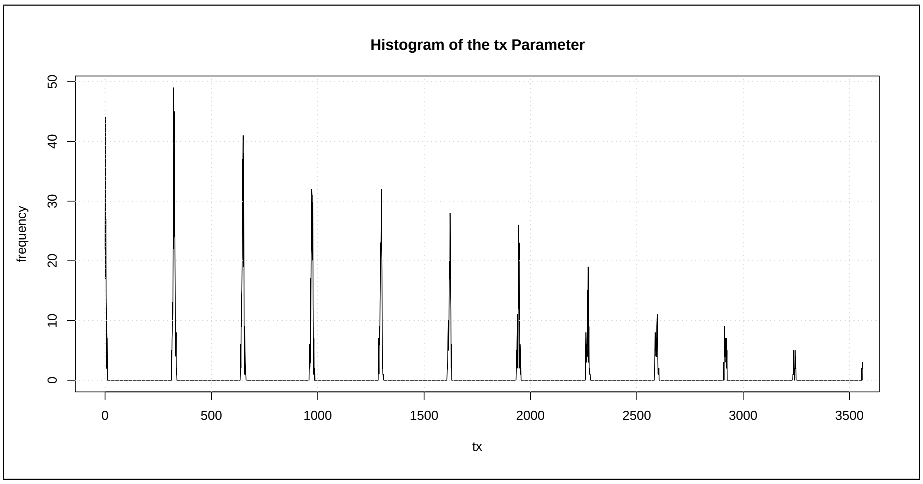


Figure 5.5: Histogram of the values of the parameters t_x. One can observe the peaks with the constant distance that is equal to the HPS distance in pixel.

distances for HPS. All the reference histograms are compared with the measured histogram using Jenson-Shannon-Divergence (JSD) [105]. The reference histogram with the smallest JSD gives us the HPS distance.

5.2.3 CPS Code Comparison

In this section, the image based method to compare two patterns is explained. The purpose of this method is to check if the patterns from two different pages are the same or not. Simply speaking, approximations of the base patterns are extracted from the two images and compared with each other. If the two base patterns are the same, the same printer was used.

Automating this process is less trivial than it may seem at a first glance: first of all, there is the problem of extracting a unique base pattern: the separating distances do not give any information about where the base pattern starts and ends, thus leading to ambiguities when extracting a base pattern. Moreover, only the HPS and VPS distances are known, but not the exact size of the base pattern. An example for this ambiguity is shown in Figure 5.6. Most CPS patterns seem to have marks that show the start and the end of a pattern. Hard coding these marks for all different pattern types is no option, therefore, a different solution is needed. Another problem with the base pattern extraction comes from missing and extra dots: these may be due to noise, dithered areas or dots being printed on text regions. Thus, an area of the width and the height determined by the HPS

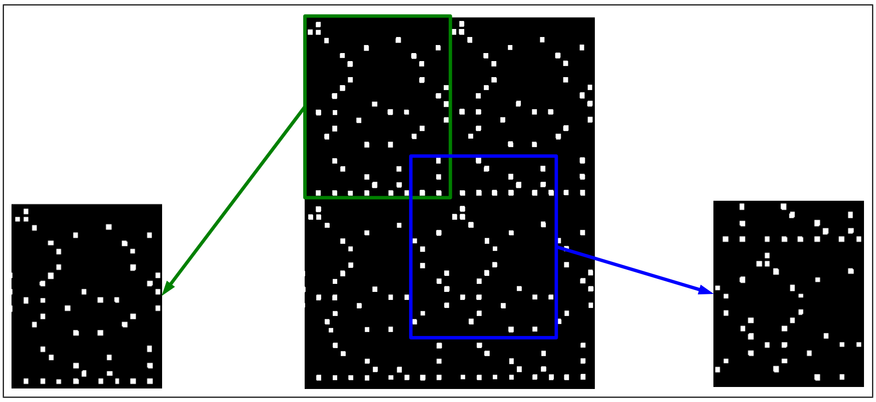


Figure 5.6: Example demonstrating the prototype ambiguity: from the dot image in the center, several possible prototypes might be extracted if the start and end points are not known.

and VPS distances will not work.

Therefore, instead of a base pattern, a *prototype* is generated that allows to deal with the afore mentioned problems. This prototype has the width and the height of the HPS and VPS distance respectively. It also incorporates information about the frequency of the dots it contains. Also, the prototypes are not compared directly with each other in order to make a decision if both patterns are identical. Instead, the prototype from the first image is matched to the second image to find out how well its dots are explained by the second image and vice versa. If all relevant dots can be "explained", the documents are considered to be printed on the same printer. This avoids the problem of aligning the different prototype patterns to each other, a problem resulting from the missing start and end point information.

In the next section, the prototype generation will be explained.

Prototype Pattern

As the tracking dots extraction is not perfect in a sense to deliver all and only tracking dots, extracting a prototype pattern on a random position in the page may lead to a prototype having extra dots or missing dots.

The method to extract a reliable prototype uses a similar approach as for the computation of the HPS and VPS distances (Section 5.2.2): first a prototype pattern is extracted from a random location. Then, this pattern is matched to the dots all over the page, similarly as for the VPS and HPS computation. But this time, no restriction on the x or y position is made.

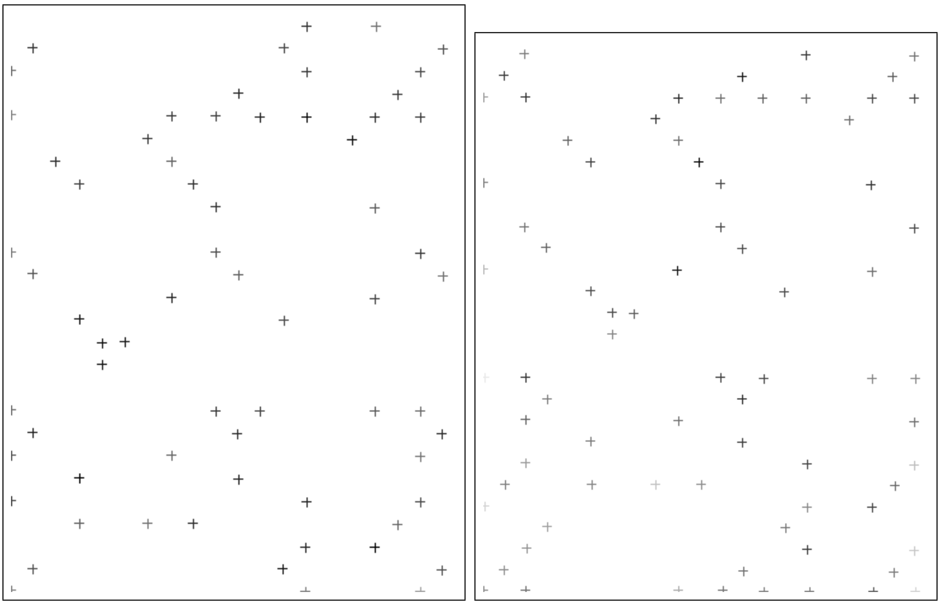


Figure 5.7: Examples of extracted prototypes. The darker the cross is the higher its frequency of occurring in a match.

For each match, a matching list defining what point from the prototype is matched to what point from the matched area is obtained. If two dots match, they are clustered together and their frequency count is increased. If not, a new prototype dot is created.

As the process depends on the choice of the first prototype and as, just as in case of the search patterns, the region where the prototype pattern is extracted from does not necessarily show a good example of the pattern, this process is repeated several times (for a practical application, five iterations showed to work well enough).

From these matchings it is possible to compute how frequent a dot at a certain position has been encountered. Dots belonging to the base pattern pattern will have high frequencies whereas noise dots will be much less frequent.

A visualization of the model is depicted in Figure 5.7. The darker the cross, the higher its measured frequency.

Prototype Comparison

In this step, the extracted prototype of one image is verified against the dots in the other image. A decision has to be taken whether the prototype fits the image or not. The idea to solve this is the following: if a dot is frequently present in one prototype pattern, the other image should also show this dot. If a dot is less frequent in one prototype, it should also be less frequent or even missing in the other image.

Thus, the differences of frequencies of the pattern dots are analyzed. A threshold has to be fixed defining what differences are considered as significant. Consider a normal CPS pattern on a standard page: it should be easily possible to find multiple occurrences of the pattern on that page. It might be that one or the other dot might be missing but it is unlikely that by chance always the same dot is missing. On the other side, if random noise dots appear, it is unlikely that they will appear in the same position for different repetitions of the pattern. Therefore, the threshold is set to a high value: infrequent noise will be discarded and for the relevant pattern dots, minor variations in frequency will not influence the outcome. A pattern is defined as different if there is at least one significant difference in frequency between the prototype pattern and the matched prototype pattern in the image.

5.2.4 Evaluation

No public dataset was available for testing the proposed method. In cooperation with the Electronic Frontier Foundation (EFF), the first dataset for evaluating methods on CPS codes was generated[3]: in an attempt to gain more information about the CPS codes, the EFF invited the visitors of their web site to send print-outs of the provided test documents, together with additional information as e.g. the manufacturer, the printer model and its serial number. In a first step, these documents had to be scanned. The scanner used for this task is a Fujitsu *fi-4120C2* automatic document feeding scanner with a maximum optical scanning resolution of 600 dpi and a color depth of 24 bit. All documents were scanned using 600 dpi in full color mode.

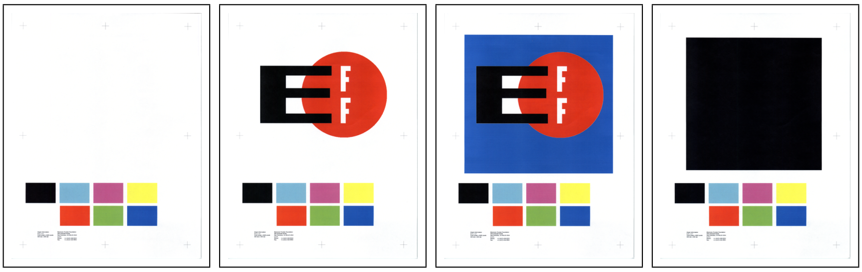
A first sample set of 68 sets of test print-outs has been scanned. Each set consists of 8 pages from the EFF printer test set sheets[4]. An example of such a sample set of 8 images is given in Figure 5.8. On a set level, the ground truth has been generated manually. It contains the following information:

- manufacturer of the printer / copier

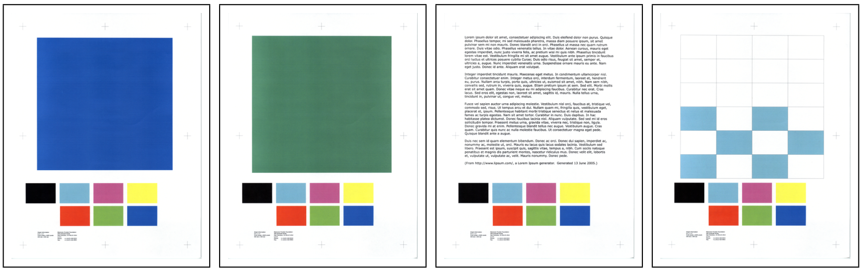
---

[3] The dataset can be downloaded from http://madm.dfki.de/downloads-ds-mic, last retrieved on July, 12th 2010

[4] http://www.eff.org/wp/investigating-machine-identification-code-technology-color-laser-printers#testsheets, last retrieved on July, 12th 2010



(a) Sheet 0 (b) Sheet 1 (c) Sheet 2 (d) Sheet 3



(e) Sheet 4 (f) Sheet 5 (g) Sheet 6 (h) Sheet 7

Figure 5.8: Test sheets from the Electronic Frontier Foundation.

- serial number of the printer / copier used to generate the print-outs
- presence or absence of dots
- manually measured horizontal pattern separation distance
- manually measured vertical pattern separation distance
- vertical pattern separation distance according to the classification given in the paper by Tweedy [110], if available
- miscellaneous information

Evaluation of CPS Code Classification

For the evaluation of the classification based on the HPS and VPS distances, the different classes have to be defined: Tweedy [110] proposed thirteen different classes of VPS distances. For the 0.32 and 0.64 inch distances, Tweedy distinguished two different subclasses based on different visual characteristics. As these subclasses have the same VPS distance, these are merged into the same class for the current

```
0.48 0.96
        Epson Aculaser C1900
        Epson AcuLaser C900
        Epson AcuLaser C1500
        Konica Minolta Magiccolor 2300 DL
        Minolta QMS Magicolor 2210
        Minolta QMS Magicolor 2300DL
        Konica Minolta Magicolor 2300 DL
        Konica Minolta Magicolor 2430 DL
        Minolta QMS Magicolor 2430 DL
        Minolta Magicolor 2300 DL
        Minolta QMS Magicolor 2300 DL
        Konica Minolta Magicolor 2430 DL
        Minolta QMS Magicolor 2300 DL
        Minolta QMS Desklaser 2200
        Konica Minolta Magicolor 2430 DL
0.54 0.69
        HP Color Laserjet 3700DN
        HP Color LaserJet 4700
        HP Color LaserJet 3700DN
        HP Color LaserJet 2840
        HP Color LaserJet 2600N
        HP Color LaserJet 2605DN
        HP Color LaserJet 4600
        HP Color LaserJet 4600
        HP Color LaserJet 2550N
        HP Color LaserJet 3700
        HP Color LaserJet 4650
        HP Color LaserJet 4600
        HP Color LaserJet 4600DN
        HP Color LaserJet 2500
```

```
0.69 0.54
        Lexmark C910
        Lexmark C912
        Kyocera C2630D
        HP Color LaserJet 5500
        HP Color LaserJet 5500DTN
0.64 1.28
        Epson Aculaser C1100
        Epson Aculaser C3000
        Dell 3110CN
        Dell 5100CN
        Xerox Docucolor 1632
1.28 0.64
        Xerox Phaser 790
        Xerox DocuColor 6060
        Xerox DocuColor 2045
        Xerox DocuColor 12
0.96 0.48
        Minolta  DialtaColor CF 2001
        Minolta-QMS Magicolor 7300
        Konica Bizhub C252
        Konica Minolta Bizhub 350C
2.54 0.16
        Canon CLC-iR C3220-C1
-1 -1
        Xerox Phaser 8400N
        Xerox Tektronix Phaser 7750DN
        OKI C9200 Model N31061A
        Samsung CLP-500/XAA
        Brother HL-2700 CN
        OKI C5150n
```

Figure 5.9: Excerpt of the list of printer classes in the evaluation data set. The HPS and VPS class is given at the top, followed by the printers in this class. An other list of printers can be found in [110].

experiments. Thus, eleven different classes are considered: no CPS pattern, 0.16, 0.32, 0.48, 0.50, 0.54, 0.64, 0.69, 0.96, 1.20 and 1.28 inch distances. A list of printers corresponding to the different VPS distances is given in [110]. A list of HPS and VPS pairs and the associated printers can be found in Figure 5.9.

As two different measures are computed per page, three different accuracy rates are being computed: the accuracy of computing the correct HPS distance, the accuracy of computing the correct VPS distance and the accuracy of correctly computing both VPS and HPS distances. A list of the different pairs of HPS and VPS distances with more than two occurrences is presented in Table 5.1. In total 16 different pairs could be identified in the dataset.

Sheet 6 (Figure 5.8(g)) of each sample contains the most realistic document type for the proposed scenario, namely a page containing mainly text regions. As not all samples are complete. in total 67 images have been used.

The computation of the HPS and VPS distances is done using the method

HPS distance [in inch]	VPS distance [in inch]	# Occurrences
0.48	0.96	15
0.54	0.69	14
-1.00	-1.00	11
0.69	0.54	5
0.64	1.28	5
1.28	0.64	4
0.96	0.48	4
-1.00	0.16	2

Table 5.1: List of pairs of HPS and VPS distances occurring more than once. Documents without dots are represented with the pair $(-1.0, -1.0)$. A -1.0 in any other row means that during ground truth generation, the respective HPS or VPS distance was not measurable due to a sparse pattern (Section 5.2.5).

above. The classification is done using a simple threshold: if the difference between the ground truth distance and the computed distance is less than 0.02 inch, it is considered to be in the same class. If no tracking dots are present, no reasonable matches for the search patterns can be found. The document is then considered to be CPS code free.

Evaluation of CPS Code Comparison

The test setup for the evaluation of the CPS code comparison is as follows: a set of image pairs was defined to compare print-outs with same HPS and VPS distances. For each sample set, Sheet 3 was compared with Sheet 6 and Sheet 6 was compared with Sheet 3 of the numerically next sample set. If Sheet 3 is not present for a sample set, the next Sheet in numerical ascending order is taken (Sheet 4, etc.). By this method a balanced set of 94 comparisons was generated. The full list of files that have been compared is presented in Appendix B. The threshold which decides if a dot significantly differs from one pattern to the other is set to 0.9.

5.2.5 Results for CPS Code Classification

The results for the test on the 67 images containing text and graphics are shown in Table 5.2. It can be seen, that the results for the VPS distance are slightly better than those for the HPS distance. An analysis of the errors showed the following problems:

- **diffuse patterns:** some patterns (generated mostly by Canon machines) have an irregular appearance that shows a clearly distinguishable VPS distance but a much less clearer HPS distance. For these patterns, repetitions in

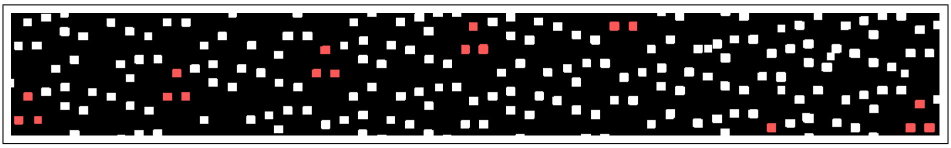


Figure 5.10: Example of a diffuse pattern. It can be seen that the pattern is repeating in horizontal direction but that the repetitions have a small offset. A characteristic set of dots is colored in red to make the repetitions more easily detectable.

horizontal direction are slightly shifted from one repetition to the next. An exact horizontal repetition was not detectable. But due to the error margin given for the matching, it will find a HPS distance that is actually not the correct one when sticking to the definition. This is mostly observed for the 0.16 and 0.32 inch patterns. An example can be found in Figure 5.10.

- **bad print quality:** a few documents present printing defects that can be observed when using an old drum: this leads to toner spreading all over the page, resulting in many small dots everywhere that cannot be easily distinguished from the CPS dots.
- **sparse patterns:** on several print-outs the pattern only appears around printed areas. Thus the large white areas where the dots are easily identifiable do not contain any dots. This significantly reduces the number of dots for reliable matching, leading to misdetection.

In only three cases, both VPS and HPS distances could not be correctly extracted. In all other cases at least one of the distances was correctly computed, while the other distance was returned either as "no dots present" or as a distance that does not fit any of the classes and that can thus be easily intercepted. Four HPS distance errors are due to the diffuse patterns that Canon printers present. However, in three of the four cases the VPS distance was correctly computed.

# of doc. images	VPS & HPS corr.	HPS corr.	VPS corr.
67	57	59	62
Accuracy	85.1%	88.0%	92.5%

Table 5.2: Results on the 67 document images of type 6 from the EFF-DFKI dataset. The VPS distance is detected correctly in 92.5% of the documents.

The problem of diffuse patterns could be solved by accurately deskewing the scanned image before processing. The allowed variation during horizontal matching of the search pattern could be reduced, leading to a more accurate estimation of the HPS distance. Another approach would imply verifying if the diffuse patterns appear only in combination with the 0.16 and 0.32 inch VPS distances. In these cases, the HPS distance information could be discarded for printer class detection.

Sparse patterns represent a major problem: depending on the document content, there may be enough dots left to detect a repetitive pattern. For some sample sets, the ground truth had even to be updated as the automatic method correctly found a pattern that had not been detected while manually generating the ground truth. In most cases, however, there are only very few dots left which makes it hard for an automated method to detect a recurring pattern. During manual ground truth generation knowledge about the printer type often helped in steering the search for the HPS and VPS distances into the right direction. Unfortunately this information is not available in a real world scenario. Concerning the bad print quality, a more sophisticated document cleaning could be tried.

5.2.6 Results for CPS Code Comparison

The evaluation of the accuracy on the CPS code comparison returned 83 correct decisions and 11 wrong decisions, thus an accuracy of 88.3%. An analysis of the results showed that all of the 11 errors were due to comparisons that should have returned "same prototype" but did in fact return "different prototype".

A detailed analysis of the errors made showed the following two issues:

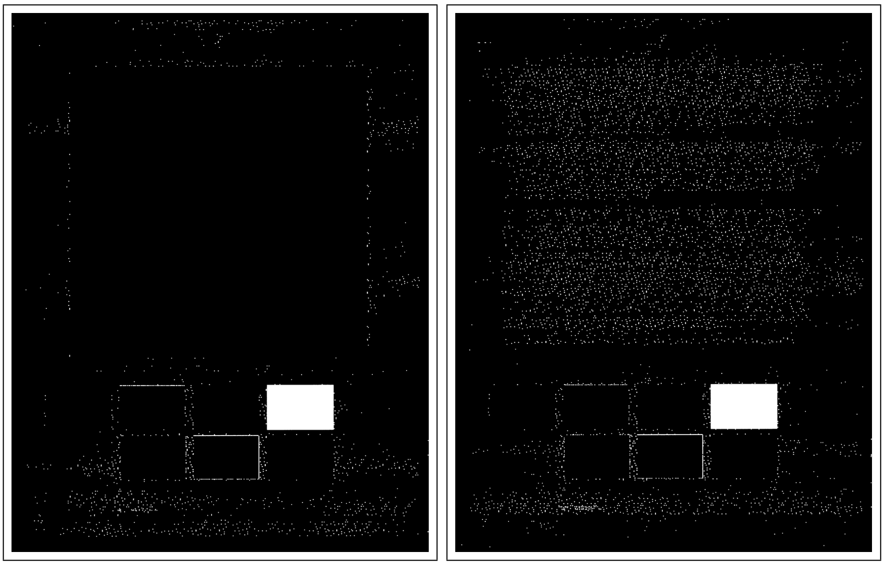
- **sparse patterns:** seven out of the eleven errors are due to sparse patterns. For these patterns the dots seem only to be present around page content areas, so not in large white spaces. As only few dots are present and this mostly in areas where dots may be missing due to overlaid text, it is hard to compute a good prototype estimate. Examples of sparse patterns can be seen in Figure 5.11.

- **time variant CPS codes:** several Xerox printers are known to have a time variable coding that includes the date and the time when the print-out was generated [37]. The remaining four errors are all due to this problem. An example of two different patterns extracted from two images of the same sample set can be found in Figure 5.12[5]. It is clear that the patterns have common parts but that they consistently differ in places that are known to encode the time information. Interestingly enough there are also sample sets (e.g. sample set 0015, Xerox Phaser 790) of print-outs from Xerox printers that do not encode date or time information.

Resolving the problem with Xerox printers could by done by using a mask to ignore dots encoding time and date information. Another possibility would be to

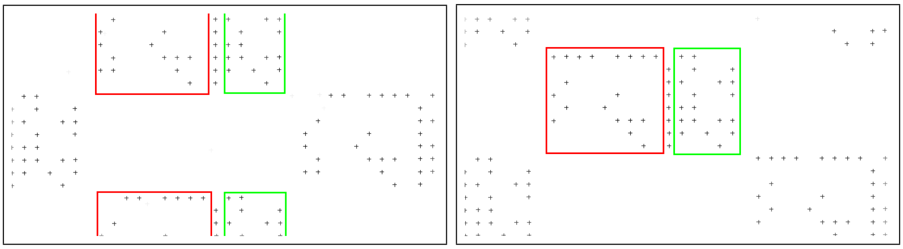
---

[5] Decoding information has been obtained by using the applet from the EFF: http://w2.eff.org/Privacy/printers/docucolor/, last retrieved on March, 25th 2010



(a) 0057-0004 (b) 0057-0006

Figure 5.11: Example of sparse patterns. It should be noticed, that for Sheet 0057-0004 only few dots are present. For the image containing Sheet 6, many more dots can be observed, especially in the text area.



(a) 0057-0004 (b) 0057-0006

Figure 5.12: Examples of differing patterns from the same Xerox printer. It can be seen that there is a considerable overlap in both patterns but some parts differ. The red rectangle shows the region where the time and date information is encoded. This region shows differences between the two patterns. The green region shows the serial number information. Using the decoding applet provided by the EFF, the following information for the left pattern could be extracted: *Printer serial number: 029246 [or 37029246], Date: June 20, 2005, Time: 08:38.* For the right pattern contains the following information: *Printer serial number: 029246 [or 37029246], Date: June 20, 2005, Time: 08:40.*

decode the pattern and use the serial number for comparison of two printers, as the decoding method for this CPS pattern type is known.

The problem of the sparse patterns is much harder to solve: if only few dots are present, no stable prototype pattern can be generated. In this case the system could try to extract the prototype patterns and display these to an operator who could then decide if the document should be analyzed further. Only comparing the HPS and VPS distance could also be done, as these can more reliably be detected even in presence of sparse dots.

5.2.7 Discussion

In this section, it could be shown that the counterfeit protection system codes generated by most color laser printers and color copiers can be used in an automatic setup for detecting the printer class a document was printed on. If a forged document is processed that claims to come from an invoice source that is known to have used a color laser printer from manufacturer A, and if now this document is classified as coming from a printer of manufacturer B, the invoice will be send to an operator for further verification.

This approach is extended to detect if two print-outs have the same CPS pattern. In that case it can be concluded that exactly the same printer has been used to generate both print-outs.

These two methods build an important barrier for document forgeries: a perpetrators would have to have at least a printer form the same printer class, or, at worst, the person would need to have access to the original printer to circumvent these barriers.

A useful extension of the proposed method of CPS code based authentication using comparison of two patterns, would be an automated decoding for Xerox type patterns, as these are the only patterns where the decoding procedure is known. This would make comparison of these patterns working even if date and time information is encoded.

In the next section, a different approach for authentication is presented. The idea is to detect distortions that come from scanning or printing. This is done by comparing positions of foreground content from genuine documents to the positions of the corresponding parts in the questioned document. This approach is described in Section 5.3.

5.3 Authentication using Positional Variations

In the previous section, CPS codes were used for authentication. These codes, however, are only produced by color laser printers and copiers. In this section an authentication approach is presented that does not require the CPS codes. Instead, a novel feature, positional variations, are used for document authentication. The idea is to detect distortions that are commonly produced by scanning and printing processes. This is done by comparing the positions of foreground elements in constant text parts. In the considered scenario of invoice processing it can be observed, that most documents have constant text parts for invoices from the same source. If positions of elements in theses parts differ from the model to the questioned document, this is a strong hint that the document has been tampered.

The forgery experiment in the Chapter 1 showed that the type of expected forgeries are from the group of “Print, Paste & Copy” (PPC), ”Scan, Edit & Print“ (SEP) and ”Reverse Engineered Imitation“ (REI) forgeries. The first category of forgeries are created by printing text on a new sheet that will be pasted on the genuine document. Then, a copy of this document is made. The second category is the type of forgeries generated by scanning a document, modifying it with standard image manipulation software and printing it out again. The forgeries from the last category are produced by rebuilding the genuine invoice in a layout or word processing software.

In all three scenarios, positional variations were observed: reversely engineering a document is a hard task as the positions of all the components depend on many different variables: font types, font sizes, line spacings, etc. For the SEP and REI forgeries it could be observed that even by scanning and printing (or by copying) distortions are introduced. These are not directly visible but when comparing two pages closely against each other it can be observed that not all parts of the image match as expected.

The approach works as follows: observing a number of original invoices from one invoice party allows to build a model signature of the non-variable part of an invoice. A new invoice is checked against this model signature and if it is significantly different, it is considered as potentially forged. The doubtful document could then be forwarded to a human operator for further inspection.

After a pre-processing step performing binarization [94] and skew correction (Chapter 2), first an intrinsic document signature is constructed. The genuine invoice images from an invoicing party are accurately aligned to one reference invoice (Section 5.3.1). For the reference image a binary map representing the areas of fixed content is created. Then, a signature for this invoicing party is built based on an analysis of variations in positions and sizes of the bounding boxes of the connected components among the aligned invoices (Section 5.3.2). Once a signature is constructed for an invoicing party, the originality of the new invoices

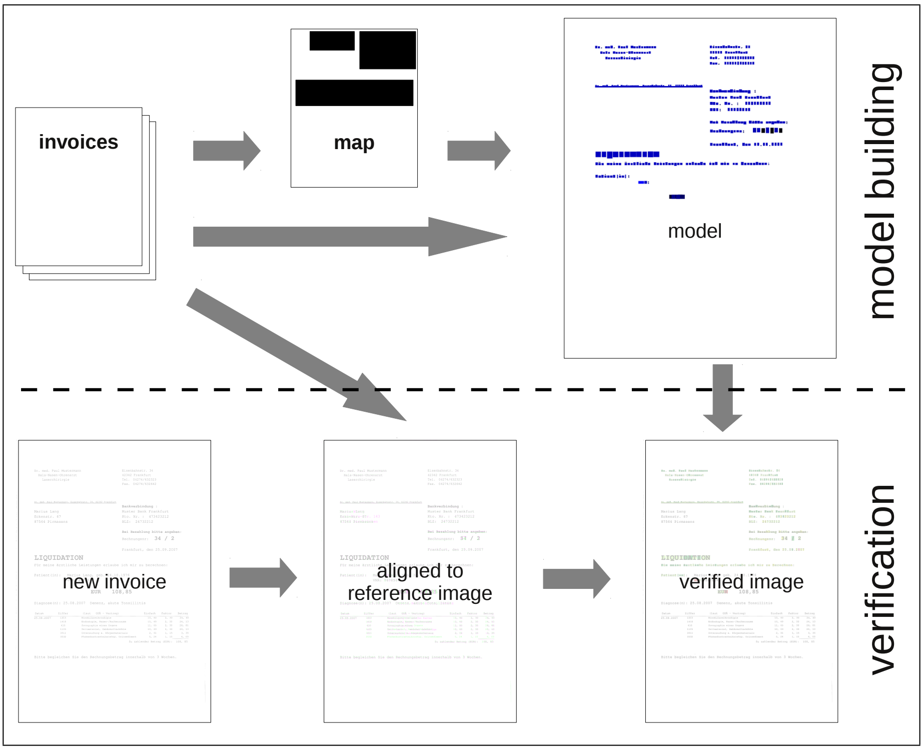


Figure 5.13: Visualization of the approach using positional variation modeling. First a model is generated from several genuine invoices (training set) and a map defining what areas are constant. The verification aligns the questioned invoice to the reference image from the training set. Finally, bounding boxes of the connected components are verified against the model.

from that source can be verified by comparing it to the signature. A diagram showing the different parts is given in Figure 5.13.

5.3.1 Alignment

The first step is to accurately and robustly align the images. The alignment of two document images aims at identifying the transformation parameters that allow to overlay both images.

Different techniques have been proposed in literature for image registration and alignment. The approaches for general image registration [128] are not well suited for binary document image registration because binary documents lack the color and texture features that are typically used in image registration. Nakai et al. [77] and Liang et al. [61] have proposed image registration techniques for document images, but in that case only alignment/registration of the same document under different kinds of distortions is considered.

For this application, the document image-matching technique described in the authors previous work [115] is used to align two images from the same invoicing party. This technique is tolerant to changes in the two documents to be matched and is hence a good candidate for this scenario. It uses an optimal branch-and-bound search algorithm, called RAST [17] (Recognition by Adaptive Subdivision of Transformation Space). This method allows robust and accurate finding of the globally optimal parameters describing the transformation needed to align both images. Since the RAST algorithm finds the globally optimal alignment of the two images, it is expected that it will align the two images based on their static part.

The quality function is defined as the number of model points matching an image point under the error bound ϵ.

The RAST algorithm uses a branch-and-bound search for quickly finding a global optimum, for the given quality function. The method uses a priority queue containing parameter subspaces ordered by their upper bound quality. The highest upper bound quality subspace is divided into two new subspaces, by splitting it into two parts of equal size. For each part, the new upper bound quality is determined and both subspaces are added into the priority queue. These steps are repeated until a stopping criterion is met.

For applying RAST, first an initial parameter space (also called transformation space) has to be defined. Let $[t_{x_{min}}, t_{x_{max}}] \times [t_{y_{min}}, t_{y_{max}}] \times [a_{min}, a_{max}] \times [s_{min}, s_{max}]$ be the initial search space, where t_x stand for translation in x direction, t_y translation in y direction, a for the rotation angle and s for the scale. As the pages are being deskewed before processing, the rotational dimension can be discarded by setting $a_{min} = a_{max} = 0.0°$. A more detailed description of RAST can be found in [17, 20].

As image points, the centers of the bounding boxes of the connected components are chosen. Matching all points from one image to all points of a second image can be time consuming. In order to speed up the computation of the upper bound for the quality, a filtering step is added before the branch-and-bound search: to avoid comparing connected components that are not similar at all, Fourier descriptors for the contour of the connected components have been extracted [40]. These describe the shape of the connected component. For example it is not reasonable to match a round character as an "o" to an "x". In order to be invariant to scale and rotation, the images of the connected components are downscaled to a fixed size and the phase is discarded to obtain rotation invariance for the Fourier Descriptors. For each model point (connected component) only the n most similar image points are considered for the quality estimation.

5.3.2 Positional Variation Modeling

The modeling of the positional variations is done on a connected component level. The input of the modeling step consists of several sets of bounding boxes of connected components coming from a genuine document. A map defining the constant areas in a document type is also given as input. The question to be solved is how to generate a model out of this information that represents the positional variations of the genuine connected components.

All the connected components that are not included in the areas defined by the map are discarded. A bounding box of a connected component is defined by four parameters: (x_l, y_l, x_h, h_y), where (x_l, y_l) define the lower left and (x_h, y_h) the upper right corner of the bounding box of the connected component. Modeling the positional variations can thus be done on the level of these four parameters.

It is reasonable to assume that the values for these parameters are normally distributed: their mean will be around the "ground truth" value and due to noise, binarization artefacts and other distortions small variations may occur.

In order to build the model, the bounding boxes of the connected components of the different genuine samples have to be clustered in order to compute the statistics of the different connected components. This clustering is done by checking if two bounding boxes overlap.

$$D_{\text{ov}}(\mathbf{c}_i, \mathbf{c}_j) = 1 - \frac{2 \times \text{Ov}(\mathbf{c}_i, \mathbf{c}_j)}{\text{area}(\mathbf{c}_i) + \text{area}(\mathbf{c}_j)}$$

where $\text{Ov}(\mathbf{c}_i, \mathbf{c}_j)$ is the number of overlapping pixels of both connected components and $area(c)$ is the number of pixels of connected component c. This will return 1.0 if two bounding boxes are identical and 0 if two bounding boxes do not overlap at all.

After this clustering the mean and standard deviation for each parameter of each bounding box cluster is estimated. A simplified visualization of the model is depicted in Figure 5.14. The blue ares show bounding boxes that have low variability (stable bounding boxes) and the dark areas show components with more variability.

Assuming a perfect match, the model and the images would contain exactly the same number of connected components in the constant document areas. The likelihood of a genuine document image I composed of connected components $\mathcal{C} = (c_1, \ldots, c_n)$ to be genuine (not forged: $\neg f$) could be computed as follows:

$$p(I|\neg f) = p(c_1, \ldots, c_n|\neg f) = p(c_1|\neg f) \ldots p(c_n|\neg f) \tag{5.2}$$

assuming independence of the parameters of the connected components:

$$p(c|\neg f) = p(x_l, y_l, x_h, y_h|\neg f) = p(x_l|\neg f)p(y_l|\neg f)p(x_h|\neg f)p(y_h|\neg f) \tag{5.3}$$

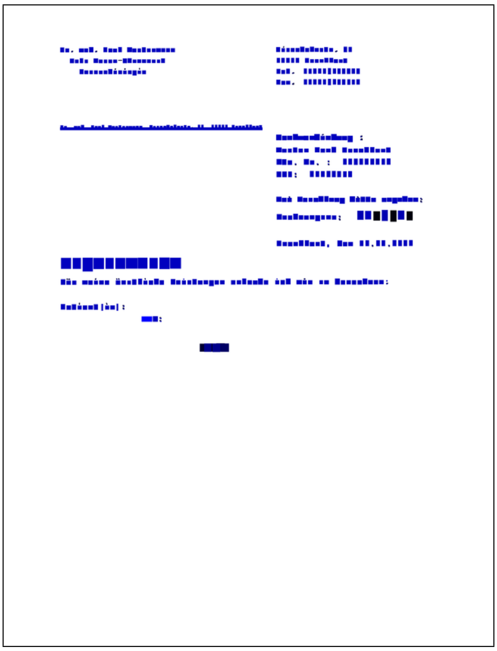


Figure 5.14: Visualization of the model. The background is painted white. The blue regions represent the bounding boxes of the document foreground. The lighter blue the boxes are the more "stable" this box is.

where

$$p(x_l|\neg f) = \mathcal{N}(x_l, \mu_{x_l}, \sigma_{x_l}) \tag{5.4}$$

and $p(y_l|\neg f)$, $p(x_h|\neg f)$ and $p(x_h|\neg f)$ are modeled accordingly.

The authentication of a document is done by first aligning the image to the reference image from the model. Then the likelihood in Equation 5.3.2 is computed. In the end, a decision is needed whether the questioned document should raise an alarm or not. As modeling of the variations for forged documents is complex due to missing data, a thresholding approach is used: for the training set used to build the model, the mean μ_L and the standard deviation σ_L of the computed likelihoods are computed. A threshold is fixed on the basis of the confidence intervals.

As the connected components depend on print and scan quality, the question of robustness against merging and breaking connected components arises. As the scanning process can be optimized by the operator of the system, the remaining source of merged and broken components is the invoice generation process. It may happen that the printer of the person creating the invoice is low on ink or the paper was changed which could lead to more ink smearing. These problems will result in merged and broken connected components and will lead also to new connected components that are not covered by the model.

In order to deal with this problems, the following approach is chosen: only connected components that can be "explained" by the model (components where the bounding box overlaps with one model component) are taken into account. This could potentially lead to the problem that a forgery where only one component matches the model would have a high likelihood. A preliminary classification step is added based on the matching quality obtained by the alignment procedure: if this alignment quality is to low, the document will be rejected as forgery. The rejection scheme is defined by modeling the matching quality as normally distributed $\mathcal{N}(\mu_q, \sigma_q)$ and setting a low threshold at $\pm 10.0 \times \sigma_q$, where μ_q is the mean and σ_q the standard deviation of the matching quality computed on the model training set.

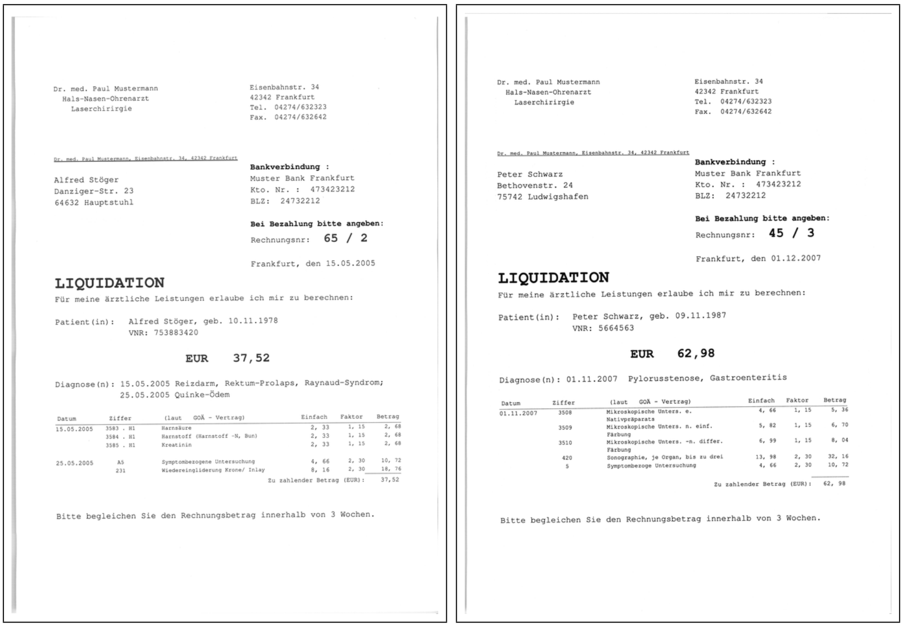
5.3.3 Evaluation

In order to test the performance of the proposed signature, a dataset with forged document samples is needed. As to the author's best knowledge no public dataset is available containing genuine and tampered documents from one and the same invoice party, a new dataset of invoices was generated. These sample documents were created by a student using a word processing software. Next, one document was randomly picked and given to other students. Their task was to copy the document as accurate as possible using the text editor or layout software of their choice. The number of genuine documents is 40, the number of REI forgeries is 12. An example of a genuine and a forged document is given in Figure 5.15.

As the dataset was relatively small, an n-fold cross validation was run with $n = 4$. Thus, four different training sets were generated out of the 40 genuine invoices, each training set consisting of 10 images. The first image in alphabetical order of the file names of each training set was chosen as reference image where the remaining nine images were aligned to. The other 30 genuine images formed the genuine part of the respective test set. To each of these test sets the forged invoices ($n = 12$) were added.

A second test has been done in order to measure the performance of the method on the SEP forgery scenario: instead of remaking the whole document using a word processor, the forger could just scan an original invoice, make changes using an image editor and print the changed invoice. As scanning and printing an original invoice distorts it slightly, the proposed method should be able to detect these cases. An example of such a distortion can be found in Figure 5.16.

To simulate this scenario the same four-fold cross validation setup was chosen. Out of the 40 genuine invoices, 5 were copied on different multi function printers (MFP) to simulate effects of scanning and printing the original invoice. A list of the MFPs used in this test is given in Table 5.3. In total 8 different MFPs were



Dr. med. Paul Mustermann
Hals-Nasen-Ohrenarzt
Laserchirirgie

Eisenbahnstr. 34
42342 Frankfurt
Tel. 04274/632323
Fax. 04274/632642

Dr. med. Paul Mustermann, Eisenbahnstr. 34, 42342 Frankfurt

Alfred Stöger
Danziger-Str. 23
64632 Hauptstuhl

Bankverbindung :
Muster Bank Frankfurt
Kto. Nr. : 473423212
BLZ: 24732212

Bei Bezahlung bitte angeben:
Rechnungsnr: **65 / 2**

Frankfurt, den 15.05.2005

LIQUIDATION

Für meine ärztliche Leistungen erlaube ich mir zu berechnen:

Patient(in): Alfred Stöger, geb. 10.11.1978
VNR: 753883420

EUR 37,52

Diagnose(n): 15.05.2005 Reizdarm, Rektum-Prolaps, Raynaud-Syndrom;
25.05.2005 Quinke-Ödem

Datum	Ziffer	(laut GOÄ - Vertrag)	Einfach	Faktor	Betrag
15.05.2005	3583 . H1	Harnsäure	2, 33	1, 15	2, 68
	3584 . H1	Harnstoff (Harnstoff -N, Bun)	2, 33	1, 15	2, 68
	3585 . H1	Kreatinin	2, 33	1, 15	2, 68
25.05.2005	A5	Symptombezogene Untersuchung	4, 66	2, 30	10, 72
	231	Wiedereingliderung Krone/ Inlay	8, 16	2, 30	18, 76
				Zu zahlender Betrag (EUR):	37,52

Bitte begleichen Sie den Rechnungsbetrag innerhalb von 3 Wochen.

Dr. med. Paul Mustermann
Hals-Nasen-Ohrenarzt
Laserchirirgie

Eisenbahnstr. 34
42342 Frankfurt
Tel. 04274/632323
Fax. 04274/632642

Dr. med. Paul Mustermann, Eisenbahnstr. 34, 42342 Frankfurt

Peter Schwarz
Bethovenstr. 24
75742 Ludwigshafen

Bankverbindung :
Muster Bank Frankfurt
Kto. Nr. : 473423212
BLZ: 24732212

Bei Bezahlung bitte angeben:
Rechnungsnr: **45 / 3**

Frankfurt, den 01.12.2007

LIQUIDATION

Für meine ärztliche Leistungen erlaube ich mir zu berechnen:

Patient(in): Peter Schwarz, geb. 09.11.1987
VNR: 5664563

EUR 62,98

Diagnose(n): 01.11.2007 Pylorusstenose, Gastroenteritis

Datum	Ziffer	(laut GOÄ - Vertrag)	Einfach	Faktor	Betrag
01.11.2007	3508	Mikroskopische Unters. e. Nativpräparats	4, 66	1, 15	5, 36
	3509	Mikroskopische Unters. n. einf. Färbung	5, 82	1, 15	6, 70
	3510	Mikroskopische Unters. -n. differ. Färbung	6, 99	1, 15	8, 04
	420	Sonographie, je Organ, bis zu drei	13, 98	2, 30	32, 16
	5	Symptombezoge Untersuchung	4, 66	2, 30	10, 72
				Zu zahlender Betrag (EUR):	62, 98

Bitte begleichen Sie den Rechnungsbetrag innerhalb von 3 Wochen.

Figure 5.15: The left image is an "original" document from the dataset. The image to the right represents a sample of a forged document (REI type forgery).

used to obtain 40 copied invoices. The copies were scanned using the same scanner as for the other invoices. The threshold is computed in the same manner as done in the test before.

5.3.4 Results

The results of the test on the first dataset containing genuine documents and manually generated REI forgeries are shown in Figure 5.18. It shows the box plot of the classification error rate for the different folds relative to the threshold which was fixed according to the different confidence intervals depending on σ_t. It can be seen that for low values, the accuracy is quite low. Accuracies over 96% are achieved setting the threshold $\sigma_t > 4 \times \sigma_L$. The best mean accuracy of 97.0% is obtained for settings of $\sigma_t = 6 \times \sigma_L$. An example of an aligned and a resulting image is shown in Figure 5.17.

Figure 5.19 shows the results on the copy dataset. In contrast to the test on the pseudo-invoices, the maximum accuracy of 93.5% is obtained for $\sigma_t = 3.5 \times \sigma_l$. For higher values, the accuracy decreases slightly.

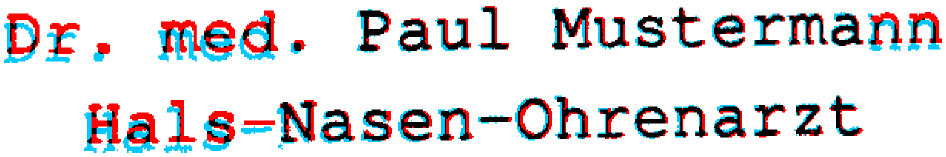


Figure 5.16: Example of distortions induced by copying a genuine document. The originally black pixels from the copies invoice are painted in blue, the black pixels from the original document are painted in red. If blue and red pixels overlap, these are painted black. It can be seen that the copying process moves blocks up and down: left part the copy is too far down, the middle part fits well and the right part is again too far down.

Manufacturer	**Model**	**Type**
Brother	1010e	Desktop fax machine
Canon	l220	Desktop fax machine
Nashutec	MP 1500 SP	Workgroup MFP
Ricoh	1224c	Workgroup MFP
Ricoh	Aficio MPC 2500	Workgroup MFP
Ricoh	Aficio MPC 3000	Workgroup MFP
Samsung	SCX 4216f	Desktop fax machine

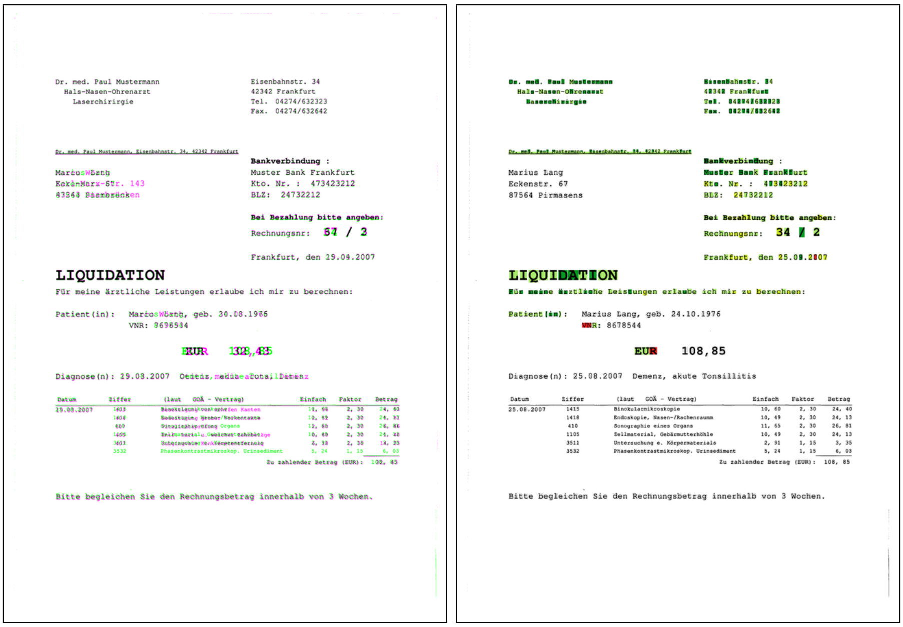
Table 5.3: List of the different MFP machines used to generate the test set for the SEP scenario. It contains small desktop devices as well as A3 sized workgroup MFPs.

The two plots in Figure 5.20 show the matching qualities and computed likelihoods for the different class of samples. It can be clearly seen that the manual forgeries tend to have the worst matching quality as well as the lowest likelihoods, followed by the copies. Some copies are close to genuine invoices but still allowing to detect most of them as forgery.

5.3.5 Discussion

In the previous sections a method was presented to detect forgeries on the basis of the distortions that are introduced by scanning and printing or by reengineering of a document. It is shown that a model-based approach for detecting and measuring the positional variations of fixed page components can effectively be used to detect these distortions and to use them for document authentication. Accuracies of up to 97% are reported on classification between genuine and forged documents.

A few aspects of the method have to be dealt with in future: the map defining the constant areas has to be automatically generated. One possibility would be to generate this map by an automated learning step: given a set of invoices from the same source, the matching result could be used to identify “stable” components



(a) Alignment

(b) Resulting image

Figure 5.17: Example of the alignment and the resulting images. The alignment image shows both images colored differently. Overlapping foreground pixels are painted black. The resulting image shows the analyzed bounding boxes. The different colors show the different levels of variations.

without the need of an additional map.

Also it could be analyzed whether the method can be extended to use relative variations of components to each other: the distortions do not only affect the absolute position of constant page parts but also the positions and distances between other foreground components. It has to be analyzed if variations in the relative positions of variable documents part can also be used for forgery detection.

5.4 Conclusion and Future Work

In this chapter, document authentication methods were presented that use intrinsic document features only. The first approach uses the counterfeit protection system codes that are generated by most color laser printers and color copiers for detecting the printer class a document was printed on. Using these CPS codes automatic printer class identification and printer identification is presented. Accuracies of up

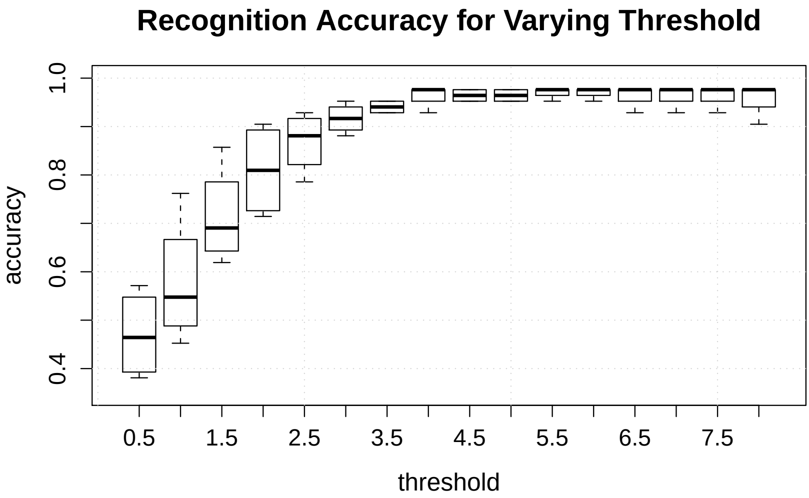


Figure 5.18: Results of the test on the pseudo-invoices. The best mean accuracy 97.0% is obtained for $\sigma_t = 6.0$

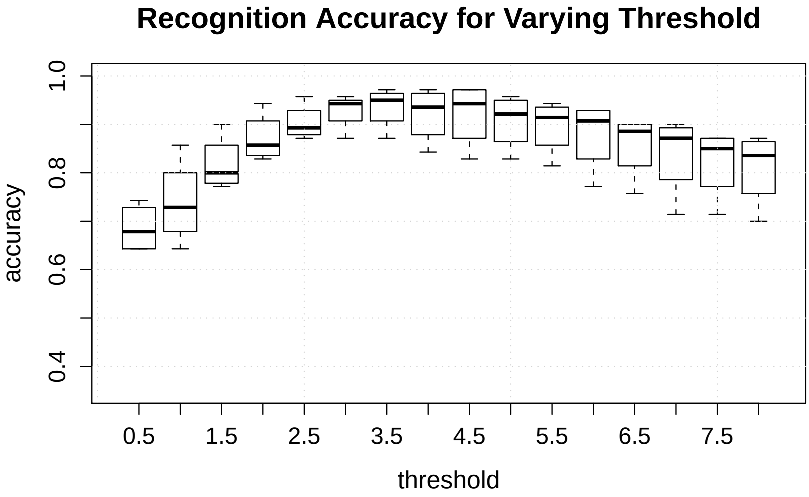
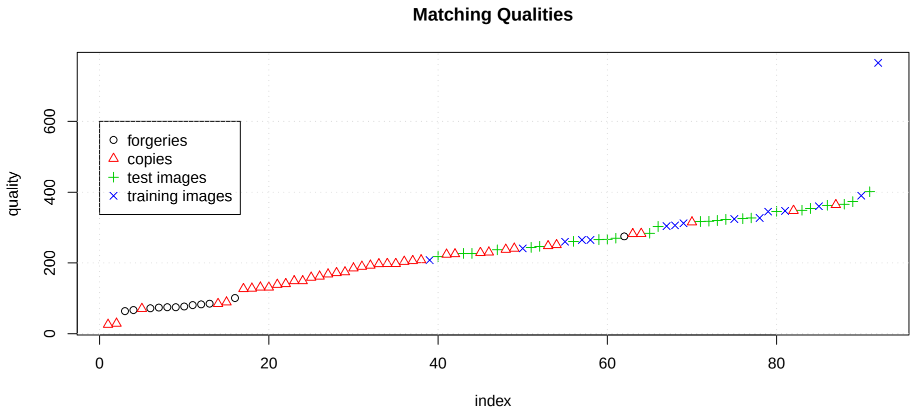


Figure 5.19: Results of the test on the copied invoices. The best mean accuracy of 93.5% is obtained for $\sigma_t = 3.5$



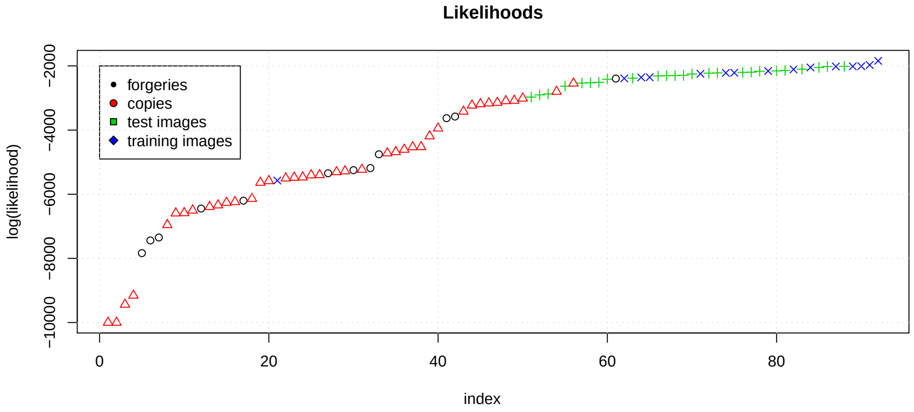


Figure 5.20: Sorted qualities and log-likelihoods for the different samples. It can be seen that the generated forgeries have lowest scores in both cases.

to 92.5% and 88.3% were obtained during evaluation of the respective methods.

The second approach uses the distortions that are introduced by scanning and printing processes to the forged document. Using fixed page content areas these variations can be measured and effectively be used for document authentication: if the position of a component significantly differs from the position of that component in the model, the document could be forged. This approach proved that in 97% of the cases it was able to correctly determine if the document was genuine or not.

The authentication methods presented in this chapter are thus important obstacles for perpetrators: if the invoice source is using a color laser printer producing CPS codes, the person will need access to the same printer to make a forgery that will pass the CPS authentication check. Also, most attempts involving a standard scanners and printers will fail, due to the distortions introduced by these devices.

Chapter 6

Line Examination for Document Security

In Chapter 5, intrinsic document features have been used for document authentication. These features require prior knowledge about questioned the document in order to authenticate them. Prior knowledge, however, is not always available.

If no prior knowledge is available, it is possible to use document source independent features for a general plausibility check of the document's contents. In this chapter, two methods are presented for this task: automatic verification of text-line skew angle and text-line alignment variations. For both features general assumptions can be made that hold for most document types. If a questioned document violates these assumptions, it will be reported to an operator for further investigation. The following contributions are presented in this chapter[1]:

1. First approach for automated document forgery analysis using text-line skew angle and alignment variations.
2. New data sets for evaluation of forgery detection methods.
3. An in depth evaluation of the proposed methods that shows its usefulness in the context of document security (area under the ROC curve (AUC) score of $AUC = 0.89$, $n = 191$).

---

[1] This chapter is based on the author's work in
[117] J. van Beusekom, F. Shafait, and T. M. Breuel. Automatic line orientation measurement for questioned document examination. In *Proc. of the 3rd Int. Workshop on Computational Forensics*, volume 5718 of *Lecture Notes in Computer Science*, pages 165–173, The Hague, The Netherlands, August 2009, and
[120] J. van Beusekom, F. Shafait, and T. M. Breuel. Document inspection using the text-line alignment. In *Proc. of the 9th IAPR Workshop on Document Analysis Systems*, Boston, MA, USA, June 2010. accepted for publication.

6.1 Introduction

In every day life, document authentication is an important task as many documents present a potential value. A typical example are bank notes. When handling bank notes, most people quickly authenticate them by verifying easy to detect security signs as e.g. holograms, structured print and special inks. This is seamlessly done by many people as they have got used to the money and its security features.

Assuming the scenario of a person handling an unseen bank note: the person handling this bank note will not know exactly what features to look for. This person might either just trust the source, or look for features he might know from previously encountered bank notes: he might e.g. look for a watermark of a metallic stripe insider the paper, as all of the notes he has seen so far contained these security features.

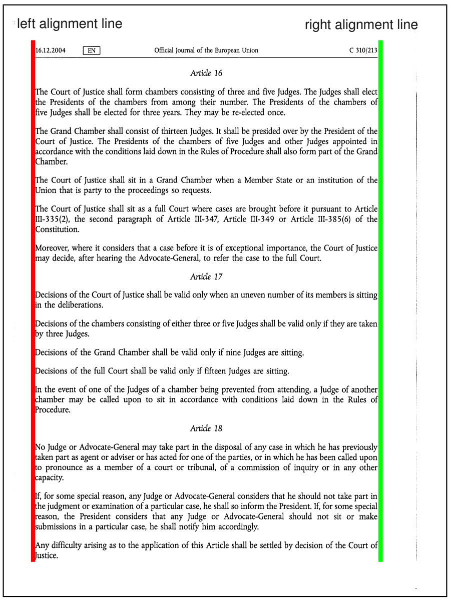
For less secured documents, authentication is only feasible if the characteristic features of the document are known. Assume e.g. that a perpetrator had a car accident: the car has been repaired and the person receives the invoice from the shop. In order to defraud money from the insurance company, he adds some replacement parts to the invoice that actually haven't been replaced or increases the price of single entries. This can be done by printing the new items on a new sheet and paste the part on to the original invoice to make a copy of that (PPC forgery). Finally, he sends it to the insurance company and hopes to defraud the extra money. The resulting invoice will look similar to a genuine invoice.

The insurance company receives the invoice and notices, that it has no previous invoice from that specific shop in its archive to compare the new invoice with. Even in this case, it is still possible for the insurance adjuster to do a plausibility check, just as for the example of the bank notes. For printed documents it is possible to find features that are common to most documents and that are potentially being distorted in case of document forgery. These features can be extracted and checked. Two of these generic features can be measured on text-line level, the text-line skew and the text-line alignment.

One observation from the forgery experiment in Appendix A is that for the PPC forgeries, text-line skew and alignment variations were higher than for genuine documents. These features are also being used in questioned document examination [65] and are therefore chosen as candidates for doing plausibility checks on documents.

Several methods have been presented for extracting text-lines [63, 52], but to the author's best knowledge this is the first attempt to use the skew angle and the alignment in an automated setup for document security purposes.

In this Chapter, an automatic plausibility check of printed documents using text-line skew and alignment measures is presented. The features as well as the statistical models to perform the plausibility check are presented in 6.2. Evaluation


left alignment line
right alignment line

16.12.2004 EN Official Journal of the European Union C 310/213

Article 16

The Court of Justice shall form chambers consisting of three and five Judges. The Judges shall elect the Presidents of the chambers from among their number. The Presidents of the chambers of five Judges shall be elected for three years. They may be re-elected once.

The Grand Chamber shall consist of thirteen Judges. It shall be presided over by the President of the Court of Justice. The Presidents of the chambers of five Judges and other Judges appointed in accordance with the conditions laid down in the Rules of Procedure shall also form part of the Grand Chamber.

The Court of Justice shall sit in a Grand Chamber when a Member State or an institution of the Union that is party to the proceedings so requests.

The Court of Justice shall sit as a full Court where cases are brought before it pursuant to Article III-335(2), the second paragraph of Article III-347, Article III-349 or Article III-385(6) of the Constitution.

Moreover, where it considers that a case before it is of exceptional importance, the Court of Justice may decide, after hearing the Advocate-General, to refer the case to the full Court.

Article 17

Decisions of the Court of Justice shall be valid only when an uneven number of its members is sitting in the deliberations.

Decisions of the chambers consisting of either three or five Judges shall be valid only if they are taken by three Judges.

Decisions of the Grand Chamber shall be valid only if nine Judges are sitting.

Decisions of the full Court shall be valid only if fifteen Judges are sitting.

In the event of one of the Judges of a chamber being prevented from attending, a Judge of another chamber may be called upon to sit in accordance with conditions laid down in the Rules of Procedure.

Article 18

No Judge or Advocate-General may take part in the disposal of any case in which he has previously taken part as agent or adviser or has acted for one of the parties, or in which he has been called upon to pronounce as a member of a court or tribunal, of a commission of inquiry or in any other capacity.

If, for some special reason, any Judge or Advocate-General considers that he should not take part in the judgment or examination of a particular case, he shall so inform the President. If, for some special reason, the President considers that any Judge or Advocate-General should not sit or make submissions in a particular case, he shall notify him accordingly.

Any difficulty arising as to the application of this Article shall be settled by decision of the Court of Justice.

Figure 6.1: Example document with the left and the right alignment lines.

and results are presented in Section 6.3 and 6.4. The chapter concludes with Section 6.5.

6.2 Text-Line Features for Document Security

For checking the plausibility of a document, two following two features are used: the skew angle of text-lines and the alignment of text-lines. Measuring of the text-line's skew angle is straightforward using the method presented by Breuel [18] (details are presented in Chapter 2).

To measure the alignment of a text-line, the following approach is used: first, the left and right *alignment lines* are computed. These lines are defined as the left and right margin lines where justified, left and right aligned text start or end on. A visualization of the alignment lines is depicted in Figure 6.1.

After having extracted the alignment lines, the distance between the start and end point of a text-line to the respective alignment line is computed. This distance is used to perform a plausibility check.

In Section 6.2.1 the use of the skew angles of text-lines for the proposed application is explained. Section 6.2.2 explains the details of the method using the alignment of text-lines for plausibility checks of documents. In Section 6.2.3 both

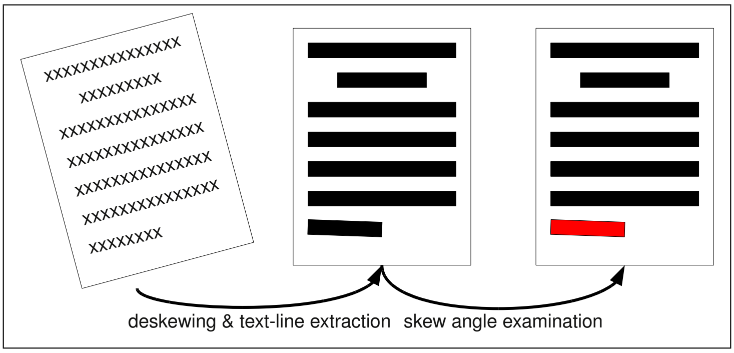


Figure 6.2: Visualization of the text-line skew examination: the binarized document is deskewed. The text-lines are examined if their skew angles are abnormally high or not.

features are combined into one framework.

6.2.1 Plausibility Check using Skew Angles

Using the skew angle of text-lines for identifying suspicious documents is a well-known technique in questioned document examination [65]. Questioned document examiners do this step mostly manually using standard image manipulation software. In this section, the first approach to automate this process is presented.

The main idea of the process is sketched in Figure 6.2. The binarized document is correctly oriented and deskewed using the method presented in Chapter 2. Next, the text-line skew angles are extracted. These are checked if they are within the "natural" variation of text-line skew. Then, the text-line is considered as valid. Else, the line is reported as an "implausible" line.

In the next section, the statistical modeling of the skew angle variations and its application to the plausibility check are explained.

Text-Line Skew Variation Model

In order to define what "normal" and "abnormal" skew angles are, the natural variation of text-line skew angles has to be measured. Although in the electronic representation, all text-lines are exactly parallel, the transfer to the paper medium and again back to an electronic image format adds text-line skew variations that have to be considered before being able to make a decision on the skew angle:

- **printing:** variations introduced by printing can be speckles, noise and distortions.

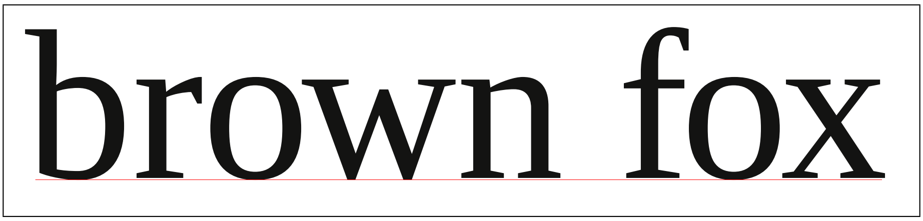


Figure 6.3: Examples showing typographic enhancements. Note that the characters "r", "n", "f"and "x" are located slightly above the horizontal red line which connects the bottom points of the characters with a round part at the base-line level, like "b" and "o".

- **digitization:** digitization of a paper-based document leads to many variations coming from the scanning [48] or from subsequent image processing, as e.g. the regularly observed page stretching while using flatbed scanner.
- **typographic enhancements:** optical correction is often used in typography to improve the visual appearance of fonts on lines. This leads to characters starting slightly bellow the base-line, which will reduce the accuracy of the measurement. An example is shown in Figure 6.3.

All these effects can lead to text-line skew variations that are frequently observed and that should thus not be considered as "abnormal".

In Figure 6.4 a histogram of measured skew angles is shown. It can be seen, that the distribution is peaked around 0°, that its variation is quite low and that its shape should be reasonably well approximated by a Gaussian distribution with parameters $\mathcal{N}(\mu_\alpha, \sigma_\alpha)$ being the mean and the standard deviation of the skew angle α.

Using this model, a simple threshold based method to decide whether a text-line skew angle is suspicious or not could be applied: using the confidence intervals, the threshold could be set to $\pm 3 \times \sigma_\alpha$, so every skew angle outside the 99.7% confidence interval will be reported.

This approach has two drawbacks:

- **no prior:** due to the missing prior, forged text-lines are considered to be as frequent as genuine ones. It is, however, more likely that genuine text-lines are much more frequent than forged ones.
- **text-line length:** for longer text-lines, the skew angle can be measured much more accurately than for shorter ones. Also, the skew angles of shorter lines are more sensitive to noise. In an extreme case, for a line with a length of 100 px consisting of two connected components, one supplemental pixel at the bottom of one component would change the angle by at least ±0.57°.

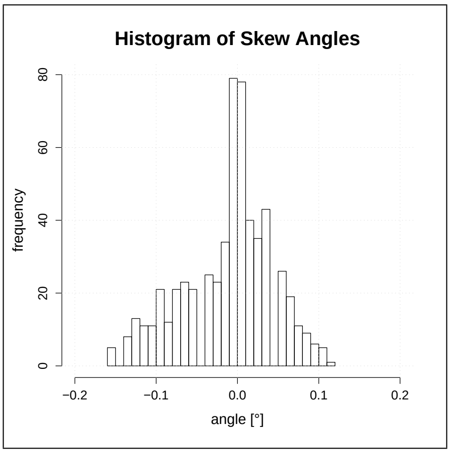


Figure 6.4: Histogram of skew angles measured on deskewed pages. The vast majority of measured skew angles is close to 0°, although, small variations can be observed.

For longer lines with more connected components the effect of the noise pixel will be lower.

To solve the first problem, a Bayesian formulation of the problem is proposed:

$$P(f|\alpha) = \frac{p(\alpha|f) \times P(f)}{p(\alpha)} \tag{6.1}$$

where $P(f|\alpha)$ is the posterior of having a forged text-line given the observation of its skew angle α, $p(\alpha|f)$ is the likelihood of observing the skew angle α knowing it is a forged text-line. $P(f)$ is the prior for observing a forged text-line and $p(\alpha)$ is the probability of observing the skew angle α (for forged as well as for original text-lines).

In the ideal case, statistics on the parameters could be used to extract reliable estimates. This, however, is only possible for genuine documents, as in a practical setup, no training data for forged documents is available. Even worse, to the authors best knowledge there is no public dataset with forged documents that could be used to extract the necessary information. For original documents, the information can partially be computed. Thus, the posterior of having a genuine text-line given the observed skew angle α is written as:

$$P(\neg f|\alpha) = \frac{p(\alpha|\neg f) \times P(\neg f)}{p(\alpha)} \tag{6.2}$$

In this case, $p(\alpha|\neg f)$ is estimated from training data consisting of genuine documents. As mentioned above, it is modeled as a Gaussian distribution $\mathcal{N}(\mu_\alpha, \sigma_\alpha)$.

$P(\neg f)$ is considered as a sensitivity parameter of the system that can be tuned by the operator according to his needs. Still, $p(\alpha)$ is not known due to the missing data from forged documents. The information of interest is which of both posteriors is higher and thus, the normalizing factor can be ignored. Consequently, a text-line is classified as a forged one if:

$$\hat{P}(f|\alpha) > \hat{P}(\neg f|\alpha) \tag{6.3}$$

where $\hat{P}(\neg f|\alpha) = p(\alpha|\neg f) \times P(\neg f)$ and $\hat{P}(f|\alpha) = p(\alpha|f) \times P(f)$.

As no measurements can be done on the skew angles of forged documents an assumption of the likelihood term has to be made: the assumption is that the observed skew angles for forged text-lines are uniformly distributed in a symmetric interval around the mean of 0°.

To solve the second problem concerning the measurement accuracy for lines of different length, the following solution is adopted: instead of estimating the parameters μ_α and ϵ_α over all the text-lines, the parameters are estimated for a certain text-line length interval. Ideally, the interval would have the size of one pixel. This would require very large amounts of data to obtain reasonable estimates of the parameters. Instead, a window of size $2 \times W$ is defined for which the parameters for a given line length l are computed around in the interval $[l - W, l + W]$. The size of the interval has to be fixed by the operator. In principal, it is best to have small sized intervals. But this is only possible if enough training data is available. If only few samples are available, a higher values has to be chosen in order to have robust model parameter estimates.

Finally, the two terms that need to be evaluated and compared are:

$$P(\neg f|\alpha) = \frac{\mathcal{N}(\alpha, \mu_{\alpha,l}, \sigma_{\alpha,l}) \times P(\neg f)}{p(\alpha)} \tag{6.4}$$

and

$$P(f|\alpha) = \frac{\mathcal{U}(\alpha, -w_{\alpha,l}, +w_{\alpha,l}) \times P(\neg f)}{p(\alpha)} \tag{6.5}$$

where l is the text-line length, $\mu_{\alpha,l}, \sigma_{\alpha,l}$ the estimated mean angle and standard deviation for the text-lines with length $[l - W, l + W]$ and where $-w_{\alpha,l}, w_{\alpha,l}$ are the left and right bound of the interval uniform distribution for the forged skew angles. Ignoring the normalizing factor, a text-line is classified as forged if

$$\mathcal{U}(\alpha, -w_{\alpha,l}, +w_{\alpha,l}) \times P(f) > \mathcal{N}(\alpha, \mu_{\alpha,l}, \sigma_{\alpha,l}) \times P(\neg f) \tag{6.6}$$

or as genuine in the other case.

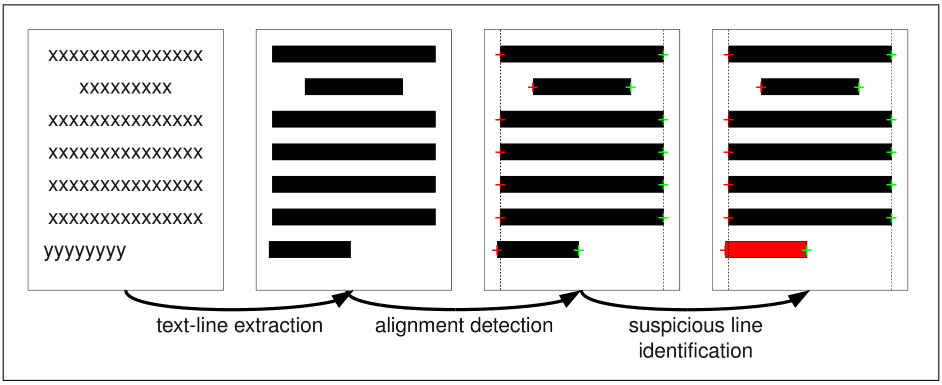


Figure 6.5: Visualization of the text-line alignment examination: the text-lines are extracted from the binarized document image. Then the left and right alignment lines are computed. Finally, each text-line is examined whether it shows normal alignment or not.

6.2.2 Plausibility Check using Text-Line Alignment

The examination of the alignment property of text-lines has also been previously used in questioned document examination [65]. The main idea of the process is sketched in Figure 6.5.

In the first step, the text-lines are extracted from the binarized document image. Then the left and right alignment lines are computed. These are finally used to examine whether the text-lines are normally aligned or not. In the next section, the alignment detection is explained.

Alignment Line Computation

After the text-line extraction, the alignment lines have to be detected. Four different alignment types are commonly distinguished in typesetting:

- **left aligned:** text-lines start at the left margin
- **right aligned:** text-lines end at the right margin
- **justified:** text-lines start at the left margin and end at the right margin
- **centered:** text-lines do neither touch the right nor the left margin. The gaps between both margins are equal in size.

In practical document security applications, centered lines are unlikely to be forged, as they are not frequent and mostly, they do not contain any valuable information. Therefore, the focus in this work lies on the left and right alignment of the text-lines.

In order to find the alignment lines, the starting and the end points of the text-lines have to be analyzed. The left alignment line is defined as a vertical line where left-aligned and justified text-lines have their starting point. The right alignment line is analogously defined as the vertical line where right-aligned and justified text-lines have their end point.

Finding these lines is done using RAST line finding [21], similar to the text-line finding explained in Section 2.3.3. The key difference is that no descender line has to be modeled and that the parameterization of the line is different. Instead of the (r, θ) parameterization, where r is the y-intercept and θ is the rotation angle with respect to the horizontal, the polar representation $\vartheta = (|\overrightarrow{n}|, \beta)$ is adopted, where $|\overrightarrow{n}|$ is the norm of the vector that is normal to the line and that points to the origin, and β is the rotation angle of $\overrightarrow{n}$.

The advantage of this parameterization is that it is well suited for searching for vertical lines. Consider feature points $\{x_1, x_2, \cdots, x_n\}$: the quality function is written as:

$$\hat{\vartheta} := \arg\max_{\vartheta} Q_{x_1^n}(\vartheta) \tag{6.7}$$

where

$$Q_{x_1^n}(\vartheta) = Q_{x_1^n}(|\overrightarrow{n}|, \beta) = \sum_{i=1}^{n} (q_{(|\overrightarrow{n}|,\beta)}(x_i)) \tag{6.8}$$

and

$$q_{(|\overrightarrow{n}|,\beta)}(x) = \max\big(0, 1 - \frac{d^2_{(|\overrightarrow{n}|,\alpha)}(x)}{\epsilon^2}\big) \tag{6.9}$$

and where $d_{(|\overrightarrow{n}|,\beta)}(x)$ is the Euclidean distance of a point x to the line defined by parameters $(|\overrightarrow{n}|, \beta)$.

The line finding is run twice, each time with different feature points: once with the starting points and once with the end points of text-lines. The left and right alignment lines are considered as the respective resulting highest quality lines, thus the lines where most start or end points of text-lines lie on.

Text-Line Alignment Model

The alignment feature of a text-line is measured by the distance of the text-line's start and end point to the corresponding alignment line. A visualization can be found in Figure 6.6. For each text-line two different distances are computed. To simplify the explanation, in the following, only the left alignment distance is considered. The modeling and computation of the right alignment distance is done analogously.

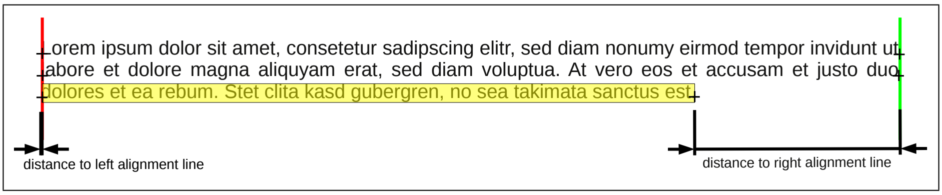

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut
labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo
dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est
distance to left alignment line
distance to right alignment line

Figure 6.6: Visualization of the text-line alignment measure: the crosses represent the start and end points of each line. The vertical red and green lines represent the left and right alignment lines. The distances are visualized for the last text-line only.

Again, just as for the text-line skew angle variance modeling, there is the problem of missing training data for forged documents. Also, the modeling of genuine documents features is not as straightforward as in case of the skew angle based method: for the skew angle it is clear, that the angles to be encountered are close to 0°. For the alignment distances, this is not true. Indented text-lines e.g. will lead to measurements that are significantly off from the normal variations for left aligned text-lines and these should not be considered as potentially forged lines.

Therefore, instead of assuming a Gaussian distribution as done for the skew angles, the distribution of the alignment distances is measured from a training set of genuine text-lines. A Bayesian classification approach is chosen:

$$P(\neg f|d_l) = \frac{p(d_l|\neg f) \times P(\neg f)}{p(d_l)} \tag{6.10}$$

where $p(d_l|\neg f)$ is the likelihood of observing the distance d_l of the left alignment line to the start of the text-line given the information that the text-line is original. As discussed above, assuming a Gaussian distribution would be a bad choice. Therefore, the real distribution is measured on a training set.

For the forged case this becomes:

$$P(f|d_l) = \frac{p(d_l|f) \times P(f)}{p(d_l)} \tag{6.11}$$

The problem to be solved is how to model the likelihood $p(d_l|f)$. Again, training data is not available. The assumption is that the variability around the correct alignment distance is higher for forged lines than for original ones. However, this distribution is not Gaussian, as, just as for genuine text-lines, forged text-lines may start or end somewhere further away from the corresponding alignment line.

The distances are therefore modeled as a mixture of a Gaussian and a uniform distribution:

$$p(d_l|f) = P(j) \times p(d_l|f, j) + (1 - P(j)) \times p(d_l|f, \neg j) \tag{6.12}$$

where $P(j)$ is the prior of observing a justified text-line, $p(d_l|f,j)$ the likelihood of observing distance d_l given the fact that the text-line is forged and justified and $p(d_l|f,\neg j)$ the likelihood of observing distance d_l given the fact that the text-line is forged and not justified.

As argued before, $p(d_l|f,j)$ is modeled as being normally distributed $\mathcal{N}(\mu_{d_l}, \epsilon_{d_l})$. For the observed distance in a case of a non justified text-line, a uniform distribution is assumed $\mathcal{U}(d, 0, L_{max})$ where L_{max} is the longest expected text-line. The prior $P(j)$ is estimated from the training set as it is expected to be the same for forged and genuine text-lines.

So far the modeling of one of the two alignment distances has been explained. Next, these two measures have to be combined into one common framework. Two different measures are generated for each text-line: d_l and d_r, representing the distances of the left and right text-line point to the left and right alignment line respectively. In the end, one decision is needed. In order to combine these two features into one decision, the presented model is extended:

$$P(\neg f|d_l, d_r) = \frac{p(d_l, d_r|\neg f) \times P(\neg f)}{p(d_l, d_r)} \tag{6.13}$$

and

$$P(f|d_l, d_r) = \frac{p(d_l, d_r|f) \times P(f)}{p(d_l, d_r)} \tag{6.14}$$

where d_l and d_r are the distances of the left and right text-line point to the left and right alignment line respectively, $p(d_l, d_r|\neg f)$ is the likelihood of observing d_l, d_r knowing that the text-line is not forged and $p(d_l, d_r|f)$ is likelihood of observing the afore mentioned distances knowing that the text-line is forged.

For classification, the normalizing term is discarded. Assuming independence of d_l and d_r, the two terms are rewritten as:

$$\hat{P}(\neg f|d_l, d_r) = p(d_l|\neg f) \times p(d_r|\neg f) \times P(\neg f) \tag{6.15}$$

and

$$\hat{P}(f|d_l, d_r) = p(d_l|f) \times p(d_r|f) \times P(f) \tag{6.16}$$

where now $p(d_l|\neg f)$ and $p(d_r|\neg f)$ are measured from a training set and $p(d_l|f)$ and $p(d_r|f)$ are approximated by the mixture distribution described above. The prior is used as a sensitivity parameter to tune the system.

6.2.3 Integration of Skew and Alignment

In the previous two sections, the skew angle and the alignment features were considered separately for detecting implausible text-lines. In this section, the combination of both features into a combined framework is presented.

Two different combination schemes have been tested: the first scheme is a simple voting based on the two separate classification results: a text-line is classified as forgery if one of the two classifiers presented in the previous two sections "fires". This is a restrictive approach that should prevent that many forged text-lines are missed.

The second combination scheme extends the statistical model presented in the previous section to include the text-line skew angle α and the alignment distances d_l and d_r.

$$P(\neg f|\alpha, d_l, d_r) = \frac{p(\alpha, d_l, d_r|\neg f) \times P(\neg f)}{p(\alpha, d_l, d_r)} \tag{6.17}$$

and

$$P(f|\alpha, d_l, d_r) = \frac{p(\alpha, d_l, d_r|f) \times P(f)}{p(\alpha, d_l, d_r)} \tag{6.18}$$

The normalization factor is discarded and independence of the observations is assumed, leading to the following simplification:

$$\hat{P}(\neg f|\alpha, d_l, d_r) = p(\alpha|\neg f) \times p(d_l|\neg f) \times p(d_r|\neg f) \times P(\neg f) \tag{6.19}$$

and

$$\hat{P}(f|\alpha, d_l, d_r) = p(\alpha|f) \times p(d_l|f) \times p(d_r|f) \times P(f) \tag{6.20}$$

Thus, a text-line is classified as forged if $\hat{P}(f|\alpha, d_l, d_r) > \hat{P}(\neg f|\alpha, d_l, d_r)$ and vice-versa.

6.3 Evaluation

Evaluation of the proposed approaches showed to be a challenging task. On the one hand, no public real-world dataset could be found to do a meaningful evaluation on. On the other hand, apart from the observations made during the forgery experiment in Section 1, no statistics could be found on the methods used by amateur document forgers. For these reasons, new datasets had to be generated.

- the **Two-pass Print** 300 **dpi (TP300)** dataset contains 43 document images that were generated using a two pass printing process: in the first pass, an electronic document page was printed. Then, an electronic document was generated having extra text located in the whitespace at the end of the printed document page. This extra text was printed in a second pass on the paper. Finally, this paper was scanned using a resolution of 300 dpi. Images 001 to 015 were used as a training set for the method development. The remaining images were used as test set. Three different laser printers were used in different combinations, assuring that the second pass was always printed on the same printer as the first pass. This test set focusses on the scenario of a forger who wants to add additional text-lines to an original by printing them onto the original page.

- the **Print, Paste and Copy 300 dpi (PPC300)** dataset contains 49 document images that were generated using a manual modification process: in the first pass, an electronic document page was printed. Then, on a new sheet, a supplementary text-line was printed. This was cut out manually and pasted over an existing text-line on the printed page. A copy of this page was made. The copy has been scanned using a resolution of 300 dpi.
- the **Two-pass Print LaserJet (TPLJ)** dataset contains 64 document images that were generated using a similar two pass printing procedure as for the TP300 dataset. The first difference is that instead of deliberately adding extra text at the end of an integral document page, the page was printed in two passes, the first one printing all but the last part (e.g. leaving out the last paragraph) and the second pass exclusively prints the last paragraph. Again, the same printer was used for the two passes (a HP Laserjet 2100tn). The focus of this test set is again the two print process, but this time with much more constraint setup in order to find out what degree of exactness is possible under ideal circumstances. The only source of distortions that will be present are coming from the interaction with the printer hardware.
- the **Two-pass Print Color LaserJet (TPCLJ)** dataset contains 50 document images that were generated in exactly the same way as the TPLJ dataset, except that a different printer was used (a HP Color Laserjet 4650dn). The goal of this dataset is the same as for TPLJ, but this time with a different printer.
- the **Originals** dataset contains 30 document images that were generated by printing and scanning pages from an electronic document. This dataset is used to learn the distributions of the features for genuine text-lines.

All images have been binarized and deskewed before further processing. The deskewing method from Chapter 2 has been modified to use the median angle instead of the angle of the text-line with highest quality, to avoid deskewing the page by an angle of a forged text-line. It is reasonable to assume that for a given page, the number of forged text-lines is lower than the number of genuine text-lines. In the first part (Section 6.4.1) it will be shown, that the pre-processing of the document (scanning, binarization and deskewing) is stable and that it is thus likely that the distortions present in the measurements have been introduced by the forging procedure. The hypothesis about the distributions of the features for original text-lines are also verified. Therefore the text-line alignment distances and the skew angles have been extracted automatically from the *Originals* dataset.

The second part (Section 6.4.2) consists of evaluating the skew angle of text-lines for the plausibility check. The tests were run on all four test sets. For the TP300 and the PPC300 dataset, an extensive evaluation on the influence of both parameters $\sigma_{d_\alpha} \in [0.5, 8.0]$ and $P(f) \in [0.0, 1.0]$ has been done. The method has also been evaluated on the TPLJ and TPCLJ dataset.

The third test setup (Section 6.4.3) repeats the tests from the second setup for the alignment feature. The parameters of interest are $\sigma_d \in [1.0, 30.0]$ and $P(f) \in [0.0, 1.0]$. In both tests, the prior was sampled in with higher density (step size 0.005) in the start ($[0.0, 0.1]$) and end ($[0.9, 1.0]$) regions of the interval and with a step size of 0.02 for the middle part of the interval ($]0.1, 0.9[$).

The fourth part (Section 6.4.4) deals with the evaluation of the combination of both features into a combined framework by using reasonable parameter settings found in the isolated case. Both methods of classifier combination have been tested. For the statistical approach, the parameters $\sigma_\alpha = 2.5$ and $\sigma_d = 20.0$ were fixed according to the settings giving good performance from the previous experiments and sole the parameter $P(f)$ was varied. Tests were done on all four test datasets.

For the evaluation of the alignment distances, centered text-lines are being ignored as they do not fit the proposed model. In the case of the combination of both approaches, if a centered text-line is encountered, only the skew angle is considered to make a decision. Each line that does not start or end within a fixed threshold of six pixels from an alignment line and that has its middle point on the center alignment line is being considered as a centered text-line.

As evaluation measures, the receiver operating characteristic (ROC) [35] curves and the area under the ROC curve (AUC) are used.

6.4 Results

In the following section, the results of the different evaluation steps are presented. Hypothesis validation is presented in Section 6.4.1. Evaluation of the different features and the combination of both features are given in Sections 6.4.2, 6.4.3, and 6.4.4.

6.4.1 Hypothesis Validation

In this part, the hypothesis concerning the distributions of the skew and alignment text-line features are analyzed. The features were extracted on the *Originals* dataset containing 30 deskewed and genuine document images. The histogram of the extracted text-line skew angles is shown in Figure 6.7. It can be seen, that the vast majority of the extracted skew angles lies around 0°.

The histograms of the left and right alignment distances are shown found in Figure 6.8. As expected, the majority of the text-lines start at the left and end at

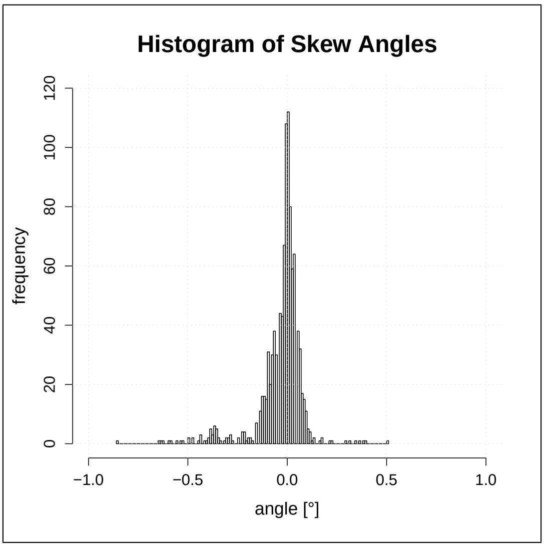


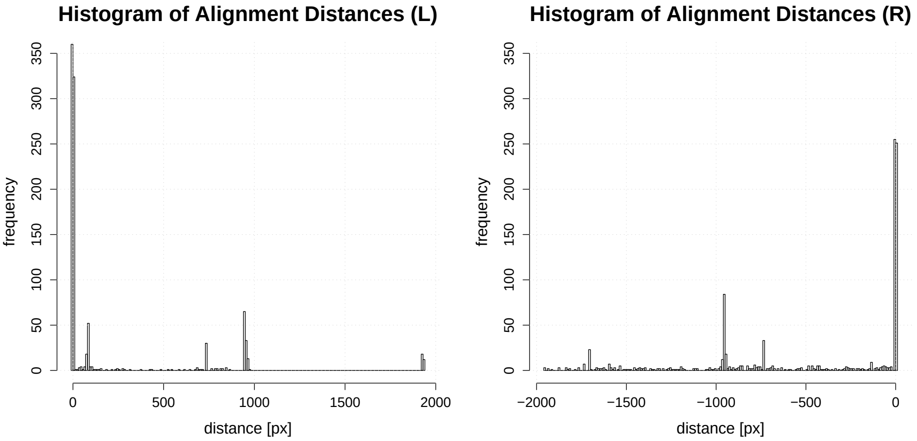
Figure 6.7: Histogram of skew angles measured on deskewed pages. The vast majority of measured skew angles is close to 0°. The text-lines were extracted from 30 genuine document pages.

the right alignment line. Also, peaks coming from indented text-lines are observed. It becomes clear, that a frequency measurement-based statistics for computing the likelihood $p(d_{l|r}|\neg f)$ is a reasonable choice.

A last test was done to test if the variations that are being measured do not come from the scanning or the pre-processing steps. Therefore, a small sample set of 21 document pages has been printed once. These pages have been repeatedly scanned, in total 15 times on the same scanner. For each run, the pre-processing pipeline has been run to obtain deskewed images. Also, the mean and standard deviation of the text-line skew angles and alignment distances have been measured. The variance of these parameters over all runs has been measured. Results can be found in Table 6.1. It can be seen that the variation between the different runs is quite low. It can thus be concluded, that the pre-processing steps are stable and that the variations measured in the test datasets are likely to be induced by document generation process rather than by the digitization process.

6.4.2 Results on Text-Line Skew Angles

Results on the TP300 and the PPC300 dataset are presented in Figure 6.9. It can be seen, that in both cases for the parameter $\sigma_\alpha = 2.5$ the best results in



(a) Left Alignment Distances

(b) Right Alignment Distances

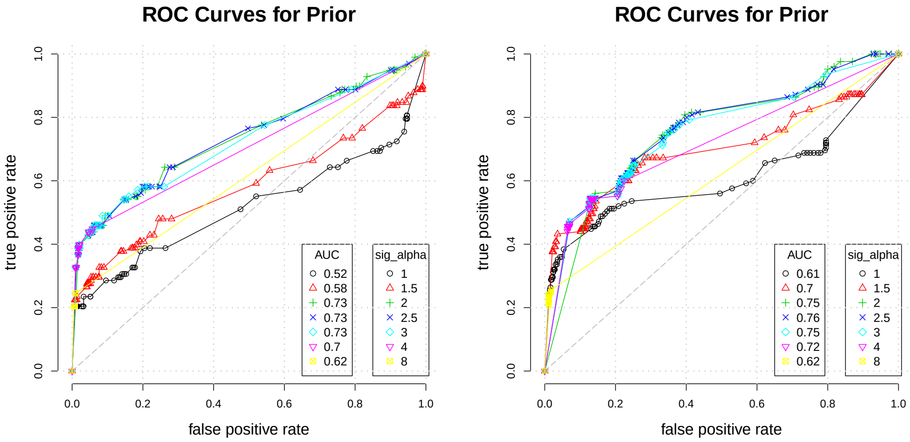
Figure 6.8: Histogram of the distances of the left and right line points to the corresponding alignment lines. As expected, the majority of lines start on the left and end on the right alignment line. Also, peaks can be identified that are likely to belong to indented lines.

terms of area under the ROC curve (AUC) is found. Also, small values of σ_α give bad performance. This is due to the high number of false positives (original text-line detected as a forged one), as the threshold for identifying a skew angle as suspicious is really low. Vice-versa, high values of $\sigma_\alpha > 3$ lead to a high number of false negatives (forged text-lines detected as original ones). It can also be seen, that for these high values, there are practically no points in the center and end part of the ROC curve, leading to the conclusion that for these parameters settings, the prior influences the outcome only for extreme values.

It can also be seen that the performance on the two pass printing forgeries is slightly lower compared to the manual forgeries. This goes along with the observation that the skew angle variations on the manually forged text-lines are much higher than for the two pass printed text-lines.

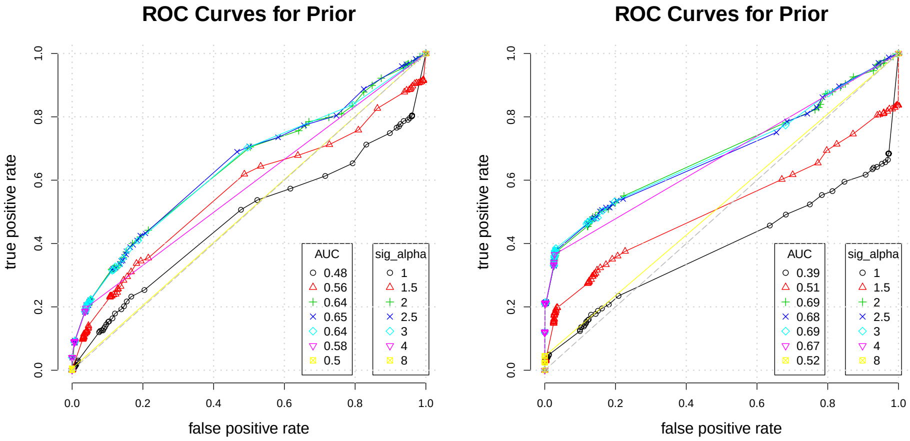
The results on the TPLJ and TPCLJ dataset are shown in Figure 6.10. The overall performance is comparable to the performance obtained for the TP300 dataset. Also, the characteristic gap of graph points in the middle of the curve is observed.

An analysis of the errors made by the proposed text-line skew examination gave the following results:



(a) TP300 (b) PPC300

Figure 6.9: Results for the test on the skew angle of text-lines on the TP300 and PPC300 datasets. The performance is best for $\sigma_\alpha = 2.5$.



(a) TPLJ (b) TPCLJ

Figure 6.10: Results for the test on the skew angle of text-lines on the TPLJ and TPCLJ datasets. It can be seen, that the performance is best for $\sigma_\alpha = 3.0$.

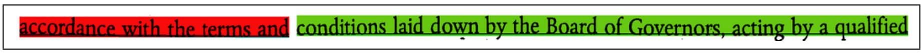
	mean α	std α	mean d_l	std d_l	mean d_r	std d_r
run-01	-0.004	0.112	96.7	363.3	-435.1	577.0
run-02	-0.006	0.106	92.3	363.5	-434.8	576.9
run-03	-0.010	0.117	96.2	362.1	-434.8	576.9
run-04	-0.004	0.107	96.3	363.6	-435.1	577.3
run-05	-0.010	0.115	96.5	363.5	-434.8	577.1
run-06	-0.002	0.104	94.0	362.8	-434.3	577.5
run-07	-0.012	0.110	92.0	364.5	-435.1	577.0
run-08	-0.010	0.109	95.3	363.6	-434.7	576.6
run-09	-0.007	0.116	95.2	362.5	-434.0	577.2
run-10	-0.005	0.108	95.9	363.4	-435.5	576.7
run-11	-0.004	0.106	92.2	363.5	-434.4	576.5
run-12	-0.010	0.111	96.2	363.2	-435.0	576.7
run-13	-0.012	0.123	95.5	363.5	-434.8	576.7
run-14	-0.015	0.112	96.5	363.4	-434.8	576.8
run-15	-0.011	0.105	90.8	364.5	-434.9	576.9
stddev	0.004	0.005	1.99	0.62	0.36	0.27

Table 6.1: Measurements of the features for different runs of the pre-processing pipeline consisting of scanning, binarization and deskewing. The low standard deviations show that the pre-processing step does not introduce any significant variations of the measurements.

- **line splits:** for the manually generated forgeries of the PPC300 dataset, it occurs, that the forged text-line is split into two text-lines by the text-line finding algorithm, due to the narrow constraints set on the positional error of the points lying on the text-line. It is observed, that pasting pieces of paper containing a single text-line on an existing document is a task that is likely to introduce not only skew of the text-line but also a certain degree of curl, leading to the split text-line during extraction. An example image illustrating the problem can be seen in Figure 6.11(a).

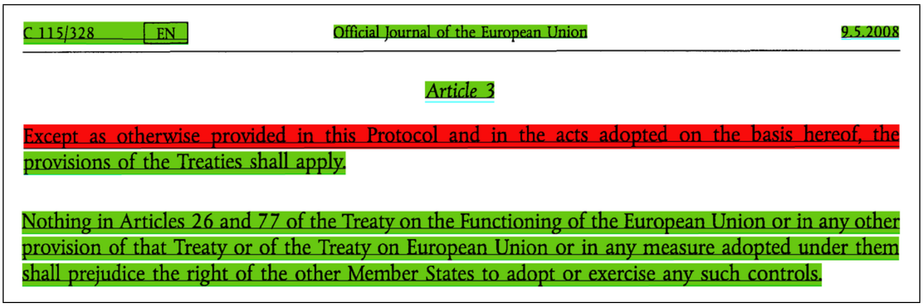
- **errors in deskewing:** in rare cases, when more forged text-lines are present than original ones, the deskewing will rotate the page in a way such that the majority of the text-lines are horizontally aligned. In this case it happens, that the forged text-lines are considered as original ones and vice versa. An illustration is shown in Figure 6.11(b).

- **limited discriminative power of the feature:** in several cases, there is no significant difference in skew angle of forged and original text-lines. In these cases, the method basing on skew angle measurement alone cannot detect the forgery. An example is given in Figure 6.11(c).



accordance with the terms and conditions laid down by the Board of Governors, acting by a qualified

(a) Split line error: due to the curvature of the line, the text-line finding split the line into two parts, leading to a true positive and a false negative count for this text-line.



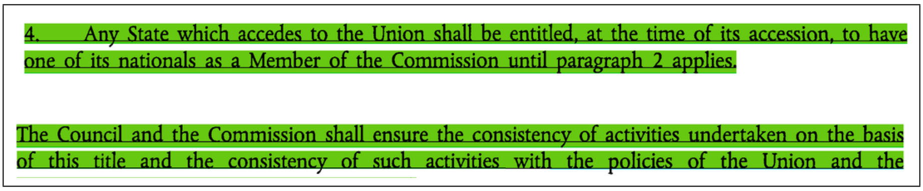
C 115/328 EN Official Journal of the European Union 9.5.2008

Article 3

Except as otherwise provided in this Protocol and in the acts adopted on the basis hereof, the provisions of the Treaties shall apply.

Nothing in Articles 26 and 77 of the Treaty on the Functioning of the European Union or in any other provision of that Treaty or of the Treaty on European Union or in any measure adopted under them shall prejudice the right of the other Member States to adopt or exercise any such controls.

(b) Deskewing error: as the forged part contains more text-lines than the original lines, the deskewing process rotated the page such that the forged lines are perfectly horizontal. A genuine text-line is therefore classified as forged line.



4. Any State which accedes to the Union shall be entitled, at the time of its accession, to have one of its nationals as a Member of the Commission until paragraph 2 applies.

The Council and the Commission shall ensure the consistency of activities undertaken on the basis of this title and the consistency of such activities with the policies of the Union and the

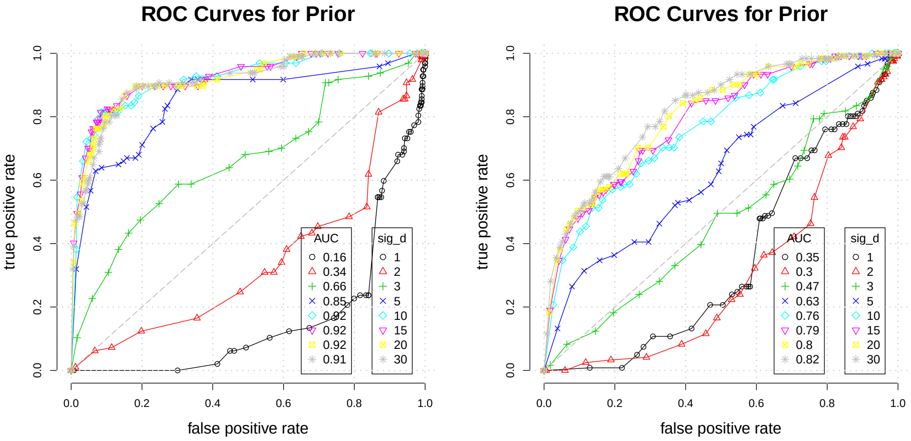
(c) Error due to the discriminative limitations of the feature. The last two lines have no significant difference in skew angle compared with the first two lines and are therefore classified as genuine.

Figure 6.11: Example of errors for the skew-base classification of forged text-lines. Green lines have been classified as valid text-lines, red ones have been classified as forged lines by the system.

The first two error types reduce the accuracy of the proposed method in the current evaluation setup, but in a real-world scenario, these problems would be less important, as in both cases suspicious lines would be reported to the human operator who can easily see that there are issues related to that document page. For the last problem, no solution can be found relying on the skew angle alone. Other features have to be added to detect these forgeries.

6.4.3 Results on Text-Line Alignment

The results on the TP300 and the PPC300 dataset can be found in Figure 6.12. The figures show different ROC curves for different values of σ_d and a varying



(a) TP300

(b) PPC300

Figure 6.12: Results for the test on the alignment feature. Low values of σ_d lead to a bad performance. Values of 15.0 or higher work well for the analyzed dataset.

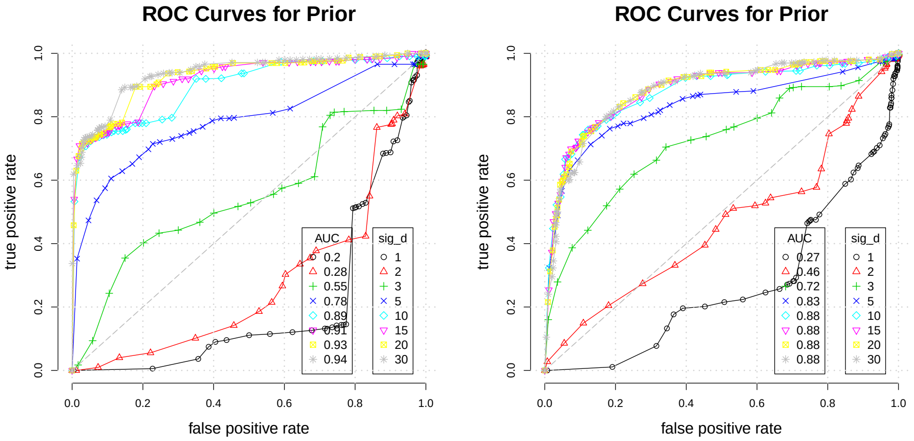
prior $P(f) \in [0.0, 1.0]$. For reasonable values of σ_d, the performance is good. For values of $\sigma_d > 10$ the performance does not improve significantly anymore.

In contrast to the previous results on the skew angle based approach, the results on the TP300 dataset are better than for the PPC300 dataset. This is likely due to the increased amount of control over the alignment in the manual forgery process, than in the case of the two pass print forgeries, where the alignment is only partially controllable by adjusting the document page in the paper tray accordingly.

The results on the TPLJ and TPCLJ dataset can be found in Figure 6.13. Again, the performance is better than on the manual forgeries, although slight differences between the two printers can be observed.

An inspection of the errors made by the system showed the following reasons for failure:

- **typographic enhancement** problems, as e.g. italics at the start of the line or different characters that reach outside the line boundaries, as e.g. an italic “J” at the start of the line: for optical correction reasons characters may stick outside of the normal text-line borders, leading to false positives as can be seen in Figure 6.14(a).
- **wrong alignment line:** when more indented text-lines are present than regular ones or if the number of forged text-lines is higher than the number of original text-lines, the wrong alignment lines may be found leading to



(a) TPLJ (b) TPCLJ

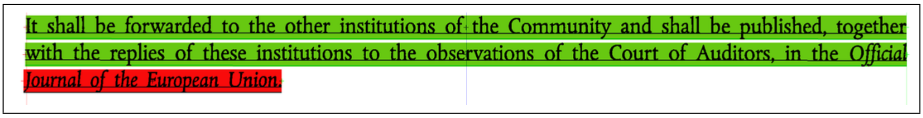
Figure 6.13: Results for the test on the skew alignment of text-lines on the TPLJ and TPCLJ datasets. It can be seen, that a good performance is obtained for values of $\sigma_d \geq 15.0$ approximatively.

misclassification of genuine lines. This rare error leads to both false positive and false negative classifications. An example is shown in Figure 6.14(b).

- **errors in text-line finding:** in rare cases text-lines extend to border noise or are split into two parts.
- **indented text-lines:** text-lines from enumerations are slightly indented, leading to occasional false positives, as can be seen in Figure 6.14(d)
- **limited discriminative power of the feature:** in some cases the left and right alignment of the forged text-lines is so accurate that a detection using the alignment feature is not feasible. This problem leads to false negative classifications. An illustrative example is given in Figure 6.14(c).

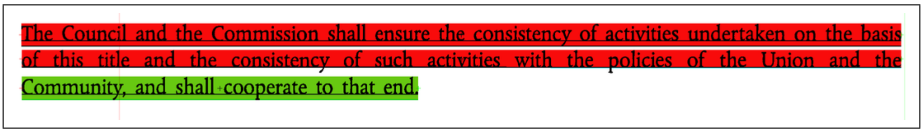
The problem with the optical corrections and with enumerations could be solved by integrating optical character recognition information to the process. Text-lines starting with numbers, italic characters or ending with characters presenting these problems could be treated differently, e.g. by allowing more variation for these text-lines.

The problem of extracting the wrong alignment line could be encountered by extending the method to not only extract one alignment line but to extract more



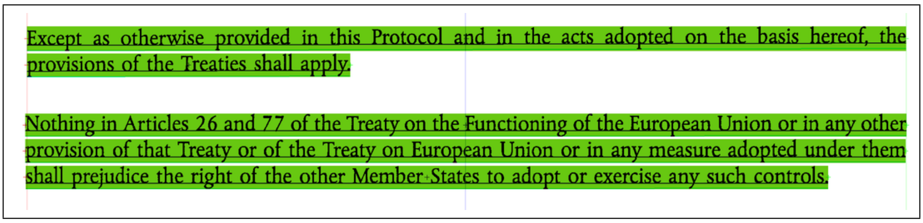
It shall be forwarded to the other institutions of the Community and shall be published, together with the replies of these institutions to the observations of the Court of Auditors, in the *Official Journal of the European Union.*

(a) Error leading to a false positive due to optical corrections on the italic character *J*. The italic "J" extends outside of the text-line boundary.



The Council and the Commission shall ensure the consistency of activities undertaken on the basis of this title and the consistency of such activities with the policies of the Union and the Community, and shall cooperate to that end.

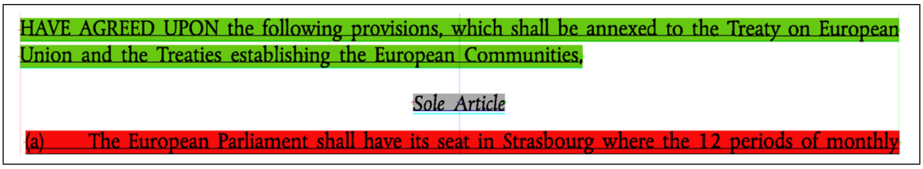
(b) Error due to a wrong alignment line. The last line is falsely classified as not forged. The upper two lines are still correctly identified as forged because of the bad alignment on the right side.



Except as otherwise provided in this Protocol and in the acts adopted on the basis hereof, the provisions of the Treaties shall apply.

Nothing in Articles 26 and 77 of the Treaty on the Functioning of the European Union or in any other provision of that Treaty or of the Treaty on European Union or in any measure adopted under them shall prejudice the right of the other Member States to adopt or exercise any such controls.

(c) Error due to the discriminative limitations of the feature. The last three lines are not recognized as forged lines. The last three lines show forged lines with accurate alignment to genuine text-lines.



HAVE AGREED UPON the following provisions, which shall be annexed to the Treaty on European Union and the Treaties establishing the European Communities,

Sole Article

(a) The European Parliament shall have its seat in Strasbourg where the 12 periods of monthly

(d) Error due to indentation: the line of the enumeration is slightly indented and is therefore falsely classified are a forged text-line.

Figure 6.14: Example of errors for the alignment-based classification of forged text-lines.

and compare them pairwise by computing their distance. This feature could then be used to detect suspicious alignment line constellations.

6.4.4 Results on the Combination of Both Features

The results for the combination using the voting scheme that classifies a text-line as forged if at least one of the two separate classifiers does so can be found in Figures 6.15 for TP300 and PPC300 and in Figure 6.16 for TPLJ and TPCLJ test sets. It can be seen that in three of the four cases, the combination outperforms the best of the single classifiers on the respective dataset, using the AUC as performance measure.

For the test on the combination using the statistical model, the parameters σ_α and σ_d where fixed with values that proved to perform well in the previous tests: $\sigma_\alpha = 2.5$ and $\sigma_d = 20.0$.

The results for the TP300 and the PPC300 datasets can be found in Figure 6.17. The results for printer specific tests on TPLJ and TPCLJ dataset are shown in Figure 6.18.

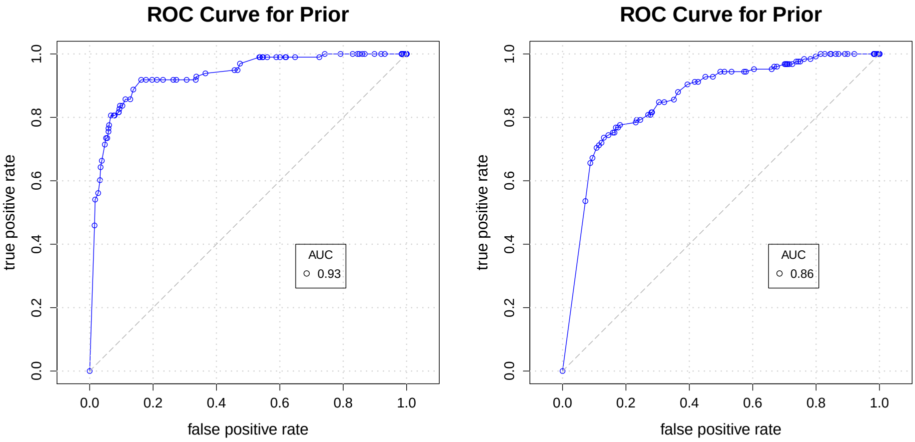
In comparison to the results of the single features, using the AUC as a measure, the combination of the two features does not significantly change the performance. One can also see that for some datasets, the combination slightly outperforms the best single feature (PPC300, TPCLJ) but for the other two cases using only the alignment would have given a slightly better result in terms of AUC.

6.5 Conclusion & Future Work

In this chapter, the first approach for automatic plausibility checks for forgery detection on printed documents has been presented. The text-line skew angle and alignment features have been integrated into a statistical framework for automatically detecting implausible skew angles or alignment distances. Extensive evaluation of the proposed methods on different datasets has been done to show the usefulness of the approach.

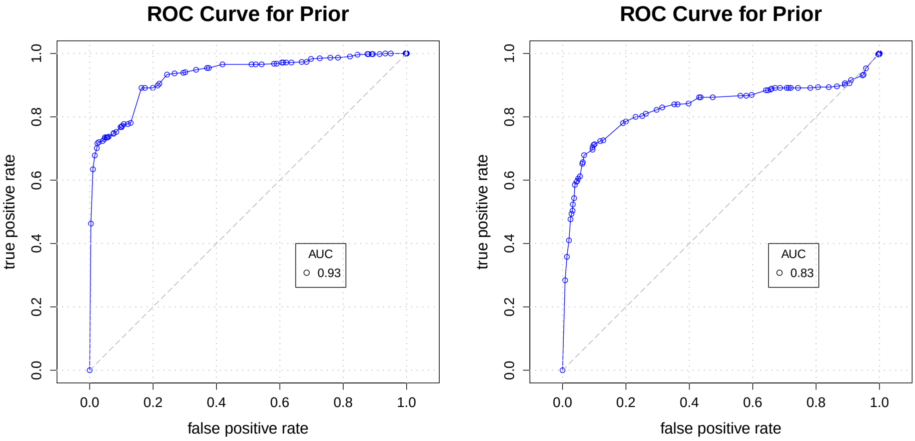
Future work on the proposed method will have to analyze how to extend the method to multi-level indented text and columns as they commonly occur in invoices. Instead of extracting only one alignment line for each border, several alignment lines could be extracted and compared pairwise. A similar model on the distances between alignment lines could be elaborated as for the distances of text-lines to the alignment lines.

Another important aspect is the extension of the evaluation using real-world datasets or at least datasets that have been generated by a wide variety of different persons and methods, in order measure the impact of the method in the different forging scenarios.



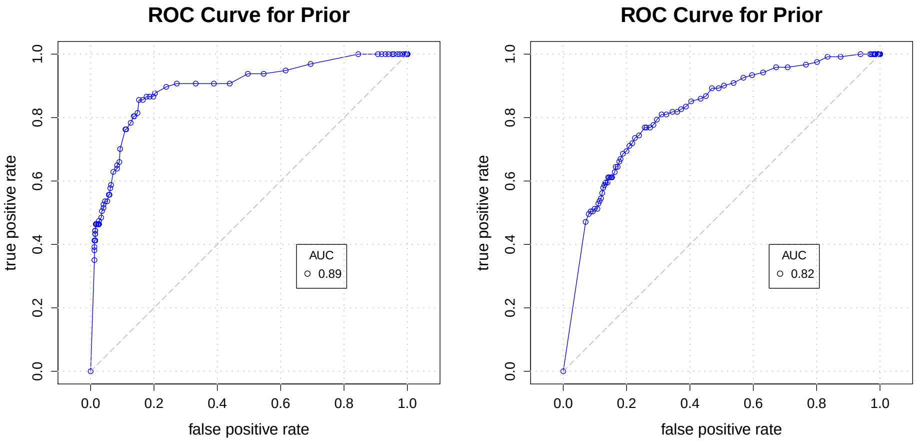
(a) TP300 (b) PPC300

Figure 6.15: Results for the test on the combination of the alignment and the skew angle feature using the voting scheme for combining the results of both classifiers.



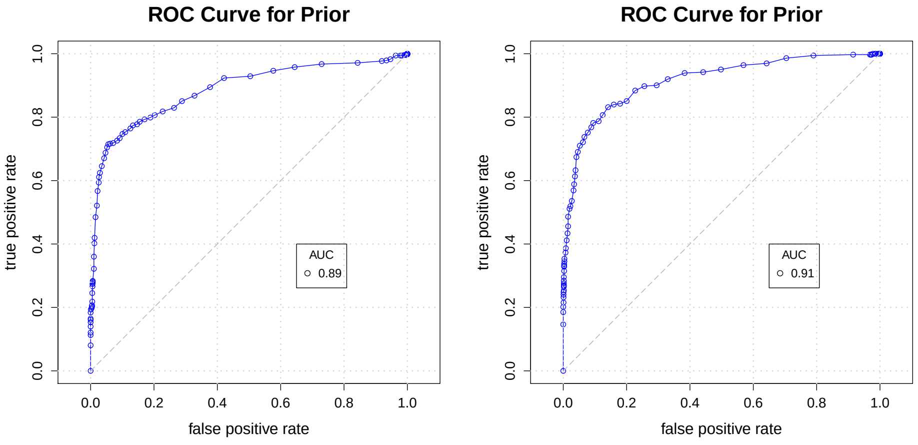
(a) TPLJ (b) TPCLJ

Figure 6.16: Results for the test on the combination of the alignment and the skew angle feature using the voting scheme for combining the results of both classifiers.



(a) TP300 (b) PPC300

Figure 6.17: Results for the test on the alignment feature on the TP300 and PPC300 dataset, using the statistical model to combine the skew angle and the alignment distance features.



(a) TPLJ (b) TPCLJ

Figure 6.18: Results for the test on the skew alignment of text-lines on the TPLJ and TPCLJ datasets, using the statistical model to combine the skew angle and the alignment distance features.

Chapter 7

Discussion

The methods presented in this thesis address the increasingly common problem of document forgery by standard printing and scanning means. The core concern of this thesis was to find out if it is possible to automatically or semi-automatically detect altered documents. Experimental results in Chapters 5 and 6 show clearly that this is possible: it could be shown in Chapter 5 that it is even feasible to detect copies of genuine documents as altered documents. This proves that the proposed methods can be used to effectively filter forgeries generated using standard printing and scanning technology.

The question that arose in Chapter 1 concerning the features to use for detecting forged documents was answered in Chapters 5 and 6. Using document intrinsic features, two model-based approaches for document authentication are presented in Chapter 5. It could be shown that the counterfeit protection system codes generated by color laser printers are a strong feature, which can reliably be used to detect if the same printer or printer type was used for two print-outs. A more general approach for document authentication uses the positional variations of characters to measure the distortions introduced by scanning and printing. Forgeries and even most copies could be detected as not being genuine. Finally, text-line alignment and skew were presented in Chapter 6 as additional features for alteration detection. It could be shown that even if no previous knowledge about the questioned document is available, verification of the text-line consistency in a document can be successfully applied for forgery detection.

The question concerning the requirements for the document pre-processing was covered in Chapters 2, 3 and 4: to allow accurate measuring of the text-line skew angles, an integrated approach for orientation and skew detection has been presented in Chapter 2. It could be shown that this method provides competitive results on standard datasets while having the advantage that it can be computed with only little overhead as it uses text-line information that is needed anyway in later processing steps. Also, in Chapter 3, a document cleanup method was presented to specifically remove marginal noise by detecting the page content's bounding rectangle, the *page frame*. A comprehensive and multi-facet evaluation scheme has been introduced. This proved the good performance of the developed approach. After document cleanup, logical labeling of the document contents has been presented in Chapter 4. A case-based reasoning approach has been followed for developing a flexible yet accurate labeling method. This can be used to extract meta-data from documents, which is suited for finding the corresponding model in the model-based forgery detection approaches.

Despite of all these achievements, it could be argued that a professional forger can generate a falsified document that the proposed system cannot detect. This may be true and therefore, automatic document forgery detection remains a challenge: the scanning and printing techniques will further evolve, and so have to do the counterfeit detection techniques. This thesis presented a first important step into the direction of making document digitization systems secure against forgeries. The long-term goal is to develop a wide variety of approaches that, combined into one toolkit, are able to analyze numerous features automatically and condense the results into a single decision, whether a questioned document should raise an alert or not. The toolkit will then also support the operator in verifying the outcome.

In order to continue on the path of the development of such a toolkit, new features will be analyzed and integrated: questioned document examiners have a wide variety of optically measurable features, such as font comparison, text-line spacing, paper analysis, etc. and it has to be analyzed which of these can reliably be computed in an automatic process. But also the existing methods can be improved: the positional variation modeling has to automatically detect the areas on which the scanning and printing distortions can be measured. Also the use of CPS codes has to be extended in a way to decode patterns for which the decoding scheme is known. Then print-outs from printers encoding the time and date in the pattern can be identified. An important step will also be large-scale tests in practical settings: although the data sets used throughout this thesis have been generated in a way to represent real-world forgeries well, real world data is useful to further improve the presented approaches. Furthermore, new features and further directions of development can be identified when working on real-world data.

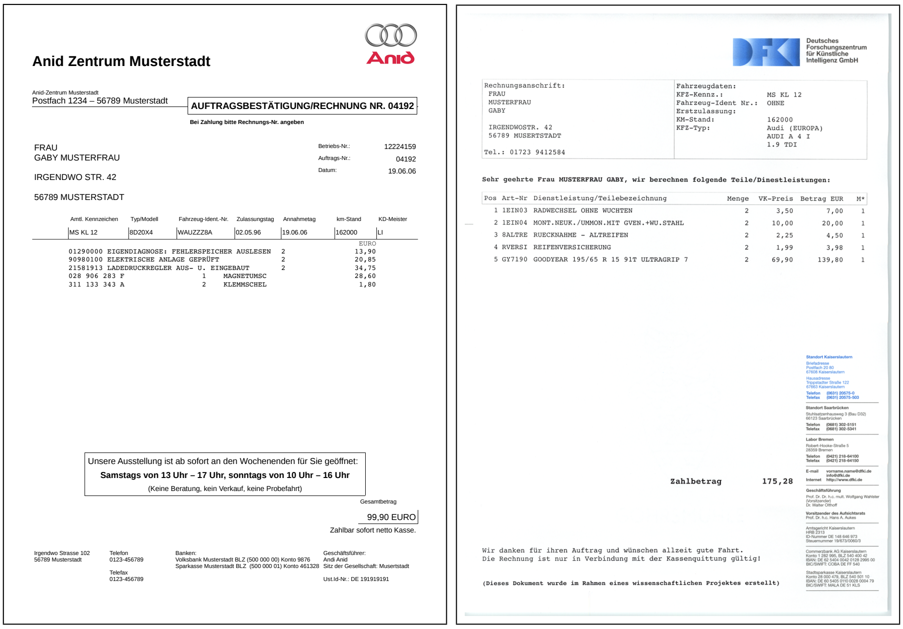
Appendix A

Survey on Forgery Techniques

One major problem in the development of techniques for forgery detection is that it is not known how the man on the street would forge a document. It is reasonable to assume that he will use standard hard- and software to solve this task, but even with this restriction there are many possible scenarios leading to various kinds of defects.

In an effort to get an overview on how people would forge documents using standard hard- and software a small experiment was setup. In this experiment people were given two invoices of and the following description:

> *Have a look at the invoice and consider the following scenario: a person with criminal energy, Frau Gaby Musterfrau, can't have enough money and thinks of defrauding money from her insurance company. She recently had a technical problem with the car that is covered by her car insurance company. The idea she came up with is to somehow forge the original invoice and increase the total amount of money stated in the invoice. She will then send the invoice to the insurance company and get more money back than she actually paid to have the car repaired.*
>
> *Frau Gaby Musterfrau knows that the insurance company does not check the invoice with the repair shop. But that when something looks suspicious, the insurance company will do so. In that case, Gaby Musterfrau would be in trouble because she knows it is illegal to defraud the insurance company.*
>
> *Your Task: Play the role of Frau Gaby Musterfrau and forge the invoice with the goal of increasing the total amount of money.*



Anid Zentrum Musterstadt

Anid

Anid-Zentrum Musterstadt
Postfach 1234 – 56789 Musterstadt

AUFTRAGSBESTÄTIGUNG/RECHNUNG NR. 04192

Bei Zahlung bitte Rechnungs-Nr. angeben

FRAU
GABY MUSTERFRAU

IRGENDWO STR. 42

56789 MUSTERSTADT

Betriebs-Nr.: 12224159
Auftrags-Nr.: 04192
Datum: 19.06.06

Amtl. Kennzeichen	Typ/Modell	Fahrzeug-Ident.-Nr.	Zulassungstag	Annahmetag	km-Stand	KD-Meister
MS KL 12	8D20X4	WAUZZZ8A	02.05.96	19.06.06	162000	LI

```
                                                          EURO
01290000 EIGENDIAGNOSE: FEHLERSPEICHER AUSLESEN  2        13,90
90980100 ELEKTRISCHE ANLAGE GEPRÜFT              2        20,85
21581913 LADEDRUCKREGLER AUS- U. EINGEBAUT       2        34,75
028 906 283 F                 1    MAGNETUMSC             28,60
311 133 343 A                 2    KLEMMSCHEL              1,80
```

Unsere Ausstellung ist ab sofort an den Wochenenden für Sie geöffnet:
Samstags von 13 Uhr – 17 Uhr, sonntags von 10 Uhr – 16 Uhr
(Keine Beratung, kein Verkauf, keine Probefahrt)

Gesamtbetrag
99,90 EURO
Zahlbar sofort netto Kasse.

Irgendwo Strasse 102
56789 Musterstadt

Telefon
0123-456789

Telefax
0123-456789

Banken:
Volksbank Musterstadt BLZ (500 000 00) Konto 9876
Sparkasse Musterstadt BLZ (500 000 01) Konto 461328

Geschäftsführer:
Andi Anid
Sitz der Gesellschaft: Musertstadt

Ust.Id-Nr.: DE 191919191

(a) Anid Invoice

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Rechnungsanschrift:
FRAU
MUSTERFRAU
GABY

IRGENDWOSTR. 42
56789 MUSERTSTADT

Tel.: 01723 9412584

Fahrzeugdaten:
KFZ-Kennz.: MS KL 12
Fahrzeug-Ident Nr.: OHNE
Erstzulassung:
KM-Stand: 162000
KFZ-Typ: Audi (EUROPA)
AUDI A 4 I
1.9 TDI

Sehr geehrte Frau MUSTERFRAU GABY, wir berechnen folgende Teile/Dinestleistungen:

Pos	Art-Nr	Dienstleistung/Teilebezeichnung	Menge	VK-Preis	Betrag EUR	M*
1	1EIN03	RADWECHSEL OHNE WUCHTEN	2	3,50	7,00	1
2	1EIN04	MONT.NEUK./UMMON.MIT GVEN.+WU.STAHL	2	10,00	20,00	1
3	8ALTRE	RUECKNAHME - ALTREIFEN	2	2,25	4,50	1
4	RVERSI	REIFENVERSICHERUNG	2	1,99	3,98	1
5	GY7190	GOODYEAR 195/65 R 15 91T ULTRAGRIP 7	2	69,90	139,80	1

Zahlbetrag 175,28

Wir danken für ihren Auftrag und wünschen allzeit gute Fahrt.
Die Rechnung ist nur in Verbindung mit der Kassenquittung gültig!

(Dieses Dokument wurde im Rahmen eines wissenschaftlichen Projektes erstellt)

Standort Kaiserslautern
Briefadresse
Postfach 20 80
67608 Kaiserslautern
Hausadresse
Trippstadter Straße 122
67663 Kaiserslautern
Telefon (0631) 20575-0
Telefax (0631) 20575-503

Standort Saarbrücken
Stuhlsatzenhausweg 3 (Bau D32)
66123 Saarbrücken
Telefon (0681) 302-5151
Telefax (0681) 302-5341

Labor Bremen
Robert-Hooke-Straße 5
28359 Bremen
Telefon (0421) 218-64100
Telefax (0421) 218-64150

E-mail vorname.name@dfki.de
info@dfki.de
Internet http://www.dfki.de

Geschäftsführung
Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster
(Vorsitzender)
Dr. Walter Olthoff

Vorsitzender des Aufsichtsrats
Prof. Dr. h.c. Hans A. Aukes

Amtsgericht Kaiserslautern
HRB 2313
ID-Nummer DE 148 646 973
Steuernummer 19/673/0060/3

Commerzbank AG Kaiserslautern
Konto 1 282 995, BLZ 540 400 42
IBAN: DE 62 5404 0042 0128 2995 00
BIC/SWIFT: COBA DE FF 540

Stadtsparkasse Kaiserslautern
Konto 28 000 479, BLZ 540 501 10
IBAN: DE 60 5405 0110 0028 0004 79
BIC/SWIFT: MALA DE 51 KLS

(b) DFKI Invoice

Figure A.1: Genuine invoices handed to the candidates for the forgery test

The candidates were not given any information about how they should or could forge the document. As the candidates did not perform the forgery under supervision, they were asked not to discuss any ideas with their colleagues in order to avoid biasing the results into one direction. However, it could not be verified if the candidates sticked to this request.

The invoices that they were given were generated by the author using a word processing software. Genuine bills from previous car repairs were taken as a basis in order to have a realistic layout and to avoid copyright issues in case of publishing the dataset. Figure A.1 shows the two genuine invoices. The "Anid" invoice was printed on normal white paper. The "DFKI" invoice was printed on old stationary from the DFKI with a logo on top and business related information at the bottom right part of the page.

As the scenario implies that the test candidates are given a paper version of the invoice and hand back a forged paper document, sending the task around by email was not possible. Therefore, only a limited number of people could be asked for participation. Approximately 40 samples (cover letter, DFKI invoice and Anid invoice) have been distributed. A small award was advertised for the best [1] forgery of each invoice.

Unfortunately only 14 candidates delivered forgeries. In total 25 forgeries were obtained, as some candidates submitted several forgeries using different approaches. The approaches can be divided into the following categories:

- **Print, Paste & Copy (PPC) Forgeries:** these forgeries are generated by replacing a part of the invoice. This is done by printing the new text (presenting higher values or more items) on an empty sheet and pasting this part onto the genuine bill. This is than copied using a color copier.
- **Reverse Engineered Imitations (REI) Forgeries:** this approach follows the idea of generating an editable document by imitating the genuine invoice. The genuine invoice is scanned and used as a template to generate a new document by retyping all the text, putting the logos in place, etc.
- **Scan, Edit & Print (SEP) Forgeries:** in this category the forgeries are generated by digitizing the invoice and manipulating the digital image. Here, mostly only numbers were modified, e.g. increasing the price of a tire or increasing the number of tire from two to four.

An overview of the frequencies of the different approaches is given in Table A.1. The first line gives the number of samples that have been received for each class of forgery. The second line contains the number of candidates that provided a forgery of that type. It can be noted, that the SEP forgery approach is by far the most commonly used approach. An analysis of the background of the test candidate gives a possible explanation: out of the fourteen candidates, ten are computer scientists or study computer science, with an image processing background. The other four have a less technical background. In the table, this information is broke down to the number forgery types per background. This information is given in the last two rows of the table. It can be seen that the computer scientists tend to make REI and SEP forgeries, whereas PPC forgeries are mainly produces by persons with non-technical backgrounds.

Manual inspection of the documents has been done to see what peculiarities to forgeries show. First, the distortion is examined. Therefore, two unique horizontally overlapping feature points and two vertically overlapping feature points are selected and their distances are measured using a ruler. The results for the "Anid" invoice can be found in Table A.2

The quality of the logos was also analyzed. Great differences in quality could be observed. Examples of the original and forged logos can be found in Figure A.2.

---

[1] "Best" has not been concretely defined. It was just said that the most genuine looking invoice would win

	PPC	REI	SEP
# samples	3	5	17
# of candidates	3	3	9
# candidates with computer science background	0	2	8
# candidates with other background	3	1	1

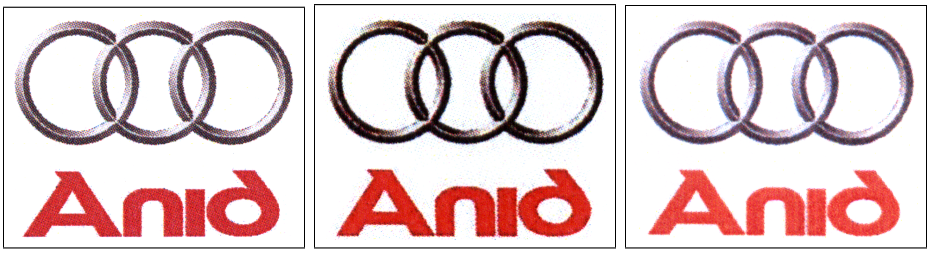
Table A.1: Distribution of the forgery types, the forgery samples and the background of the test candidates.

sample	Height[cm]	Width[cm]	Δ_W[cm]	Δ_H[cm]
Original	19.12	19.00	0.00	0.00
SEP-01	18.58	18.70	0.54	0.30
SEP-02	18.36	18.46	0.76	0.54
SEP-03	18.58	18.68	0.54	0.32
SEP-04	18.10	17.54	1.02	1.46
SEP-05	19.60	18.96	-0.48	0.04
SEP-06	18.60	18.40	0.52	0.60
SEP-07	17.96	17.60	1.16	1.14
SEP-08	19.10	19.02	0.02	-0.02
SEP-09	19.10	18.96	0.02	0.04
PPC-01	19.62	18.99	-0.5	0.01
PPC-02	19.61	18.84	-0.49	0.16
REI-01	18.80	18.10	0.32	0.90
REI-02	18.50	17.90	0.62	1.10

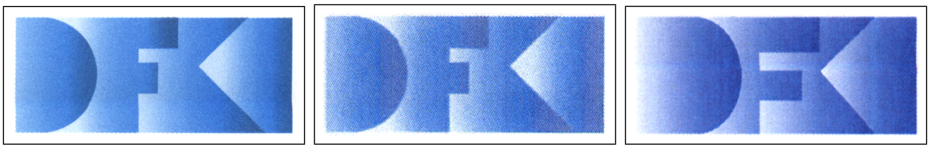
Table A.2: Distortion of the forged documents. The distance between two vertically overlapping and two horizontally overlapping features points were measured. It can be seen that for all forgery types, small variations are noticeable.

Concerning the font consistency there was only one forgery where at a first glance different fonts could be identified. Text-line skew and alignment variations were most significant in the PPC forgeries. Alignment inconsistencies for single characters could be seen in some SEP forgeries.

Concerning the printing techniques, all of but one forgeries have been printed using a color laser printer or a MFP machine that produces CPS codes. One PPC forgery had been copied on a black and white laser printer as for that person no color copier was available.



(a) Genuine Anid logo (b) Forged (1) (c) Forged (2)



(d) Genuine DFKI logo (e) Forged (1) (f) Forged (2)

Figure A.2: The leftmost logo shows the genuine logo. The other two are examples for forged logos. In (b) JPEG artifacts can be observed. In (c) it can be seen that the color saturation is less high. The logo in (e) shows unusual color variations in the originally dark blue areas. The forger that generated the page with logo (f) generated a REI forgery using a digital version of the DFKI logo downloaded from the Internet: the gradients are better than for (e) but the blue tone is slightly different and also the appearance of the single characters.

Appendix B

List of Comparisons for Evaluation on CPS

This appendix contains the list of all file comparisons that were used for the evaluation of the CPS pattern comparison in Section 5.2.4. The first two entries are the file names to be compared, followed by a one (same pattern) or a zero (different patterns). The last two numbers are the ground truth HPS and VPS distances in pixels.

```
#===== 0.96 x 0.48 (VPS x HPS)  =====
0001/0005.png 0001/0008.png 1 288 576
0001/0008.png 0005/0005.png 0 288 576
0005/0005.png 0005/0008.png 1 288 576
0005/0008.png 0011/0005.png 0 288 576
0011/0005.png 0011/0008.png 1 288 576
0011/0008.png 0025/0005.png 0 288 576
0025/0005.png 0025/0008.png 1 288 576
0025/0008.png 0050/0005.png 0 288 576
0050/0005.png 0050/0008.png 1 288 576
0050/0008.png 0052/0005.png 0 288 576
0052/0005.png 0052/0008.png 1 288 576
0052/0008.png 0057/0006.png 0 288 576
0057/0006.png 0057/0008.png 1 288 576
0057/0008.png 0059/0005.png 0 288 576
0059/0005.png 0059/0008.png 1 288 576
0059/0008.png 0060/0005.png 0 288 576
0060/0005.png 0060/0008.png 1 288 576
0060/0008.png 0061/0007.png 0 288 576
0061/0007.png 0061/0008.png 1 288 576
0061/0008.png 0063/0005.png 0 288 576
0063/0005.png 0063/0008.png 1 288 576
0063/0008.png 0064/0005.png 0 288 576
0064/0005.png 0064/0008.png 1 288 576
0064/0008.png 0066/0005.png 0 288 576
0066/0005.png 0066/0008.png 1 288 576
0066/0008.png 0067/0005.png 0 288 576
0067/0005.png 0067/0008.png 1 288 576
0067/0008.png 0001/0005.png 0 288 576

#===== 0.16 x 0.32 (VPS x HPS) =====
0003/0005.png 0003/0008.png 1 192 96
0003/0008.png 0009/0005.png 0 192 96
0009/0005.png 0009/0008.png 1 192 96
0009/0008.png 0037/0005.png 0 192 96
0037/0005.png 0037/0008.png 1 192 96
0037/0008.png 0055/0005.png 0 192 96
0055/0005.png 0055/0008.png 1 192 96
0055/0008.png 0056/0005.png 0 192 96
0056/0005.png 0056/0008.png 1 192 96
```

```
0056/0008.png 0003/0005.png 0 192 96

#===== 1.28 x 0.64 (VPS x HPS) =====
0004/0005.png 0004/0008.png 1 384 768
0004/0008.png 0018/0005.png 0 384 768
0018/0005.png 0018/0008.png 1 384 768
0018/0008.png 0021/0005.png 0 384 768
0021/0005.png 0021/0008.png 1 384 768
0021/0008.png 0034/0005.png 0 384 768
0034/0005.png 0034/0008.png 1 384 768
0034/0008.png 0044/0005.png 0 384 768
0044/0005.png 0044/0008.png 1 384 768
0044/0008.png 0053/0005.png 0 384 768
0053/0005.png 0053/0008.png 1 384 768
0053/0008.png 0004/0005.png 0 384 768

#===== 0.54 x 0.69 (VPS x HPS) =====
0006/0005.png 0006/0008.png 1 414 324
0006/0008.png 0033/0005.png 0 414 324
0033/0005.png 0033/0008.png 1 414 324
0033/0008.png 0042/0005.png 0 414 324
0042/0005.png 0042/0008.png 1 414 324
0042/0008.png 0045/0005.png 0 414 324
0045/0005.png 0045/0008.png 1 414 324
0045/0008.png 0006/0005.png 0 414 324

#===== 0.69 x 0.54 (VPS x HPS) =====
0008/0005.png 0008/0008.png 1 324 414
0008/0008.png 0010/0005.png 0 324 414
0010/0005.png 0010/0008.png 1 324 414
0010/0008.png 0019/0005.png 0 324 414
0019/0005.png 0019/0008.png 1 324 414
0019/0008.png 0022/0005.png 0 324 414
0022/0005.png 0022/0008.png 1 324 414
0022/0008.png 0023/0005.png 0 324 414
0023/0005.png 0023/0008.png 1 324 414
0023/0008.png 0024/0005.png 0 324 414
0024/0005.png 0024/0008.png 1 324 414
0024/0008.png 0028/0005.png 0 324 414
0028/0005.png 0028/0008.png 1 324 414
0028/0008.png 0029/0005.png 0 324 414
0029/0005.png 0029/0008.png 1 324 414
0029/0008.png 0030/0005.png 0 324 414
0030/0005.png 0030/0008.png 1 324 414
0030/0008.png 0031/0005.png 0 324 414
0031/0005.png 0031/0008.png 1 324 414
0031/0008.png 0032/0005.png 0 324 414
0032/0005.png 0032/0008.png 1 324 414
0032/0008.png 0046/0005.png 0 324 414
0046/0005.png 0046/0008.png 1 324 414
0046/0008.png 0048/0005.png 0 324 414
0048/0005.png 0048/0008.png 1 324 414
0048/0008.png 0049/0005.png 0 324 414
0049/0005.png 0049/0008.png 1 324 414
0049/0008.png 0008/0005.png 0 324 414

#===== 0.64 x 1.28 (VPS x HPS) =====
0015/0005.png 0015/0008.png 1 768 384
0015/0008.png 0039/0005.png 0 768 384
0039/0005.png 0039/0008.png 1 768 384
0039/0008.png 0040/0005.png 0 768 384
0040/0005.png 0040/0008.png 1 768 384
0040/0008.png 0041/0005.png 0 768 384
0041/0005.png 0041/0008.png 1 768 384
0041/0008.png 0015/0005.png 0 768 384
```

Appendix C

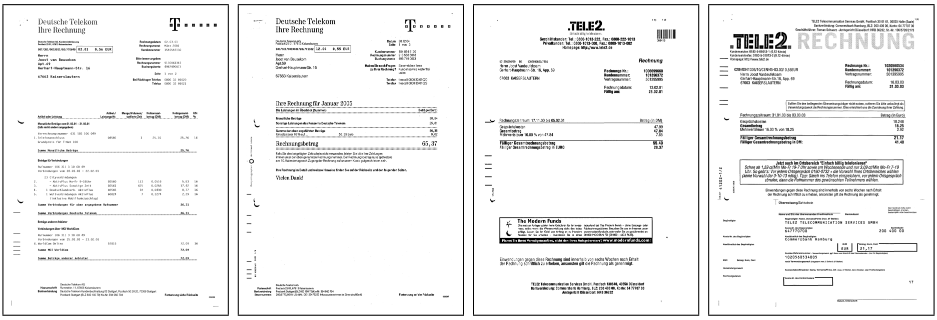
Ground-Truth Generation for Logical Labeling

To the author's best knowledge, there is no public dataset containing document images of invoices. In order to evaluate the logical labeling method in Chapter 4, a new dataset with corresponding ground truth had to be generated.

The ground truth information needed for the logical labeling contains two parts:

- **Block ground truth:** the block ground truth contains the geometric position of the different text blocks in the document image. Due to translational variances introduced to the document image during scanning and due to the varying block lengths, these blocks vary in position and in size.
- **Label ground truth:** for each block it has to be known what label it is assigned.

As manually labeling of ground truth is a cumbersome and time-intensive work, a semi automatic approach was chosen. The semi-automatic procedure is illustrated in Figure C.2. In the first step, not shown in the figure, document images are deskewed using the method described in Chapter 2. The second step is the manual step where for each layout type, one prototype is chosen. For this prototype an image map is generated that contains filled rectangles. Each rectangle stands for one text-block. The color of the rectangle is associated to a label (e.g. black being the label "invoice body"). Now each image from one layout type is aligned to its prototype. This is done using the document image alignment technique introduced in Section 5.3.1. The corresponding text-blocks in the aligned image are computed by merging all connected components that overlap with an image map ground truth text-block. Finally, the back transformation of the computed rectangles is done.



(a) Telekom old (b) Telekom new (c) Tele2 old (d) Tele2 new

Figure C.1: Layout type in invoice dataset. The two images to the left show the old and the new Telekom invoices. The images to the right show the old and the new Tele2 invoices.

	Telekom old	Telekom new	Tele2 old	Tele2 new
# samples	16	41	15	23
percentage	16.8	43.2	15.8	24.2

Table C.1: Distribution of layout types in the invoice dataset in absolute values (first row) and in [%] (second row).

The dataset consists of 95 documents from two different invoice sources. As both invoice sources changed the layout of their documents in the considered time interval, two layout type had to be distinguished for each invoice source, leading to four different layout types. One example of each layout type can be found in Figure C.1. The types have been named according to the invoice sources. Table C.1 shows how many layouts of which type are in the dataset.

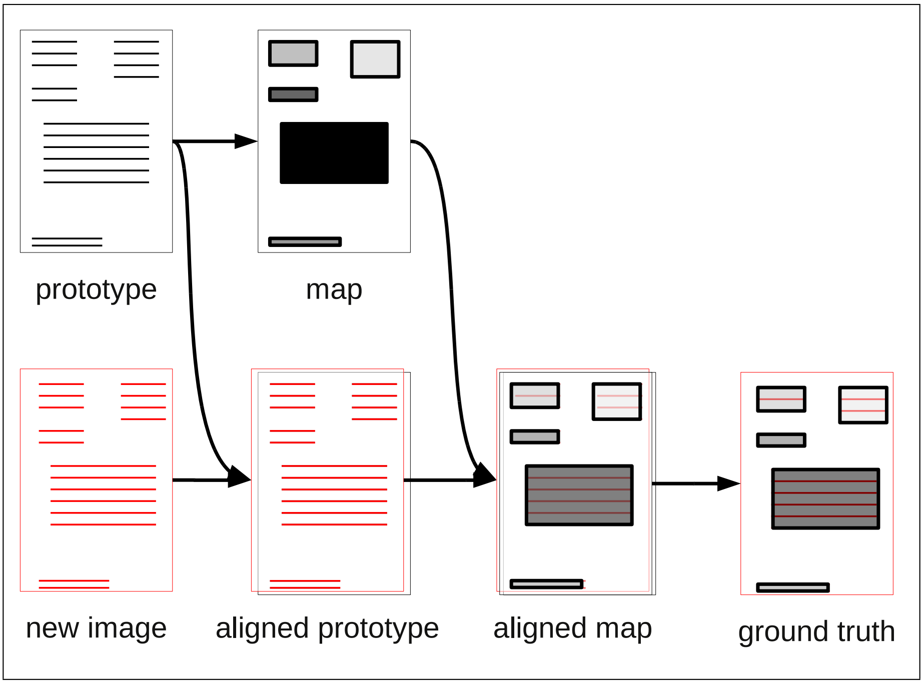


Figure C.2: Illustration of the ground truth generation for logical labeling. For each layout type, one prototype is selected. A ground truth map is generated for that prototype. Each document of that layout type is then aligned to the prototype. Using the computed transformation parameters, the position of the ground truth map in the original image is computed and the ground truth rectangles are computed by taking the minimum bounding rectangle that includes all connected components included into the rectangle of the image map.

Bibliography

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.

[2] M. Aiello, C. Monz, L. Todoran, and M. Worring. Document understanding for a broad class of documents. *Int. Jour. on Document Analysis and Recognition*, 5(1):1–16, 2002.

[3] G. N. Ali, P.-J. Chiang, A. K. Mikkilineni, J. P. Allebach, G. T. C. Chiu, and E. J. Delp. Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices. In *Proc. of the Int. Conf. on Digital Printing Technologies*, pages 511–515, New Orleans, LA, USA, September 2003.

[4] I. Amidror. A new print-based security strategy for the protection of valuable documents and products using moire intensity profiles. In *Proc. of SPIE Optical Security and Counterfeit Deterrence Techniques IV*, volume 4677, pages 89–100, San Jose, CA, USA, January 2002.

[5] A. Amin and S. Fischer. A document skew detection method using the hough transform. *Pattern Analysis and Applications*, 3(3):243–253, 2000.

[6] A. Antonacopoulos. Local skew angle estimation from background space in text regions. In *Proc. of the 4th Int. Conf. on Document Analysis and Recognition*, pages 684–688, Ulm, Germany, August 1997.

[7] A. Antonacopoulos, B. Gatos, and D. Bridson. Page segmentation competition. In *Proc. of the 9th Int. Conf. on Document Analysis and Recognition*, pages 1279–1283, Curitiba, Brazil, September 2007.

[8] V.N. Aradhya Manjunath, A. Rao, and G.H. Kumar. Language independent skew estimation technique based on gaussian mixture models: A case study on south indian scripts. In *Proc. of the 2nd Int. Conf on Pattern Recognition and Machine*, pages 487–494, Kolkata, India, December 2007.

[9] H.B. Aradhye. A generic method for determining up/down orientation of text in roman and non-roman scripts. *Pattern Recognition*, 38(11):2114–2131, 2005.

[10] B. T. Ávila and R. D. Lins. Efficient removal of noisy borders from monochromatic documents. In *Proc. of the 1st Int. Conf. on Image Analysis and Recognition*, pages 249–256, Porto, Portugal, September 2004.

[11] B. T. Ávila and R. D. Lins. A fast orientation and skew detection algorithm for monochromatic document images. In *Proc. of the 5th ACM symposium on Document engineering*, pages 118–126, New York, NY, USA, November 2005.

[12] H. S. Baird. *Background Structure in Document Images*, pages 17–34. World Scientific, Singapore, 1994.

[13] G. R. Ball, R. Stittmeyer, and S. N. Srihari. Writer verification in historical documents. In *Proc. of SPIE Document Recognition and Retrieval XVII*, volume 7543, pages 1–8, San Jose, CA, USA, January 2010.

[14] D. S. Bloomberg, G. E. Kopec, and L. Dasari. Measuring document image skew and orientation. In *Proc. of SPIE Document Recognition and Retrieval II*, pages 302–316, San Jose, CA, USA, February 1995.

[15] P. Bose, J.-D. Caron, and K. Ghoudi. Detection of text-line orientation. In *Proc. of the 10th Canadian Conf. on Computational Geometry*, pages 98–99, Montréal, Québec, Canada, August 1998.

[16] T. M. Breuel. Recognition by Adaptive Subdivision of Transformation Space: practical experiences and comparison with the Hough transform. In *Proc. of the IEE Colloquium on 'Hough Transforms'*, pages 71–74, May 1993.

[17] T. M. Breuel. A practical, globally optimal algorithm for geometric matching under uncertainty. *Electronic Notes in Theoretical Computer Science*, 46:1–15, 2001.

[18] T. M. Breuel. Robust least square baseline finding using a branch and bound algorithm. In *Proc. of SPIE Document Recognition and Retrieval IX*, pages 20–27, San Jose, CA, USA, January 2002.

[19] T. M. Breuel. Two geometric algorithms for layout analysis. In *Proc. on the 5th IAPR Workshop on Document Analysis Systems*, pages 188–199, Princeton, NY, USA, August 2002.

[20] T. M. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 90(3):258–294, 2003.

[21] T. M. Breuel. On the use of interval arithmetic in geometric branch-and-bound algorithms. *Pattern Recognition Letters*, 24(9–10):1375–1384, 2003.

[22] T. M. Breuel. Binary morphology and related operations on run-length representations. In *Proc. of the 3rd Int. Conf. on Computer Vision Theory and Applications*, pages 159–166, Funchal, Madeira, Portugal, January 2008.

[23] T. M. Breuel. The OCRopus open source OCR system. In *Proc. of SPIE Document Recognition and Retrieval XV*, volume 6815, pages 0F1–0F15, San Jose, CA, USA, January 2008.

[24] R. S. Caprari. Algorithm for text page up/down orientation determination. *Pattern Recognition Letters*, 21(4):311–317, 2001.

[25] R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena. Geometric layout analysis techniques for document image understanding: a review. Technical Report 9703-09, IRST, Trento, Italy, 1998.

[26] S. Chen, M. Y. Jaisimha, J. Ha, R. M. Haralick, and I. T. Phillips. Reference manual for the UW english document image database. Technical report, Seattle University, Washington, 1993.

[27] S. Chen and S. Srihari. A new off-line signature verification method based on graph. In *Proc. of the 18th Int. Conf. on Pattern Recognition*, pages 869–872, Hong Kong, China, August 2006.

[28] J. L. C. Chim, C.-K. Li, N. L. Poon, and S.-C. Leung. Examination of counterfeit banknotes printed by all-in-one color inkjet printers. *Journal of the American Society of Questioned Document Examiners*, 7(2):69–75, 2004.

[29] L. Cinque, S. Levialdi, L. Lombardi, and S. Tanimoto. Segmentation of page images having artifacts of photocopying and scanning. *Pattern Recognition*, 35(5):1167–1177, 2002.

[30] Fujitsu Cooperation. Datasheet fujitsu fi-5900c sheet-fed scanner. http://www.fujitsu.com/downloads/COMP/fcpa/scanners/fi-5900c_datasheet.pdf, accessed on 18.12.2009.

[31] A. K. Das, S. K. Saha, and B. Chanda. An empirical measure of the performance of a document image segmentation algorithm. *Int. Jour. on Document Analysis and Recognition*, 4(3):183–190, 2002.

[32] A.K. Das and B. Chanda. A fast algorithm for skew detection of document images using morphology. *Int. Jour. on Document Analysis and Recognition*, 4(2):109–114, 2001.

[33] E. Del Ninno, G. Nicchiotti, and E. Ottaviani. A general and flexible deskewing method based on generalized projections. In *Proc. of the 9th Int. Conf. on Image Analysis and Processing*, pages 632–638, Florence, Italy, September 1997.

[34] A. Dengel and G. Barth. ANASTASIL: Hybrid knowledge-based system for document image analysis. In *Proc. the 11th Int. Joint Conf. on Artificial Intelligence*, pages 1249–1254, Detroit, MI, USA, August 1989.

[35] R. O. Duda, Hart P. E., and Stork D. G. *Pattern Classification*, page 49. John Wiley & Sons, 2001.

[36] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[37] Electronic Frontier Foundation. Is your printer spying on you? http://www.eff.org/issues/printers, accessed on 18.12.2009.

[38] K. C. Fan, Y. K. Wang, and T. R. Lay. Marginal noise removal of document images. *Pattern Recognition*, 35(11):2593–2611, 2002.

[39] G. Ford and G. R. Thoma. Ground truth data for document image analysis. In *Symposium on Document Image Understanding Technology*, pages 199–205, Greenbelt, MD, USA, April 2003.

[40] G. H. Granlund. Fourier preprocessing for hand print character recognition. *IEEE Trans. on Computers*, C–21(2):195–201, 1972.

[41] J. Hails. *Criminal Evidence*, page 150. Thomson Learning, 2004.

[42] N. A. Hampp, M. Neebe, T. Juchem, M. Wolperdinger, M. Geiger, and A. Schmuck. Multifunctional optical security features based on bacteriorhodopsin. In *Proc. of SPIE Optical Security and Counterfeit Deterrence Techniques V*, volume 5310, pages 117–124, San Jose, CA, USA, January 2004.

[43] H. Hamza, Y. Belaid, and A. Belaid. A case-based reasoning approach for invoice structure extraction. In *Proc. of the 9th Int. Conf. on Document Analysis and Recognition*, pages 327–331, Curitiba, Brazil, September 2007.

[44] H. Hamza, Y. Belaid, and A. Belaid. Case based reasoning for invoice analysis and recognition. In *Int. Conf. on Case-based Reasoning*, pages 404–418, Belfast, Northern Ireland, August 2007.

[45] B. P. K. Horn. *Robot Vision*, pages 69–71. MIT Press, 1987.

[46] D. Ittner. Automatic inference of textline orientation. In *Proc. of the 2nd Annual Symposium on Document Analysis and Information Retrieval*, pages 123–133, Las Vegas, NV, USA, April 1992.

[47] Sauvola J. and Kauniskangas H. Mediateam document database ii, cd-rom collection of document images, 1999. http://www.mediateam.oulu.fi/downloads/MTDB/, accessed on 29.12.2009.

[48] T. Kanungo and R. M. Haralick. An automatic closed-loop methodology for generating character groundtruth for scanned documents. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(2):179–183, 1999.

[49] D. Keysers, F. Shafait, and T. M. Breuel. Document image zone classification - a simple high-performance approach. In *Proc. of the 2nd Int. Conf. on Computer Vision Theory and Applications*, pages 44–51, Barcelona, Spain, March 2007.

[50] N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. Delp. Survey of scanner and printer forensics at purdue university. In *Proc. of the 2nd Int. Workshop on Computational Forensics*, volume 5158 of *Lecture Notes in Computer Science*, pages 22–34, Washington, DC, USA, August 2008.

[51] J. Kim, D. X. Le, and G. R. Thoma. Automated labeling in document images. In *Proc. of SPIE Document Recognition and Retrieval VIII*, pages 111–122, San Jose, CA, USA, January 2001.

[52] K. Kise, M. Iwata, A. Dengel, and K. Matsumoto. A computational geometric approach to text-line extraction from binary document images. In *Proc. of the 3rd IAPR Workshop on Document Analysis Systems*, pages 346–355, Nagano, Japan, November 1998.

[53] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.

[54] S. Klink, A. Dengel, and T. Kieninger. Document structure analysis based on layout and textual features. In *Proc. of the 4th IAPR Workshop on Document Analysis Systems*, pages 41–52, Rio de Janeiro, Brazil, December 2000.

[55] Donald Ervin Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*, pages 74–87. Addison-Wesley, 1994.

[56] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[57] D. X. Le, G. R. Thoma, and H. Wechsler. Automated borders detection and adaptive segmentation for binary document images. In *Proc. of the 13th Int. Conf. on Pattern Recognition*, pages 737–741, Vienna, Austria, August 1996.

[58] D.S. Le, G.R. Thoma, and H. Wechsler. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition*, 27(10):1325–1344, 1994.

[59] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[60] C. K. Li and S. C. Leung. The identification of color photocopiers: A case study. *Journal of the American Society of Questioned Document Examiners*, 1(1):8–11, 1998.

[61] J. Liang, D. DeMenthon, and D. Doermann. Camera-based document image mosaicing. In *Proc. of the 18th Int. Conf. on Pattern Recognition*, pages 476–479, Hong Kong, China, August 2006.

[62] J. Liang and D. Doermann. Logical labeling of document images using layout graph matching with adaptive learning. In *Proc. on the 5th IAPR Workshop on Document Analysis Systems*, pages 212–223, Princeton, NY, USA, August 2002.

[63] J. Liang, R. M. Haralick, and I. T. Phillips. A statistically based, highly accurate text-line segmentation method. In *Proc. of the 5th Int. Conf. on Document Analysis and Recognition*, pages 551–555, Bangalore, India, September 1999.

[64] J. Liang, I. T. Phillips, and R. M. Haralick. Performance evaluation of document structure extraction algorithms. *Computer Vision and Image Understanding*, 84(1):144–159, 2001.

[65] Brian S. Lindblom and Robert Gervais. *Scientific Examination of Questioned Documents*, pages 238–241. Taylor and Francis, 2006.

[66] R.D. Lins and B.T. ÃĄvila. A new algorithm for skew detection in images of documents. In *Proc. of the 1st Int. Conf. on Image Analysis and Recognition*, pages 234–240, Porto, Portugal, September 2004.

[67] S. Lu and C. L. Tan. Automatic document orientation detection and categorization through document vectorization. In *Proc. of the 14th ACM Int. Conf. on Multimedia*, pages 113–116, New York, NY, USA, September 2006.

[68] S.J. Lu, J. Wang, and C.L. Tan. Fast and accurate detection of document skew and orientation. In *Proc. of the 9th Int. Conf. on Document Analysis and Recognition*, pages 684–688, Curitiba, Brazil, September 2007.

[69] Y. Lu and C.L. Tan. A nearest-neighbor chain based approach to skew estimation in document images. *Pattern Recognition Letters*, 24(14):2315–2323, 2003.

[70] V.N. Manjunath, G.H. Kumar, and P. Shivakumara. Skew detection technique for binary document images based on hough transform. *Int. Jour. of Information Technology*, 3(3):194–200, 2006.

[71] S. Mao, J. W. Kim, and G. R. Thoma. Style-independent document labeling: Design and performance evaluation. In *Proc. of SPIE Document Recognition and Retrieval X*, pages 14–22, Santa Clara, CA, USA, January 2003.

[72] S. Mao, A. Rosenfeld, and T. Kanungo. Document structure analysis algorithms: a literature survey. In *Proc. of SPIE Document Recognition and Retrieval X*, pages 197–207, Santa Clara, CA, USA, January 2003.

[73] S. Mao and G. R. Thoma. Bayesian learning of 2d document layout models for automated preservation metadata extraction. In *Proc. of the 4th IASTED Int. Conf. on Visualization, Imaging, and Image Processing*, pages 329–334, Marbella, Spain, September 2004.

[74] A. K. Mikkilineni, P.-J. Chiang, G. N. Ali, G. T. C. Chiu, J. P. Allebach, and E. J. Delp. Printer identification based on graylevel co-occurrence features for security and forensic applications. In *Proc. of SPIE Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 430–440, San Jose, CA, USA, February 2005.

[75] MIT Media Lab. Initiative to stop the use of tracking dots. http://www.seeingyellow.com, accessed on 18.12.2009.

[76] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 7(25):10–22, 1992.

[77] T. Nakai, K. Kise, and M. Iwamura. A method of annotation extraction from paper documents using alignment based on local arrangements of feature points. In *Proc. of the 9th Int. Conf. on Document Analysis and Recognition*, pages 23–27, Curitiba, Brazil, September 2007.

[78] Association of Certified Fraud Examiners. 2008 report to the nation on occupational fraud and abuse. Technical report, Association of Certified Fraud Examiners, Inc., 2008.

[79] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993.

[80] O. Okun, M. Pietikäinen, and J. Sauvola. Document skew estimation without angle range restriction. *Int. Jour. on Document Analysis and Recognition*, 2(2-3):132–144, 1999.

[81] O. Okun, M. Pietikainen, and J. Sauvola. Robust document skew detection based on line extraction. In *Proc. of the 11th Scandinavian Conference on Image Analysis*, pages 457–464, Kangerlussuaq, Greenland, June 1999.

[82] O. Okun, M. Pietikainen, and J. Sauvola. Robust skew estimation on low-resolution document images. In *Proc. of the 5th Int. Conf. on Document Analysis and Recognition*, pages 621–624, Bangalore, India, September 1999.

[83] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[84] J. L. Parker. An instance of inkjet printer identification. *Journal of the American Society of Questioned Document Examiners*, 5(1):5–10, 2002.

[85] C. Pateras and G. T. Toussaint. Bamboo fields: A new proximity graph and its application to text-line orientation estimation in document analysis. In *Proc. of the 3rd Workshop on Proximity Graphs*, pages 123–133, Starkville, MS, USA, December 1994.

[86] W. Peerawit and A. Kawtrakul. Marginal noise removal from document images using edge density. In *4th Postgraduate Workshop on Information and Computer Engineering*, Phuket, Thailand, January 2004.

[87] I. T. Phillips. User's reference manual for the UW english/technical document image database III. Technical report, Seattle University, Washington, 1996.

[88] W. Postl. Detection of linear oblique structures and skew scan in digitized documents. In *Proc. of the 8th Int. Conf. on Pattern Recognition*, pages 687–689, Paris, France, October 1986.

[89] D. Pu, G. R. Ball, and S. N. Srihari. A machine learning approach to off-line signature verification using bayesian inference. In *Proc. of the 3rd Int. Workshop on Computational Forensics*, volume 5718 of *Lecture Notes in Computer Science*, pages 125–136, The Hague, The Netherlands, August 2009.

[90] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *Proc. of the 7th Int. Conf. on Computer Vision*, pages 1165–1173, Corfu, Greece, September 1999.

[91] Y. Rangoni, F. Shafait, J. van Beusekom, and T. M. Breuel. Recognition driven page orientation detection. In *Proc. of the 16th Int. Conf. on Image Processing*, Cairo, Egypt, November 2009.

[92] M. Rieß. *Das schriftvergleichende Gutachten und seine Bedeutung in Verfahren nach der Strafprozeßordnung.* PhD, Fakultät für Philosophie, Psychologie und Erziehungswissenschaft der Universität Mannheim, 1988. ISBN 3-7950-0913-8.

[93] R. Safabakhsh and S. Khadivi. Document skew detection using minimum-area bounding rectangle. In *Proc. of the the Int. Conf. on Information Technology: Coding and Computing*, pages 253–258, Washington, DC, USA, March 2000.

[94] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[95] S. Schoen. Decoding identifying printer information. http://events.ccc.de/camp/2007/Fahrplan/events/1976.en.html, accessed on 18.12.2009.

[96] L. Schomaker and M. Bulacu. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *IEEE Trans. Pattern Analysis Machine Intelligence*, 26(6):787–798, 2004.

[97] L. Schomaker, M. Bulacu, and K. Franke. Automatic writer identification using fragmented connected-component contours. In *Proc. of the 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 185–190, Tokyo, Japan, Oktober 2004.

[98] M. Schreyer. Intelligent printing technique recognition and photocopy detection for forensic document examination. In *Proc. of Informatiktage 2009*, volume S-8, pages 39–42, Bonn, Germany, 2009.

[99] C. Schulze, M. Schreyer, A. Stahl, and T. M. Breuel. Evaluation of graylevel-features for printing technique classification in high-throughput document management systems. In *Proc. of the 2nd Int. Workshop on Computational Forensics*, volume 5158 of *Lecture Notes in Computer Science*, pages 35–46, Washington, DC, USA, August 2008.

[100] C. Schulze, M. Schreyer, A. Stahl, and T. M. Breuel. Using dct features for printing technique and copy detection. In *Proc. of the 5th Int. Conf. on Digital Forensics*, pages 95–106, Orlando, FL, USA, January 2009.

[101] F. Shafait, D. Keysers, and T. M. Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Proc. of SPIE Document Recognition and Retrieval XV*, volume 6815, pages 681510–681510, San Jose, CA, USA, January 2008.

[102] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(6):941–954, 2008.

[103] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel. Page frame detection for marginal noise removal from scanned documents. pages 651–660, Aalborg, Denmark, June 2007.

[104] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel. Document cleanup using page frame detection. *Int. Jour. on Document Analysis and Recognition*, 11(2):81–96, 2008.

[105] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

[106] P. J. Smith, P. O'Doherty, C. Luna, and S. McCarthy. Commercial anti-counterfeit products using machine vision. In *Proc. of SPIE Optical Security and Counterfeit Deterrence Techniques V*, volume 5310, pages 237–243, San Jose, CA, USA, January 2004.

[107] R. Smith. An overview of the Tesseract OCR engine. In *Proc. of the 9th Int. Conf. on Document Analysis and Recognition*, pages 629–633, Curitiba, Brazil, September 2007.

[108] A. L. Spitz. Correcting for variable skew in document images. *Int. Jour. on Document Analysis and Recognition*, 6(3):192–200, 2004.

[109] N. Stamatopoulos, B. Gatos, and A. Kesidis. Automatic borders detection of camera document images. In *Proc. of the 2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, pages 71–78, Curitiba, Brazil, September 2007.

[110] J. S. Tweedy. Class characteristics of counterfeit protection system codes of color laser copiers. *Journal of the American Society of Questioned Document Examiners*, 4(2):53–66, 2001.

[111] J. van Beusekom, D. Keysers, F. Shafait, and T. M. Breuel. Distance measures for layout-based document image retrieval. In *Proc. of the 2nd Int. Workshop on Document Image Analysis for Libraries*, pages 232–242, Lyon, France, April 2006.

[112] J. van Beusekom, D. Keysers, F. Shafait, and T. M. Breuel. Example-based logical labeling of document title page images. In *Proc. of the 9th Int. Conf. on Document Analysis and Recognition*, pages 919–923, Curitiba, Brazil, September 2007.

[113] J. van Beusekom, Y. Rangoni, and T. M. Breuel. Trainable multiscript orientation detection. In *Proc. of SPIE Document Recognition and Retrieval XVII*, volume 7543, pages 1–10, San Jose, CA, USA, January 2010.

[114] J. van Beusekom, M. Schreyer, and T. M. Breuel. Automatic counterfeit protection system code classification. In *Proc. of SPIE Media Forensics and Security XII*, San Jose, CA, USA, January 2010.

[115] J. van Beusekom, F. Shafait, and T. M. Breuel. Image-matching for revision detection in printed historical documents. In *Proc. of the 29th DAGM Symposium on Pattern Recognition*, pages 507–516, Heidelberg, Germany, 2007.

[116] J. van Beusekom, F. Shafait, and T. M. Breuel. Document signature using intrinsic features for counterfeit detection. In *Proc. of the 2nd Int. Workshop on Computational Forensics*, volume 5158 of *Lecture Notes in Computer Science*, pages 47–57, Washington, DC, USA, August 2008.

[117] J. van Beusekom, F. Shafait, and T. M. Breuel. Automatic line orientation measurement for questioned document examination. In *Proc. of the 3rd Int. Workshop on Computational Forensics*, volume 5718 of *Lecture Notes in Computer Science*, pages 165–173, The Hague, The Netherlands, August 2009.

[118] J. van Beusekom, F. Shafait, and T. M. Breuel. Resolution independent skew and orientation detection for document images. In *Proc. of SPIE Document Recognition and Retrieval XVI*, volume 7247, San Jose, CA, USA, January 2009.

[119] J. van Beusekom, F. Shafait, and T. M. Breuel. Combined orientation and skew detection using geometric text-line modeling. *Int. Jour. on Document Analysis and Recognition*, 2010.

[120] J. van Beusekom, F. Shafait, and T. M. Breuel. Document inspection using the text-line alignement. In *Proc. of the 9th IAPR Workshop on Document Analysis Systems*, Boston, MA, USA, June 2010. accepted for publication.

[121] R.L. van Renesse. Ordering the order, a survey of optical document security features. In *Proc. of SPIE Conference on Practical Holography IX*, pages 268–275, San Jose, CA, USA, February 1995.

[122] R.L. van Renesse. Paper based document security-a review. In *European Conf. on Security and Detection*, pages 75–80, London, UK, April 1997.

[123] R.L. van Renesse. Hidden and scambled images - a review. In *Proc. of SPIE Optical Security and Counterfeit Deterrence Techniques IV*, volume 4677, pages 333–348, San Jose, CA, USA, January 2002.

[124] R.L. van Renesse. Protection of high security documents - developments in holography to secure the future market and serve the public. In *Proc. of Holo-Pack.Holo-Print*, Vienna, Austria, November 2006.

[125] D. Xi, M. Kamel, and L. Seong-Whan. Skew estimation and correction for form documents using wavelet decomposition. In *Proc. of the 2nd Int. Conf. on Image Analysis and Recognition*, pages 182–190, Porto, Portugal, September 2004.

[126] F. Xie, Jiang Z.-g., and Wang L. Skew detection for form document based on elongate feature. In *Proc. of the 6th Int. Conf. on Energy Minimization Methods in Computer Vision and Pattern*, pages 127–136, Ezhou, China, August 2007.

[127] H. Zhou and Z. Liu. Page frame segmentation for contextual advertising in print on demand books. In *Proc. of 2nd IEEE Workshop on Internet Vision*, pages 17–22, Miami, FL, USA, June 2009.

[128] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.

Curriculum Vitae

Name: Joost van Beusekom

Date of Birth: February 28, 1981

Place of Birth: Luxembourg, Luxembourg

Education

School Education	Lycée de Garçons de Luxembourg, *Diplôme de fin d'études classiques*, 2000
University Education:	University of Kaiserslautern *Diploma in Computer Science (University of Kaiserslautern)*, 2006

Academic and Professional Experience

Feb. 2006 – May 2006	Researcher with DFKI GmbH, Kaiserslautern, Germany
Jun. 2006 – present	Researcher with IUPR research group at University of Kaiserslautern, Kaiserslautern, Germany