Deciphering Algorithms for Degraded Document Recognition

by

Chi Fang

A dissertation submitted to the
Faculty of the Graduate School of the
State University of New York at Buffalo
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

July 18, 1997

Committee Members

Chairman:	Dr. Jonathan J. Hull Ricoh California Research Center
Members:	Dr. Sargur N. Srihari Computer Science, SUNY at Buffalo
	Dr. Deborah K. Walters Computer Science, SUNY at Buffalo
Outside Reader:	Dr. Paul Gader Dept. of Computer Engineering and Computer Science University of Missouri at Columbia

This dissertation was defended on June 30, 1997.

To my parents and sister, for their lasting love.
And to Catherine. Her love, encouragement, understanding
and sacrifice made this possible.

Acknowledgements

I would like to take this opportunity to express my gratitude to the people whose love, friendship and encouragements not only supported me through the completion of this thesis, but also made the otherwise tedious graduate school life a whole lot more fun, colorful and rewarding.

Dr. Jonathan Hull, my thesis advisor, has always been the source of imagination. The many discussion sessions with Jon in his pleasant office at UB first led me into the wonderful field of OCR and then resulted in the evolution of my rich but vague academic interests into a concrete and focused research project. Being in a research field characterized by highly inter-disciplinary idea exchanges among computer science, psychology, linguistics and cognitive science, it is more than lucky to have an advisor always ready with insights and expertise, and most of all, with the heart and patience.

I would also like to thank Dr. Deborah Walters, member of the dissertation committee and my advisor during the first year of my graduate study in the Computer Science Department at UB, for her guidance of my research in vision and cognitive science and her continuous support throughout my thesis work. Her encouragements and understanding have always been the source of strength.

I would like to thank Dr. Sargur Srihari, member of my dissertation committee and director of the CEDAR center, for his guidance in the field of pattern recognition and his support for my thesis. The CEDAR center under his guidance provided an excellent diverse environment for me to learn the advanced technologies in OCR and pattern recognition.

Prof. Lin Chen, director of the Beijing Laboratory of Cognitive Sciences and my former academic advisor at the University of Science and Technology of China, led me into the field of vision and cognitive science. From him I learned the beauty of science and the excitement of inter-disciplinary idea exchange.

Kee Yip, Darryl Troxel and Mohan Kumar of TRW Enterprise Solution had been my supervisors at TRW. Thanks for their friendship, encouragements and help that made it possible for me to continue the research after leaving UB.

Robin Lo, Sheng Lee and Max Anzai of the System Research and Development Division, Ricoh Corp. provided me valued support to finish this thesis.

Thanks to Dr. Tao Hong for his friendship. The many discussions with him helped form some of the concrete ideas in this thesis.

Many colleagues and fellow graduate students at the CEDAR research center at UB have given me their valued supports. Thanks goes to Yanhong Li, Gongwei Wang, Jianchuan Gong, Dar-Shyang Lee, Liang Li, Anda Zhao, Jeffery Zhou, James Lii, Steve Lam, Victor

Zandy, Siamake Khoubyari, Ravindra Asundi, Debashish Niyogi, Thomas Pilarski and Afshin Shafie.

Special thanks to faculty of the Department of Computer Science, for providing the excellent graduate program and a diverse and free academic environment.

Finally I would like to extend my gratitude to Eloise Benzel, Gloria Koontz and Sally Elder of the Department of Computer Science, Eugenia Smith, Heike Berlind, Natalie Leaver and Rita Hamilton of CEDAR center, for their kindness and help through the last five years.

Abstract

The research presented in this thesis provides new solutions to two fundamental problems in document recognition. The first problem is *character segmentation*. Touching characters and character fragmentation caused by image degradation are difficult problems for current document recognition systems. The second problem that this thesis addresses is the dependence of today's document recognition systems on extensive *font training*. These two problems are shown to be the main reasons that cause performance breakdown for most of today's commercial document recognition systems.

Our research provides solutions to the two problems by seeking alternative approaches that can recognize degraded documents with high accuracy and robust performance. Reliable performance on degraded documents is achieved by avoiding these two difficult problems in the recognition approaches.

We propose to consider the computational task of document recognition as a process of finding the mapping between the visual patterns in the input document and their linguistic identities. Deciphering algorithms are proposed to decode the mapping by combining the visual constraints from the input document and the various levels of linguistic constraints from a language base. This deciphering approach to document recognition works on different levels of language granularities, both on the character level as well as on the word level. Therefore, document recognition can be achieved by decoding characters using character level linguistic constraints. It can also be achieved by the direct decryption of words using word level linguistic constraints.

A modified character-level deciphering algorithm is proposed based on an existing research. The modifications to an original algorithm extend a character-level deciphering algorithm to documents that contain touching characters. It also provides a solution to errors in character pattern clustering.

A word-level deciphering approach to document recognition is also proposed in this thesis. Word-level language constraints are used to first decrypt the words from a selected portion of the input text that has relatively more reliable word n-gram statistics. Word decryption is considered as a process of constraint satisfaction and is implemented by a probabilistic relaxation algorithm. After the decryption of words in a selected portion of the input text, font information is learned and then used to re-recognize the rest of the input text which was not used for deciphering. The re-recognition of the rest of the input text is achieved by a hypothesis testing algorithm.

The applicability of the proposed deciphering approaches to practical document recognition tasks is demonstrated by experimental results of applying the proposed approaches to scanned documents. The test documents used for the experiments contain image degradations including touching characters, character fragmentation, as well as individual character deformations.

Contents

Chapter 1

Introduction

Ever since the emergence of computer technology in the 1940's, a long-standing objective has been to build automatic reading machines with the power, tolerance and efficiency of a human reader [20, 156, 64, 111, 106, 63]. Research on computer document recognition has been an important part in the efforts to understand human reading and build automatic systems with matching performance.

From the point of view of information processing, document recognition can be considered as a mapping from the input of visual patterns distributed across a page to the output of an ordered list of linguistic identities. As shown in Figure 1.1, a typical computer document recognition system usually involves three general stages. Each of these stages may vary in complexity or extent depending on individual implementations. The first stage is the image preprocessing stage which prepares an improved visual input for the following classification stages. Tasks that are completed in this stage usually involve image noise removal, image enhancement, text block location and deskewing. Character segmentation or word segmentation may also be performed in this stage. The results of the image preprocessing stage are then passed to the second stage where character recognition or word recognition is carried out with pattern classification techniques. Classification results from the second stage may

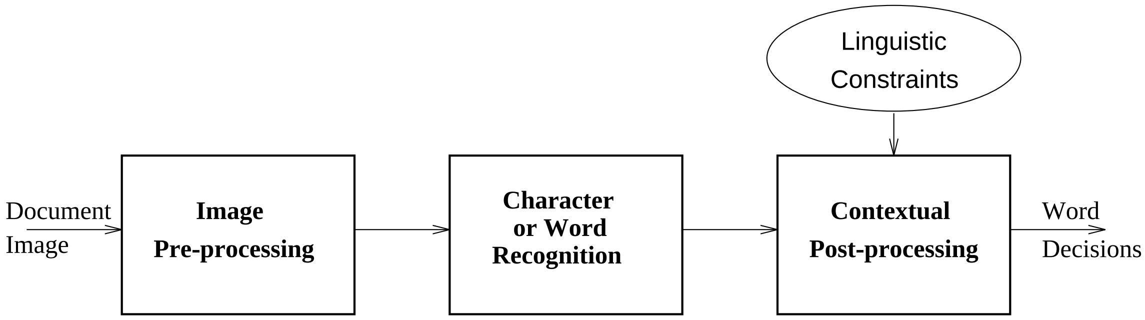


Figure 1.1: The processing stages in a document recognition system.

then be passed to the third stage where linguistic constraints are used to resolve ambiguities or mistakes in the classification results. Certain early implementations may not use any linguistic constraints at all to do post-processing. Whatever the results from the classification stage will be the final results of the recognition for those systems.

1.1 The Motivation

Efforts in building machine systems for document recognition have always been influenced by man's understanding of how human beings read documents. The seemingly effortlessness nature of human reading has allured many scholars into grossly under-estimating the complexity and difficulties in building machine systems for document recognition [154, 132]. Despite three decades of extensive research, solid progress has only been made in limited areas such as developing recognition algorithms for isolated machine-printed characters. Machine systems for document recognition still have limited capabilities for recognizing degraded documents [124, 138, 140, 139, 137].

Research and experiments have shown that the performance breakdown of commercial document recognition systems under real application situations is caused mainly by two reasons. The first is the difficulty in dealing with touching characters and character fragmentation that are abundant in real-world documents as a result of document degradations [26, 15, 30, 142, 113, 22]. Examples of touching characters and character fragmentation in documents are shown in Figure 1.2 and Figure 1.3. Touching characters and character fragmentation make it difficult to correctly segment character images for individual classification, and therefore poses severe difficulty to conventional document recognition systems that are critically dependent on character segmentation.

The second major problem that affects the performance of document recognition systems is that a recognition algorithm usually needs to be trained on many different fonts that can appear in an input document. Image degradation in documents makes effective font training even more difficult. The need for large amounts of font training is a significant impediment to the development of high performance commercial recognition systems [84, 7, 5, 8, 174].

A biological reconnaissance survey of the project site was conducted by Andrew C. Sanders of the University of California, Riverside, in August 1988. The survey, as well as consultations with the California Native Plant Society and a review of the California Natural Diversity Data Base, failed to turn up any evidence of threatened or endangered plant or animal species at the Second Afterbay site; however the Orange-throated Whiptail lizard (Cnemidophorus hyperythrus), a candidate species for Federal listing, was observed approximately ¾ of a mile to the south of the power plant in sage scrub habitat. The activities related to lining the Second Afterbay will not encroach on this area, but to minimize any potential impact, specific mitigation measures will be adopted to ensure that this species will not be affected by the lining activities.

A November 1986 archeological and historical site survey of the Second Afterbay area was conducted by Dr. Russell Barber of the California State University, San Bernardino. He concluded that the project area contained no cultural resources or sites potentially eligible for inclusion in the National Register of Historic Places.

MITIGATION MEASURES

To help mitigate some of the significant environmental effects identified in the attached Environmental Checklist, the following mitigation measures are being considered for incorporation as part of the proposed project:

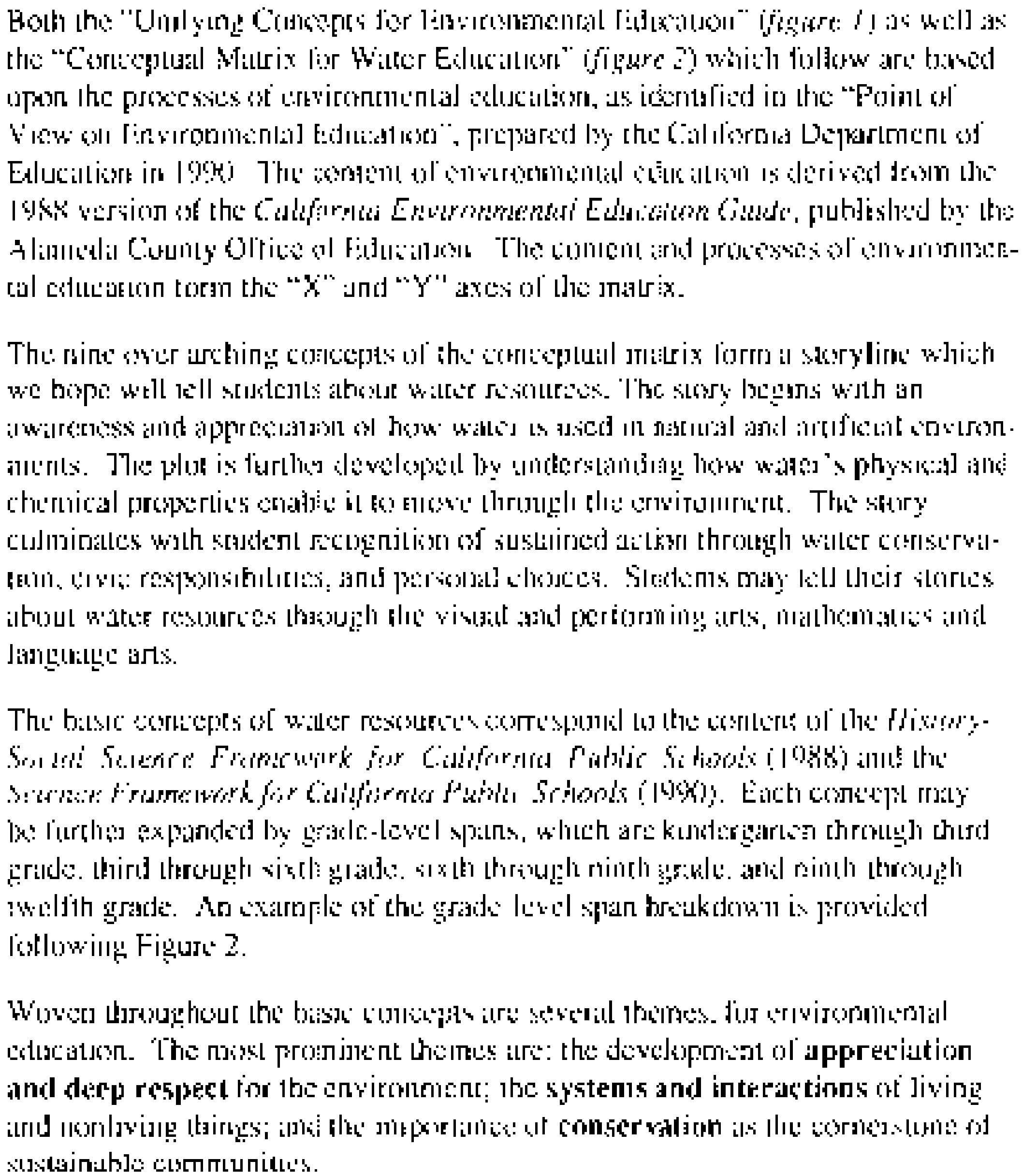
- Air Quality - During construction, contractors will be required to sprinkle unpaved construction areas with water at least twice per day to reduce particulate emissions by approximately 50 percent. Also, as required, applications for an Authority to Construct and Permit to Operate, along with other appropriate permits for operation of stationary and mobile pollutant generating machinery and equipment will be submitted to or appropriated from the South Coast Air Quality Management District.

- Plant Life - A comprehensive revegetation plan will be prepared and incorporated as part of the project. A compliance monitoring program will be implemented to ensure success of the revegetation effort and adherence to specified permit requirements and stipulations developed during the environmental review process.

- Noise - Construction contracts will limit the use of heavy equipment and the transport of construction materials to between the hours of 8 a.m. and 5 p.m. Noise barriers placed around stationary equipment to reduce noise levels at nearby subdivision areas. Construction contracts and DWR inspectors will ensure that OSHA safety regulations are observed.

- Transportation/Circulation - Double truck/trailers will be used, when practical, to reduce the amount of trips required to transport construction materials to the project site. Also, when practical, Palm Avenue will be used to bring in heavy equipment and required materials.

Figure 1.2: Touching characters in words caused by image degradation.



Both the "Unifying Concepts for Environmental Education" (*figure 1*) as well as the "Conceptual Matrix for Water Education" (*figure 2*) which follow are based upon the processes of environmental education, as identified in the "Point of View on Environmental Education", prepared by the California Department of Education in 1990. The content of environmental education is derived from the 1988 version of the *California Environmental Education Guide*, published by the Alameda County Office of Education. The content and processes of environmental education form the "X" and "Y" axes of the matrix.

The nine over arching concepts of the conceptual matrix form a **storyline** which we hope will tell students about water resources. The story begins with an awareness and appreciation of how water is used in natural and artificial environments. The plot is further developed by understanding how water's physical and chemical properties enable it to move through the environment. The story culminates with student recognition of sustained action through water conservation, civic responsibilities, and personal choices. Students may tell their stories about water resources through the visual and performing arts, mathematics and language arts.

The basic concepts of water resources correspond to the content of the *History-Social Science Framework for California Public Schools* (1988) and the *Science Framework for California Public Schools* (1990). Each concept may be further expanded by grade-level spans, which are kindergarten through third grade, third through sixth grade, sixth through ninth grade, and ninth through twelfth grade. An example of the grade-level span breakdown is provided following Figure 2.

Woven throughout the basic concepts are several themes for environmental education. The most prominent themes are: the development of **appreciation and deep respect** for the environment; the **systems and interactions** of living and nonliving things; and the importance of **conservation** as the cornerstone of sustainable communities.

Figure 1.3: Characters fragmentation caused by image degradation.

There have been extensive research efforts over the past three decades in seeking solutions to these two problems. The recent years have also witnessed the emergence of a growing number of commercial recognition systems that deliver limited solutions to these two problems to a certain extent. Nevertheless, breakthroughs in the general solution to these two problems have been few [27, 28, 142, 39, 3, 116, 117, 118]. Most of the past efforts were focused on exploring alternative solutions to character segmentation, trying to improve segmentation results in some special application areas or under specific conditions. The majority of today's commercial systems are still restricted to a few selected fonts, and their performance usually degenerates quickly on degraded documents.

While it has been shown that these two problems are extremely difficult to solve, theoretical as well as commercial incentives for their solutions remain high.

When put into a broader perspective, the difficulties with character segmentation and font training in document recognition are not isolated nor accidental. They often reflect other problems familiar to researchers in a few related branches of science. For decades *figure-ground segregation*(or image segmentation) has been an elusive problem that perplexed researchers in the fields of computer vision, cognitive science and psychology[107, 9, 145, 85, 35]. This difficulty in separating what is meaningful to recognition from what is irrelevant background noise lies at the core of the computational model that forms the base of today's computer vision systems [109]. Character segmentation is just a reflection of this fundamental difficulty in the task of document recognition. With real-world documents that contain many touching characters and character fragmentation, targeting the right image area that corresponds to a single character rather than including fragments of its touching neighbors is a demanding task. This requires a good understanding of both image representation and pattern recognition.

Similarly, the problem of font training in document recognition parallels the issue of information representation in computer vision and human vision. Visual information representation in computer vision has long been troubled by the fundamental difficulty of having a generic representation for a certain object that is concise so that it is computationally feasible, while at the same time is general and sufficient enough to capture the rich varieties of visual patterns of the object in the real world [107, 110]. This is an issue of concern not only for designing general theoretical frameworks in computer vision, but also for developing commercial computer vision systems for specific task domains. The biological systems for visual information processing that exist in our human brain are the only proof that such an ideal and working representation of visual information does exist, but finding the mechanism and formalizing it mathematically remains a challenging issue.

In light of the fact that character segmentation and font training are the two major problems that impair the performance of today's document recognition systems, and general solutions to these two problems are difficult to find, a potential alternative to the solution of degraded document recognition points to approaches that can achieve recognition without the reliance on character segmentation or font training. That is, instead of trying to find solutions to these two problems directly, the potential for solution to document recognition may lie in approaches that can *avoid* these two problems. This is the direction that the research presented in this thesis takes. The goal of this research is to seek new approaches to degraded document recognition by using non-traditional deciphering algorithms that recognize characters or words by solving the equivalent of a substitution cipher.

As will be explained in more detail later in the thesis, the adoption of an deciphering approach is based on renewed understandings of the computational task of document recognition and the visual and linguistic constraints involved. We show that the deciphering approach can provide well-founded solutions to the difficult problems in document recognition.

1.2 Difficulty in Character Segmentation

As was pointed out early in this chapter, character segmentation under image degradation has being one of the major problems affecting the performance of today's document recognition systems. In this section, we take a closer look at the problem of character segmentation. We show why character segmentation is important to today's document recognition systems, and why solution to this problem has been difficult to find.

A conventional system for word recognition has an architecture similar to that shown in Figure 1.4. It usually involves the *sequential* stages of character segmentation, character recognition, word recognition and, in some cases, contextual verification, whose results are sometimes feed back to the previous stages to generate new candidates that better fit the linguistic context.

One problem with this conventional approach to word recognition is that it involves the unavoidable stage of character segmentation. The basic idea of character segmentation is to dissect the image patterns of words into individual characters so that they can be fed to character recognizer for individual recognition. In the simplest case, one would expect that text is printed with a clear gap between adjacent characters, and therefore character breaks will be indicated by uninterrupted vertical white space, as shown in Figure 1.5. In reality, however, character breaks are frequently distorted by the compound effect of fonts, printing quality, nonuniform contrast, binarization, and image artifacts(Figure 1.6). These degradations pose serious difficulties to character segmentation. For example, in the case of low contrast or insufficient ink, characters tend to be fragmented, usually resulting in *over-segmentation* in which segmentation results include fragments of characters(Figure 1.7). While in the case of ink smudging, font serifs and ligatures, many adjacent characters may be touching each other, resulting in character under-segmentation(Figure 1.8).

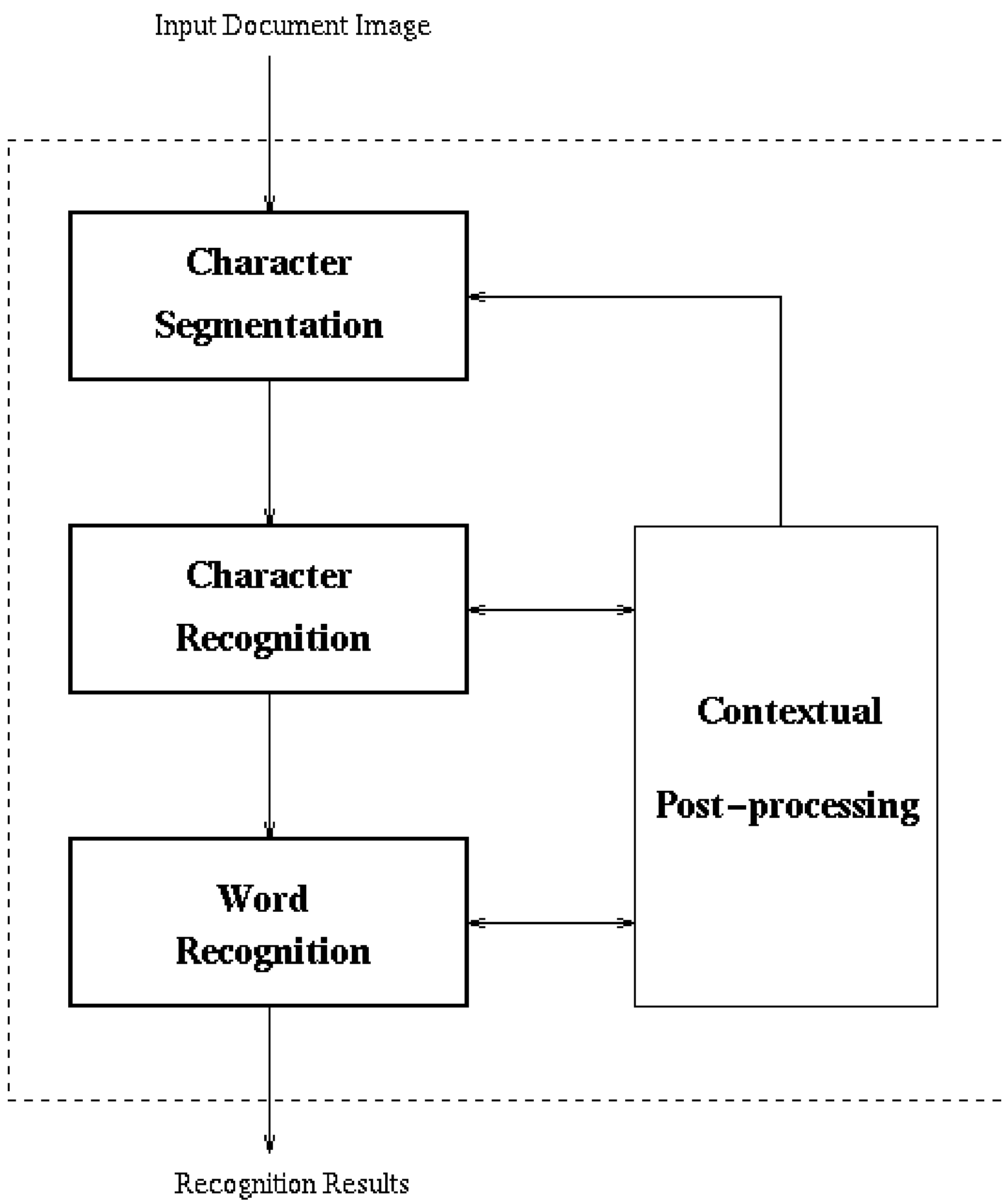


Figure 1.4: Conventional system for document recognition.

A character segmenter that that fails to show good performance in presence of image degradation usually degrades the performance of a segmentation-based system.

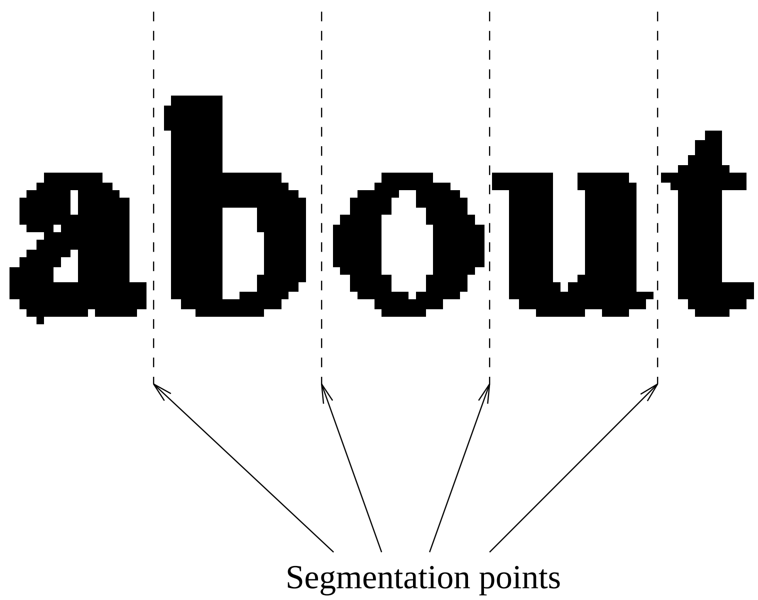


Figure 1.5: Ideal word image for character segmentation.

A combination of techniques has been used to handle character segmentation under image degradation. Early methods for character segmentation were based mainly on projection profile analysis and pitch estimation, which work fine on typewriter fonts that have equal spacing between characters. Connected component analysis was used to deal with font serifs, ligatures and special decorative type-fonts. More recent developments stress the integration of segmentation with recognition and using recognition results and linguistic context as feedback to improve segmentation decisions made earlier.

Despite many satisfactory results reported for constrained domains, character segmentation in the presence of image degradation still remains one of the most difficult problems in document recognition and the weakest link in a practical OCR system [30, 141, 113, 22].

The proposed Devil Canyon Second Afterbay Lining Project is to be located at the southern base of the San Bernardino Mountains near the mouth of Devil Canyon in western San Bernardino County. The proposed project is part of a previously authorized action by the Department of Water Resources dealing with enlargement of the East Branch of the California Aqueduct. This enlargement project involves, among other things, the construction of the Devil Canyon Second Afterbay – an 800-acre-foot reservoir associated with the Devil Canyon Powerplant complex.

The Second Afterbay will allow for variable flows through the powerplant, thereby enabling it to generate more pollution-free power during peak electric demand periods each day. This, in turn, will provide increased power sale revenues to the State Water Project and reduce the costs charged by the Department to water agencies that receive water passing through the powerplant.

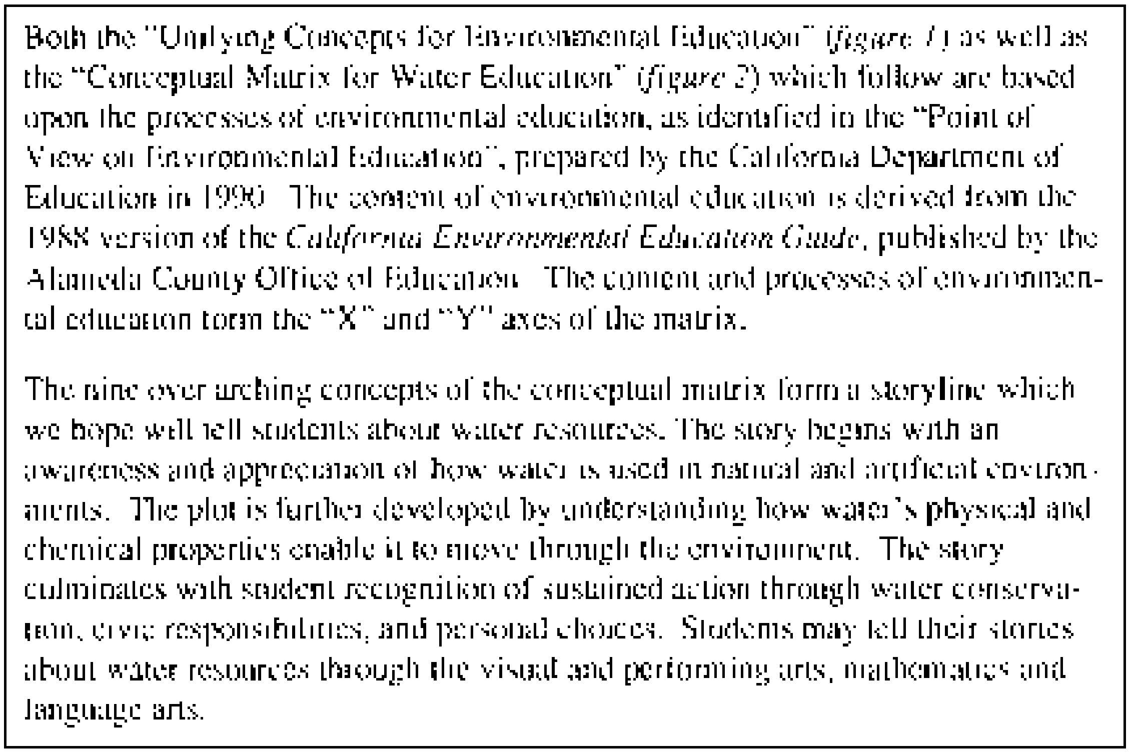
of unimpaired runoff at four stations in the Sacramento Valley. The sum of the forecast runoff at the Sacramento River above Bend Bridge near Red Bluff, Feather River inflow to Lake Oroville, Yuba River at Smartville, and American River inflow to Folsom Reservoir, is the Four Basin Index (Figure 2). The Four Basin Index also designates normal and subnormal snowmelt runoff years. These forecasts are published by DWR in the Bulletin 120 series, "Water Conditions in California," on February 1, March 1, April 1, and May 1 of each year. A fall data summary is also published each October. Delta water quality standards are predicated on the year's classification of the Four Basin Index. Wet years dictate more stringent water quality standards, whereas drier years allow relaxation of the standards.

The 1983 water year was forecast as wet with a final predicted unimpaired runoff of 38.3 million acre-feet (MAF) in May 1983. The actual runoff was 37.7 MAF. The April to July snowmelt runoff was predicted to be 14.2 MAF, while the actual runoff was slightly less at 13.7 MAF (Table 1).

This report will describe the importance of striped bass, Morone saxatilis, that depend upon the environment of the Sacramento-San Joaquin River Estuary, changes in their population that have taken place in the last 25 years, and what is known of the causes of these changes. We will emphasize the major decline in the population that became most evident after 1976.

We describe the adult population and the factors that affect it; followed by information on juveniles; and finally, a summary of what we believe has happened to the stock.

Figure 1.6: Image degradations.



Both the "Unifying Concepts for Environmental Education" (*figure 1*) as well as the "Conceptual Matrix for Water Education" (*figure 2*) which follow are based upon the processes of environmental education, as identified in the "Point of View on Environmental Education", prepared by the California Department of Education in 1990. The content of environmental education is derived from the 1988 version of the *California Environmental Education Guide*, published by the Alameda County Office of Education. The content and processes of environmental education form the "X" and "Y" axes of the matrix.

The nine over arching concepts of the conceptual matrix form a storyline which we hope will tell students about water resources. The story begins with an awareness and appreciation of how water is used in natural and artificial environments. The plot is further developed by understanding how water's physical and chemical properties enable it to move through the environment. The story culminates with student recognition of sustained action through water conservation, civic responsibilities, and personal choices. Students may tell their stories about water resources through the visual and performing arts, mathematics and language arts.

Figure 1.7: Text that usually results in character over-segmentation and an example of character over-segmentation.

The proposed Devil Canyon Second Afterbay Lining Project is to be located at the southern base of the San Bernardino Mountains near the mouth of Devil Canyon in western San Bernardino County. The proposed project is part of a previously authorized action by the Department of Water Resources dealing with enlargement of the East Branch of the California Aqueduct. This enlargement project involves, among other things, the construction of the Devil Canyon Second Afterbay – an 800-acre-foot reservoir associated with the Devil Canyon Powerplant complex.

The Second Afterbay will allow for variable flows through the powerplant, thereby enabling it to generate more pollution-free power during peak electric demand periods each day. This, in turn, will provide increased power sale revenues to the State Water Project and reduce the costs charged by the Department to water agencies that receive water passing through the powerplant.

Figure 1.8: Text that usually results in character under-segmentation and an example of character under-segmentation.

1.3 Contextual Linguistic Information in Document Recognition

Document recognition has been considered a sub-area in the broader field of computer vision. But unlike most of the tasks in vision such as scene recognition or facial recognition, document recognition has one unique feature. In addition to the fundamental visual processing that are common to most vision tasks, document recognition has the added dimension of linguistic processing. The precise roles of linguistic knowledge and linguistic constraints in the task of performing recognition of documents are still to be found, but their uniqueness and importance are evident.

Over the past thirty years, the field of document recognition has changed from considering itself mainly a visual classification task to its recent formulation that stresses more heavily on the use linguistic constraints in addition to visual recognition. The fact that OCR(Optical Character Recognition) was used in the 60s and 70s to refer to this field reflected the limited understanding of the important role of linguistic processing at that time. It had long been accepted that character recognition could be achieved by visual analysis of the character image patterns using techniques developed in pattern recognition. Word recognition and thus document recognition were simply the natural outcome after every character was individually recognized and their results combined.

This view that document recognition was essentially a *sequential* task that involved mainly visual analysis began to be challenged as more and more fundamental problems in this field became realized. One of the problems is about the relationship between character segmentation and character recognition. It is like a "chicken-and-egg" dilemma in that on the one hand, character recognition cannot be achieved without first segmenting the characters; while on the other hand, because of image degradation, correct decisions on character segmentation are usually hard to make without character or word recognition results becoming available to verify the many segmentation hypotheses. This is especially true with degraded documents that include extensive touching characters and character fragmentation (Figure 1.9). Another problem that challenged the view of considering document recognition as mostly a visual processing task is the difficulty in dealing with the similar and competing character candidates produced by the visual classification. By depending on visual pattern classification alone, it is often extremely difficult to make the right decision in accepting one candidate while rejecting the rest for a character. The existence in English of many visually similar characters('c' and 'o,' 'o' and 'e,' 'h' and 'b,' 't' and 'r,' and 'o' and '0' and '1' and 'l'), as well as the existence of characters that replicate exactly parts of other characters('n' vs. 'm,' 'c' vs. 'd' and 'l' vs. 'd') aggravate this problem, so does the introduction of image degradation in documents. Problems such as these question the claim that document recognition can be achieved by essentially sequential visual processing.

Early applications of linguistic processing in OCR were mainly to help decide which character candidates from visual classification make better sense in a linguistic context. Character n-gram statistics or a dictionary were usually used to model the linguistic constraints. Later developments along this line included using linguistic constraints at the word and sentence levels such as word n-gram statistics, word collocation statistics, and the transition probabilities of word categories [162, 160, 43]. New statistical tools and language models such as the Hidden Markov Model and grammatical parsers that were developed and achieved successes in speech recognition and natural language understanding were also introduced into OCR [133, 135]. More recent developments include incorporating linguistic processing within the recognition process itself, not only to post-process results produced by visual character classification, but also to generate new character or word candidates [152]. Using linguistic constraints and incorporating them into visual recognition has helped to disambiguate similar candidates and achieved better segmentation and recognition results, and has therefore

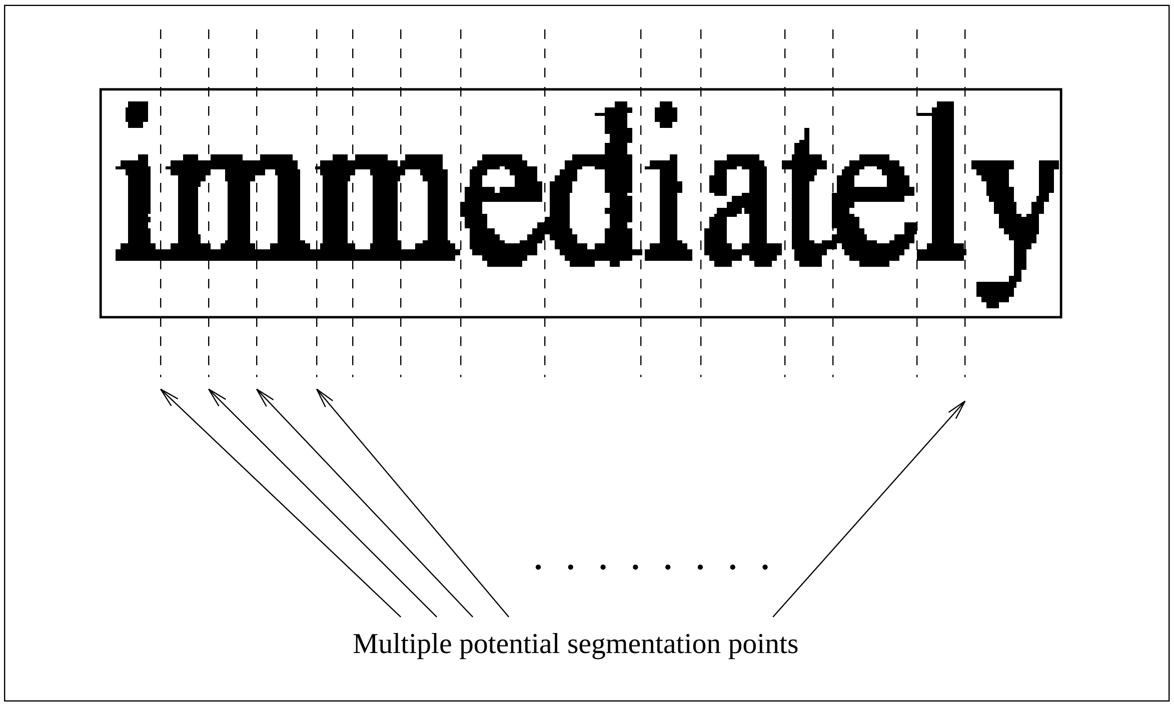


Figure 1.9: The inter-dependency of character segmentation and recognition.

greatly increased the performance of document recognition systems when applied to degraded documents.

The improved understanding of the importance of linguistic processing in document recognition has also paralleled the recognition of the importance of using contextual information in many other aspects of computer vision and image understanding [11, 9, 158, 159]. Research in using contextual information in pattern recognition had resulted in new classification frameworks that incorporated different levels of contextual constraints inside the discrimination function for high classification accuracy and more meaningful interpretation of results [52, 164]. Psychological studies in human reading have also shown the critical role of linguistic context in human reading, especially when visual information is impaired [130, 82, 36, 146, 150].

Developments in the field of OCR, computer vision, patterns recognition, as well as human reading have shown that the use of contextual information is more than an ad-hoc remedy to deficiencies in visual recognition. Rather, we believe the need for contextual information and linguistic processing is regulated by the nature of the computational task of document recognition itself. Visual information and classification alone are not enough to achieve reliable recognition of documents. It is even less so when the critical visual information is impaired under image degradation. A machine system that does not use context has a fundamental limitation. Even the view of considering linguistic processing as a subsidiary addition to the dominating visual classification is not enough to solve the challenging problems encountered in the real world. Potential solution to the difficult problems in document recognition relies on further understanding of the computational task of document recognition itself. It also relies on a better understanding of the roles visual and linguistic information play in document recognition and how they interact. They are the source from which new and more powerful algorithms and implementations can originate. Seeking these new understandings as well as the associated algorithms and implementations to solve the difficult problem of degraded document recognition are the focus of the research presented in this thesis.

1.4 The Potential for New Approaches

As pointed out earlier, the aim of this research is to seek approaches that are able to recognize degraded documents without the need for character segmentation or font training. Although the field of document recognition has seen increasing use of linguistic constraints for post-processing and contextual verification, the potential of designing new recognition approaches that combine visual and linguistic constraints to achieve font-independent and segmentation-free recognition have been overlooked. The potential for new approaches is based on the following observations.

First of all, for an extended passage of text, there exists a correspondence between the distribution of visual character(or word) patterns across a document page and the distribution of characters(or words) in a matching language base. This correspondence exists in individual characters and individual words. The simplest case is that with very high probability the most frequently appearing visual patterns of characters in an extended flow of text will correspond to the letter "e", and the most frequently appearing visual pattern of words will correspond to the word "the". The correspondence between visual patterns and their linguistics identities also exist in distributions of multiple characters and words, in the forms of character n-grams and word n-grams, or character and word transition probabilities. Figure 1.10 shows some examples of correspondence in character and word n-grams statistics.

Because this correspondence between the distribution of visual patterns in an input document page and the distribution of their linguistic identities in a matching language base is *independent* of font and in fact varies little from document to document, it is possible to design a deciphering-like algorithm that is able to decrypt this correspondence, and therefore achieve document recognition without the need for font training. We believe that rather than depending on font training that tries to respond to font variations, designing algorithms that are able to uncover the stable correspondence between visual patterns and their linguistic identities is a better solution to the problem. This is an important step forward in designing robust document recognition systems that achieve recognition by the decryption of the underlying invariant correspondence which is immune to font variations and image degradation.

Secondly, since the correspondence between visual patterns and their linguistic identities shows up at different levels of linguistic granularities, it is possible that deciphering-like algorithms can be designed accordingly so that they work at different levels of linguistic granularities. For example, a deciphering algorithm can work at the character level to decrypt individual characters by using character-level language constraints such as a lexicon or character n-gram statistics. A deciphering approach can also work above the character level to directly decode words by relying on word-level linguistic constraints. This ability to directly recognize words instead of building on character recognition results leads to the possibility of having a document recognition system that is free of character segmentation, and therefore avoids altogether the difficult problem of segmenting touching characters and character fragmentation in degraded documents.

Visual Patterns	**Relative Frequency of Visual Patterns**		**Relative Frequency in Language Database**	
th	**"th"**	0.05	th	0.0495
he	**"he"**	0.024	he	0.0237
in the	**"in the"**	0.043	in the	0.0435
of the	**"of the"**	0.032	of the	0.0328
trade war	**"trade war"**	0.015	trade war	0.0147

Figure 1.10: The correspondence between the distribution of visual patterns in a document and the distribution of linguistic identities from a language database.

1.5 Document Recognition as a Computational Task

The field of document recognition has been closely associated with studies in computational vision. The two major difficult problems of character segmentation and font training, for which we seek to find solutions in this thesis, also find close counterparts in computational vision. In his formulation of a computational approach to vision, Marr [107] proposed a three-level methodology for the understanding of any machine system that carries out an information processing task. As shown in Figure 1.11, when trying to understand a computational task, it is important to seek answers separately at different levels of abstraction. On the one extreme is the *computational theory level*, in which the performance of the system is characterized as a mapping from one kind of information to another. The most important question on this level is *what* does the system compute. The next level of *representation and algorithm* is about the choice of representation for the input and output and the algorithm to be used to transform one into the other. The important issue of concern at this level is *how* does the system compute the mapping. At the other extreme is the *hardware implementation level* which is concerned with *how* the algorithm and representation are implemented physically.

Computational theory is of critical importance because an algorithm is likely to be understood more readily by understanding the nature of the problem being solved than by examining the mechanism(and the hardware) in which it is embodied. Take for example the understanding of the behavior of flying. It doesn't matter very much whether the flying is associated with a modern airplane or with a bird, the principal issue in understanding flying is the issue of aerodynamics, rather than the structure and organization of feathers or the components in an airplane. The differences in hardware implementation details make sense only when the computational issue of aerodynamics is thoroughly understood. To build a competent and working system, it is equally important to seek separate explanations of a

Computational Theory

What is the goal of the computation

Why is it appropriate

What is the logic of the strategy by which it can be carried out

Representation and Algorithm

How can this computational theory be implemented

What is the representation for the input and the output

What is the algorithm for the transformation

Hardware Implementation

How can the representation and algorithm be realized physically

Figure 1.11: The three levels at which any machine carrying out an information processing task must be understood(from [Marr82]).

computational task at different levels that can be linked together into a cohesive whole.

In the case of document recognition, it is important to break down the whole issue of document recognition and ask the right questions at the appropriate levels, so that we can base our research on the established achievements in this field and keep from wasting time and resources on repetitive efforts that may have already been investigated under different umbrellas. For example, whether to adopt a deciphering approach or a traditional segmentation-and-recognition approach to document recognition is a difference at the representation and algorithm level. Whether to use a probabilistic relaxation model, a Hidden Markov Model, or even a simulated annealing model to implement the deciphering algorithm is an issue of algorithmic implementation and thus still at the algorithm level. But the problem of whether to use an artificial neural network to implement simulated annealing is an issue at the hardware and implementation level. Although each of these issues is important, it needs to be treated appropriately with due significance within a more global perspective in the computational framework of document recognition.

Our interest in this thesis is to seek an alternative answer to the question of *how* the mapping between visual patterns and their linguistic identities is computed in recognizing a document. And for this we propose the deciphering approaches. It needs to be pointed out that it is not our intention in this research to pursue the computational framework of document recognition that perhaps human beings use in reading or document recognition.

1.6 Objectives of This Research

The primary goal of this research is to design new approaches that can recognize documents with reliable performance. The approaches should be able to recognize real degraded documents with high accuracy, and should be able to accomplish stable and reliable recognition without the reliance on font training or character segmentation. In addition to this primary goal, this research also intends to seek answers to the following related questions.

1. How do the proposed deciphering approaches contribute to the understanding of the computational task of document recognition at the computational theory level as well as at the representation and algorithm level?

2. What are the roles of visual and linguistic constraints in document recognition? How can linguistic constraints at different levels of linguistic granularities be used to compensate for the deficiency in visual information in case of image degradation? And what are the algorithmic frameworks in which visual and linguistic information can be optimally combined and used in document recognition?

3. How the deciphering approaches work at different levels of linguistic granularities? Is character-level deciphering still applicable to documents that include extensive touching characters where the correspondence between visual patterns and their linguistic identities is impaired? How far can the deciphering approaches go and what are their implications on building recognition approaches that are font-independent and segmentation-free?

4. How do the proposed deciphering approaches relate to approaches that have been applied before to document recognition such as *constraint satisfaction*, *combinatorial optimization* and *hypothesis testing*? How can different algorithms such as heuristic

search, probabilistic relaxation and simulated annealing be used to implement deciphering, and what are their effects on the performance of the system?

5. How does the system perform on real scanned documents with image degradations? How can system performance be further improved?

1.7 Outline of the Dissertation

In the remaining chapters of this dissertation, we will first discuss background information concerning past research in several important areas that are related to our research. We will then present the deciphering approaches, followed by detailed presentations and discussions of each of the proposed approach. Experimental results of applying the proposed deciphering approaches to scanned documents will be presented and discussed. Finally, observations and conclusions will be drawn and further improvements will be pointed out.

Document recognition is a highly inter-interdisciplinary research field that is related closely to other research fields such as vision, linguistics, pattern recognition, digital image processing, natural language understanding, psychology, cognitive sciences, as well as mathematics and statistics. Chapter 2 reviews the important research in some of these areas that are related to the topics in the thesis. Special attention will be given to three topics. The first is about the numerous research efforts in the past on character segmentation algorithms and the extreme difficulty of this problem under image degradation. The second topic is about past research efforts in designing document recognition approaches that are intended to be segmentation-free. And the third topic is about experimentation on the application of character-level deciphering algorithms to OCR.

Chapter 3 presents our proposed approaches to degraded document recognition that use deciphering algorithms to uncover the mapping between visual patterns and their linguistic identities. Outlines of two deciphering models at different levels of linguistic granularities are presented. The first one is a modified character level deciphering algorithm that has the added ability to handle extensive touching characters and tolerates clustering errors. The second one is a novel approach that applies the deciphering concept at the word level to decrypt words directly using word-level linguistic constraints.

The following three chapters are devoted to detailed descriptions of the algorithmic implementations of the proposed approaches presented in Chapter 3.

Chapter 4 discusses modifications to a previous character-level deciphering algorithm for document recognition that are able to handle touching characters and are also tolerant to mistakes made at the clustering stage.

Chapter 5 describes a word-level deciphering approach that solves a selected portion of the input text by using a word-level probabilistic relaxation algorithm. Font information is then learned from the solved portion of the text and used to re-recognize the rest of the text.

One important characteristic of the proposed word-level deciphering algorithm is its ability to learn font information of the input document dynamically through the decryption of part of the input text and use the learned font information to recognize the rest of the document unavailable to the cipher solution. Chapter 6 describes a hypothesis testing algorithm that achieves reliable and efficient word recognition using the learned font information.

Finally, Chapter 7 summarizes the approaches presented in this thesis. It also discusses the results and implications of this research and points out some potential directions in which the deciphering approaches can be improved.

Chapter 2

Background

In this chapter, important research and developments in the past that were related to our proposed approaches are reviewed. We will be focusing on research in three areas. They are character segmentation in traditional document recognition systems, segmentation-free approaches to document recognition, and deciphering algorithms for OCR.

Character segmentation has been an important and difficult problem in document recognition for decades. It affects to a great extent the overall performance of document recognition systems that are based on character segmentation and character classification. Section 1 reviews the important developments in character segmentation from the voluminous research published in this field. The advantages as well as shortcomings of the various approaches are pointed out. The intention here is to show the critical importance of character segmentation in conventional document recognition systems and the difficulty in finding a general solution. The group of approaches to document recognition that seek alternatives to the conventional method of character segmentation and character classification form the general category of segmentation-free methods. Developments in segmentation-free approaches to document recognition are reviewed in Section 2. Section 3 is dedicated to the review of a special group of methods that recognize characters by using deciphering algorithms that solve the equivalent of a substitution cipher. Special attention is given to character level deciphering algorithms that use heuristic search, probabilistic relaxation, and simulated annealing algorithms to implement the deciphering. They are the bases on which we developed our own approaches. Finally, Section 4 summarizes this review and draws some conclusions.

2.1 Character Segmentation

Character segmentation is a process that seeks to decompose an image of a sequence of characters into sub-images of individual symbols. Its decision, that a pattern segmented from the image is that of a character, can be right or wrong. Wrong character segmentation decisions often make a major contribution to the error rates of document recognition systems.

Character segmentation and character classification are the two major processing stages in traditional document recognition systems. The more classical view was that they were essentially two sequential steps. Character segmentation is first done independent of character classification, and results of character segmentation are then used by the character classification stage to produce the final recognition output.

In reality, however, character segmentation and character classification show strong dependency. On a more local scale, good character segmentation decisions require knowledge on what constitutes a character pattern, which is basically a classification problem based on pattern similarities. On a more global scale, a good character segmentation decision is one that satisfies contextual constraints within the word, or even constraints beyond the word on the sentence level. For example, an image pattern could equally be a 'm,' an "in," or a "ni," especially when image degradations are involved. But usually only one of these options will constitute a contextually acceptable result(Figure 2.1). The interaction between character segmentation and character classification is the most important aspect to consider when designing an approach to character segmentation.

Therefore, the review of character segmentation methods in this section will be based on how segmentation and classification interact in the overall process of recognition.

Figure 2.1: Character segmentation and contextual constraints.

2.1.1 Segmentation Independent of Classification

In the late 1950s and early 1960s, OCR was usually applied to constrained printing or writing with well-separated characters that facilitated segmentation. Examples were bank check fonts that were designed with strong leading edge features to indicate the accurate position of each character [44], as well as forms with printed boxes where hand-printed characters are supposed to be enclosed (Figure 2.2). Therefore, character segmentation was a relatively easy task and considered only a secondary issue in an OCR system. The following methods for character segmentation represent the state of the art at that time.

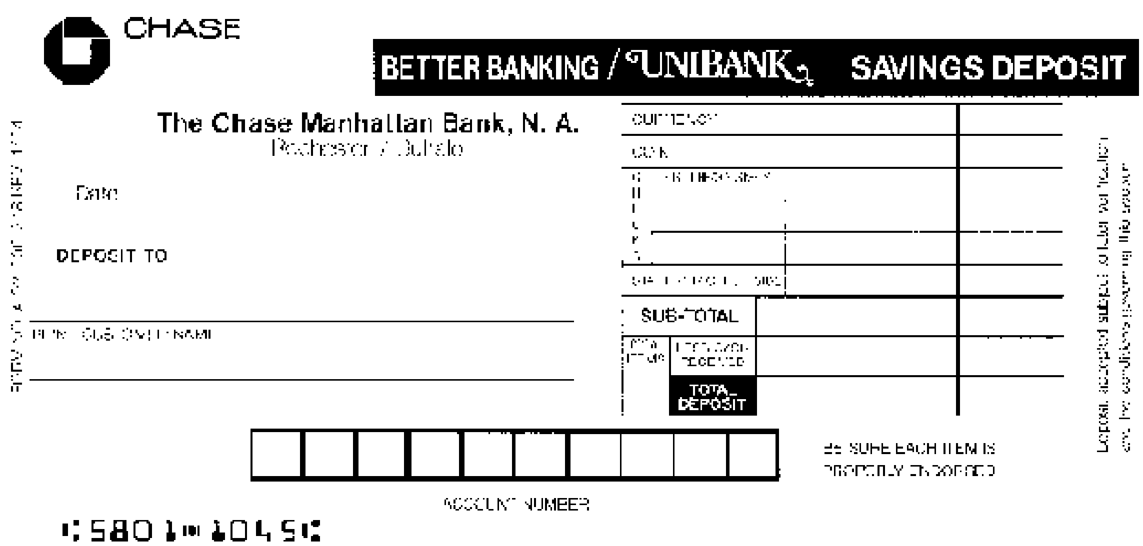


Figure 2.2: An example of bank checks and forms with specially designed font and boxes to help OCR.

2.1.1.1 White Space and Pitch Estimation

Vertical whitespace often serves to separate successive characters. The notion of detecting the vertical white space between successive characters has naturally been an important concept in dissecting images of characters.

In many machine printed applications involving limited font sets each character occupies a block of fixed width. The pitch, or number of characters per unit of horizontal distance, provides a basis for estimating segmentation points. Pitch estimation also permits correct segmentation in case of touching characters and character fragmentation.

Combining white space location and pitch estimation can usually produce reliable segmentation results on prints of well-spaced font with fixed width. Examples of commercial applications can be found in [67, 69].

2.1.1.2 Projection Analysis

The vertical projection of a word image consists of a simple running count of black pixels in each column. It can help detect white space between characters. It can also help to locate vertical strokes. For example, it was used in [119] to segment Kanji hand-printed address. Columns where the projection fell below a predefined threshold were considered candidates for segmentation points.

In the case of touching characters or text lines where neighboring characters overlap horizontally, the projection often contains a minimum at the proper segmentation points. Baird [6] proposed a method in which the ratio of the second derivative of the projection to its height was used as a criterion for choosing segmentation points. This criterion avoids considering points along a thin horizontal line as potential segmentation points. Lu's algorithm [103] improved on this method by using a peak-to-valley function that favors low valleys with high peaks on both sides.

Different pre-filtering methods were proposed by Tsujimoto and Liang [168, 102] that were intended to produce better projection for locating boundaries between touching characters. Moreover, the upper and lower profiles of the patterns were also used in combination with the projection to get improved segmentation point candidates [6].

Applications of projection analysis for character segmentation on italic fonts or on text that involves slanted characters can be found in [50], where the projections were implemented at two-degree increments between -16 and 16 degrees from the vertical. Segmentation cuts were also implemented along the projection angle.

2.1.1.3 Connected Components Analysis

Pitch-based methods usually cannot be applied when the width of characters is variable. Also, projection analysis has limitations when applied to text with slanted characters or with handwriting, where even non-touching characters may not be separable along a straight vertical line (Figure 2.3). These situations call for more sophisticated methods.

A common approach is connected components analysis that analyzes connected black regions. To account for touching characters and character fragmentation, further processing may be necessary to combine or split components into character images.

One method is based on analysis of bounding boxes which indicate the location and dimensions of each connected component. The distribution of bounding boxes on a text page carries information that can help the proper segmentation of non-cursive characters. By using information regarding the size, aspect ratios, and adjacency relationships of the bounding boxes to trigger the merging or splitting of connected components, this method can usually achieve accurate character segmentation. One example was given in [29] where knowledge of the symbols was also used to help segment handwritten post codes.

Experimental comparison of character segmentation by projection analysis vs. segment-

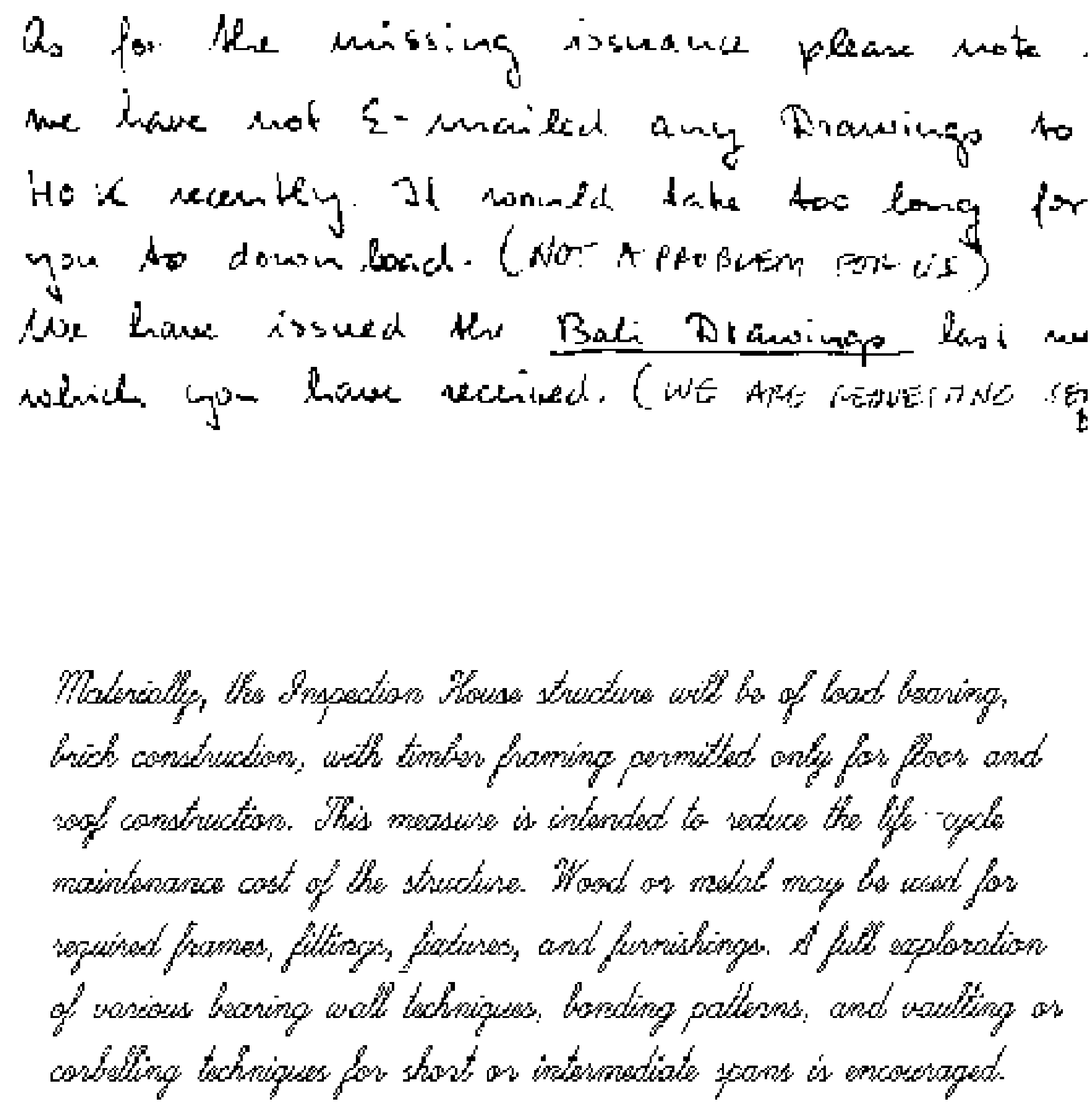


Figure 2.3: Examples of handwriting and slanted text.

ation by connected component analysis is reported in [173]. Both segmenters were tested on a large database of hand-printed digits(272870 in total). Segmentation by connected component analysis was shown to produced significantly better results over segmentation by projection analysis.

2.1.1.4 Dissection By Intersection Analysis

To achieve better segmentation results on text with touching characters, more detailed processing is necessary. The intersection of two connected characters can give rise to special image features. Consequently dissection methods have been developed to detect these special image features and then dissect the text lines into character sub-images. Such methods often work as a follow-on to bounding box analysis. Only image components that failed dimensional tests need to be subjected to detailed examination.

A further concern is that simple splitting along a straight-line usually won't be able to achieve good segmentation results with text that has overlapping or slanted characters. Accurate segmentation for these cases needs more detailed analysis and knowledge of the shape of the patterns to be split to determine the appropriate segmentation path.

These different points have been addressed by a number of authors. Parui [123] used contour analysis to help detect the likely segmentation points. Shridhar [153] proposed an algorithm for digit segmentation which not only detected likely segmentation points, but also computed the appropriate segmentation path. Lecolinet [99, 98] proposed a technique to detect connected components that have to be further segmented. The basic idea of this technique is based on the concept of pre-recognition in which connected component analysis is followed by a simple recognition step whose role is to determine whether the connected component is a single, connected, or broken character pattern. Pattern splitting is then carried out on connected components classified as touching characters by locating characteristic landmarks indicating likely segmentation points. Methods for finding the optimal segmentation path were also reported in [32, 55, 172].

2.1.1.5 Landmark Detection

Image features such as character ascenders and descenders usually can serve as landmarks for segmentation of word images. Methods of character segmentation based on detection of ascenders and descenders has been applied to both printed text as well as cursive writings. An example of applying landmark detection in character segmentation is given in [64].

2.1.2 Re-segmentation after Classification Using Context

The identities of characters or words in the text are subject to contextual constraints on different linguistic levels. These contextual constraints can be used to re-evaluate the segmentation decisions made earlier. Bozinovic [16] used a Markov model to represent splitting, merging as well as misclassification in the recognition process. The system seeks to correct the errors by minimizing an edit distance between recognition output and words in a given lexicon. Casey proposed a non-Markov system that uses a spell-checker to correct repeatedly made merge and split errors in a complete text, rather than in individual words [23].

The "grapheme" methods first segment an input image into "graphemes" which are sub-images that are not necessarily individual characters. A following recognition stage modifies the pre-segmentation decisions by using a mapping function that uses contextual constraints to combine or split the pre-segmented "graphemes." An example of applying grapheme method for the segmentation of off-line cursive script is reported in [148].

Most dissection techniques for cursive script are based on the fact that lowercase characters are usually linked by lower ligatures. A simple way to locate these ligatures is to detect the minima of the upper outline of words. Examples of techniques along this line are given in [42, 105, 70, 89, 47].

Sennhauser [152] proposed a blackboard-based system which integrated three contextual knowledge sources to provide feedback and generate hypotheses for recognition and re-segmentation. The three knowledge sources include a spelling checker, a general letter n-gram statistics model, and another specific n-gram model for the processing of numbers and digits. Integration of multiple contextual knowledge sources does not increase the implementation complexity since each knowledge source can be implemented as a module of its own which is then added to the blackboard-based recognition framework.

2.1.3 Recognition-based Segmentation

The principal of recognition-based segmentation is quite different from the segmentation approaches discussed above. No feature-based dissection algorithm is employed. The image is divided systematically into many tentative overlapping pieces, from which the optimal character segmentation is to be derived from the recognition results.

Early work on this method was described in [92, 25], in which the concept of a moving window of variable width was used to produce tentative segmentation candidates, which were then confirmed or rejected based on character recognition results. Both serial and parallel optimization schemes were used. The serial implementation recognizes words iteratively in a left-to-right fashion. The parallel implementation proceeds in a more global way. It generates a lattice of possible feature-to-letter combinations. The final decision is found by choosing an optimal path through the graph.

A method that combined dynamic programming and a neural network for the optimal selection of segmentation hypotheses was reported in [19]. In another research, a neural network model was used to implement the concept of "selective attention" in human attention and reading, and was applied to compute the optimal segmentation [83].

Character or word-level linguistic constraints are usually used in the recognition-segmentation stage. Contextual constraints could be represented in the form of a lexicon. They could also be represented statistically as letter or word transition probabilities, which are usually used in Hidden Markov Models(HMMs) to exploit contextual constraints. The basic idea of Hidden Markov Models is to model English text as a Markov process which allows transition probabilities to be assigned to various letter combinations or n-grams. The probabilities are usually calculated from the n-gram frequencies of a language base. HMMs can also be used to tackle the problem of character segmentation. An HMM can be trained to learn the letter segmentation behavior because contextual constraints, such as word and letter frequencies, and syntactic rules, can be supported by transition probabilities between letter states. Examples of using HMMs for the optimal segmentation and recognition of text were reported in [60, 31, 151, 12, 96].

Non-HMM methods such as probabilistic relaxation and hypothesis testing were also used in recognition-based segmentation. Hayes used a probabilistic relaxation method to read off-line handwritten words [65]. The model worked on a hierarchical description of words derived from a skeletal representation. Relaxation was performed on the stroke graph and on the letter graph where all the possible segmentations were kept. Letter n-gram statistics were used to rule out unlikely combinations. Examples of using a hypothesis testing and verification algorithm to recognized hand-printed or on-line cursive words were reported in [75, 75, 13, 121]

2.2 Word Recognition Without Character Segmentation

The extensiveness of research reported in designing competent character segmentation algorithms to deal with degraded documents also reflects the difficulty in achieving this goal. This has stimulated research that seeks alternative recognition approaches that recognize the words directly, eliminating the character segmentation stage altogether. These approaches are often referred to as segmentation-free methods.

Segmentation-free methods usually belong to two general categories. The first are the so called holistic word recognition algorithms which perform recognition directly on image patterns corresponding to words. The second category of methods still seek to recognize individual characters, but the characters are recognized without first having to segment out each character. These methods usually involve the detection and recognition of special sub-character features which are then registered with their locational information to derive a recognition decision on a character. A character is recognized when a set of neighboring features optimally alarms the existence of a character. For this reason, this category of methods are referred to in the rest of the thesis as *feature registration* methods.

2.2.1 Holistic Word Recognition

In holistic word recognition methods, recognition is based on the comparison of a collection of simple features extracted from the whole word against a lexicon of feature representations that represent the shapes of words from a lexicon. Two basic steps are usually involved. The first step performs feature extraction, and the second step performs comparison between the feature representation of the unknown word and those of potential words from the lexicon.

Recent developments in this area are concerned mainly with the way in which comparison between hypotheses and references are performed. Recent comparison techniques are more flexible and account the word pattern variabilities better. These techniques are generally based on *dynamic programming* algorithms with optimization criteria based either on distance measurements or on a probabilistic framework.

Dynamic programming was employed in [129] for check recognition. Words were represented by a list of features indicating the presence of ascenders, descenders, directional strokes and closed loops. Paquet [122] and Leroux [100] used a similar scheme, but with different features. Features used in Paquet's method were based on the notion of "guiding points," which are the intersection of letters and the median line of the word. Whereas features used in Leroux's method were derived from the contours of word image patterns.

Hidden Markov Models are also used in holistic word recognition [115, 61]. Angular representation is used in Nagy's system to represent the feature, while structural image primitives are used in Gilloux's system.

Hull [74] proposed a holistic approach to recognize machine-printed text. Here the method is used to recognize commonly occuring words in a document. Clustering is used to separate groups of matching words from the text. Conventional OCR and linguistic analysis can then be applied to recognize the groups. This approach, instead of treating all words as members of a lexicon, separates out of the text those words that belong to its lexicon.

Because word recognition decisions are solely based on comparison with prototype feature representations, holistic recognition approaches require the prototype feature representations to be general and robust that they withstand variations in fonts, sizes, as well as different image degradations. This often limits its use to specific application domains where the prototype lexicon can be statically defined, like bank check recognition. More general applications require the mechanism to dynamically generate the prototype feature representation to accommodate new word candidates as well as image variations.

2.2.2 Feature Registration Method

A different approach to segmentation-free recognition can be described as feature registration methods. They differ from the holistic recognition methods in that the actual recognition and representation of features are on the sub-character level. The recognition of a character is triggered by detecting a combination of sub-character features at the correct locations.

Chen [30] proposed an approach that was based on the concept of occluding object recognition in computer vision [34], in which objects are recognized and *then* segmented from the image. Characters are treated as touching or occluding objects that are subject to special constraints on their poses. To recognize a character, the algorithm searches the image for the maximal subset of character features that have a spatial relationship consistent with the character feature model. When such a subset of consistent features is found, the character is recognized. Each character is modeled by a set of geometric features, each geometric feature described by a set of attributes such as its vertical and horizontal positions and stroke length. Feature matching is implemented by a pre-organized indexing-and-voting mechanism. This indexing-and-voting mechanism is organized according to the geometric features of the reference characters so that an image feature of an unknown character can be rapidly indexed to its matching model features. The collection of votes is thresholded to determine the character with the highest probability at each image position. Confidence in each recognized character is based on the number and nature of the features found in the image. Figure 2.4 illustrates the basic concepts in this approach.

A similar approach was proposed by Rocha [141, 143] in which each subgraph of features that matches a previously defined character prototype is recognized anywhere in the word even if it corresponds to a broken character or to a character touching another one. The characters are detected in the order defined by the matching quality, so the most illegible characters can use robust parameters estimated in previous characters. A final check concentrates efforts on poorly matched features that might be illegible symbols. Each subgraph that is

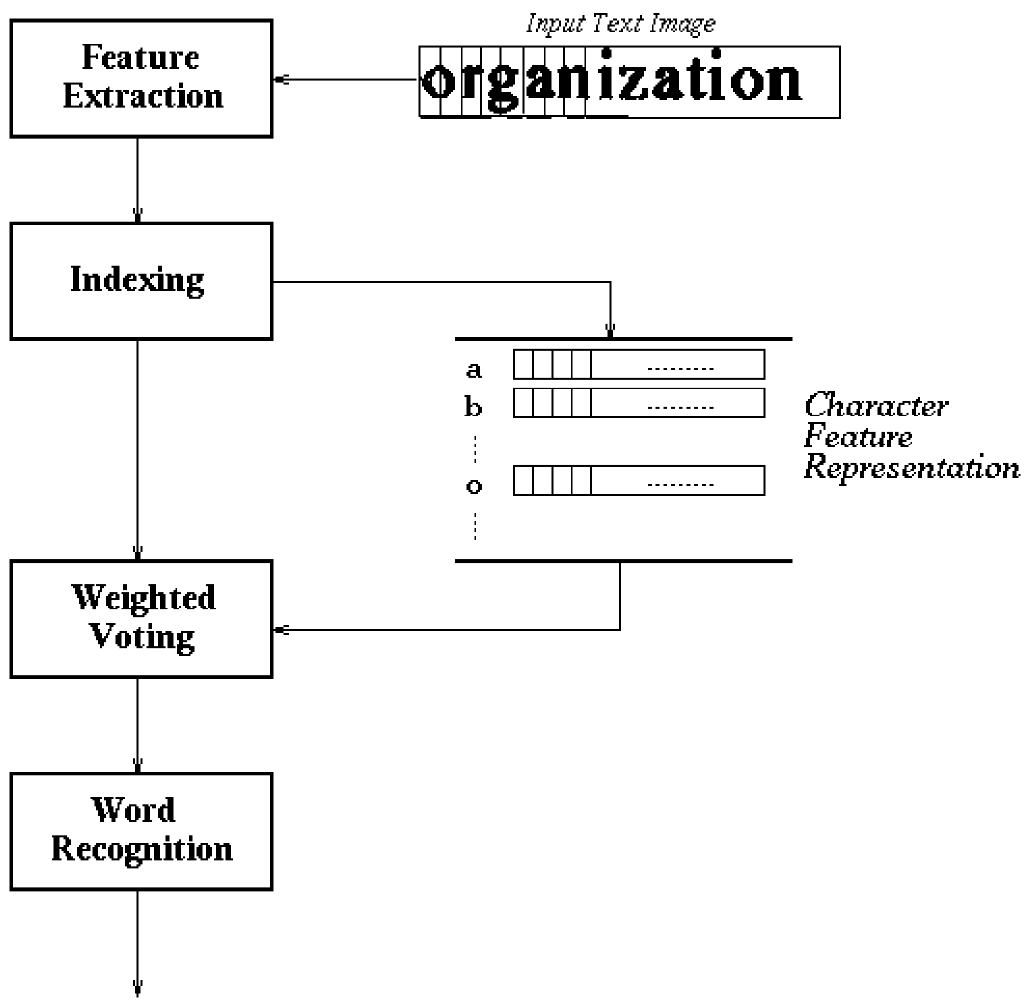


Figure 2.4: A segmentation-free approach(from [Chen92]).

recognized is introduced as a node in a directed net that compiles different alternatives of interpreting features in the feature graph. A path in the net represents a consistent succession of characters in the word. The method allows the recognition of characters that overlap, or that are underlined. A final search for the optimal path under certain criteria gives the best interpretation of the word features.

A different approach using letter prototype alignment was proposed for cursive handwriting recognition [41]. The basic idea of the alignment approach is to decompose the recognition process into two stages: First, the transformations between the viewed object and each one of the candidate models or prototypes are determined. This is the alignment stage. The alignment transformation may be computed from a small number of corresponding *anchor* features in the image and the model. In the second stage, the model that minimize the distance measure is found [170]. For character recognition, the anchor points are first extracted by tracing the image contour. Then strokes are detected and recognized by prototype alignment, using affine transformations computed from anchor point correspondence. Letter hypotheses are formed by detecting potential instances of each of the character prototypes. In addition to the true instances, several false ones are usually found. Finally, a best-first search is performed to compose admissible strings out of the set of all detected letter instances.

A segmentation-free classifier for machine-printed numeric fields that used a neural network approach can be found in [54]

2.3 Character Recognition as Substitution Cipher Decoding

Efforts to design autonomous reading machines motivated the early application of substitution cipher decoding to OCR [20, 21, 4]. The basic idea with this approach is to make use of language statistics and assign alphabetic labels to visual character patterns so that the visual character repetition pattern in the input text passage best matches the letter repetition pattern provided by a language model.

As shown in Figure 2.5, this method usually consists of two stages. First, a clustering algorithm classifies the visual character patterns into distinct groups by means of a distance measure. Each group ideally corresponds to a single letter in the alphabet. The passage of input text is thus considered a substitution cipher with the true alphabetic identity of each character pattern being substituted with a group identity assigned after the clustering stage. Secondly, a deciphering algorithm is applied to permute assignments of alphabetic identities to different groups until the repetition statistics of the character patterns best match that of the language model. Because character patterns in the input text passage are used only to form distinct pattern groups rather than for direct character recognition, and alphabetic identities are decided solely on the basis of maximizing the correspondence of linguistic statistics between the cipher text and the language model, this approach to OCR is font-independent and no font training is needed.

The many reported methods under this general category differ from each other in the specific kinds of linguistic statistics that are used as constraints and in the individual ways the deciphering algorithms are implemented.

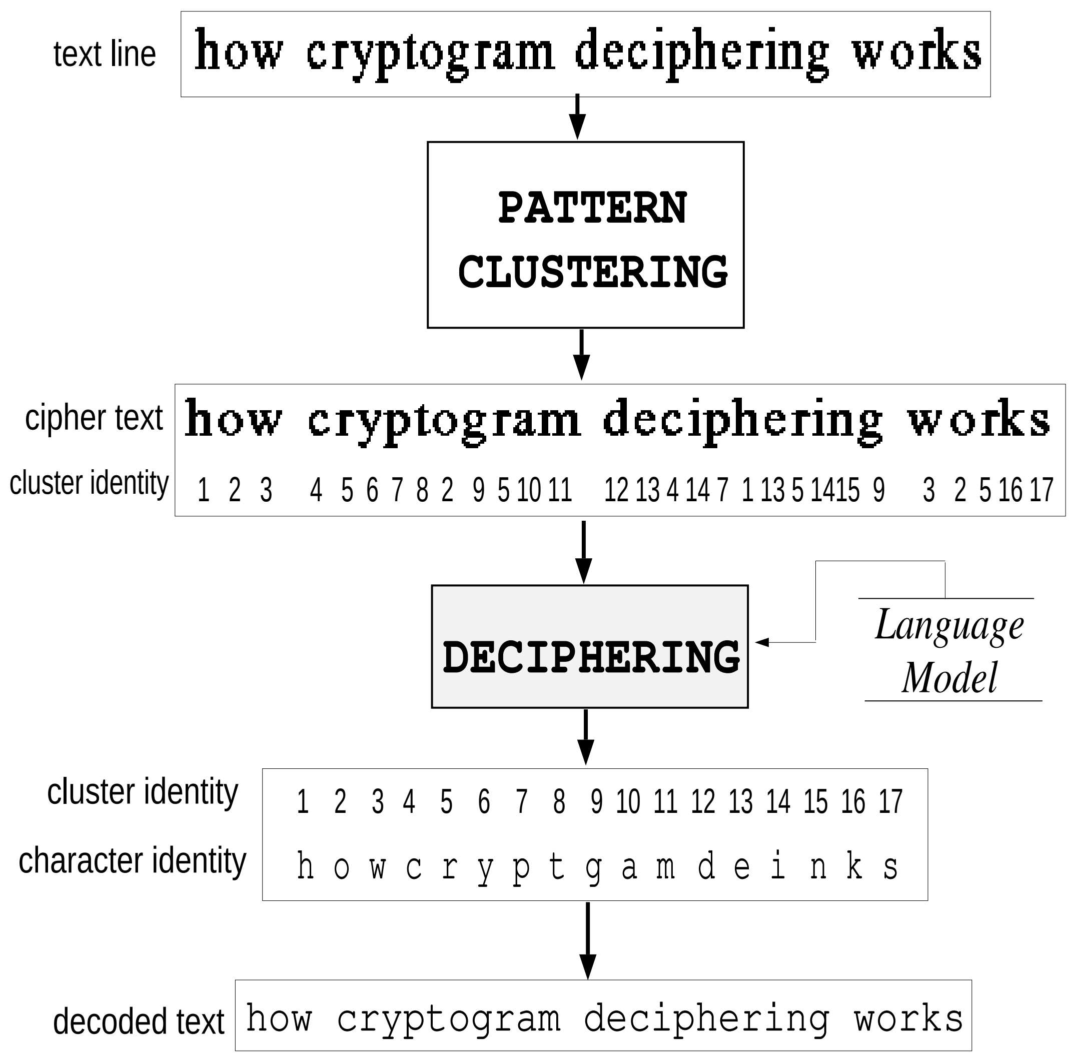


Figure 2.5: OCR as decoding a substitution cipher.

2.3.1 Character n-grams and Dictionary Lookup

One group of methods are based on using character n-gram statistics as constraints. Examples that use character bigram and trigram statistics are [157, 120]. Some other works also used a stationary Markov model, examples are [155, 57, 56, 95].

Two problems exist with using character n-gram statistics. First, a substantial amount of cipher text is needed for statistically reliable n-gram estimation. Second, a non one-to-one mapping from the set of character pattern clusters to the set of alphabetic letters resulting from bad clustering presents a serious difficulty for n-gram techniques.

A solution to these two problems is to use a dictionary instead of character n-grams as constraints. In this deciphering framework, breaking a substitution cipher corresponds to a particular mapping from the set of character pattern clusters to the alphabetic letter set such that all the cipher-words are legitimate words in the dictionary.

Among the published works that use dictionary lookup [23, 112, 114, 104, 134], Casey [23] presented an algorithm based on adaptive optimization of the assignment with the help of pattern matching provided by an on-line spelling checker. This algorithm showed some limited ability to handle two-character touching patterns, but it lacks the support for handling more extensive multi-character touching patterns. Lucks [104] proposed an automated algorithm that employs an exhaustive search in a large on-line dictionary for words that satisfy constraints on word length, letter position and letter repetition pattern. A similar method was presented by Ramesh et al. [134] that exploits letter repetition patterns in cipher words and uses a dictionary search that matches the repetition patterns. They also suggested the use of letter frequencies to further cut down candidates for cipher words after the dictionary search.

Nagy's work on decoding a substitution cipher using dictionary lookup represents the most systematic treatment of this problem [112, 114]. The concept of "*cover set*" was proposed to exploit the powerful constraints of character repetition patterns distributed across the text passage. Two search algorithms based on a depth-first search and a bread-first search are presented that use linguistic and problem-specific heuristics to constrain the search space.

The above works can be categorized as character-level deciphering because character-level language constraints are used and characters are directly identified by the algorithm. Word and document recognition decisions are built upon character recognition results. No attempt has been made to directly identify words within a deciphering framework by exploiting word-level linguistic constraints such as word unigram or word n-gram statistics that are abundant in a text passage. Almost all the methods use either character n-gram statistics or dictionary lookup. Little attempt has been made to combine the two sources of information together in the decryption process. Also, most of the algorithms use only linguistic constraints after the clustering stage. No attempt has been made to exploit the visual constraints of characters and combine them with language constraints in the deciphering process. Moreover, no method has attempted to address systematically the problem of touching characters inside a character deciphering framework and develop a deciphering algorithm that is tolerant to clustering errors that frequently happen with degraded documents.

Our proposed approaches are aimed at investigating solutions to the above problems.

2.3.2 Deciphering as Relaxation Labeling and Application to OCR

Relaxation algorithms have been applied to image processing in the past [126, 144, 176]. They are iterative parallel classification algorithms that estimate the class membership probabilities of each element in a graph structure based on those of its neighbors and the constraints that exist among the neighbors. The process is iterated until a satisfactory classification is achieved. A new formulation of relaxation based on probability theory [126, 125] paves the way for more general application of relaxation algorithms.

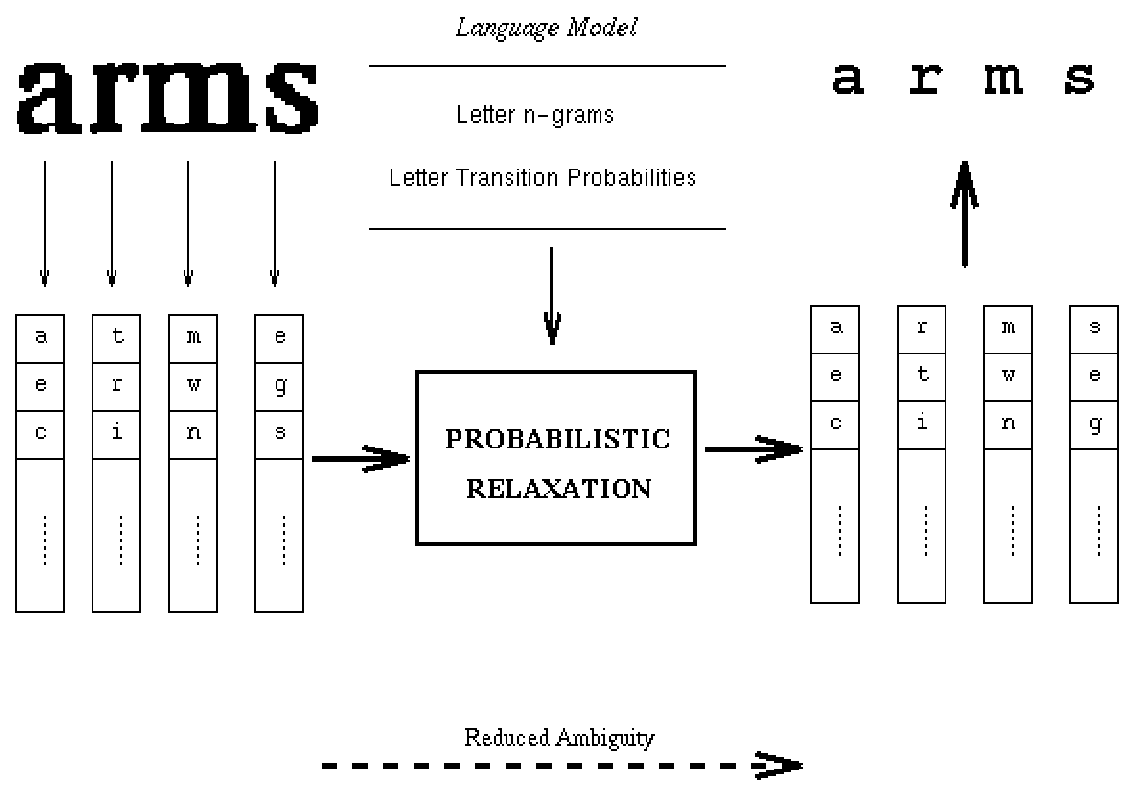


Figure 2.6: OCR as probabilistic relaxation.

Peleg and Rosenfeld developed an approach that uses a relaxation labeling algorithm to break a substitution cipher with application to OCR [127]. In this approach, every cipher text code is assigned probabilities of representing plain-text letters. These probabilities are updated in parallel for all cipher-text codes, using joint letter probabilities (trigrams). Iterating the updating scheme results in improved estimates that finally lead to the breaking of the cipher(Figure 2.6).

Later similar studies in applying a relaxation algorithm for breaking a substitution cipher resulted in improved updating rules [91, 79] and deciphering algorithms for substitution ciphers where word boundaries may have been concealed [79, 90]. Relaxation labeling was also applied to contextual word recognition where initial probabilities of character patterns being certain alphabetic letters are provided by a character recognizer and relaxation labeling is used for post-processing by subjecting character identity probabilities to letter n-gram constraints provided by a language model [62].

Relaxation labeling algorithms for decoding a substitution cipher rely on using character level language constraints in decoding character identities. A challenging direction of research which has not been investigated is to use a relaxation labeling algorithm for the direct identification of word identities by using word level language constraints provided by a language model. An immediate advantage of this approach is that it renders a natural solution to the difficult problem of recognizing words with extensive touching patterns, to the extent that all the characters inside a word could be touching together in case of severe image degradation. The major difficulty here is that the existence of relatively stable and reliable language constraints on the character level, such as character n-gram statistics or a vocabulary of common words, usually will not scale up to the word level. This issue has to be addressed before a deciphering approach can be applied on the word level to directly identify words. Investigation in this direction constitutes one of the major objectives of the approach proposed here.

2.3.3 Combinatorial Optimization and Simulated Annealing in OCR

Experiments have recently been reported that consider character recognition as a process of combinatorial optimization. A simulated annealing algorithm can be used to solve the optimization. Examples can be found in [48, 49].

An instance of a combinatorial optimization problem can be described as a pair$(\mathcal{R}, \mathcal{F})$, where $\mathcal{R}$ the configuration space, is a finite set of configurations, and $\mathcal{F}$ is a cost function that attaches a real number(cost) to each configuration $A \in \mathcal{R}$. The problem is to find a configuration A_{opt} such that $\mathcal{F}(A_{opt})$ is the global minimum. Optimization algorithms that provide a global minimum usually require a prohibitively large amount of time. However the quality of the solution can be traded off against the time required to obtain the solution by using approximation algorithms. Simulated annealing is an approximation algorithm for finding a global minimum which is based on randomization techniques.

Simulated annealing combines the simplicity of an iterative improvement algorithm with the power of randomization techniques to search through the configuration space and settle with uniform probability on the set of global minima. In an iterative improvement(or local search) algorithm, there exists a generation mechanism which is a simple means of generating a transition from a given configuration to another one. The generation mechanism defines a neighborhood structure on the set of configurations: the neighborhood of a configuration A is the set of configurations $\mathcal{N}$ that are reachable from A in exactly one transition. At each step of the algorithm an element of the neighborhood is selected but a move occurs only if the the cost function decreases. Each output of the generation mechanism selects one element of the neighborhood. Starting from an initial configuration the local search uses the output of the generation mechanism to consider the transition(the next configuration). The transition is acceptable if the cost function is decreased in the proposed configuration. The algorithm terminates if no neighboring configuration with less cost can be found. The algorithm, although simple and fast, has the disadvantage of finding only a local minimum.

Randomization is introduced in simulated annealing and it allows transitions to happen occasionally even if the cost function is deteriorating. The acceptance criterion for a transition employs a probabilistic rule called the *Metropolis Criterion*. The Metropolis Criterion, in conjunction with the method of simulated annealing, was introduced by Metropolis *et al.* in 1953 as an algorithm for simulating the evolution of a solid in a heat bath. Starting from a configuration A_i with associated cost $F_i = \mathcal{F}(A_i)$ and generating configuration A_j with its associated cost F_j, the transition is acceptable with probability $P(A_j acceptable)$,

$$P(A_j acceptable) = \begin{cases} 1 & F_j \leq F_i \\ \exp((F_i - F_j)/T) & F_j > F_i \end{cases}$$

T is the control parameter and is analogous to temperature in the annealing process of metals. For large value of T, large deteriorating of the cost function are accepted, and for $T = 0$, the algorithm reduced to a simple local search algorithm. The algorithm starts from an initial configuration with a relatively large value of T. For a given value of T a number of transitions are considered and then T is lowered until it reaches zero.

General information about simulated annealing and it various applications can be found in [97, 1].

Forsyth and Safavi-Naini [48, 49] proposed an elegant formulation of decoding a substitution cipher as a combinatorial optimization problem and suggested using a simulated annealing algorithm to find the optimal solution that corresponds to the breaking of the cipher. A configuration of this combinatorial optimization problem is defined as a specific mapping from the cipher alphabet to the plain-text alphabet. The cost for a configuration is defined as the discrepancy of relative character bigram frequencies measured from across the cipher text and from a language database, as a result of the specific mapping. Satisfactory results have been reported when the matching language database is correctly chosen and the input text is long enough to provide reliable character n-gram statistics.

Again, as in the case of using relaxation labeling for decoding a substitution cipher, applying a simulated annealing algorithm to the direct identification of words by using word-level language constraints poses a challenging research direction that could significantly improve OCR performance.

2.4 Summary and Conclusion

We have reviewed research in three areas. They represent the important developments over the past three decades in OCR and document recognition that are related to the research in this thesis.

Traditional systems for document recognition and most of today's commercial systems rely on character segmentation as an important processing stage. Unfortunately many of the available algorithms for character segmentation tend to be fragile and easily affected by image degradation. Poor results in character segmentation tend to significantly affect the overall recognition performance of the whole system. Efforts that attempt to seek better character segmentation results have represented a substantial portion of the research in OCR and document recognition. Those important segmentation approaches were briefly reviewed in this chapter and revealed the scale of the efforts as well as the level of difficulty in finding a general solution.

It is sometimes rather frustrating to see such a seemingly simple task of segmenting a character from a word image could be so difficult to achieve. But, as had been pointed out in Chapter 1, the difficulty in finding a general solution to character segmentation is not accidental. Rather, it is a reflection of the fundamental difficulty of *figure-ground segregation* in vision. Researchers in the fields of computer vision, image processing, visual psychology as well as cognitive science have long sought to find the mechanism in human beings that achieve figure-ground segregation effortlessly. Extensive research efforts related to the computational simulation of figure-ground segregation have been reported in the areas of perceptual organization, texture discrimination and shape decomposition [108, 128, 101, 14, 88]. But even today, knowledge of the mechanisms as well as their computational formulation are still at the very primitive stage. To make things even less optimistic, sometimes even the human brain seems to be fooled by abnormal environments and make wrong decisions in figure-ground segregation, as is illustrated by the examples in Figure 2.7. Computational implementation of image segmentation is never an easy task. Therefore, seeking alternative approaches to document recognition that can avoid the difficulty of character segmentation seems to be a viable and promising solution.

Segmentation-free approaches to document recognition are a relatively new development in document recognition. As is shown in the above review, the state-of-the-art techniques in this area follow two general tracks. The first stresses holistic recognition directly on the word image themselves, while the second chooses to use the feature registration approach where sub-character features are detected across the text page and characters are recognized by feature grouping at the right locations.

Nevertheless, there are inherent flaws with the holistic word recognition approach. This method is based on feature representation of words and on comparison between the feature representations of the input word and those of the lexicon words. Therefore, despite the availability of fixes that can accommodate for minor font variations, this method is basically font-dependent. Moreover, there is the issue of of having to have a feature representation for words that can differentiate between similar but different words(e.g. "he" vs. "be," "hut" vs. "but," "alter"vs. "after," "barking" vs. "banking" and etc.), and can still tell that a word is the same after undergoing image degradation. These are issues in document recognition that still don't have satisfactory solutions.

The same observation also applies to the feature registration approach. The fact that sub-character features are detected across the text page independent of the notion of character locations is what makes this method segmentation-free. But some other difficult issues still demand solutions. For example, after the feature detection stage, formation of character hypotheses still depends on the character representation being used. A character classification system that depends on a chosen character visual representation is still font-dependent to a

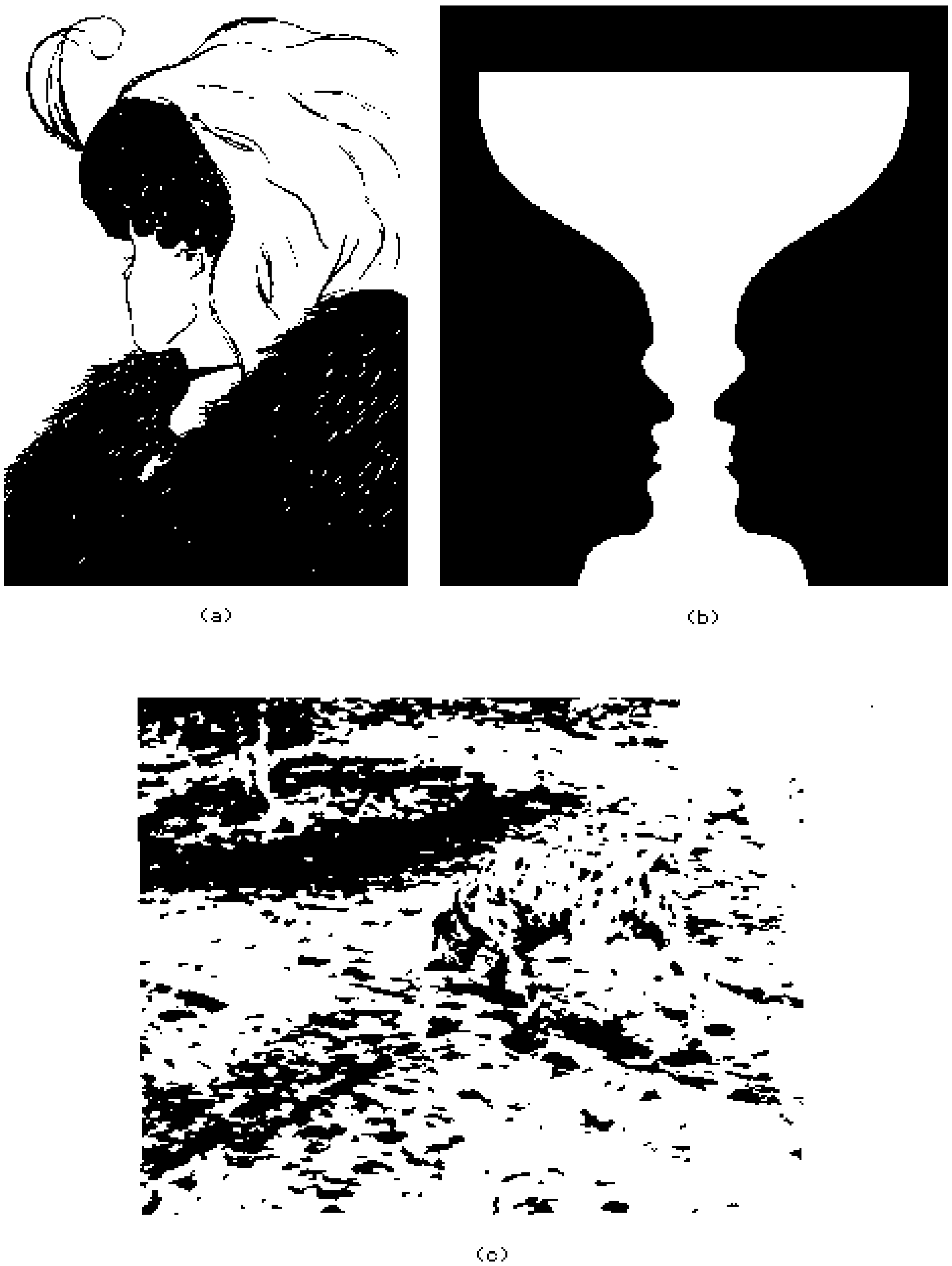


Figure 2.7: Examples that illustrate the difficulty in figure-ground segregation in human vision. a) wife or Mother-in-law? b) vase or opposing faces? c) see a sniffing dog?

large extent. Moreover, this method also has to address the issue of differentiating similar character patterns and deal with character deformations.

Therefore, it appears that a genuine font-independence and tolerance to image degradation are what are lacking in today's segmentation-free approaches to documentation recognition. We believe, as had been pointed out earlier in Chapter 1, the potentials of having a genuine font-independent recognition method for document recognition reside in adopting recognition approaches that can directly recognize words by uncovering the correspondence between the visual patterns and their linguistic identities. This is the direction our proposed word-level deciphering algorithms will be taking.

The success of deciphering algorithms that use character-level linguistic constraints has shown its power of rendering font-independent recognition methods, and also the potential for handling touching characters or character fragmentation by either enhancing the character level deciphering algorithm or by applying deciphering directly on the word level. These will be the focus of the approaches proposed in the next chapter.

Chapter 3

A Computational Framework for Document Recognition By Deciphering

As had been pointed out in the previous chapters, the main focus of this research is to seek novel solutions to the two fundamental problems of character segmentation and font training. Research and development efforts in the past have produced limited improvements for the solutions of these two problems mostly under specific conditions.

The research presented in this thesis seeks to solve these two problems by designing alternative approaches that can recognize documents with robust performance and do so by avoiding these two problems. In short, our research is focused on *segmentation-free* and *font-independent* approaches to document recognition. This approach is not only theoretically sound, but also delivers high recognition performance on degraded documents.

We propose to consider the computational task of document recognition as a process of finding the mapping between the visual patterns in the input document and their linguistic identities. Deciphering algorithms are proposed to decode the mapping by combining the visual constraints from the input document and the various levels of linguistic constraints from the language base. This deciphering approach to document recognition works on different levels of language granularities, both on the character level as well as on the word level. Therefore, document recognition can be achieved by decoding characters using character level linguistic constraints; it can also be achieved by the direct decryption of words using word level linguistic constraints.

With a deciphering approach, image patterns of characters and words are recognized not by visual classification but by the decryption of the font-independent correspondence between the visual patterns and their linguistic identities. This makes a deciphering approach font-independent. Furthermore, the potential that decryption could be applied above the character level to directly recognize the words leads to a recognition approach that is free of character segmentation.

The system architecture of the proposed approaches is shown in Figure 3.1. A pattern classification stage uses a clustering algorithm to group connected components or word images in the input text into different clusters. A deciphering algorithm is then applied to decrypt a selected portion of the input text that is available for cipher solution. Font information is then dynamically learned from the decrypted portion of the text and used to recognize the rest of the text that is difficult to solve through decryption.

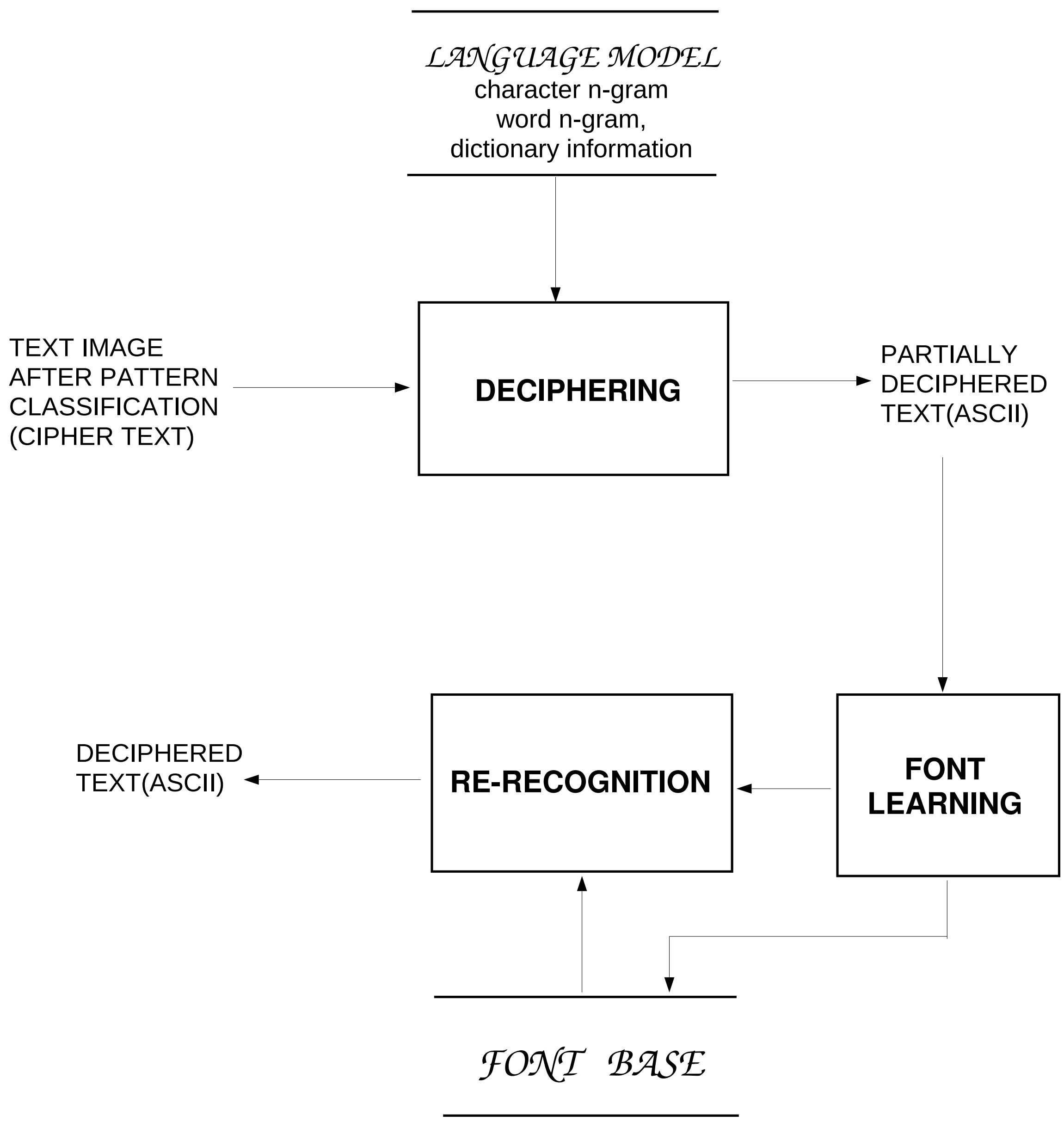


Figure 3.1: System architecture of the deciphering approaches.

3.1 Character Level Deciphering

A modified character-level deciphering algorithm is proposed that recognizes characters or touching characters by combining character level linguistic constraints with visual constraints from the input document (Figure 3.2). The modifications are based on an Breadth-First Search deciphering algorithm proposed by Nagy [112]. The modified algorithm differs from Nagy's original algorithm in four important aspects: 1)it has the added ability of handling extensive occurrence of touching characters; 2)it is tolerant to mistakes made at the clustering stage by being able to detect and correct the clustering errors, and 3)font information is learned dynamically along the deciphering process and used to help recognize long touching patterns that are difficult to solve through decryption, and 4)visual constraints are tightly integrated with language constraints throughout the deciphering process to help decode difficult long touching patterns and improve assignments when language constraints are not sufficient.

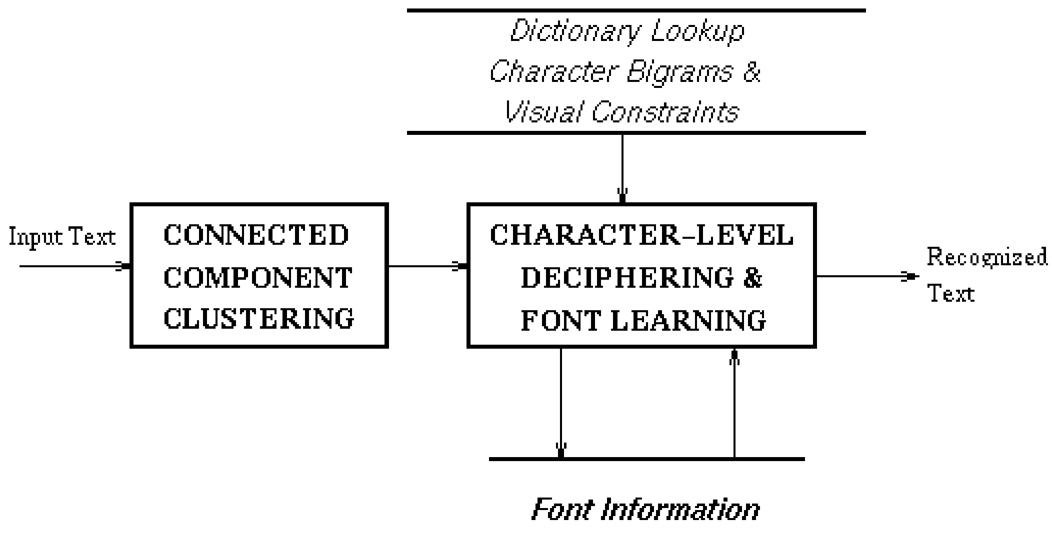


Figure 3.2: Outline of the modified character-level deciphering algorithm.

3.2 Word Level Deciphering

Before deciphering can be applied on the word level to directly recognize words, some difficult issues need to be resolved.

One of the major difficulty of applying a deciphering approach on the word level to directly recognize words is the lack of a stable and reliable word n-gram statistic pattern that needs to show in the input document. Also, unlike the existence of a dictionary that regulates the legal combinations of characters on the character level, there doesn't seem to exist a set of "legitimate English sentences" which regulates the legal combinations of words that the deciphering algorithm can use. Nevertheless, there are ways to solve these problems. They are based on the following observation.

While it is true that there does not seem to exist a content-independent and stable word n-gram repetition pattern for even extensive text passages, usually a portion of the words in the input document that are either subject-related key words or function words do repeat themselves in a way that they constitute relatively stable and reliable word bi-grams that match relatively well with those provided by a language model.

Therefore, we propose to solve a selected portion of the input text that has relatively more reliable word bi-gram statistics by using a word-level deciphering algorithm. Font information is learned from this decrypted portion of text and then used for re-recognizing the rest of the text not available for cipher solution. Figure 3.3 illustrates this idea.

A word-level deciphering approach to document recognition is proposed. Word-level language constraints are used to first decrypt the words from a selected portion of the input text that has relatively more reliable word n-gram statistics. Font information is learned from the decrypted portion of the text and then used to re-recognize the rest of the text.

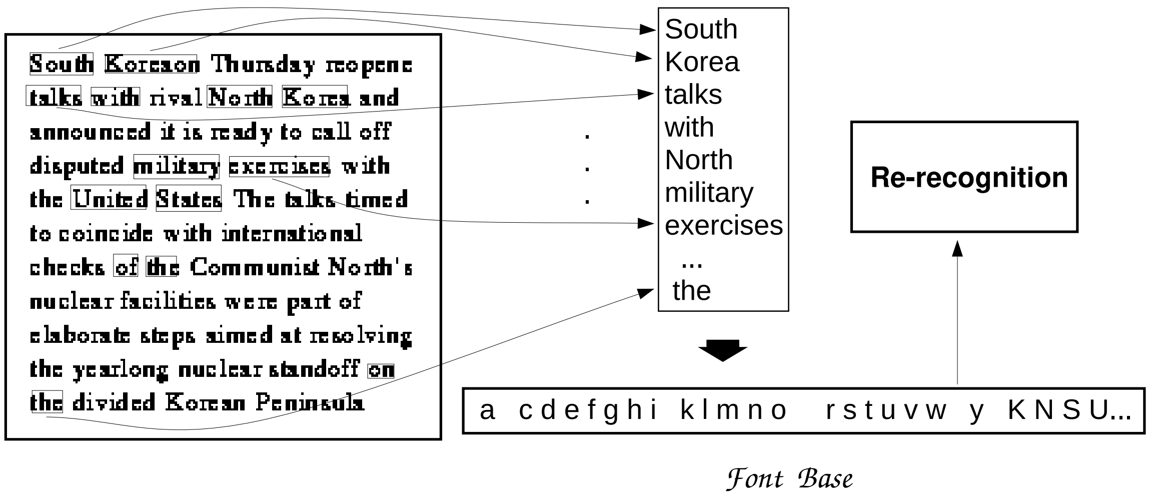


Figure 3.3: Word level deciphering.

This approach considers word deciphering as a process of constraint satisfaction and uses a probabilistic relaxation algorithm to achieve the decryption (Figure 3.4).

The following chapters discuss details of each proposed approach.

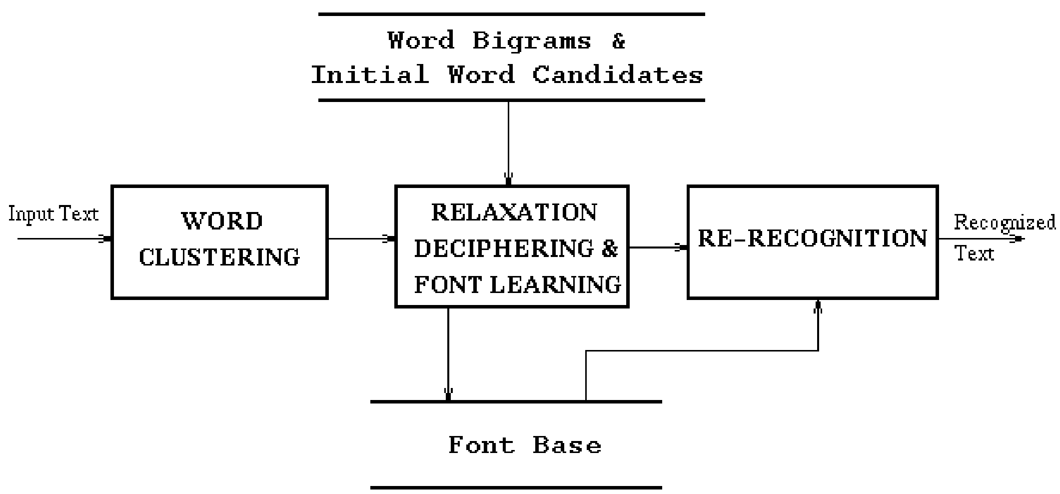


Figure 3.4: Outline of the word-level relaxation deciphering algorithm.

Chapter 4

Character Level Deciphering

The objective of a character-level deciphering algorithm for character image recognition is to assign alphabetic identities to character images such that the character repetition pattern in an input document image matches the letter repetition pattern provided by a language model. An essential preprocessing step for deciphering is a character image clustering algorithm that assigns images of equivalent characters to the same cluster. Degradation in document image quality can cause touching characters and mistakes in clustering the character images, which in turn cause difficulties for character-level deciphering algorithms.

Modifications to a previous character-level deciphering algorithm for OCR are presented in this chapter. The modifications tightly integrate visual constraints from characters and touching patterns with linguistic constraints from a language model. This allows the deciphering algorithm to recognize touching characters and to detect and reverse clustering mistakes. This results in a deciphering algorithm that has robust performance under image degradation.

4.1 Introduction

Character-level deciphering algorithms have been proposed for OCR in the past [20, 23, 112, 114, 127]. These techniques recognize character images in a printed text by solving a substitution cipher(Figure 2.5). An image pattern clustering step first converts each character on the input page into a computer readable code so that every shape corresponds to a distinct code [175, 24]. A deciphering algorithm is then applied that uses a language model to assign alphabetic labels to the cipher codes so that the character repetition pattern in the input text passage best matches the letter repetition pattern provided by the language model.

A technique that is representative of the many proposed character-level deciphering algorithms is that presented by Nagy [112]. This algorithm solves a substitution cipher by extending a tentative assignment of letters to character patterns according to the degree of match between the decrypted portion of the cipher text and a vocabulary of common words.

Image degradation in an input document causes two problems that are difficult for current deciphering algorithms to handle. One problem is the occurrence of touching characters. The other problem is that mistakes usually appear in the clustering stage. Character pattern clustering mistakes can be categorized into four types. The first type causes visually similar samples of different characters to be included in the same cluster. The second type splits samples of an identical character into different clusters. The third type involves certain touching character patterns that look like other single characters and are merged into the same cluster. Finally, the fourth type of clustering mistakes mix fragments of characters with single characters that are visually similar. Figure 4.1 illustrates each of these four types of character clustering mistakes with an example. Touching characters and clustering mistakes are hard problems with character-level deciphering algorithms because they invalidate the assumption that character repetition patterns in the input text match that of a language model, which is the basic assumption that all character-level deciphering algorithms for OCR depend upon. This is also the main reason why deciphering algorithms, though theoretically well fitted to the task of character recognition, are difficult to use for a practical OCR system that needs to deal with degraded documents.

Previous deciphering algorithms can handle clustering mistakes that split identical characters into different clusters. But it is difficult for them to handle touching characters and clustering mistakes that group different characters into the same cluster. Also, previous methods use primarily linguistic constraints such as lexicons or character n-gram statistics. No attempt has been made to also use visual constraints in the deciphering process to improve the performance.

Our modifications to a character-level deciphering algorithm are intended to address the above issues. Starting with the algorithm NONMATCHES proposed by Nagy in [112], the proposed modifications integrate visual constraints of character patterns extracted from the text page and linguistic constraints provided by a language model to decode the identities of characters and touching patterns. Short touching patterns are solved by matching the current cipher word against the dictionary with the assumption that it might include touching patterns each consisting of an estimated number of single characters, and resolving the multiple matches with constraints from the *cover set* [112]. The *cover set* of a cipher word consists of a group of other unsolved cipher words from the same document page whose identities are dependent on the decryption of the current cipher word. The more difficult long touching patterns are postponed to be processed at the end of the deciphering process when most of the single characters and short touching patterns have already been deciphered and their font information is available to help identify the long touching patterns by performing partial pattern matching.

The modifications proposed here result in an algorithm that is tolerant to both kinds of clustering mistakes mentioned above, and is able to decipher touching characters.

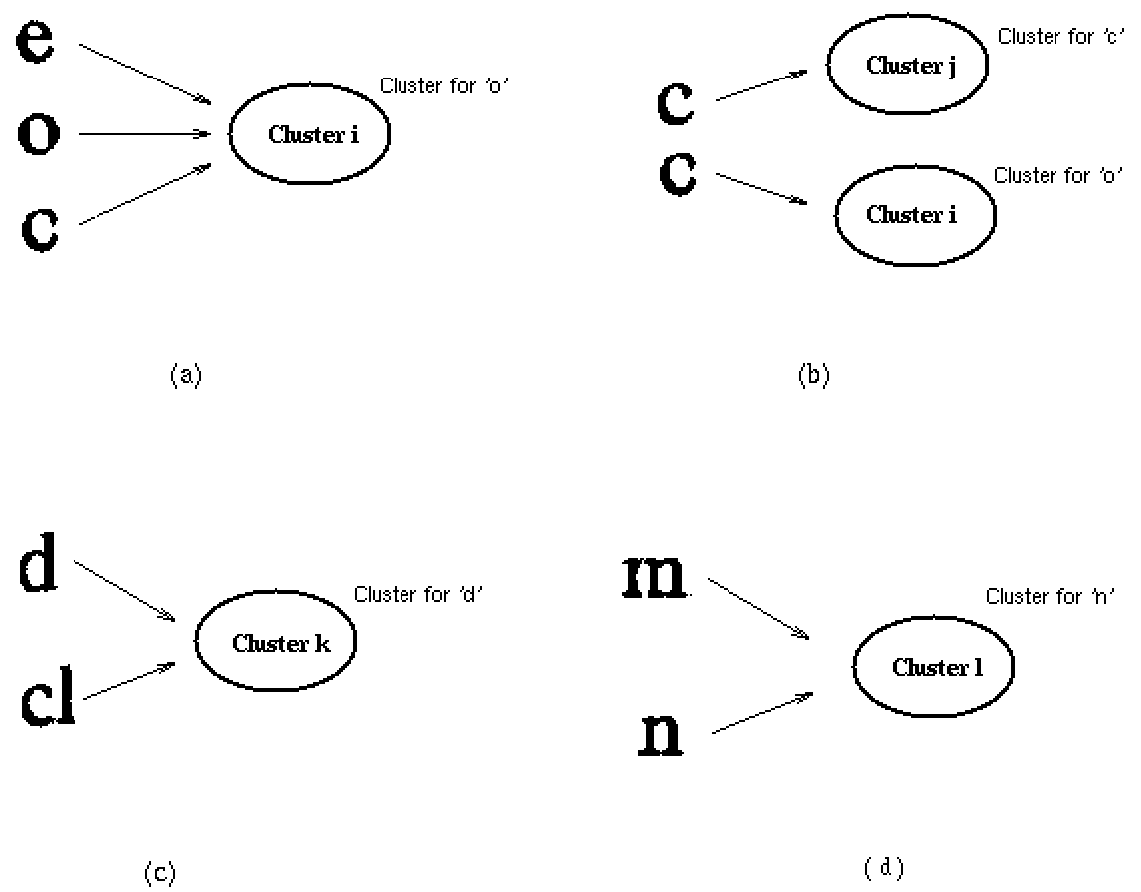


Figure 4.1: Character clustering mistakes. a) similar but different characters included in a single cluster, b) identical characters split into different clusters, c) touching characters assigned to a single-character cluster, and (d) fragment of a character being put into a cluster of another single character.

The remainder of this chapter contains three sections. Section 2 outlines the original character-level deciphering algorithm proposed by Nagy and the basic concepts used in the algorithm. Section 3 discusses how our modifications to the algorithm solve the problem of touching characters. Section 4 discusses ways to handle clustering mistakes. The modified algorithm as a whole is presented in Section 5. To test the validity of the modifications for handling touching characters and clustering mistakes under image degradation, the modified character-level deciphering algorithm is applied to both scanned documents as well as artificially created test documents that simulate various document degradations. Experimental results are presented and discussed in Section 6. Finally, conclusions and directions for further improvements are pointed out in section 7.

4.2 The Original Algorithm

Figure 4.2 is an outline of the original algorithm proposed by Nagy in [112] that our modifications are based upon.

Algorithm *NONMATCHES*(input, output)

Begin

Initialize partial assignment PA to be empty;

Repeat

1. Find the most frequent undecided cipher word CW and its associated *Cover Set*;

2. Match the selected CW against dictionary, undecided characters act as wild cards;

3. choose the assignment that results in the least number of un-matched words in the *Cover Set*;

4. Update partial assignment PA;

Until partial assignment PA is complete

End

Figure 4.2: Outline of Nagy's original deciphering algorithm.

This algorithm solves a substitution cipher by extending a tentative assignment PA of letters to character patterns according to the degree of match between the decrypted portion of the cipher text and a vocabulary of common words. To make use of the constraint that the same characters can occur in many different cipher words, the concept of a *Cover Set* associated with each current cipher word is introduced. If S denotes cipher words that are not completely decrypted by the current tentative assignment PA, a word $W1 \in S$ is said to *cover* a word $W2 \in S$ if every unassigned character pattern in $W2$ is also in $W1$. All the words covered by $W1$ form the *Cover Set* of $W1$. The idea here is that a valid new set of character assignments that cause the current cipher-word Wi to match a dictionary word should also make all the cipher-words in the cover set of Wi legal dictionary words.

Figure 4.3 shows an example of how the above algorithm deciphers character patterns using the cover set constraints in combination with dictionary matching. Let's assume that only lower case letters are involved in the input document, and somewhere along the deciphering process three character patterns 'a', 'e' and 't' have been correctly deciphered and assigned the correct letter labels, as indicated by the partial assignment of cipher codes at the top of Figure 4.3. The current cipher word is "there." Its cover set includes three words: "that," "the" and "are." The patterns for 'h' and 'r' in the current cipher word "there" are the unknown patterns that need to be deciphered. Matching the current cipher word against the dictionary with the two unknown patterns replaced by two wild cards, we obtain two matching results. The first is "there," and the second is "these." Each of them is associated with a new assignment of cipher codes to letter labels. For each dictionary match, we then count the total number of mismatches of cipher words in the cover set as a result of adopting the new assignment. In this case, the new assignment associated with "there" will result in every cipher word in its cover set finding a match in the dictionary. That is, the total number of mismatches is 0. Because the new assignment associated with "these" will make it impossible to find a match for the cipher word "are" in the dictionary, the total number of mismatches is 1. The assignment that results in the least number of mismatches among cipher words in the cover set is chosen as the deciphering result of the current cipher word, and is added to the current partial assignment. In this case, two new letter assignments that represent the decryption of the characters 'h' and 'r' are added to the current partial assignment. This decoding process repeats itself with more and more cipher words being decoded, until all the cipher codes are decrypted and assigned the right letter labels.

The proposed modifications presented in the following add to the original algorithm the abilities of handling touching characters and clustering mistakes, which until now were unsolved problems for character level deciphering algorithms.

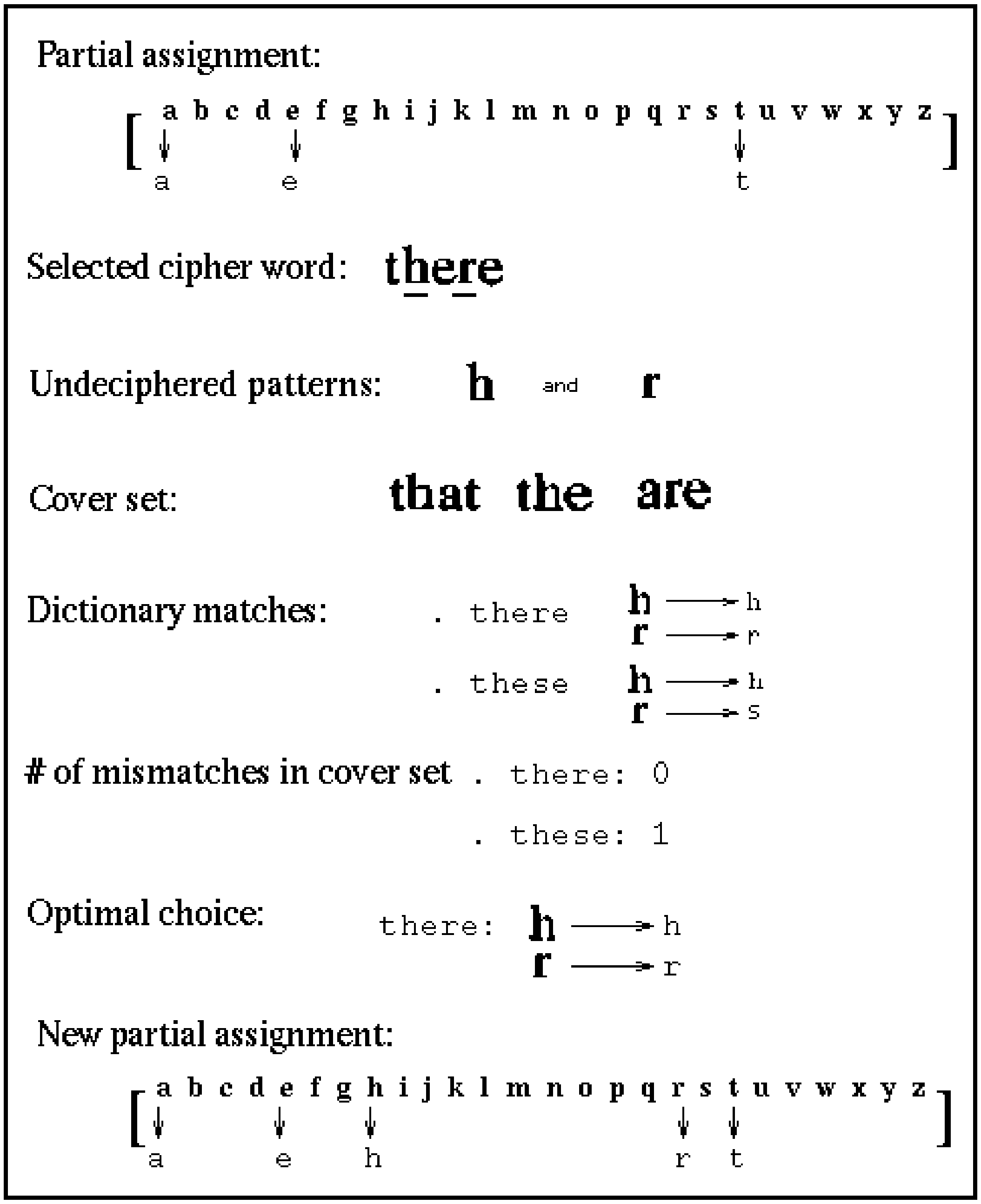


Figure 4.3: An example of using the *cover set* constraints and dictionary matching in Nagy's character level deciphering algorithm.

4.3 Dealing With Touching Characters

There are two kinds of touching character patterns according to their effect on the deciphering algorithm. Some touching character patterns are caused by image blurring, minor noise, and font specificities such as ligatures(e.g., "ff," "fl," etc.) as shown in Figure 4.4. The other kind of touching character patterns are caused by severe image degradation and result in long touching patterns such as those shown in Figure 4.5. Touching patterns in the first category are usually short, involving two or three characters. These short touching patterns are also likely to repeat themselves elsewhere in the document. Touching patterns in the second category are usually long and irregular, and they may not be repeated elsewhere in the document.

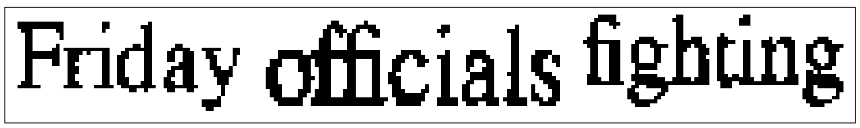

Friday officials fighting

Figure 4.4: Regular short touching characters in a degraded document.

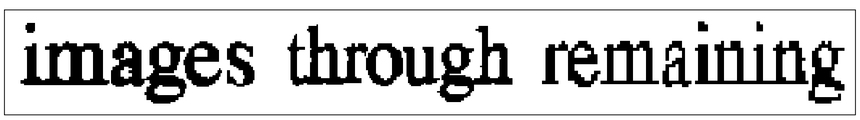

images through remaining

Figure 4.5: Irregular long touching characters in a degraded document.

4.3.1 Short Touching Patterns

Short touching patterns are solved along with single characters in the deciphering process. A potential short touching pattern is detected and labeled as such if it has a larger than usual aspect ratio and a small cluster size. A preliminary estimate N that denotes the number of characters involved in a potential touching pattern is derived by comparing the physical width of this touching pattern with the average character width. The cipher-word that includes this potential touching pattern is then matched against the words in the dictionary under the assumption that $M \in [N-l, N+l]$ characters are involved in this touching pattern. l represents the uncertainty of estimation N. The parameter l provides the algorithm with a tolerance to inaccuracies in the preliminary estimation of N at the expense of efficiency in dictionary matching.

A current cipher word is then matched against all the words in the dictionary using all such "wild cards." The match provides a set of potential assignments for the touching patterns. The optimal assignment is chosen using the *cover set* constraints and character bigram statistics.

An example of deciphering words with touching characters is shown in Figure 4.6. The word currently being deciphered is the word "Armed" in which the characters 'A,' 'e' and 'd' had already been deciphered, 'r' touches 'm' and forms the pattern "rm" that needs to be deciphered. Its cover set includes a single cipher-word "warm," where character 'w' and 'a' had already been deciphered, and the touching pattern "rm" needs to be decrypted. By comparing the average width of the connected components in the input document, the number of characters in the touching pattern "rm" is estimated to be 3. We choose the tolerance range l to be 1. Therefore 2, 3 and 4 wild-cards are used in place of the touching pattern when the cipher-word is matched against the dictionary. Figure 4.6 shows the results of the dictionary matching. Having the cipher-word "warm" in the cover set helps to decide that "rm" is the only assignment to the touching pattern that makes itself a legal dictionary word as well.

Thus the string "rm" is assigned to the touching pattern for "rm" and added to the partial assignment. Note that the final complete assignment of cipher codes to letter(or string) labels could include assignments of single characters as well as their touching patterns.

The deciphered single characters and short touching patterns are then used to build up a partial font base as the deciphering proceeds. This consists of image prototypes for deciphered characters and short touching patterns. This information is used to decipher the long touching patterns.

4.3.2 Long Touching Patterns

An example of a long touching pattern("arts") is shown in Figure 4.7. Long touching patterns cannot be deciphered the same way as short touching patterns for two reasons. First, in a cipher word that includes irregular long touching patterns, there usually are very few deciphered single characters to provide constraints for the dictionary match. This usually results in a great number of potential matches of the cipher word that provide few constraints to the final resolution of its linguistic identity. For the example in Figure 4.7, the character 'p' provides little constraint when the cipher-word is matched against the dictionary. Furthermore, there are usually no constraints in the *cover set* to resolve the many potential matches because irregular long touching patterns are not likely to repeat themselves elsewhere in the text page. This results in a *cover set* that is usually empty. For the cipher-word in Figure 4.7, the long touching pattern "arts" is not likely to appear in another cipher-word that could be included in the *cover set* of "parts" to help decide its identity. For these reasons, long touching patterns are processed after most of the single characters and short touching patterns have been deciphered and their font information is available. A recursive partial pattern matching process is applied to each long touching pattern to decide its identity based on the learned font information. The multiple matching results are then passed to a post-processing stage where dictionary and word bigram statistics are used to finally decide its linguistic identity.

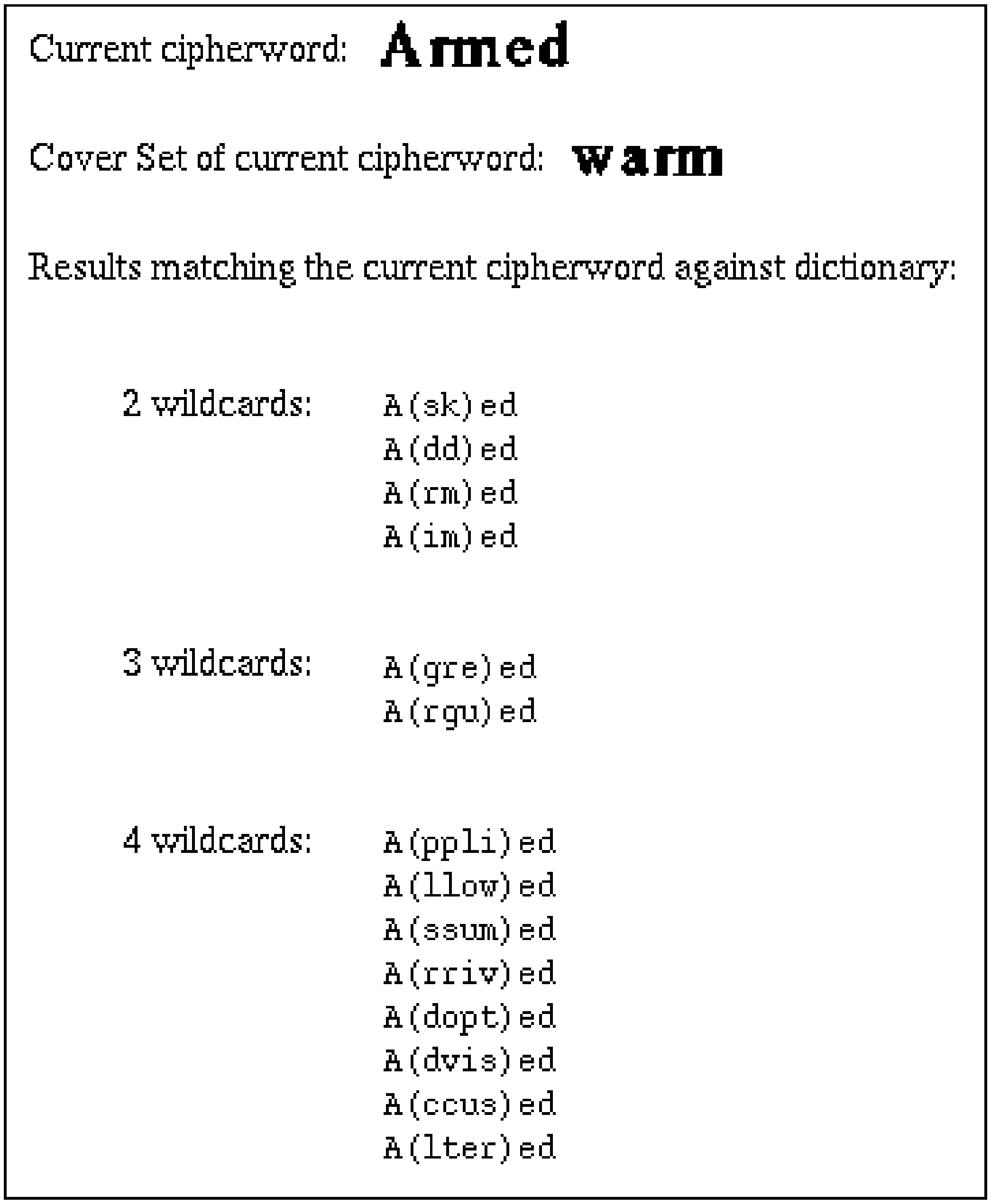


Figure 4.6: An example of deciphering word with touching characters.

Details of the recursive partial pattern matching and the post-processing stages are described in the next section.

Figure 4.7: An example of a long touching pattern.

4.3.3 Recursive Partial Pattern Matching And Post-Processing

When a cipher word that includes long touching patterns is encountered, it usually has three kinds of components: the deciphered characters or touching patterns, un-deciphered single characters, and un-deciphered touching patterns. All the un-deciphered touching patterns are inputs to the partial image matching step.

Starting from the left edge of a touching pattern, all the learned character prototypes are matched against the very left portion of the touching pattern. For each prototype, its optimal matching position is recorded together with a value that denotes the degree of match. For each match at the current character position, all the prototypes are matched again with the next potential character. This matching process is applied recursively until the right edge of the touching pattern is reached. The whole matching result is represented in a tree structure, with multiple choices at each character position and multiple choices following each particular choice at a character position.

There is a trade-off between keeping more choices at each potential character match position, which increases the recognition rate, and pruning all but the best matches, which improves runtime efficiency. Instead of just selecting the optimal choice at each position and combining them into the optimal identity for the touching pattern, our algorithm maintains the top n choices at each matching position.

All the potential touching pattern identities generated from this tree are fed to a linguistic post processor where combinations of potential touching pattern identities, together with the deciphered characters or touching patterns in the word, as well as the un-deciphered single characters acting as wild cards are matched against the dictionary. This usually results in a single match. Occasional multiple dictionary matches are resolved using word bigram statistics.

4.4 Handling Clustering Mistakes

4.4.1 Character Clustering and its Mistakes

Clustering is the process of grouping objects into sets such that the objects within a set are similar to one another while objects in different sets are different from each other. There exist many ways to measure the similarity among objects. In character recognition applications, the objects are individual character images or touching characters. They are represented either directly by the image pixels or by certain feature representations. The ideal clustering result is one in which all the instances of the same character or touching pattern are grouped into a single cluster, and each cluster includes only characters or touching patterns with the same identity. Accurate clustering performance is important to a character-level deciphering algorithm because characters are deciphered by matching the visual pattern repetition statistics calculated from pattern clustering to the letter repetition statistics provided by a language model.

The existence of visually similar characters with different identities makes it extremely difficult to achieve good clustering results. Image degradations that cause character pattern deformations aggravate this problem. There have been many research efforts that have tried to improve clustering performance by optimizing a chosen mathematical objective function [80, 2, 38]. Nevertheless, achieving accurate clustering results especially on degraded images is still a challenging issue.

Two problems need to be solved. First of all, a clustering algorithm for character patterns needs to be developed so that good clustering performance can be obtained in the presence of image degradation. Secondly, in case mistakes do happen in clustering, they should be detected and corrected by the deciphering algorithm.

4.4.2 Character Pattern Clustering

Figure 4.8 shows the outline of the character pattern clustering algorithm we used. It clusters an input set of connected components. For degraded documents, a connected component could be the image of a single character. It could also be an image of touching characters.

The algorithm generates the first cluster by putting the first connected component in it as the seed member. Each of the following connected components are matched against all the existing clusters. The numerical values that represent how well a given component matches all the existing clusters are sorted to find the best match. If the numerical value for the best match exceeds a threshold, the connected component is accepted as another member of the cluster of best match and added to that cluster. The prototype for that cluster is also updated accordingly. Otherwise, if the numerical value for the best match is less than the threshold, a new cluster is formed with the current connected component as its initial member.

The representation of the cluster prototype includes its visual image as well as information about its height, width and etc. The images of new incoming members are used to update the prototype representation. To smooth out image distortions and disparities in individual connected components, a weighted averaging approach is used to form the new prototype image. The image of each incoming connected component is first matched with the current prototype image. The image matching step that counts the percentage of overlapping pixels is a two-way matching. It is conducted by both matching the incoming pattern against the prototype and also matching the prototype against the incoming pattern. The smaller of the two values is used as the result. This is necessary because there are many instances where a character(or touching pattern) is part of another character(or touching pattern), and a high percentage of pixel coverage of the smaller pattern by the enclosing pattern could be misleading in suggesting the two patterns match. Figure 4.9 shows two such examples. Suppose in these cases 'n' and 'b' are the incoming character patterns that need to be clustered, and they are matched against the cluster for character 'h.' If only a one-way match is carried out to

Algorithm **Char_Pattern_Clustering(***Set of Connected Components***)**

begin

  create a new cluster, its only member being the first Connected Component;

  $Num\_Clusters = 1$;

  for (the rest of all the Connected Components)

    match current Connected Component with each of the existing cluster;

    find the best match;

    if (best match value above threshold)

      add current Connected Component to cluster of best match;

      update cluster prototype;

    else

      create a new cluster, its only member being the current Connected Component;

      $Num\_Clusters = Num\_Clusters + 1$;

    end-if

  end-for

end-algorithm

Figure 4.8: Outline of the character pattern clustering algorithm.

measure the percentage of pixels in 'n'(or 'b') that are covered by pixels in 'h,' then the high percentage might suggest a correct match. On the other hand, the much lower percentage of pixels in the prototype of 'h' that are covered by either 'n' or 'b' suggests a mismatch, which will be the final matching results by using a two-way match.

A new character pattern at the optimal matching position where the two overlap the most is then added to the prototype image to form the new prototype. The prototype image thus formed will have a highly weighted "central" portion that represents the stable image of a clean character, as well as a relatively low weight peripheral region that represents noise and degradations.

When evaluating the returned value that represents the degree of match between the incoming connected component and the cluster prototype, the selection of a threshold is necessary and has been a difficult problem for many clustering algorithms. Ideally, there exists a single threshold value such that a matching result greater than this indicates an acceptable match, whereas a matching result smaller than this indicates a mismatch. Unfortunately, a single threshold usually will not work with real document images for several reasons. On the one hand, similarities between characters(e.g., 'c' vs. 'e' and 'h' vs. 'b') require a higher threshold to differentiate small image details. On the other hand, image degradation causes different instances of the same character(or touching pattern) to look very different, which indicates the need for a lower threshold. In addition, there is the concern about image pattern size and the relative sensitivity to noise. For narrow characters like 'i' and 'l,' a tiny widening of the vertical stroke caused by image degradation usually results in a relatively large decrease in percentage of match, while the same degradation shows little effect on matching percentages for wider characters like 'E,' or for touching patterns. To address these issues, multiple threshold values are used in our clustering algorithm. The threshold is determined by the size of the connected component under consideration, low thresholds are used for slim characters and high thresholds are used for wide characters or touching patterns.

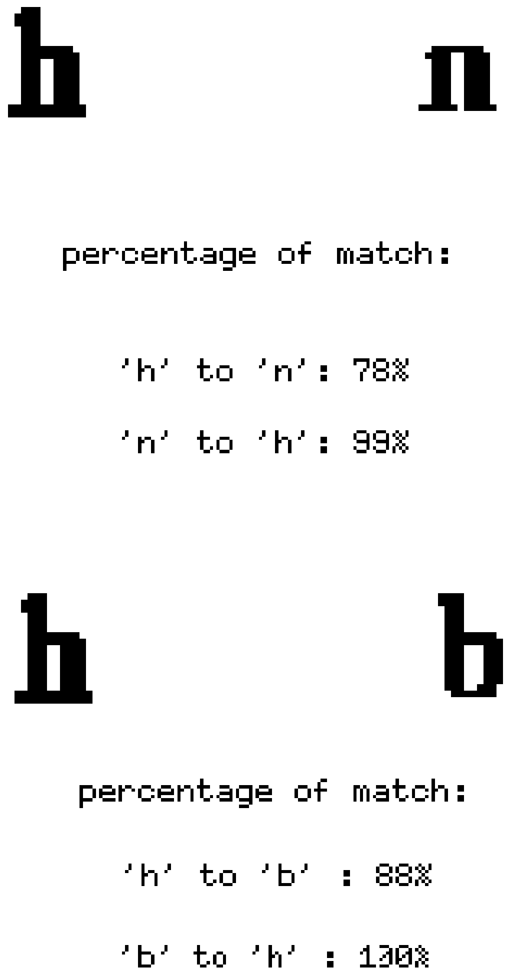


Figure 4.9: Examples of mismatches that could be considered correct if a one-way pattern match is used.

4.4.3 Detection of Clustering Mistakes

A crucial concern in the application of character-level deciphering algorithms to real-world degraded documents is the way clustering mistakes are handled. Character-level deciphering algorithms can handle clustering mistakes that split identical characters into different clusters. The following discussion concerns only with clustering mistakes that group different characters or touching patterns into the same cluster.

In the deciphering process, the letter identity of a cluster is determined when a cipher word that includes one of the cluster's members as its un-decrypted pattern is deciphered. This causes some members to be assigned incorrect letter identities if the cluster includes different characters. This, in turn, results in situations where no dictionary match can be found for cipher words that include such characters or touching patterns that have been assigned incorrect letter identities. This allows the algorithm to detect instances where characters or touching patterns are incorrectly deciphered and then identify the cluster that caused the problem.

4.4.4 Character Confusion Sets

For most fonts, there appear to exist sets of characters that closely resemble each other visually, and therefore are very likely to produce mistakes by a clustering algorithm. Examples of these characters are 'e' and 'c,' 'b' and 'h,' 'u' and 'n,' and, 'l,' 'f' and 'I.' Of course, some sets are easier to differentiate when the characters are not degraded and when certain feature representations are used by the clustering algorithm.

This heuristic of character confusion sets is used to help the deciphering algorithm to detect and reverse clustering mistakes. During the deciphering process, in case a cipher word does not match any words in the dictionary, it indicates a possible clustering mistake happened earlier. The algorithm then checks each of the deciphered characters of the cipher word to see if it is included in a character confusion set. If it is, the cipher word is matched again with the dictionary with the confusing character being assigned the other characters in the confusion set. Multiple matches are resolved with constraints from the *cover set* or from the learned font information. The optimal assignment is then used to reverse mistakes in character identity assignment and character clustering results.

Figure 4.10 shows the character confusion sets used in the algorithm. It includes four sets of visually similar characters. The first three sets each include two characters, and the fourth set includes three characters.

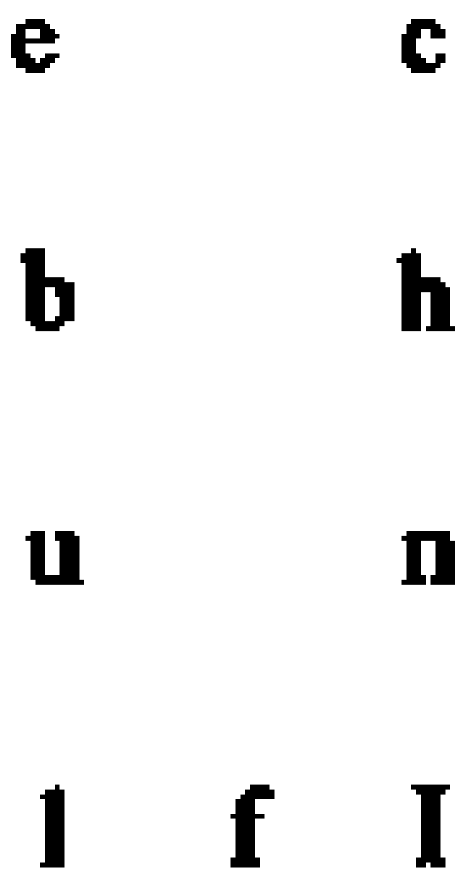


Figure 4.10: The four character confusion sets used in the deciphering algorithm.

4.5 The Modified Algorithm

An outline of the modified character-level deciphering algorithm is shown in Figure 4.11. The overall framework of a character level deciphering approach is inherited from the algorithm NONMATCHES originally proposed by Nagy in [112]. In addition to the functionalities of the original algorithm, this modified algorithm includes the ability to detect and label potential touching patterns and unusual long touching patterns. The dictionary matching algorithm is generalized to handle touching patterns as well. Font learning is implemented so that font information of the input document can be learned dynamically during the deciphering process, and used to solve the long touching patterns. Also, a recursive pattern matching algorithm is used to recognize and decipher long touching patterns with the help of linguistic constraints. To deal with the many situations where multiple competing matching results exist even with the use of cover set constraints, an algorithm is developed that uses character bigram information from the language base to resolve the multiple competing match results and to choose the optimal one. Finally, a character confusion matrix is used to exploit knowledge of visual similarities in characters. This helps to detect and reverse character clustering mistakes.

The above modifications make the algorithm tolerant to clustering mistakes and capable of handling extensive touching patterns. These added abilities extend the character level deciphering approach for OCR to real-world applications that involve degraded documents.

Initialize partial assignment *PA* to be empty;

<u>**Begin**</u>

    <u>**Repeat**</u>

        1. Find the most frequent undecided cipher word *CW* and its associated *Cover Set*, long touching patterns are processed later;

        2. Match the selected *CW* against dictionary, undecided characters or touching patterns act as wild-cards;

        3. <u>**if**</u> no dictionary match found

            <u>**then**</u>

                re-assign identities of confusing characters,

                re-match the *CW* against the dictionary;

        <u>**else if**</u> includes long touching pattern

            <u>**then**</u>

                decipher long touching pattern by visual pattern matching using learned partial font base;

        <u>**else if**</u> *Cover Set* not empty

            <u>**then**</u>

                choose the assignment that results in the least number of un-matched words in the *Cover Set*;

        <u>**else**</u>

            verify potential assignments with learned font information;

        4. Update partial assignment *PA*;

        5. Update font base;

    <u>**Until**</u> partial assignment *PA* is complete

<u>**End**</u>

Figure 4.11: Outline of deciphering algorithm.

4.6 Experimental Results And Discussions

4.6.1 Experiment Design

One of our primary interests in the modified character-level deciphering algorithm is its performance under image degradations. Specifically, we want to test the algorithm's competence in handling touching characters and its ability to deal with clustering mistakes. This is reflected in the design of the experiments. Two different sets of test documents were used for the experiments.

Set A of test documents consists of 10 artificially created document images. The original texts are excerpts from books with subjects including education, sociology, western classical music, general science, poetry, computer programming, cognitive science, and economics. The number of words in each document ranges from 88 to 135. The texts are then transformed into binary images by a text formatting program. Local pixel modifications are applied to individual word images to create long touching patterns within words by filling the white gaps between neighboring characters. Usually these long touching patterns do not occur frequently in real scanned documents. The purpose of this document set is to test the effectiveness of the proposed algorithm in recognizing long touching patterns. Figure 4.12 shows an example of a document in Set A.

Set B of test documents consists of 10 scanned images that include degradations that practical OCR systems need to deal with. These documents are generated the same way as the documents in set A. The images are then printed on clean white paper with a laser printer. The printed pages are then re-scanned at 300 pixel-per-inch(ppi) spatial resolution in 8 bit gray-scale. The final binary images are the result of thresholding the scanned gray-level images by choosing a fixed threshold that produces many distorted and touching characters.

Now is the perfect time to think
about what the military has to
offer. Although the military is
getting smaller, the Armed Forces
still need to recuit almost 400,000
young men and women for Active
and Rcscrvc positions cach ycar.

Education, training, and job
experience are important parts of
thc plan to restructurc today's
Armed Forces. They are also exactly
what tomorrow's cmployers will bc
seeking. Today's military is one of
thc most sophisticatcd and also,
technologically advanced organization
in the world a an far No big Idea warm
different zoology

Figure 4.12: An example of artificially created test document with long touching patterns.

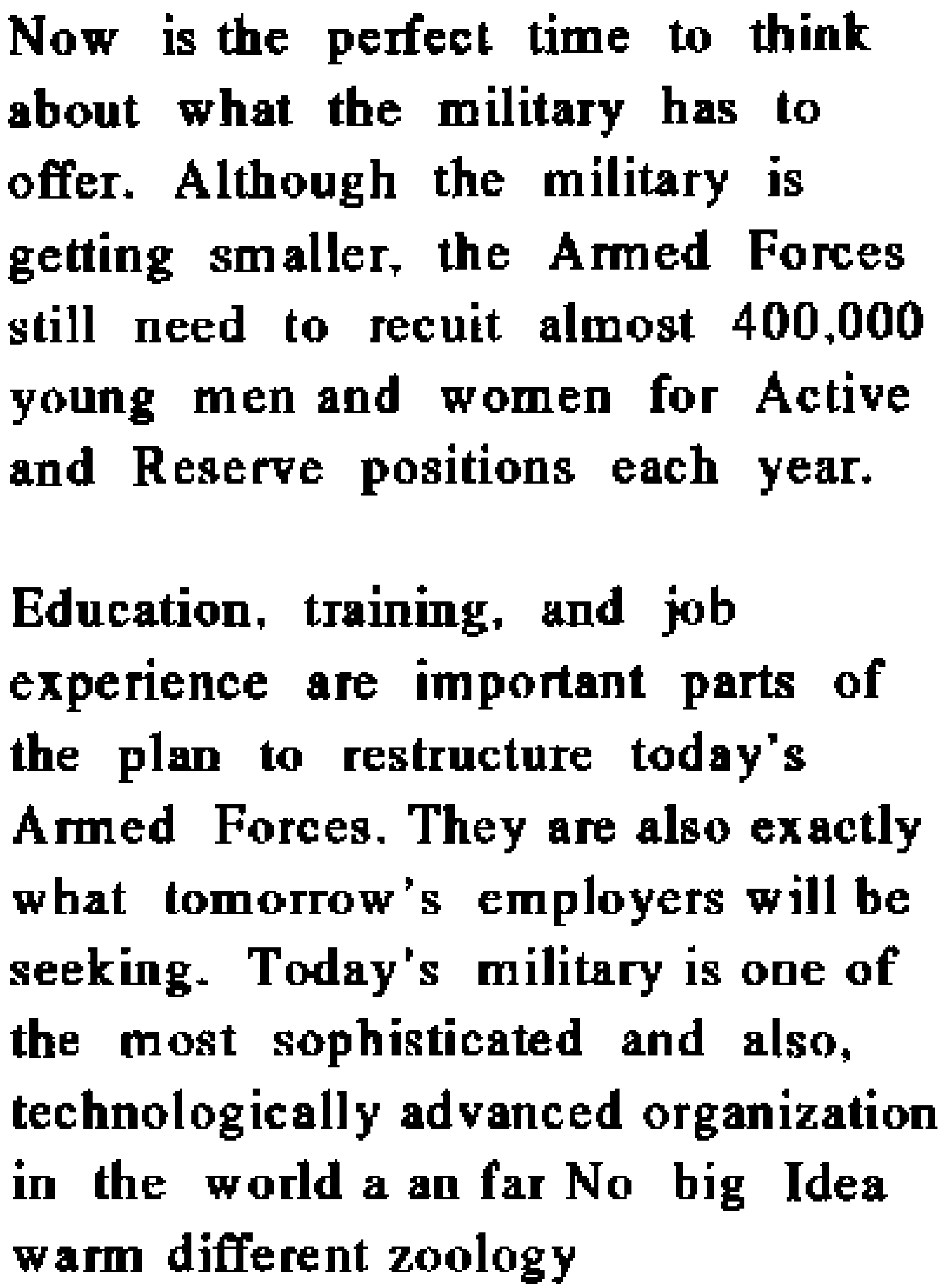
Different from the design of Set A, the design of set B is intended to test the general performance of the modified deciphering algorithm under real image degradations, as well as performance of the clustering algorithm. Figure 4.13 shows an example of a document in Set B.

The language model for the deciphering algorithm consists of a dictionary and a set of character bigram statistics. The dictionary contained all the 53121 unique English words from the Brown Corpus, which is an English text corpus that contains about one million words of running text [93]. The character bigram statistics database was also compiled from the Brown Corpus.

4.6.2 Baseline for Performance Comparison

The original Nagy's algorithm was implemented and run with input documents that contained no touching characters and no clustering mistakes. The performance of the original Nagy algorithm with perfect clustering results is a baseline for comparison with the results of running the modified algorithm on noisy document images that will be presented later.

A program was developed that implemented the Nagy NONMATCHES deciphering algorithm on a text input. It includes clustering based on character identities followed by the original NONMATCHES method. We added a few things that were not included in the NONMATCHES algorithm presented in [112]. The character bigram information compiled from the language base was used to resolve multiple competing results from dictionary matching. Multiple matching results happen when there is no cover set constraint or not enough cover set constraints are available. This often happens when deciphering is starting. A typical example is when the deciphering of a document starts with the cipher word "the" because of its high frequency in a document. None of the characters have been deciphered at this stage. Thus, the results of dictionary matching will be the set of all the three-character words in the dictionary. It is possible that sometimes the word "he" also appears in the document and is



Now is the perfect time to think about what the military has to offer. Although the military is getting smaller, the Armed Forces still need to recuit almost 400,000 young men and women for Active and Reserve positions each year.

Education, training, and job experience are important parts of the plan to restructure today's Armed Forces. They are also exactly what tomorrow's employers will be seeking. Today's military is one of the most sophisticated and also, technologically advanced organization in the world a an far No big Idea warm different zoology

Figure 4.13: An example of a scanned document with many touching patterns and deformations in individual characters.

included in the cover set of "the." This helps reduce the initial matching set to a much smaller sub-set that consists only of three-character words whose last two characters also make legitimate words in the dictionary. In other cases the cover set might be empty. These are the cases where character bigram information can help resolve competing multiple matches and choose one that is optimal as measured by character bigram constraints. Another minor change was made with the criterion for choosing the current cipher word to solve at any point. In addition to being the most frequent cipher word, the further criterion used for choosing the current cipher word is its length. Longer cipher words are prefered both because they have better chance of including deciphered characters to provide further constraints, and also because longer cipher words usually have fewer matches in the dictionary. The use of character bigram information and selecting the long cipher words are needed to start the algorithm. These constraints also make the algorithm run faster.

The 10 documents that were also used to generate test images for the modified algorithm were input to the NONMATCHES algorithm. For each document, a few "helping words" were added to the end of the original text. They helped to provide the needed cover set constraints to get the deciphering started. They also helped decipher some of the rare characters like capital characters or characters such as 'j' or 'z.' These "helping words" were added out of necessity. They are also limited in number, usually 3 to 6 for documents of about 100 words. Their use is also reasonable and well-justified since they would probably occur in a longer document. An example of the text document is shown in Figure 4.14. The subject of the text is about the role of money in earlier societies. Four "helping words" were added to the end of the original text. The word "sat" is added to help decipher the word "so," "We" to decipher "What," "Idea" to decipher "In," and "Also" to decipher "An."

Table 4.1 shows the results of running the 10 test documents. The algorithm achieves 100% correct for 7 of the test documents and produced 1 word mistake with each of the other three documents. The three error cases are, "400000" being classified as "haaaaa," "DNA" being classified as "the," and "C"(as in "C programming language") being classified as "a."

What important function does money perform for society that makes its existence so enduring? Early civilizations discovered that the use of money allowed for division of labor and specialization. It had become apparent that a community's total output was greater if a individual could specialize in a few tasks rather than being the proverbial jack-of-all-trades and master of none. An individual specializing as a cobbler, for instance, could become more expert and productive if that person did not also have to function as a blacksmith. Thus some people would specialize as hunters, others as farmers, and so on. In general, people would engage in whatever tasks were necessary for the community's well-being. Specialization is not without problems, however. sat We Idea Also

Figure 4.14: An example of the text document with the added "helping words" at the end.

The first error is because the word "400000" is not included in the dictionary. The other two errors happened because of insufficient cover set constraints. If other "helping words" like "No" or "Can" were added, they will be correctly deciphered. Cases where capital characters are deciphered as their lowercase counterparts are not classified as errors.

The above performance of applying Nagy's deciphering approach to perfect clustering inputs is a baseline for comparison with the experimental results of applying the modified deciphering algorithm to document images.

Doc	# of words	mistakes	Correct Word Rate
T0	91	400000 → haaaaa	98.9%
T1	129	none	100%
T2	117	none	100%
T3	104	DNA → the	99.0%
T4	88	none	100%
T5	112	C → a	99.1%
T6	130	none	100%
T7	135	none	100%
T8	122	none	100%
T9	100	none	100%

Table 4.1: Results of the original NONMATCHES algorithm on text inputs.

4.6.3 Results and Analysis

Table 4.2 shows the results of applying the modified character-level deciphering algorithm on the documents in Set A. Column *# of Words* indicates the total number of words in each input document. Column *# of Touching Patterns* lists the number of touching patterns included in each document. The number of touching patterns correctly deciphered by the algorithm is listed in Column *# of Touching Patterns Correctly Recognized*. The overall performance of the modified algorithm in terms of the percentage of words correctly recognized is listed in Column *Correct Word Rate*. For a comparison with Nagy's original algorithm, upon which our modifications are based, the projected performance of Nagy's original algorithm on the input documents is also listed in Column *Projected Correct Word Rate of Nagy's Algorithm*. The percentages of correct word rates in this column are calculated with the assumption that the original algorithm cannot handle touching character patterns, and thus would produce incorrect recognition results for all the words that include any touching patterns. The results show that our modifications to the original algorithm bring about 6.9 percent increase in word recognition rate.

Results of applying the modified character-level deciphering algorithm on test documents in Set B are shown in Table 4.3. As we can see, the main difference from the statistics in Table 4.2 is that extensively more touching patterns occur in documents in Set B. This causes further deterioration in Nagy's original algorithm which cannot handle touching patterns. Again, our modifications to the original algorithm bring about 35.4 percent increase in word recognition rate for the set of scanned documents.

The above experimental results show how the original algorithm deteriorate when facing degraded documents. They also show how our modifications, designed to handle touching characters and clustering mistakes, can bring back the performance to an acceptable level.

Not reflected in the above tables are the important details of how different touching patterns are treated as either short touching patterns and solved by straight-forward deciphering, or as long touching patterns and solved differently by recursive patterns matching using the learned font information. Also, details on how the algorithm detects and reverses clustering mistakes and how this contributes to the overall performance are not shown in the above tables. These important details are illustrated with some examples.

For the artificially generated document in Figure 4.12, all words are correctly deciphered

Doc	# of Words	# of Touching Patterns	# of Touching Patterns Correctly Recognized	Projected Correct Word Rate of Nagy's Algorithm	Correct Word Rate
A0	91	12	12	90.1%	96.7%
A1	129	18	16	91.5%	97.6%
A2	117	15	15	90.1%	97.3%
A3	104	11	11	90.4%	96.1%
A4	88	9	8	92.0%	95.2%
A5	112	13	13	89.3%	97.1%
A6	130	17	16	90.0%	98.1%
A7	135	25	22	86.7%	97.2%
A8	122	19	17	90.2%	96.4%
A9	100	15	13	88.0%	95.9%

Table 4.2: Test results for documents in Set A.

Doc	# of Words	# of Touching Patterns	# of Touching Patterns Correctly Recognized	Projected Correct Word Rate of Nagy's Algorithm	Correct Word Rate
B0	91	42	38	59.3%	96.7%
B1	129	56	53	61.2%	96.1%
B2	117	50	47	61.5%	97.0%
B3	104	43	39	63.5%	96.3%
B4	88	39	35	59.1%	96.6%
B5	112	41	39	68.8%	96.9%
B6	130	55	52	63.1%	95.8%
B7	135	64	60	58.5%	97.2%
B8	122	60	56	58.2%	95.3%
B9	100	51	46	56.0%	95.4%

Table 4.3: Test results for documents in Set B.

except three. The first is the word "400,000," which is not included in the dictionary. The second word is "Forces," which was recognized as "forces" because the word "Forces" was not included in the dictionary. The third incorrectly recognized word was "Education," which was recognized by the algorithm as "fducation." A character clustering mistake was responsible for this error. The character 'E' was mistakenly put in the same cluster as the character 'F' during the clustering stage. Therefore, when the word "Forces" was first recognized as "forces," the character 'E' in "Education" was also assigned an 'f' because of the clustering error. When the word "Education" was deciphered, no dictionary match could be found because the character 'E' had been assigned an 'f' already. This triggered the detection of possible clustering mistakes and the character confusion information was then consulted to explore alternative assignments. Because the character confusion set we used does not include any confusing characters for 'f,' this did not help. If the word "Forces" had correctly been recognized, 'F' and thus the 'E' in "Education" would have been assigned an 'F.' And the character confusion information with 'E' and 'F' would have helped to find the correct assignment to character 'E' and reverse the mistake.

All the single characters and touching patterns were correctly deciphered. Three of the long touching patterns ('perf' and 'ect' in "perfect," and 'arts' in "parts") were correctly solved by using font information learned from the decryption of the single characters and short touching patterns. The other long touching pattern, 'aniza' in the cipher word "organization," was solved by dictionary matching using constraints from within the cipher word itself.

For the scanned image in Figure 4.13, all but three words were correctly deciphered. One is the cipher word "training," which includes one deciphered touching pattern 'ra,' as well as two un-deciphered touching patterns 'in.' Because of clustering mistakes, 'ra' was incorrectly put in a cluster with many 'm's and assigned an 'm' when the cluster was deciphered. Since this rare case of confusing patterns was not covered by the character confusion set heuristics we used, the mistake was not detected and reversed. A similar situation happens with another unsolved word "Forces"(its first occurrence in Figure 4.13), where the deciphered touching pattern 'rc' is put in the same cluster as 're,' and the combination makes the 'F' indecipherable. The third unsolved word is the number "400,000" which was not included in the dictionary. Except for these three words, all the other words and touching patterns were correctly deciphered. Also, all the other clustering mistakes were successfully detected and corrected. Two examples of detecting and reversing clustering mistakes are shown in Figure 4.15 and Figure 4.16.

Results with the rest of the test documents in the two sets show similar patterns. In general, the modified character-level deciphering algorithm was able to decipher most of the single characters very reliably, which provides important cover-set constraints for the deciphering to proceed especially at the initial stage of the deciphering process. This was achieved by the original NONMATCHES algorithm. Our modifications extend the deciphering approach from dealing with only single characters to being capable of handling multiple touching characters. The modifications were shown to work well with most of the short touching patterns and some of the long touching patterns where constraints from the dictionary or from the cover-set were enough to uniquely decide their identities. For those long touching patterns that reside in cipher words that have little constraints from the dictionary or from the cover-set, they will be postponed to be solved by visual pattern matching using learned font information. Clustering mistakes were detected when a cipher-word finds no match in the dictionary. The character confusion set heuristics were used to help re-assign cluster identities and reverse the clustering mistakes.

Problems happen when the character similarity information represented by the character confusion heuristics is not enough to cover some delicate cases. Image degradations usually cause performance deterioration in clustering algorithms that are based on image pattern similarity, which complicates this situation. Decisions on which characters are similar to each other and thus need to be included in the character confusion heuristics are usually made subjectively by humans, without considering the similarity as seen by the clustering algorithms, which could be very different from those seen by human eyes. Therefore, alternative means

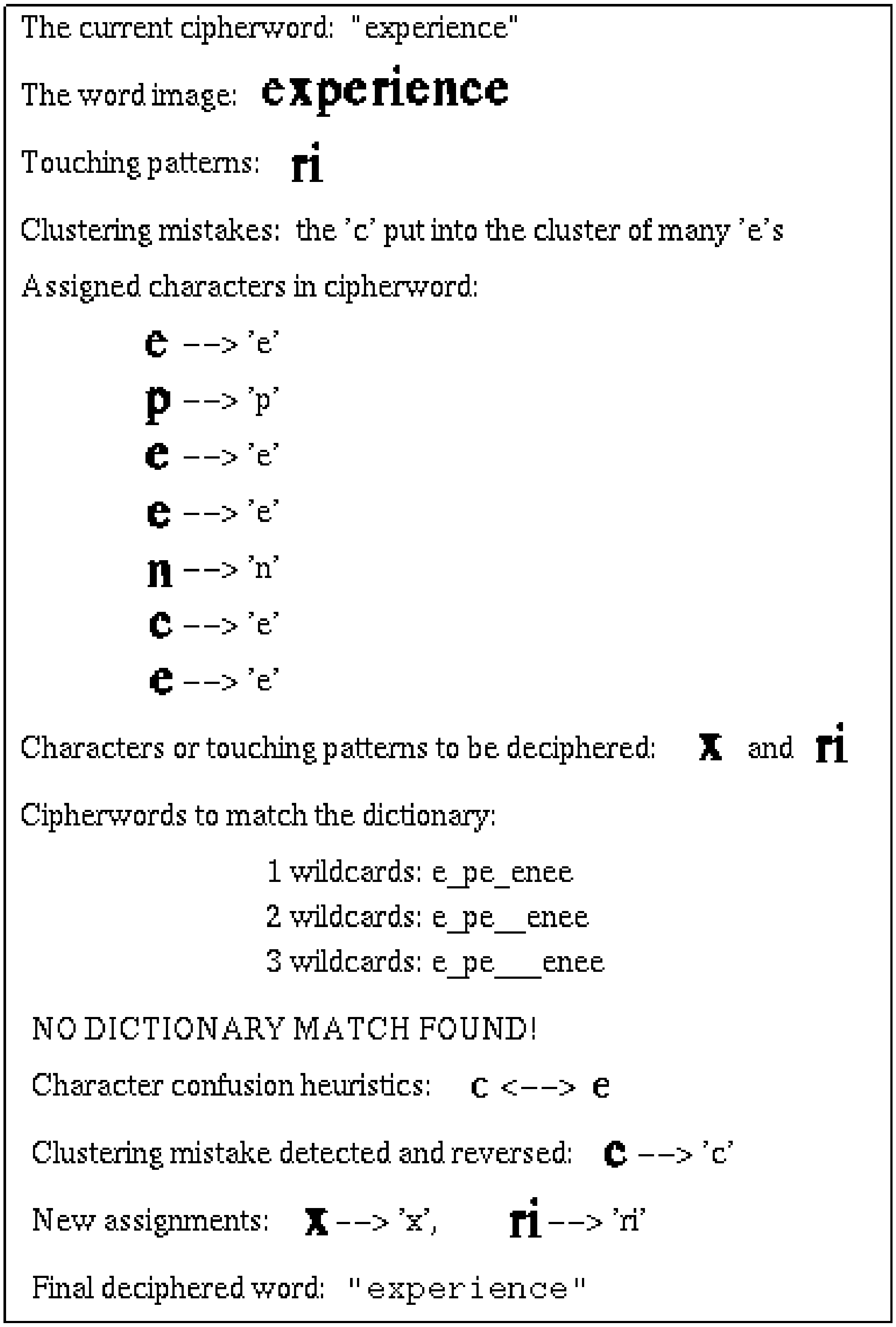


Figure 4.15: An example of detecting and reversing clustering mistakes.

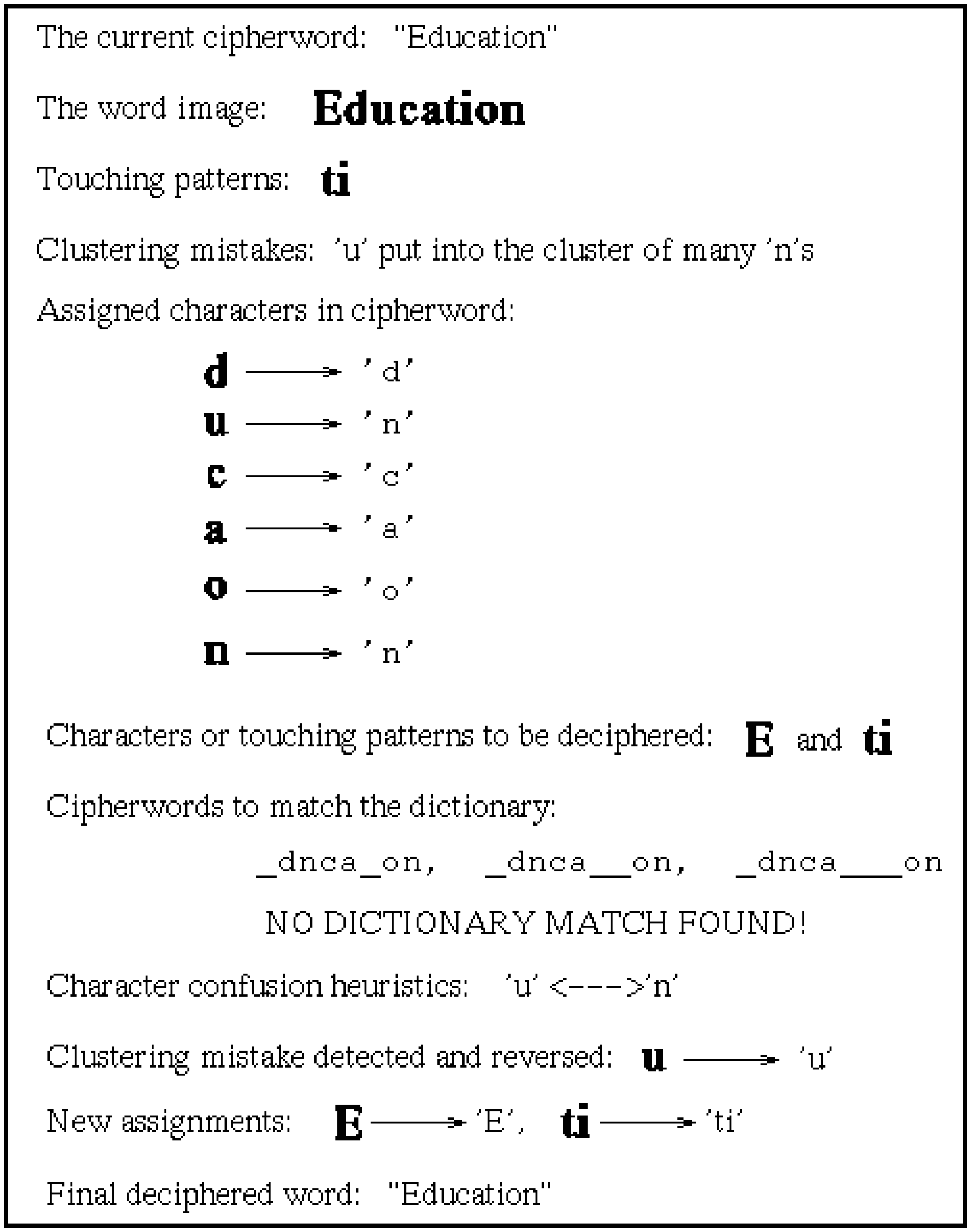


Figure 4.16: Another example of detecting and reversing clustering mistakes.

of defining the confusion sets should be investigated.

The experiments were run on a networked SunSPARC 10 station. Time efficiency of the algorithm was measured with the system running the deciphering algorithm on average load. The average time to decipher an artificially generated document in Set A is 1.8 minutes, while the average time to decipher a scanned document in Set B is 2.1 minutes. Further analysis of the time spent on different stages of the algorithm also shows that on average over 80% percent of the total running time was spent on character pattern clustering. This percentage could increase further when a more sophisticated clustering algorithm is chosen, or when the algorithm is applied to larger or more degraded documents. Image degradation tends to cause more touching patterns as well as deformed individual character patterns in the input document. This increases the number of clusters derived from the input document image, which in turn increases the time needed for each incoming pattern to do pattern matching with all the cluster prototypes. The above difference in time needed to process documents in Set A and documents in Set B is due to more severe image degradations in documents in Set B.

4.7 Conclusions and Future Directions

A modified character-level deciphering algorithm for document recognition was presented that has the ability to decode single characters as well as touching characters caused by document degradation. The touching patterns that can be recognized by the modified algorithm are indicated by both the number of touching patterns in an input document, and the number of characters that could be included in a single long touching pattern. The modified algorithm was made tolerant to mistakes in the clustering stage by detecting and correcting clustering errors during the deciphering process. Visual constraints are systematically combined with language constraints to decipher touching patterns and to detect and reverse clustering errors. The deciphering algorithm was tested on both artificially created and scanned documents that include touching characters and clustering mistakes. Satisfactory results were achieved in both cases.

Due to the small image size and different varieties of character patterns, good results in character pattern clustering are sometimes difficult to achieve. Image degradations tend to aggravate this problem. Clustering mistakes usually become more severe with degraded documents and account for the majority of deciphering errors. The current clustering algorithm used is a leader clustering algorithm, which is order dependent. One direction to increase the clustering performance is to experiment with ordering the connected components differently so that optimal clustering performance can be achieved. Also, more extensive and more delicate heuristics for representing character visual similarities are needed to handle clustering mistakes produced by a clustering algorithm. The visual similarity heuristics should represent not only visual similarities between single characters, but also the visual similarities between single characters and touching patterns(i.e., 'd' vs. "cl"), as well as visual similarities between different touching patterns(i.e., "tr" vs. "ti").

Research in spelling correction and OCR post-processing algorithms can provide useful hints about the construction of character confusion heuristics [94, 161, 86]. Most of the techniques used for text correction are based on studies of characteristics of spelling errors made by human beings [51], and therefore are not directly applicable to the correction of character identities achieved by character pattern clustering and deciphering. However, some techniques that were especially designed for the correction of OCR errors might be helpful. These techniques usually assume that most of the errors are caused by the substitution of one character for another [81, 136], and both rule-based and probabilistic approaches have been applied to represent character confusion characteristics. Usually a training phase is used to estimate a probabilistic model for character confusion by supplying an OCR device with sample texts and tabulating error statistics. The same training technique can be applied to construct character confusion heuristics that characterize typical pattern clustering algorithms. The result is expected to be a more accurate and genuine reflection of character similarities as seen by the clustering algorithms themselves, rather than the ones currently being used. Moreover, statistical contextual information such as character n-gram or word n-gram statistics can be used to select the optimal choice in case consultation with the character confusion heuristics produces multiple legal corrections of characters. We expect that further research in these directions will substantially increase the power of the algorithm in handling clustering mistakes during the deciphering process, which will be vital for achieving reliable deciphering performance especially when severe image degradation is involved.

In addition to enhancing the ability of handling clustering mistakes during the deciphering process, another area that can be improved is the clustering algorithm itself so that fewer clustering mistakes will be produced and fed to the deciphering stage. Based on the current clustering algorithm, clustering accuracy can be improved by using a structural feature representation for individual character and touching patterns. Because of small size, rich similarities and the degradations that can occur in character images, structural feature representations are usually ineffective in differentiating similar characters when used alone. But they could be very useful in differentiating similar character patterns when the pixel representation produces conflicting results. We have experimented with devising a hierarchical feature representation of characters based on psychological studies of feature detection and representation in human preattentive-attentive vision and attentive vision [166, 165, 131, 169], as well as on neural physiological studies of features detection and representation in primate visual cortex [72, 71, 10, 18, 37]. This hierarchical feature representation is computed from the skeleton representation of a character. It includes local features such as isolated spots, line segments, curve segments, line(or curve) terminators and line crossings, as well as global features such as holes. Figure 4.17 shows two examples of the hierarchical feature representations for two characters. The global features tend to remain stable across fonts and image degradation, and are therefore important in differentiating certain character patterns when most of the pixel or local feature representations fail. During pattern matching in the clustering algorithm, according to the particular pattern of image degradation in the input document, different weights can be assigned to the different feature dimensions to stress the significance of particular global or local features in the overall match. Combining this structural feature representation with the current image pixel representation can help achieve better matching results in cases where image pixel matching alone produces competing and confusing results. Examples are 'e' and 'c,' where the hole in the 'e' differentiates it from similar 'c's, and 'u' and 'n' that are differentiated from each other by the locations of line terminators.

Improvements in the clustering algorithm that are intended to achieve higher clustering accuracy also tend to increase the time needed for character clustering, especially with large input documents. A possible fix for this problem is to split the input page into multiple vertical segments, performing clustering separately on each of the segment and finally combining the results together.

It is noteworthy that the modified character-level deciphering algorithm discussed in this chapter is not able to handle the other difficult problem caused by image degradation, i.e.,

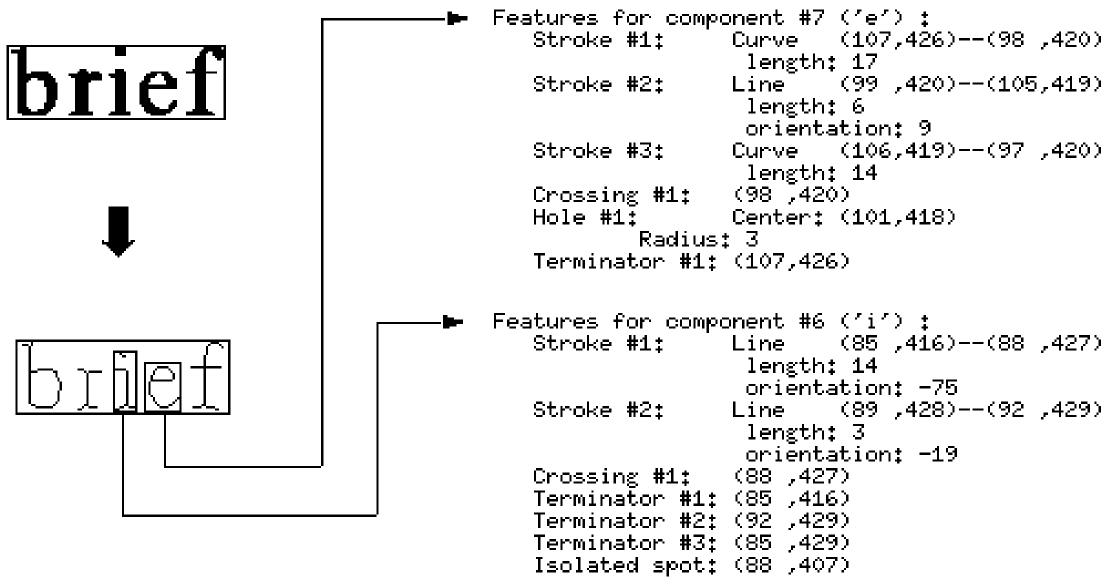


Figure 4.17: Examples of hierarchical feature representations for characters.

the problem of character fragmentation in which characters are split into multiple connected components. Combinations of different splitting possibilities for wide characters like 'm' or 'w' and the relationship of a connected component with its left and right neighboring fragments produce too complicated a situation that can be handled within a character-level deciphering approach using dictionary matching. The ultimate solution to character fragmentation is to decode the words directly, without first solving each character. This is the subject of the word-level deciphering approach presented in detail in the following chapters.

Chapter 5

Word-level Deciphering As Probabilistic Relaxation

Substitution cipher decoding has been proposed as a font-independent approach to OCR. So far most deciphering algorithms for OCR work on the character-level. Character patterns are first clustered into groups by sharing visual similarity. The original text is then translated into a cipher text with each character pattern replaced by the code that represent the cluster to which it belongs. The cipher codes are then assigned alphabetic identities such that the character repetition pattern in the input text matches the letter repetition pattern provided by a language model. The potential of using word-level language constraints in a deciphering framework to directly decode the identities of words is explored in this chapter. The intention is to solve the difficult problems of touching characters and character fragmentation. The major difficulty of using a deciphering approach on the word level is that the existence of relatively stable and reliable language constraints on the character level, such as character n-grams and a vocabulary of common words, usually does not scale up to the word level. A novel word-level deciphering approach is proposed in this chapter that first solves a selected portion of the input text by using a word-level relaxation deciphering algorithm. Font information is learned from the decrypted portion of the text and then used to re-recognize the rest of the text not available for cipher solution. Tests of the proposed approach on both artificially generated and scanned documents show satisfactory performance in presence of extensive image degradations.

5.1 Introduction

As had been pointed out earlier, current document recognition systems are usually heavily dependent on knowledge of many fonts, usually achieved through font training. Also, the existence of touching characters and character fragmentation caused by document image degradation has been an unsolved problem in document recognition systems based on character segmentation and character classification.

In Chapter 4 we discussed character level deciphering algorithms that have been proposed for OCR by solving a substitution cipher. The basic idea with using a deciphering algorithm for OCR lies in utilizing language statistics and assigning alphabetic labels to the cipher codes in a way that the character repetition pattern in the input text passage best matches the letter repetition pattern provided by a language model. Because character visual patterns in the input text passage are used only for forming distinct pattern groups rather than for direct character recognition, and the alphabetic identities of characters are decided solely on the basis of maximizing the correspondence of linguistic statistics between the cipher text and the language model, this approach to OCR is font-independent and no font training is needed.

However, character-level deciphering algorithms are also sensitive to touching characters and character fragmentation for their reliance on matching of character-level language constraints. The work presented in Chapter 4 is intented to improve the power of character-level deciphering algorithms in this respect. The modified character-level deciphering algorithm is enhanced to handle touching characters and is tolerant to clustering mistakes [45]. Nevertheless, these modifications are not able to solve the character fragmentation problem. Sensitivity to document image degradation still remains an issue for character-level deciphering algorithms.

A solution to the problem of touching characters and character fragmentation is to use word-level language constraints in a deciphering framework to directly decode the identities of words. Besides being a font-independent recognition approach, a word-level deciphering algorithm would be able to overcome sensitivity to touching characters and character fragmentation by identifying words directly instead of building up results from character recognition. Variation in visual word patterns caused by image degradation usually is not as serious a problem as in the case of character patterns in the character-level deciphering algorithms, and can usually be accounted for by a word pattern clustering algorithm.

However, until now little progress has been reported on using a deciphering algorithm on the word level for OCR. The main difficulty is that the stable and reliable language constraints that exist on the character level usually do not scale up to the word level. Word n-gram statistics are not as reliable as character n-gram statistics. Also, there does not seem to exist a set of "legitimate English sentences" to serve as constraints for word-level deciphering algorithms just as a dictionary does in a character-level deciphering algorithm.

The solution to this problem lies in the following observation. While it is true that there usually does not exist a content-independent and stable word n-gram repetition pattern for even extensive text passages, usually a portion of the words in the input text that are either subject-related key words or function words do repeat themselves in a way that they constitute relatively stable and reliable word bigrams that match relatively well with those provided by a matching language model. It is possible that this portion of the words can be recognized by a word-level deciphering algorithm using word-level linguistic constraints. Furthermore, from this recognized portion of words, font information can be learned, which can then be used to recognize the rest of the text with a more conventional OCR approach. Because of the higher occurring frequencies of the selected words in the input document, the visual images of the recognized words can provide a reliable source from which font information can be learned. It is possible that the learned font information may not cover all the character prototypes that occur in the input document, but this usually does not pose a serious problem to the following re-recognition stage that uses the learned font information to re-recognize the rest of the input text. The reasons for this are that more font information can be learned from the re-recognition stage, and also the most frequent 21 lower-case characters which are likely to be learned account for 96% of all text [93, 66].

Based on this observation, we propose a novel word-level deciphering algorithm that first solves a selected portion of the input text that has relatively more reliable word bigram statistics by using a word-level relaxation deciphering algorithm. Font information is learned from this decrypted portion of text and then used for precise re-recognition of the rest of the text. This results in a word-level deciphering approach to document recognition that overcomes the need for font training. Furthermore, sensitivity to touching characters and character fragmentation is accounted for by identifying words directly using word-level language constraints rather than building up the recognition entirely from character-level processing. This approach provides a recognition technique that has the ability to automatically adapt to a given input document and achieves high recognition performance in the presence of image degradation.

The rest of this chapter contains three sections. Section 2 discusses the details of each component of the proposed word-level deciphering algorithm. Section 3 presents and discusses the experimental results of applying this algorithm to degraded documents. Finally, conclusions and future directions for research are pointed out in section 4.

5.2 The Proposed Approach

The proposed word-level deciphering approach and its major components are shown in Figure 5.1.

At the pre-processing stages, a word segmentation and a word pattern clustering algorithm are first applied to the input text document to generate clusters of equivalent words. The ideal clustering result is that all the same words share a common cluster and different words are grouped into different clusters. Depending on the quality of the clustering algorithm being used, clustering mistakes may occur. As in the case of character image clustering, the main reason for word clustering mistakes are the visual similarities between different words(e.g., "bits" vs. "hits," "these" vs. "those," "barking" vs. "banking"). Image degradation in the input documents tends to cause more errors of this nature. Nevertheless, because word images are usually much larger in size than character images, and their similarity measures are often based on more global and stable properties, mistakes in word image clustering are usually less of a problem than mistakes in character image clustering. The proposed word-level deciphering approach does check for possible word clustering mistakes in the font learning stage where incoming characters from the decrypted words are compared with prototypes, so that mismatches caused by word clustering can be detected and clustering errors reversed.

After the word image clustering stage, each cluster is assigned an unique ID number. Each word in the text is replaced by the ID number of the cluster it belongs to, resulting in a text of cipher-words. The rest of the word-level deciphering approach in Figure 5.1 decrypts this cipher text using word-level linguistic constraints.

Frequencies of cipher-word bigrams can be measured by counting the number of occurrences of unique two-cipher-word sequences in the text. Cipher-word bigrams with relatively

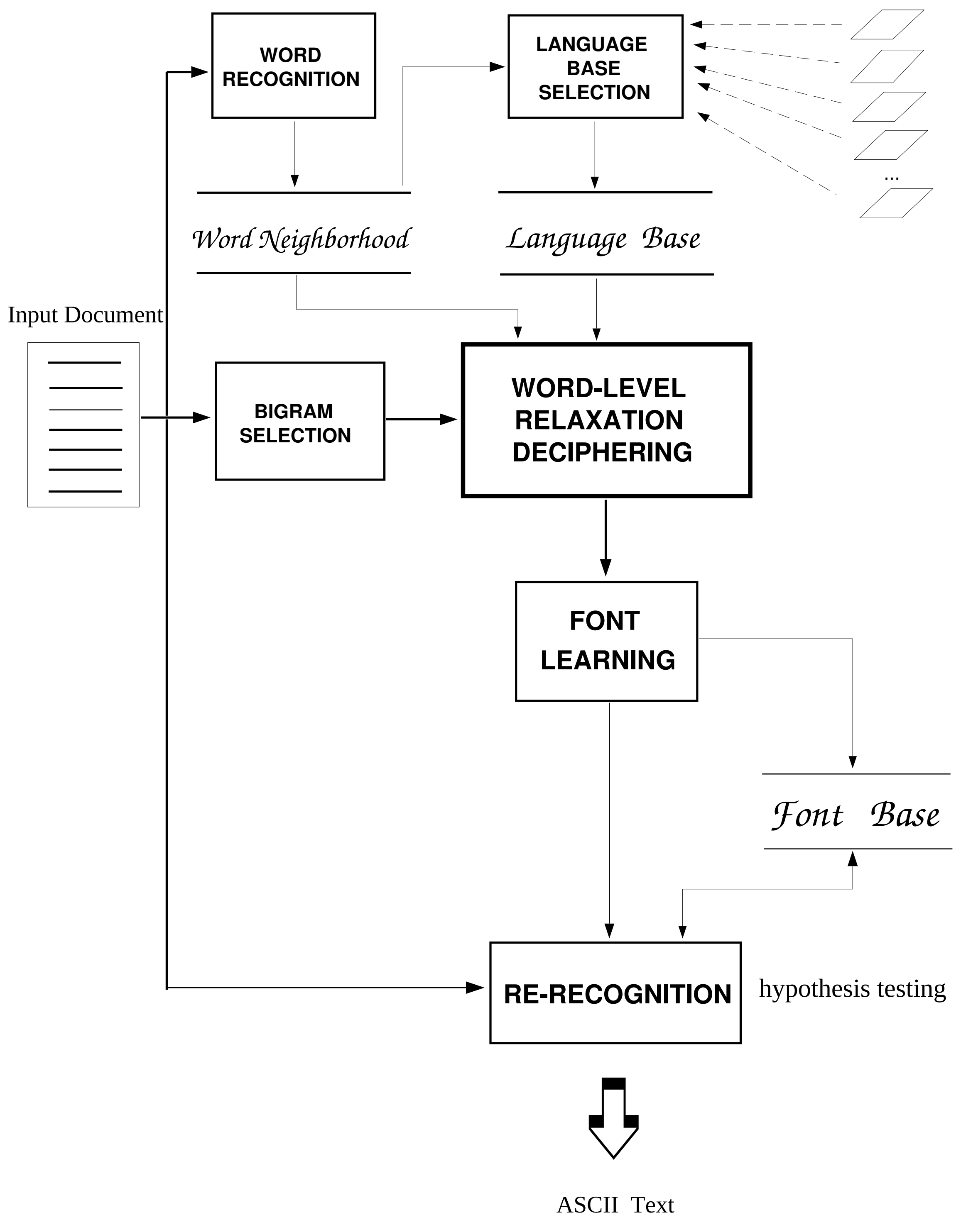


Figure 5.1: Word-level deciphering approach.

high frequency in the text are selected for deciphering using the word-level relaxation algorithm.

Two sources of information are used by the word-level relaxation algorithm. One of them is the set of candidate neighborhoods generated for each word by a preliminary word recognizer. These preliminary candidates may not be accurate or reliable enough on their own to suggest the real identities of the words, especially when severe image degradations are involved, but they provide reliable constraints for the relaxation algorithm to proceed.

Another source of information used by the relaxation algorithm are the word bigram statistics compiled from a language database which matches the input text. The matching language database is acquired by a vocabulary matching algorithm from a database of pre-classified text documents based on the sets of word candidate neighborhoods generated for each word in the input document [77, 78]. This vocabulary matching algorithm is executed on the sets of word candidate neighborhoods of an input document. Similarities between the input document and each document in the database are determined by using a modified *vector space model* [147] for information retrieval. The top N documents in the database that are most similar to the input document are selected to form the matching language database of the input document, from which the word bigram statistics is compiled and used by the word-level probabilistic relaxation algorithm. Figure 5.2 shows the top 50 most frequent word bigrams of the matching language database generated using the above vocabulary matching algorithm for an input document on a recent US-China trade dispute.

The word-level deciphering stage solves the selected portion of the text by using the above two sources of constraints within a probabilistic relaxation algorithm. Font information is learned from the decrypted portion of the text and used to re-recognize the rest of the text.

The following sub-sections contain detailed descriptions for each of the major components in the proposed approach.

1. of the
2. in the
3. United States
4. the United
5. in China
6. intellectual property
7. trade war
8. to the
9. on the
10. human rights
11. to be
12. with the
13. and the
14. in a
15. he said
16. the US
17. by the
18. more than
19. has been
20. the Chinese
21. for the
22. property rights
23. as saying
24. from the
25. a trade
26. said the
27. China has
28. China and
29. of its
30. of a
31. last year
32. in Beijing
33. at the
34. and other
35. said The
36. that the
37. said China
38. protection of
39. would be
40. of intellectual
41. Trade Representative
42. will be
43. the talks
44. it is
45. China to
46. $1 billion
47. the trade
48. the first
49. avert a
50. as a

Figure 5.2: The top 50 most frequent word bigrams of the matching language database generated using the vocabulary matching algorithm for an input document on US-China trade dispute.

5.2.1 Word Pattern Clustering

A word pattern clustering algorithm was developed to group visually similar word patterns into distinct clusters. Figure 5.3 shows the outline of the word pattern clustering algorithm being used.

The algorithm scans across the document page, text line by text line, looking for word image patterns. The first cluster is formed by putting the first word pattern into that cluster as the seed member. Each of the following word patterns is matched against the prototypes of all the existing clusters. If the best match value exceeds a threshold, the word pattern is considered another member of the cluster of best match and added to that cluster. The prototype for that cluster is also updated accordingly. If the best match value is less than the threshold, a new cluster is created with the current word pattern as its seed member. This process continues until all the word patterns in the input document are clustered.

Pattern matching of an incoming word image sample with the cluster prototypes is carried out in a way similar to that used for character patterns described in Chapter 4. It is based on measurement of the maximum percentage of overlapping pixels between the incoming word image pattern and the cluster prototype. Because of much larger image sizes, word patterns usually produce fewer matching errors than character patterns. But clustering mistakes still exist, especially among long words that differ only in one or two characters. Figure 5.4 shows an example of word clustering mistakes. Clustering mistakes like this can be detected and corrected by the "self-correction" technique in the font learning stage discussed later.

After word pattern clustering, each cluster is represented by a cluster identity code. By replacing the word patterns in the original document with the corresponding cluster identity codes, the original document is transformed into a cipher text with each cipher code representing a word identity.

Algorithm **Word_Pattern_Clustering**(*Set of Word Patterns*)

begin

    create a new cluster, its only member being the first word pattern;

    $Num\_Clusters = 1$;

    for (the rest of all the word patterns)

        match current word pattern with each of the existing cluster;

        find the best match;

        if (best match value above threshold)

            add current word pattern to cluster of best match;

            update cluster prototype;

        else

            create a new cluster, its only member being the current word pattern;

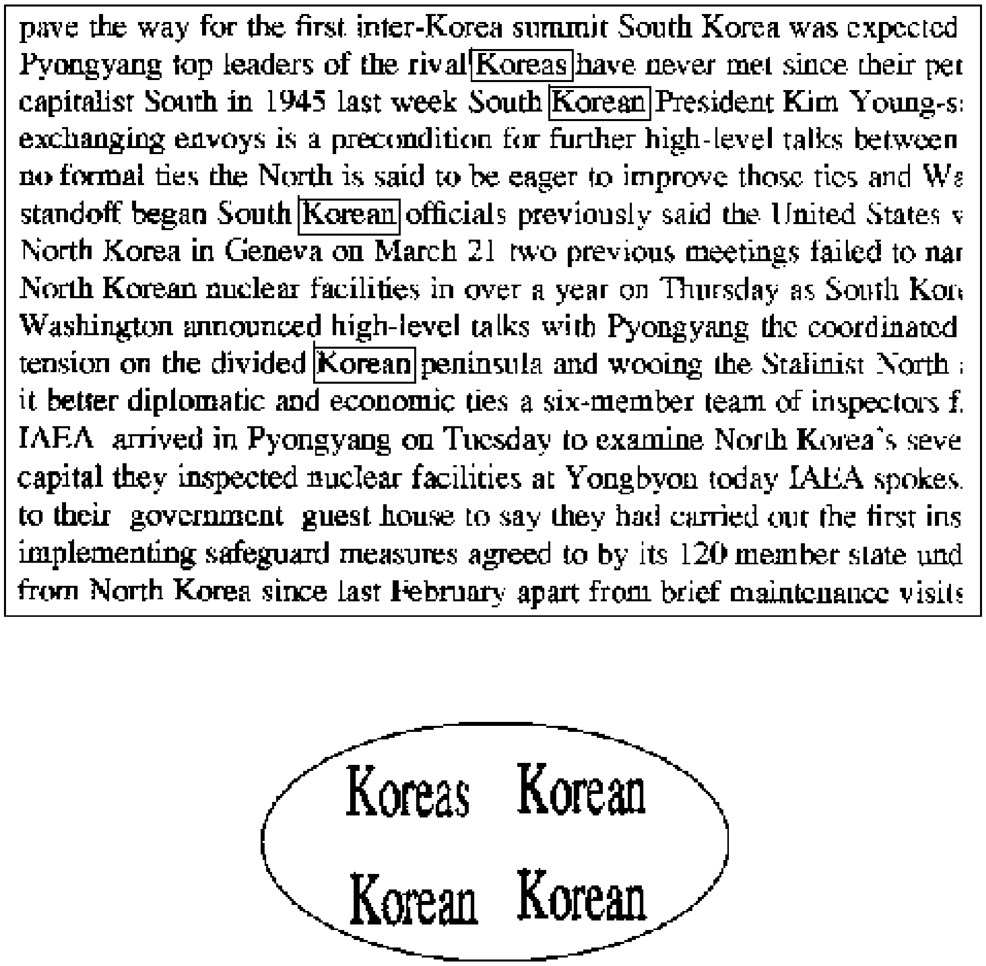
            $Num\_Clusters = Num\_Clusters + 1$;

        end-if

    end-for

end-algorithm

Figure 5.3: Outline of the word pattern clustering algorithm.



pave the way for the first inter-Korea summit South Korea was expected
Pyongyang top leaders of the rival Koreas have never met since their pe
capitalist South in 1945 last week South Korean President Kim Young-s
exchanging envoys is a precondition for further high-level talks between
no formal ties the North is said to be eager to improve those ties and Wa
standoff began South Korean officials previously said the United States v
North Korea in Geneva on March 21 two previous meetings failed to na
North Korean nuclear facilities in over a year on Thursday as South Kor
Washington announced high-level talks with Pyongyang the coordinated
tension on the divided Korean peninsula and wooing the Stalinist North
it better diplomatic and economic ties a six-member team of inspectors f
IAEA arrived in Pyongyang on Tuesday to examine North Korea's seve
capital they inspected nuclear facilities at Yongbyon today IAEA spokes
to their government guest house to say they had carried out the first ins
implementing safeguard measures agreed to by its 120 member state und
from North Korea since last February apart from brief maintenance visits

Figure 5.4: An example of errors in word pattern clustering. One "Koreas" being put into a cluster of many "Korean."

5.2.2 Selecting a portion of the text with reliable word bigram statistics

From the cipher text of the input document, the number of occurrences of each two-cipher-word sequence (cipher-word bigram) occurring in the text is calculated. Those cipher-word bigrams that have relatively higher bigram frequencies are selected from the text to be solved by the relaxation deciphering algorithm. For the present time a manually chosen threshold of bigram frequency is used for the selection. The portion of cipher-words selected this way usually consists of two categories of words: function words like "the," "of," "and" etc. as well as subject-related key words.

The choice of the threshold value affects the following deciphering stage. Generally speaking, higher threshold values will result in fewer, more reliable cipher-word bigrams being selected; whereas lower threshold values result in more, but less reliable bigrams being selected. Therefore, setting the threshold value too high will cause only a limited number of words being decrypted and the individual characters learned into the font base, an undesirable condition for the re-recognition stage which depends on using learned font information to recognize the rest of the text.

Some of the selected cipher-word bigrams will not match the word bigram statistics of the matching language base and the relaxation algorithm will fail to converge to a solution for these cases. Thus it is likely that not all the selected cipher-words will be solved by the relaxation deciphering algorithm. These unsolved cipher-words, together with the rest of the words in the input document, will be solved by the following re-recognition stage using the learned font information from the decrypted words.

5.2.3 Probabilistic relaxation deciphering algorithm

As part of the pre-processing in the proposed word-level relaxation labeling algorithm, each word pattern is transformed into a cipher code and assigned probabilities for representing plain words provided by a preliminary word recognizer. A word-level probabilistic relaxation labeling algorithm is used to update the probabilities for each word using the word bigram constraints from the matching language base.

5.2.3.1 Probabilistic relaxation and its application in OCR

Probabilistic relaxation algorithms were first applied to the problem of scene labeling in computer vision and image understanding. Usually a scene involves a set of objects, each of which can be individually identified by image classification algorithms. But the identifications of the individual objects are usually ambiguous due to image degradation or pattern complexities in real-world objects. As a result, each object in the scene are frequently associated with multiple labels representing different classification results. An early example of scene labeling was the interpretation of line segments in a line drawing of a set of polyhedra. This problem domain was also investigated in [33, 73, 171]. The ambiguities in individual object identification are difficult to resolve if each object is considered in isolation. However, the relationships among the objects can be helpful in resolving the ambiguities and achieving a scene labeling that is consistent with the scene context. One of the mathematical models that exploits contextual relationships to disambiguate the identifications of individual objects is probabilistic relaxation.

Probabilistic relaxation algorithms are iterative parallel classification algorithms where every element tries to estimate its class membership probabilities based on those of its neighbors. The process is iterated until a satisfactory classification is achieved. Detailed discussion of probabilistic relaxation algorithms can be found in [144]. The initial membership probabilities can usually be provided by preliminary recognition of each individual object.

Because of its ability of exploiting contextual constraints, probabilistic relaxation algorithms have also shown their effectiveness in OCR. Suppose that a character recognition device has scanned a string of characters $a_1, ..., a_n$, and identified them probabilistically as being various letters of the alphabet $\lambda_1, ..., \lambda_n$; let $p_i(\lambda_j)$ be the probability that character a_i is letter λ_j. We could then impose constraints on these probabilities based on the observed frequencies with which characters co-occurr. For example, labels T and H on a pair of consecutive characters a_i, a_{i+1} might reinforce one another, while the labels T and X would have a weakening effect on each other. And the relaxation process can be applied so that a recognition result representing the most context-compatible interpretation can be achieved.

5.2.3.2 Adaptation to the word-level

The relaxation updating scheme in our word-level deciphering algorithm is based on Goshtasby's relaxation procedure on the character level [62]. We changed the updating scheme so that it works on the word level by using word transition probabilities within each word bigram. The relaxation updating process is applied to each of the selected cipher-word bigrams. Iterating the updating scheme results in improved estimates of word identities that finally lead to the decryption of the selected portion of text. The core idea of the relaxation process here is to use information about the labels of highly confident words and combine it with the bigram constraints from a matching language base to correct or improve the labels of less confident words.

Figure 5.5 shows the outline of the word-level relaxation labeling algorithm.

The updating scheme of word probabilities is

$$P_i^{(k+1)}(\lambda) = \frac{P_i^{(k)}(\lambda) \cdot q_i^{(k)}(\lambda)}{\sum_\lambda [P_i^{(k)}(\lambda) \cdot q_i^{(k)}(\lambda)]}$$

where $P_i^{(k)}(\lambda)$ is the probability that cipher word i has label λ after the $(k+1)th$ iteration based on the probability that it had label λ after the kth iteration and the support it received from the labels of the neighboring words. $\gamma_{ij}(\lambda, \lambda')$ is the word transition probability, and $q_i^{(k)}(\lambda) = \sum_j \delta_{ij}\gamma_{ij}(\lambda, \lambda')P_j^{(k)}(\lambda')$ is the *contribution factor* which represents the impact to

```
Algorithm Word_Level_Relaxation(Selected_Bigram_Set)
BEGIN
    get word transition probabilities from language base
    get initial candidate neighborhood for each word
    FOR (each word bigram in Selected_Bigram_Set)
        REPEAT
            FOR (each candidate in word_0)
                compute the contribution factor
                update the probability of candidate
            FOR (each candidate in word_1)
                compute the contribution factor
                update the probability of candidate
        UNTIL(converge)
END
```

Figure 5.5: Outline of word level relaxation labeling algorithm.

its neighbors if a cipher word is assigned a certain label. λ' is the label that has the maximum $P_i^{(k)}(\lambda)$, and δ_{ij} represents weights assigned to different neighbors.

The convergence criterion of the probabilistic relaxation is satisfied when either of the following two cases happens. If the word bigrams and their associated candidate neighborhoods fail to be covered by the language model(reflected by a zero value of $\sum_{\lambda}[P_i^{(k)}(\lambda) \cdot q_i^{(k)}(\lambda)]$), the relaxation stops and the cipher-word bigram fails to be recognized; On the other hand, if for both cipher-words, one of the candidate probabilities shows dominance, reflected by a single candidate probability exceeding a threshold value of, for example, 0.95, the bigram is considered decrypted and each of the words is assigned the identity of the corresponding dominant candidates. This threshold value may be determined experimentally.

5.2.4 Font learning

For an input document, a portion of the cipher-words are recognized as a result of applying the word-level relaxation deciphering stage. Font information of the input text is learned and will be used for recognizing the rest of the words not available for cipher solution.

Font learning is implemented by segmenting the images of each deciphered word and constructing the character prototypes. The segmentation of word images into individual character images is guided by the known word identities which are already available from the word deciphering stage. The segmentation step here is different from the character segmentation task in most of the segmentation-based OCR methods where clues for word identities are not available to constrain or verify the many character segmentation hypotheses.

The representation for each character prototype includes its visual image as well as information about its height, width and etc. The segmented image of each character in the recognized word is used to update the character prototype representation. To average out image distortions and disparities in individual characters and get a good representation of the prototype image, each incoming character image first undergoes a pre-processing stage to get rid of possible surrounding white gaps or fragmented strokes from neighboring characters caused by character segmentation. Then an approach similar to a weighted averaging is used to form the new prototype image. Each incoming character image is first matched with the current prototype image, and is added to the prototype image at the optimal matching position where the two overlap the most to form the new prototype. The prototype image thus formed will have a highly weighted "central" portion that represents the stable image of a clean character, as well as a relatively lowly weighted peripheral portion that represents noise and degradations.

During the font learning process, in case an incoming character image pattern turns out to be radically different by a certain similarity measure from the current character prototype, the font learning algorithm will go on matching this character image with prototypes of all the other learned characters in the font base. If no acceptable match is found, this indicates a possible deformed incoming character image, probably resulted from a mistake made in character segmentation (such as part of the top portion of 'f' is cut away to form a deformed 'a' with the following 'a' in the word "facilities"). In this case, the deformed incoming character is simply discarded and excluded from the font learning process. This selective font learning guarantees that a badly deformed incoming character sample will not be accepted to update the font base and fonts learned are of good quality and consistency. If, on the other hand, the incoming character matches well with another character prototype in the font base, this indicates a possible mistake in the word relaxation deciphering stage or in the word clustering stage (such as a single "Koreas" clusters within many instances of "Korean," which results in this "Koreas" being assigned a "Korean" later after the cluster is decoded as "Korean" by the word relaxation stage). In this case, the suspicious incoming character pattern is re-assigned a character identity of the best match if the new assignment conforms with a legal word identity, which also causes the identity of the word being re-assigned ("Korean"→"Koreas"). This implementation of font learning achieves "self-correction" of mistakes happened earlier in the clustering stage and the relaxation stage which are difficult to be detected and corrected through other means.

Selective learning and self-correction in font learning are shown in Figure 5.6 and Figure 5.7. They, together with the weighted averaging used in the formation of character prototypes, guarantee the quality of the learned font prototypes and contribute to the overall tolerance of the approach to image degradations.

Figure 5.8 shows the outline the important parts of the font learning algorithm discussed above.

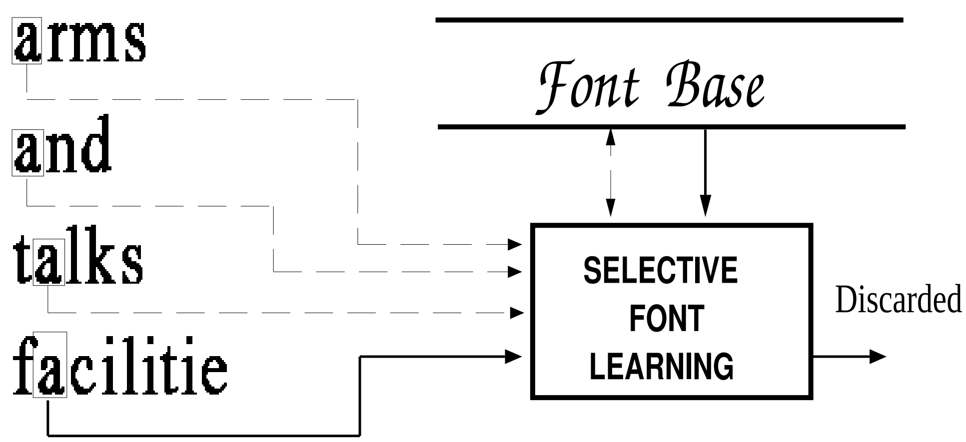


Figure 5.6: Selective font learning.

5.2.5 Re-recognition using learned font information

A hypothesis testing approach using an uniform cost search algorithm is proposed to re-recognize the rest of the words in the input text that were not decrypted by the word-level deciphering stage. The following is a brief description of this algorithm. Details of this algorithm will be discussed in Chapter 6.

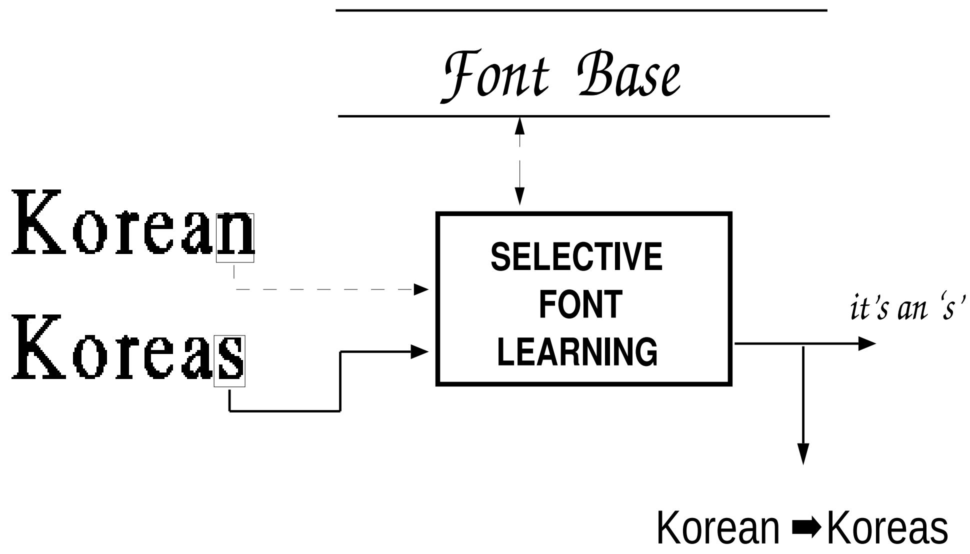


Figure 5.7: Self-correction in font learning.

As shown in Figure 5.9, the information that is available to the hypothesis testing algorithm are the word candidate neighborhood associated with each word, as well as the learned partial font base. For each word to be recognized, the algorithm starts by examining the two character image patterns at both ends of the word image, matching them with font prototypes from the learned partial font base. The match is constrained by the candidate neighborhood of this word, only prototypes of characters that could appear at both ends are to be matched with character images at both ends. Each legal combination of matches at both ends results in a updated partial word decision. Illegal character combinations are discarded. Each updated partial word decision is associated with a cost that measures the updated confidence value of this partial word decision. Each partial word decision is also associated with a reduced word candidate neighborhood that conforms to the accumulated partial decisions of this word made so far. The process of matching character images at both ends with font prototypes proceeds inwards until an unique decision can be made about the

```
Algorithm Font_Learning(Recognized_words, Font_base)
begin
    initialize Font_base;
    for (all word images from Recognized_words)
        do character segmentation;
        for (all segmented characters)
            pre-process character image;
            get prototype of character from Font_base;
            match character image against prototype image;
            if(match poorly)
                match with all other prototypes;
                if(no acceptable match)
                    discard this character image from learning;
                else
                    find the optimal match;
                    form new prototype image by weighted averaging;
                    re-assign word identity;
                    reverse possible word clustering mistakes;
                end-if
            end-if
            form new prototype image by weighted averaging;
        end-for
    end-for
end-algorithm
```

Figure 5.8: Outline of the font learning algorithm.

word identity.

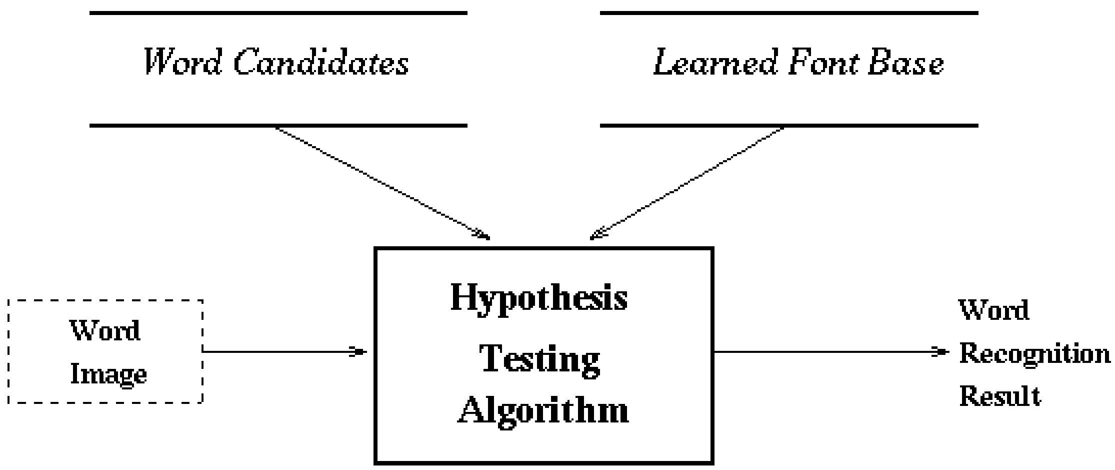


Figure 5.9: Re-recognition using an hypothesis testing algorithm.

The cost associated with each partial word decision f consists of two parts. The first part is the accumulated cost g of the series of character matches that had already happened as the matching proceeded inwards. The second part is an estimate of the potential cost h to reach a complete word decision. They basically represent the degree and the projected degree of image pattern mismatches.

Each partial word decision is saved into a open queue with its associated cost f and the reduced word candidate neighborhood. The algorithm chooses the node from the open queue that has the optimal cost f for further expansion, until a complete word decision is reached or until the open queue becomes empty, which indicates a failure of recognition.

For a partial word decision, the estimated cost h is assigned a value of 0 for most cases where there is no obvious indication of mismatch between the remaining portion of the word image and the remaining characters of its associated candidate set. However, h is assigned a large value if it becomes clear that the remaining portion of the word image doesn't match any of the candidates in the reduced candidate set.

The decision about whether the remaining portion of a word image matches its associated set of candidates is based on visual characteristics such as the overall size of the pattern, the existence of descenders and ascenders, and the existence of isolated dots at the top. The reason for choosing these visual constraints is that they are relatively stable under various image degradations.

This hypothesis testing method compensates for local character deformations by relying on an uniform cost search algorithm that seeks optimal word decision. Multiple visual constraints that are relatively stable across image degradations are used to estimate the potential cost h, which enables the algorithm to use as much as possible the visual heuristic power to rule out unlikely candidates and speed up the search.

5.2.6 Partial font base and font re-learning

It is likely that in the font learning stage, font information of some of the characters that appear in the text will not be learned because the relaxation deciphering stage only recognizes a portion of the text. This will cause difficulty in the re-recognition stage when words that include characters whose font information is not available are to be recognized. This problem is accounted for by using general geometric heuristics of characters to estimate the match. A second phase of font learning during re-recognition also helps to alleviate this problem.

In case a character is matched against the current partial word image and the character prototype is not available from the learned font base, the degree of match is estimated by comparing known visual features of the character with visual features in the targeted portions of the image pattern. This estimate is not as accurate as when the prototype is available. But this inaccuracy can usually be balanced by averaging the estimation with the more accurate character matching results at other character positions.

Prototypes of characters not available in the original font base can be learned when words that include these characters are reliably recognized by the hypothesis testing algorithm. If a word image is recognized by the hypothesis testing algorithm using only the available character prototypes, the word recognition is considered reliable, and the new character prototypes are added to the font base and used for the recognition of other words. A possible case for this to happen is when a character whose prototype is not available is positioned at the center of an input word, and this word is recognized by the hypothesis testing algorithm with the available font information before the unknown character is reached for character matching.

5.3 Experimental Results And Discussion

The proposed word-level deciphering algorithm was tested on both artificially created documents and scanned documents with varying image degradations. In the following, detailed experimental results of applying the proposed word-level deciphering algorithm to an artificially created document as well as to a scanned document will be presented first, followed by results of applying the proposed algorithm to a more extensive set of scanned documents.

5.3.1 Artificially Created Document

The document text used to test the proposed word-level deciphering algorithm consists of two pieces of news reports on the Korean Peninsula nuclear crisis collected from Clarinet. Punctuation was stripped. The text contains totally 921 words and 380 unique words. The document image was generated artificially using a text formatting package. Random noise that simulates image degradation was added to the clean image. As a result of the image degradation introduced, the preliminary word recognizer we used to generate the top ten neighborhood [68] only got a 79% correct rate for the top choice, the correct rate for top ten choices was 99.6%. A portion of the degraded document image is shown in Figure 5.10.

The matching language base that provides word bigram constraints consists of 31 recent news reports on the Korean Peninsula nuclear crisis which was generated by a vocabulary matching system [77] from a comprehensive language corpus acquired from Clarinet. The word candidate neighborhoods generated by the preliminary word recognizer were used by the vocabulary matching system to correctly locate the language base that matches the input text in content. The language base thus acquired consists of 12595 words and 2464 unique words. The word bigram set constructed from it consists of 23907 different word bigrams,

South Korea on Thursday reopened talks with rival
with the United States the talks timed to coincide w
part of elaborate steps aimed at resolving the yearlc
border village of Panmunjom the first in four mont
North's seven declared nuclear sites under a deal st
officials of the Vienna-based International Atomic
IAEA officials said the inspections which will take
diverted to weapons development in the past year
since February last year heightening suspicion that
series of talks with the United States last week the
military exercises with the South and resume high-
Ministry announced that this year's joint U.S.-Sout
with nuclear inspections North Korea has denounce
Thursday's border meeting the first since last Nove
pave the way for the first inter-Korea summit Sout
Pyongyang top leaders of the rival Koreas have ne
capitalist South in 1945 last week South Korean Pr
exchanging envoys is a precondition for further hig
no formal ties the North is said to be eager to impr
standoff began South Korean officials previously sa
North Korea in Geneva on March 21 two previous
North Korean nuclear facilities in over a year on T
Washington announced high-level talks with Pyong
tension on the divided Korean peninsula and wooir

Figure 5.10: A portion of an artificially created test document used for the word-level relaxation deciphering algorithm.

from which the bigrams with frequencies greater than a threshold are selected as the word bigram database. The frequencies of this selected set of more reliable and stable word bigrams are then normalized to word transition probabilities and used by the word-level relaxation deciphering algorithm.

301 out of the total 921 words are correctly deciphered as the result of the word-level relaxation deciphering stage. These include 24 unique words, as shown in Figure 5.11. Ten of them are function words and the other 14 are key words. This limited number of function words and key words constitute about one third of the whole input text, and the proposed word-level deciphering algorithm is able to recognize them with high reliability.

Korea	United	high-level	said
Korean	and	it	talks
North	at	last	the
South	facilities	nuclear	week
States	first	of	with
Thursday	from	on	would

Figure 5.11: Words recognized by the word-level deciphering stage.

From the recognized portion of text, the font learning algorithm is able to learn the prototypes of 25 of the 32 characters that appeared in the input text document. The results is shown in Figure 5.12.

For the re-recognition stage using the learned font information, only eight unique words

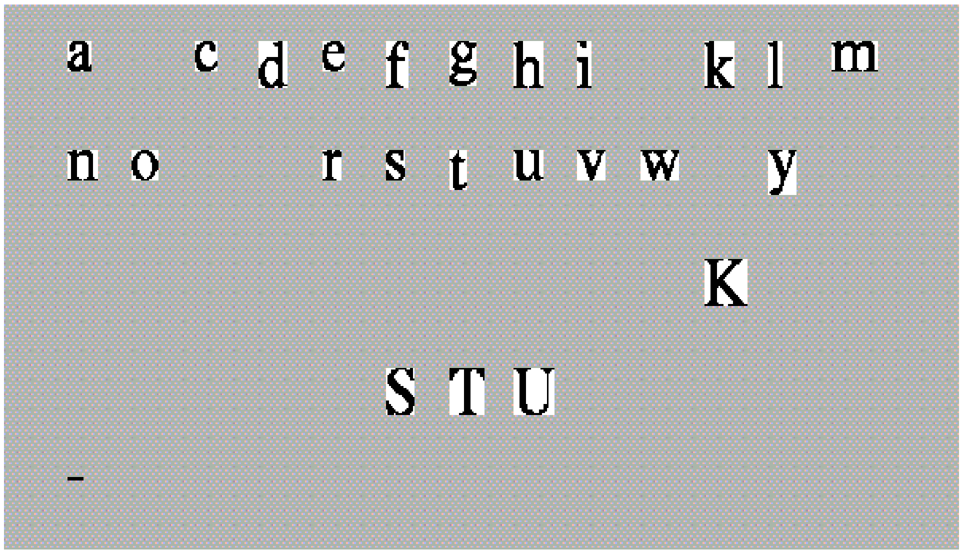


Figure 5.12: Font learned from the deciphered portion of words.

failed to be recognized correctly. The eight cases where the algorithm failed are: "be" being recognized as "he," "an" being recognized as "au," "21" failed to be correctly recognized because digits had not been recognized previously and their font information was not available, and "Hans" was recognized as "Kaus." Four other words failed to be correctly recognized because the correct candidates were not included in the initial top ten candidate neighborhood.

Some interesting observations about the hypothesis testing algorithm can be drawn from these errors. In case the word to be recognized is short, such as the words "be" and "an," the hypothesis testing algorithm seems to lose its ability to compensate for local mismatches and deformations by relying on global characteristics. This makes the word decisions heavily dependent on local character pattern matchings that are unreliable especially in presence of image degradations. Short words combined with unknown character prototypes aggravate this problem, as is shown in the case of the word "Hans."

The second phase of font learning in the re-recognition stage was able to learn 4 more character prototypes from the re-recognition results that are reliable. The updated font base is shown in Figure 5.13.

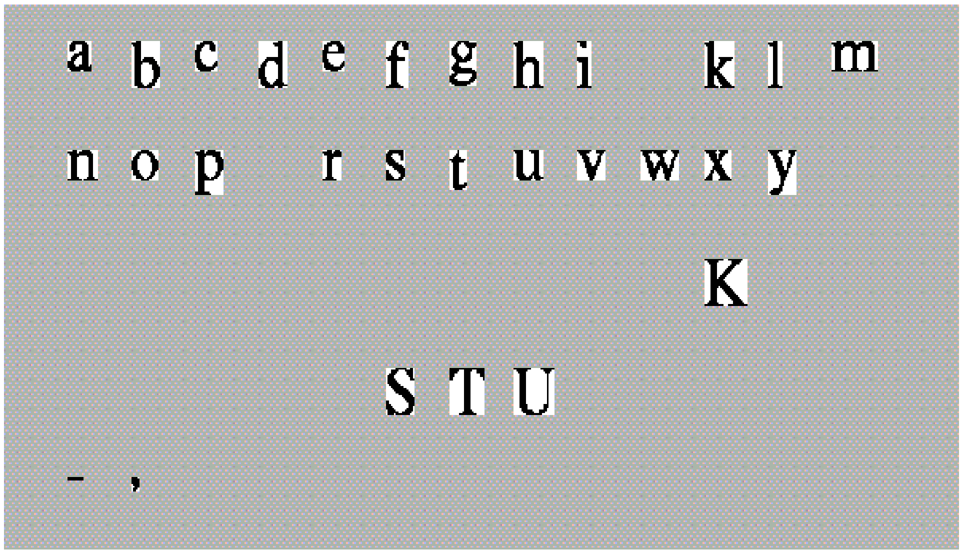


Figure 5.13: Updated font base after font re-learning in re-recognition stage.

The eight unique word mistakes resulted in 13 words that failed to be correctly recognized. The proposed word-level relaxation deciphering approach as a whole was able to achieve a correct word recognition rate of 98.6% for the artificially created degraded document.

5.3.2 Scanned Document

The text of the document was acquired the same way as discussed in the previous section for the artificially created document. The document image was first generated with the `ditroff` text formatting package in an 11*pt*. Times Roman font. The clean image was printed on plain paper using a laser printer and scanned as gray-scale image at 300 ppi resolution. The final binary image is the result of thresholding the scanned gray-scale image by choosing a threshold such that the character distortions and touching characters it created test the ability of the algorithm to handle these degradations. An enlarged portion of the final binary document image used for the test is shown in Figure 5.14.

The matching language base and the word bigrams are generated in the same way as presented in the previous section.

More severe image degradation in the scanned document caused poorer word clustering performance, which resulted in only 258 words being correctly recognized by the relaxation deciphering stage. These 258 words consist of 21 unique words, among them, 12 are key words and 9 are function words.

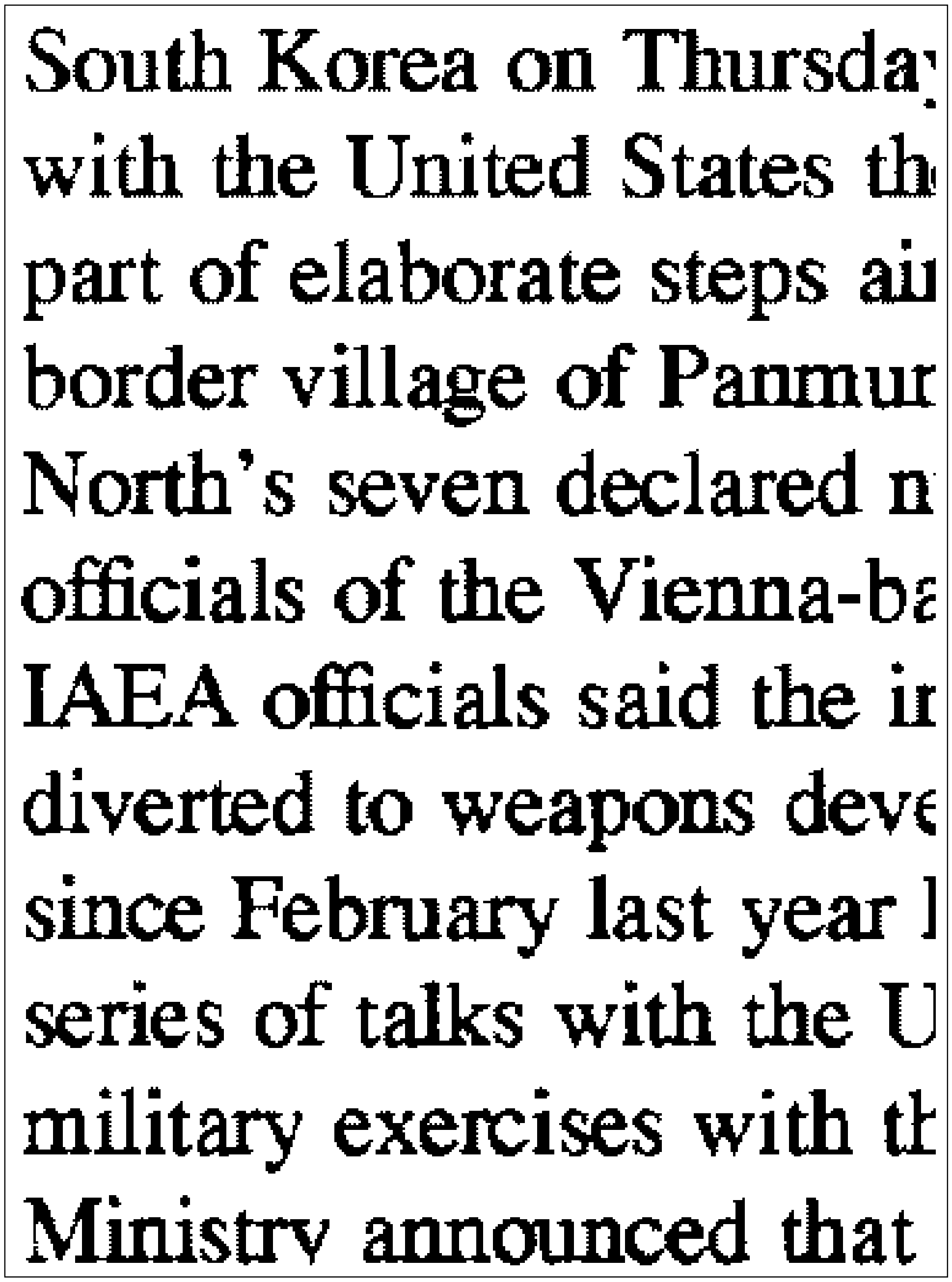
From the decrypted words, the font learning stage was able to learn 26 out of the 32 character font prototypes.

The re-recognition stage using the learned font information failed to recognize only four unique words. The four cases where the re-recognition failed are: "an" being recognized as "au," "21" failed to be correctly recognized because digits had not been recognized previously and their font information was not available, "Hans" was recognized as "Kaus," and "sufficient" was recognized as "sediment."

The errors are caused by similar reasons as in the case of the artificially created document discussed in the last section. Also, typesetting in fonts sometimes makes it difficult to deal with the problem where an 'i' following an 'f' is mistaken as part of the 'f' pattern and is completely cut off when the 'f' is recognized, as shown in Figure 5.15. This calls for more sophisticated use of character typeface and alignment information in the re-recognition algorithm.

Reliable recognition of some of the words also enabled the font re-learning phase to learn three more character and punctuation prototypes.

The four unique word mistakes in the re-recognition stage resulted in totally 8 words that were incorrectly recognized. The proposed word-level relaxation deciphering approach as a



South Korea on Thursda
with the United States th
part of elaborate steps ai
border village of Panmur
North's seven declared n
officials of the Vienna-ba
IAEA officials said the ir
diverted to weapons deve
since February last year l
series of talks with the U
military exercises with th
Ministry announced that

Figure 5.14: An enlarged portion of the scanned test document.

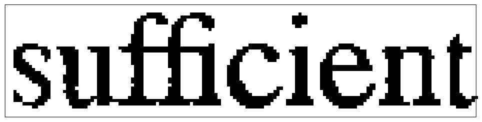


Figure 5.15: An ‘i’ following an ‘f’ is mistaken as part of the ‘f.’

whole was able to achieve a correct word recognition rate of 99.1% for the scanned document.

5.3.3 More Extensive Tests on Scanned Documents

We scanned more documents on which the proposed word-level deciphering algorithm was tested. This set of test documents includes totally 15 scanned documents of varying sizes and different degrees of image degradations. The main objective of using this set of scanned documents was to test the repeatability of the algorithm's performance on real-world scanned documents.

The 15 test documents were recent news reports extracted from the Clarinet newsgroup. They fall into three subject groups. Group 1 includes four documents related to the trade disputes between China and the United States. Group 2 includes five documents about the war in the Balkans. Group 3 consists of six documents on conflicts in Russia and the civil war in Chechnya. Punctuation was stripped from all the document texts.

The numbers of words in each document range from 885 to 973. Because of its reliance on matching the repetition pattern of visual words with the word bigram statistics in the language base, it is not surprising that word-level deciphering algorithm usually doesn't work well on short documents. It is able to produce reasonable recognition results only with documents long enough to provide reliable word bigram statistics. We did experiment with applying the word-level deciphering approach to input documents of varying sizes. Preliminary results have suggested that the proposed word-level deciphering approach starts to produce reasonable relaxation results with input documents of roughly 500 to 600 words. And usually the more words included in the input document, the better the word-level relaxation results will be. Nevertheless, the following results are only with input documents of similar sizes within a particular size range.

To generate the document images, the Unix utility `ditroff` was first used to generate the artificial images in 11*pt*. Times Roman font. The clean images were printed on plain paper using a laser printer and scanned as gray-scale images at 300 ppi resolution, which were then thresholded to generate the final binary document images. Both the scanning and the binarization process introduced image degradations that involve deformation in individual characters, touching characters as well as character fragmentation. An enlarged portion of one sample of the degraded document image from each of the three subject groups are shown in Figure 5.16, Figure 5.17 and Figure 5.18.

Three matching language databases that provide word bigram constraints for each of the subject group were constructed separately using a vocabulary matching algorithm [77, 78]. Each constructed language database consists of recent news reports from the corresponding Clarinet newsgroups. The test documents are not included in the document database from which the language bases were compiled. The sets of word bigram were then generated for each language base, from which the word bigrams with frequencies greater than a threshold were selected as the word bigram database. The frequencies of the word bigrams were then normalized to word transition probabilities and used by the relaxation deciphering algorithm. Table 5.1 shows some relevant statistics for each of the language bases compiled.

Table 5.2 summarizes the results of applying the word-level deciphering algorithm to the three groups of 15 test documents. Column *# of Words* indicates the total number of words in each document. Column *% Bigram Overlap* shows the percentage of the selected word bigrams that were solved by the relaxation stage. This percentage indicates the extent to

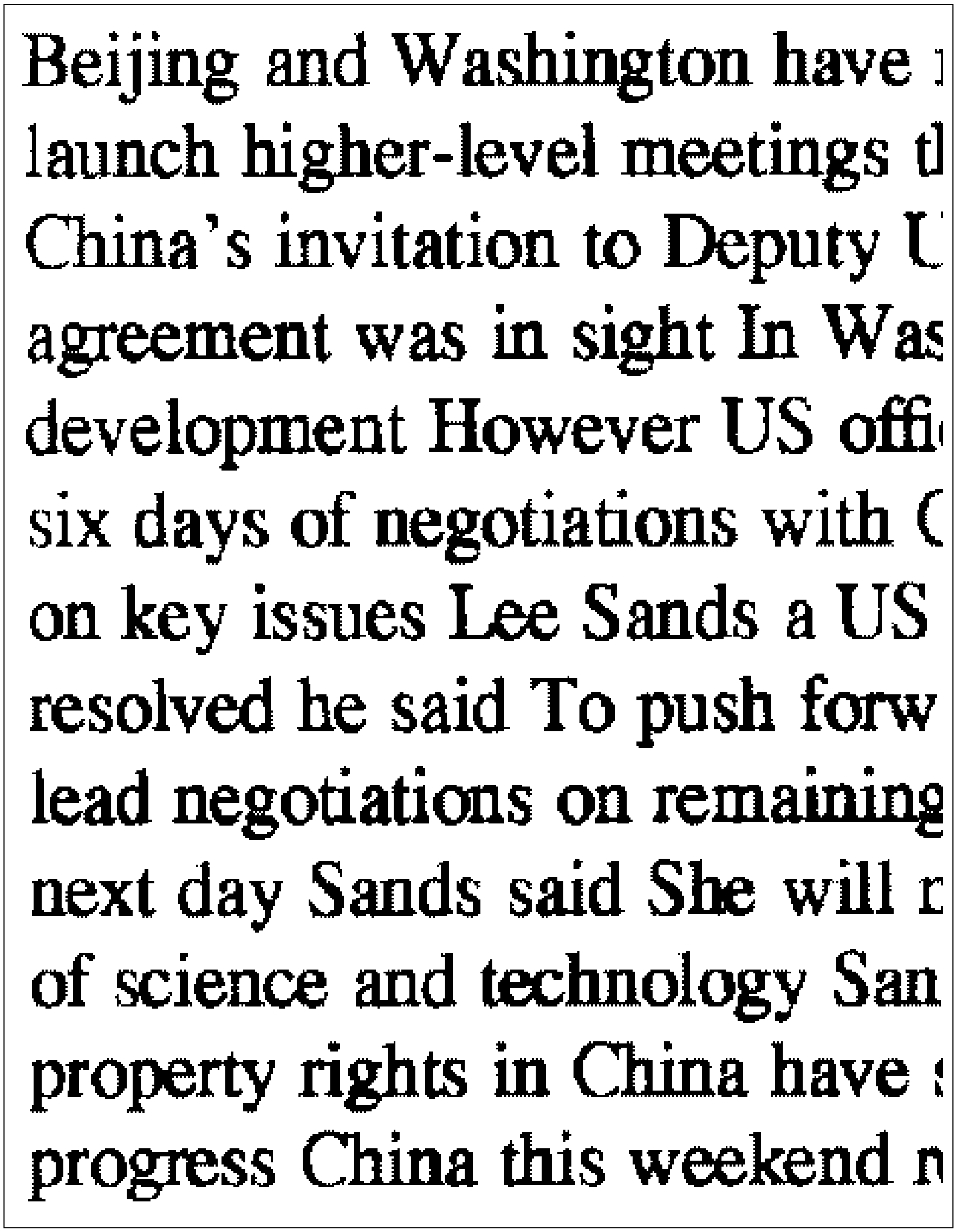

Beijing and Washington have r
launch higher-level meetings tl
China's invitation to Deputy U
agreement was in sight In Was
development However US offi
six days of negotiations with C
on key issues Lee Sands a US
resolved he said To push forw
lead negotiations on remaining
next day Sands said She will r
of science and technology San
property rights in China have s
progress China this weekend n

Figure 5.16: A portion of a scanned test document.

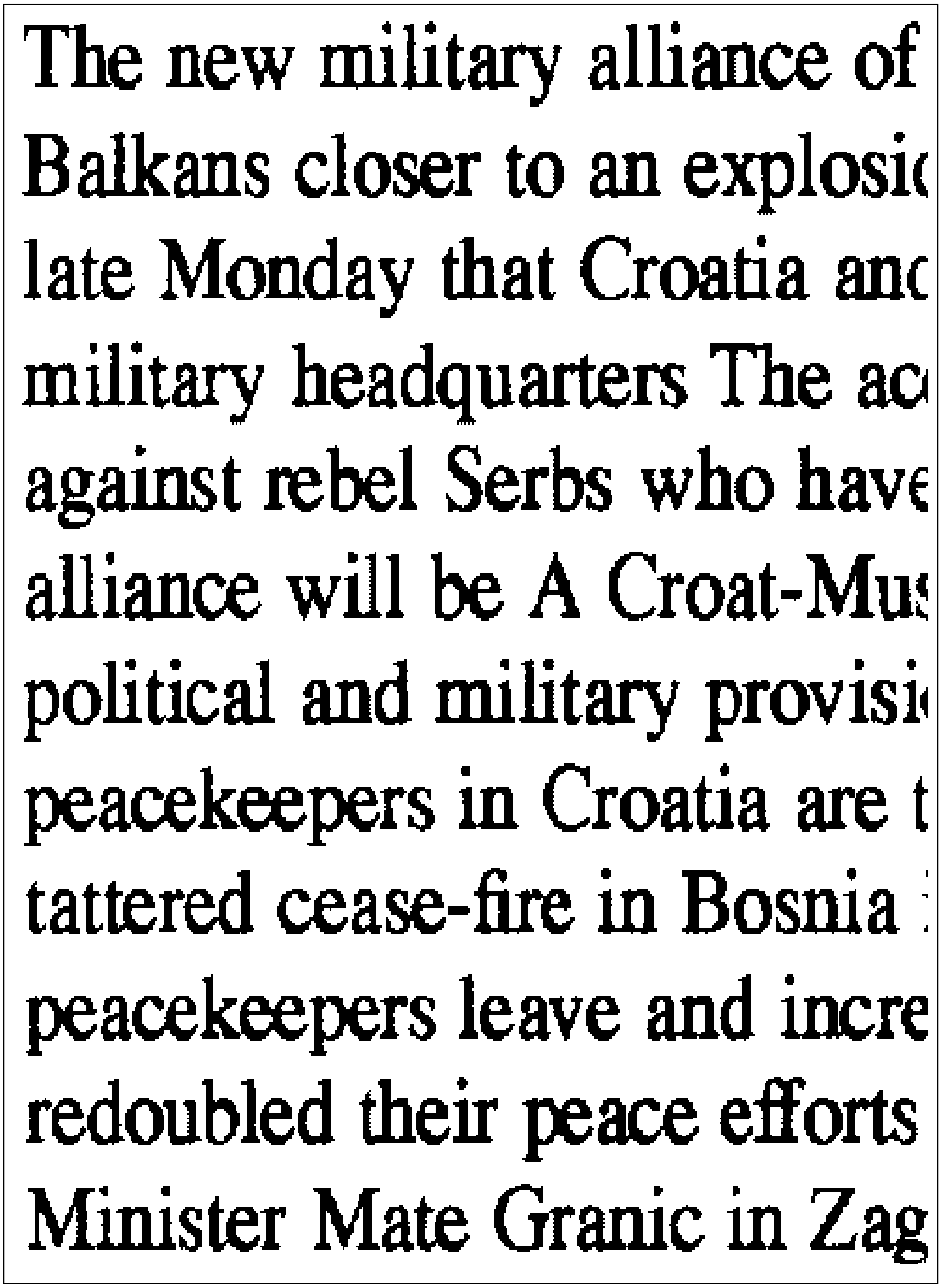
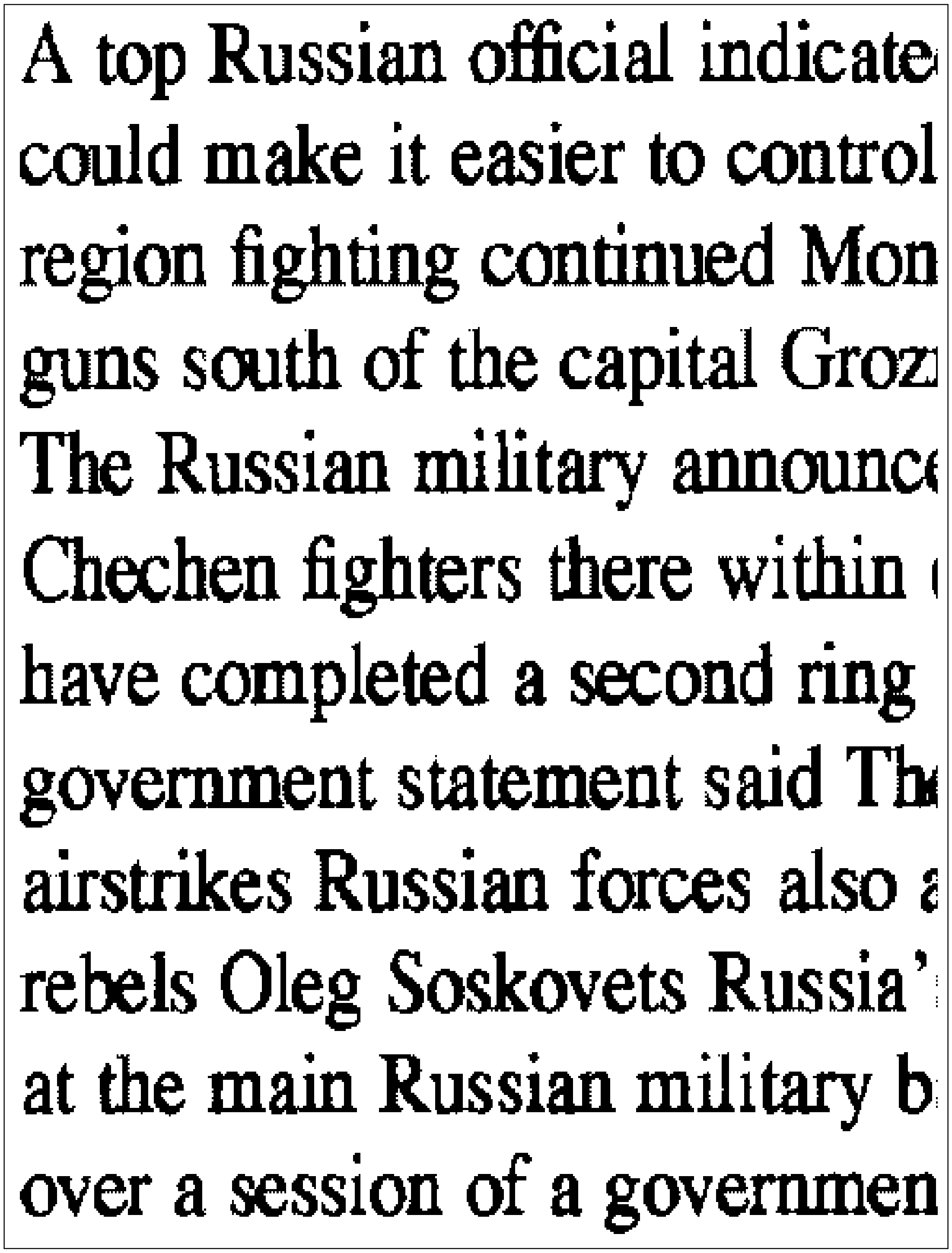

The new military alliance of
Balkans closer to an explosi
late Monday that Croatia an
military headquarters The ac
against rebel Serbs who hav
alliance will be A Croat-Mu
political and military provisi
peacekeepers in Croatia are t
tattered cease-fire in Bosnia
peacekeepers leave and incre
redoubled their peace efforts
Minister Mate Granic in Zag

Figure 5.17: A portion of a scanned test document.



A top Russian official indicate
could make it easier to control
region fighting continued Mon
guns south of the capital Groz
The Russian military announce
Chechen fighters there within
have completed a second ring
government statement said Th
airstrikes Russian forces also
rebels Oleg Soskovets Russia'
at the main Russian military b
over a session of a governmen

Figure 5.18: A portion of a scanned test document.

Subject Group	# of Words	# of Unique Words	# of Bigrams	Bigram Frequency Thresh.
China	18626	3544	36621	4
Balkans	25981	3971	46362	4
Russia	31388	5154	59514	4

Table 5.1: Statistics of language bases.

which the selected word bigrams resulted from choosing a certain bigram frequency threshold are represented by the matching language base. Column *W1* and Column *W2* stand for the number of words and unique words recognized by the deciphering stage. Column *C1* and *C2* stand for the number of characters learned after the deciphering stage and during the re-recognition stage. Column *# of Word Errors* stands for the total number of words failed to be correctly recognized. And Column *Correct Rate* represents the overall recognition rate in terms of the percentage of the total number of words in each document. For a comparison with the performance of the preliminary word recognizer used to provide the word candidates, correct word rate of the top choices produced by the preliminary word recognizer is listed in Column *P0*.

As shown by the experimental results in Table 5.2, the word-level deciphering stage is able to correctly recognize a portion of the input document which accounts for roughly 13% to 35% of all the words in the input document. Analysis of the document texts show that poor performances of the word-level deciphering stage are usually associated with input text that include many occurrences of names of persons or places, which constitute a large portion of the word bigrams selected to be decrypted by the word-level deciphering stage. Their occurrences usually are not well accounted for by the matching language model, which results in many of the selected word bigrams failed to be decrypted by the word-level deciphering stage. This is generally a problem when the word-level deciphering algorithm is applied to input documents with limited size. This problem will be less serious with input documents

Doc.	# of Words	% Bigram Overlap	W1	W2	C1	C2	# of Word Errors	P0	Correct Rate
C1(G1)	917	88.2%	206	19	25	11	13	77.8%	98.6%
C2(G1)	910	94.6%	263	25	25	10	10	76.4%	98.9%
C3(G1)	916	91.5%	184	16	22	12	13	81.1%	98.6%
C4(G1)	970	82.1%	121	11	17	16	31	75.7%	96.8%
B1(G2)	972	100%	298	40	34	4	21	81.9%	97.8%
B2(G2)	964	100%	335	46	28	8	17	77.3%	98.2%
B3(G2)	951	78.6%	184	12	17	16	15	74.9%	98.4%
B4(G2)	948	80.5%	218	14	18	12	23	79.7%	97.6%
B5(G2)	958	80.8%	147	12	16	14	30	82.0%	96.8%
R1(G3)	952	85.3%	220	16	21	13	11	77.2%	98.8%
R2(G3)	973	92.8%	241	23	27	8	15	74.2%	98.5%
R3(G3)	965	82.1%	198	15	19	11	25	78.6%	97.4%
R4(G3)	922	94.1%	233	29	29	4	17	82.5%	98.1%
R5(G3)	943	87.2%	199	17	23	12	13	80.1%	98.6%
R6(G3)	885	94.5%	294	36	29	11	15	75.2%	98.3%

Table 5.2: Word-level deciphering algorithm results on 15 test documents.

of larger sizes, where key words that are closely related to the subject as well as functional words constitute the majority of the word bigrams fed to the word-level deciphering stage. They are usually well represented by word bigram statistics from a matching language model.

For our experiments, the main reason for relatively poor overall recognition performance can be ascribed to input text that cannot be accounted for by the language model used. This often results in a very limited number of words being decrypted by the word-level deciphering stage, and only a few character prototypes being learned by the font learning algorithm following it, which in turn causes performance degradation in the re-recognition stage where the learned font information plays a crucial role. Errors in the re-recognition stage contribute to the majority of the word recognition mistakes, but not having enough font information from the word-level deciphering stage is the cause behind the scene. The font re-learning implemented inside the re-recognition stage has proven successful in alleviating this problem. Analysis of the experimental results shows that a large portion of the word errors in the re-recognition stage happened at the beginning stage of re-recognition before further font information could be learned from the recognized words.

Our experimental results have consistently shown, as predicted by the observation made at the beginning of this chapter, that for an extended input text, there does exist a portion of words with relatively reliable n-gram repetition statistics that can be recognized by a word-level deciphering approach using a matching language model. From this initial recognition, the system is able to learn font information of the input text and use it to recognize the rest of the words in the document.

The time factor of the word-level relaxation algorithm is shown in Table 5.3. In this table, the total running time of applying the word-level relaxation algorithm to an input document is broken down into five parts that correspond to the processing stages of word segmentation, word clustering, probabilistic relaxation, font learning, and word re-recognition. The total running time as well as the percentages of the total running time spent in each of the five processing stages are shown in each column. The results in this table are from running the word-level deciphering algorithm on the 15 scanned documents on a networked SunSPARC 10 station with average load. As can be observed, on average nearly 62 percent of the total running time is spent on word image clustering. Re-recognition using the learned font contributes to about 30 percent of the total running time. The other three processing stages of word image segmentation, probabilistic relaxation and font learning are relatively much less time-consuming.

Doc.	Total Time	Segmentation.	Clustering	Relaxation	Font Learning	Re-recognition
C1(G1)	4.4	5.1%	63.7%	2.3%	3.4%	25.5%
C2(G1)	4.7	4.8%	61.4%	2.5%	2.8%	28.5%
C3(G1)	4.5	4.9%	58.2%	1.9%	2.9%	32.1%
C4(G1)	5.2	5.0%	67.1%	1.8%	3.6%	22.5%
B1(G2)	5.1	5.1%	59.3%	2.1%	2.9%	30.6%
B2(G2)	5.2	4.7%	66.3%	1.9%	3.9%	23.2%
B3(G2)	4.7	4.2%	69.0%	1.7%	3.2%	21.9%
B4(G2)	4.8	4.9%	61.1%	1.5%	2.5%	30.0%
B5(G2)	5.5	5.2%	55.9%	2.1%	3.8%	33.0%
R1(G3)	5.1	5.1%	60.4%	2.3%	3.7%	28.5%
R2(G3)	5.7	4.1%	51.7%	2.1%	2.9%	39.2%
R3(G3)	5.2	4.3%	59.3%	2.0%	3.5%	30.9%
R4(G3)	5.1	5.2%	62.5%	1.7%	2.9%	27.7%
R5(G3)	4.9	4.9%	66.8%	1.9%	3.7%	22.7%
R6(G3)	4.4	4.7%	72.2%	2.1%	2.9%	18.1%
Average	5.0	4.8%	62.3%	2.0%	3.2%	27.6%

Table 5.3: Breakdown of running time of the word-level deciphering algorithm.

5.4 Conclusions and Future Directions

A novel word-level deciphering approach for document recognition is proposed in this chapter. It solves a selected portion of the input text with high reliability by using word-level language constraints within a relaxation deciphering framework. Font information is learned from the decrypted portion of the text and used to re-recognize the rest of the text that was not solved in the deciphering stage.

This word-level deciphering approach overcomes the need for font training by solving the equivalent of a substitution cipher and by learning font information dynamically from the input document. Sensitivity to touching characters and character fragmentation is reduced by decoding words directly without relying heavily on segmentation or recognition of individual characters, therefore opening up the potential for a true segmentation-free approach to document recognition. Test of the proposed approach on both artificially created and scanned documents showed reliable performance in the presence of severe image degradation.

Based on the current status of the proposed word-level deciphering algorithm, further experiments and research in the following areas are suggested.

More extensive tests of the proposed word-level deciphering approach on real scanned documents of varying sizes and with wider range of subjects are needed to get more accurate assessment of the applicability of this approach to real-word input documents. In particular, comparison study of the performance of the relaxation stage with input documents of similar subjects but different sizes will provide more quantitative assessment as to the effectiveness of the word-level deciphering approach on documents of different sizes. Also, the effects of choosing different bigram frequency threshold on the performances of the following relaxation stage as well as the font learning and re-recognition stages need to be investigated more systematically .

The language bases that we used in the experiments are compiled from related news reports from the same Clarinet newsgroup. Language bases compiled this way are "optimal" in that only texts of very closely related contents are included. The way the language bases are compiled as well as the scale of the language bases can be further experimented to observe their effects on word bigram probability relaxation. It will also be interesting to experiment with using a more comprehensive and general purpose language corpus as the language base and to see how closely it represents the word n-gram statistics in typical input documents. Comparing with using separate language bases for input documents of different subjects, using a general purpose language corpus as the language base makes the word-level deciphering algorithm more practical implementation-wise, but its feasibility need to be experimentally assessed first.

Further improvements in the performance of the word clustering algorithm are necessary to reduce the number of clustering mistakes before the result is fed to the deciphering stage, which will improve the overall recognition rate. A typical word clustering error that happens frequently is that similar words that differ only at the end of the words are grouped into the same cluster(Figure 5.19). An efficient way of dealing with this kind of clustering errors is to selectively stress the weight of pattern matching at the ending portions of the words within the overall match of the two words.

From the experimental results analysis, the time spent on word image clustering contribute to about 62 percent of the total running time of the word-level deciphering algorithm. Improvements in the word clustering algorithm in terms of time efficiency will be helpful in bring down the total running time. When processing input documents of larger sizes, a possible way to cut down clustering time is to split the input page into multiple vertical segments, performing clustering separately on each of the segment and finally combining the results together.

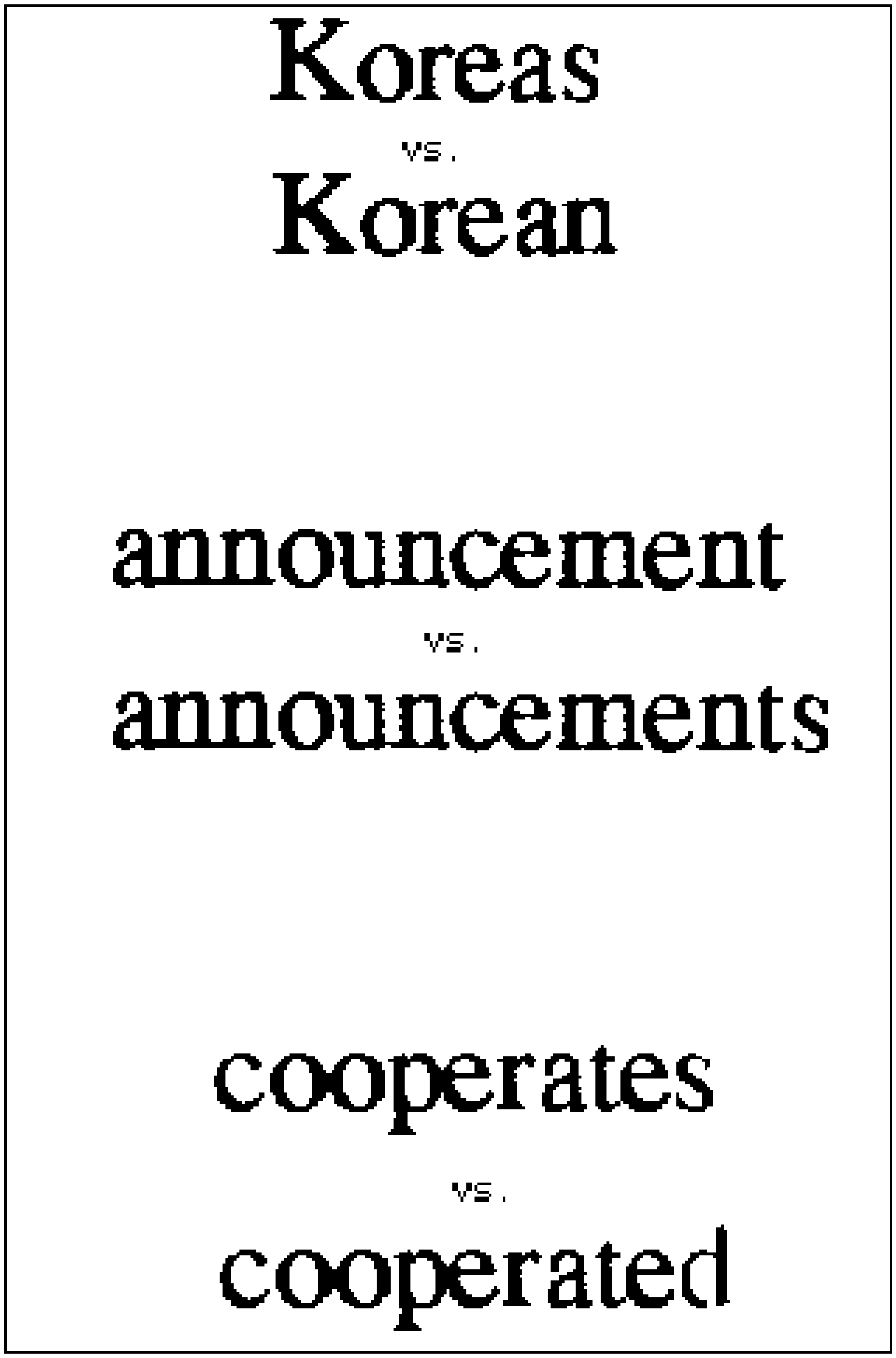


Figure 5.19: Clustering errors caused by visually similar words that differ only at the endings of word patterns.

More delicate font learning algorithm is in order to deal with real-situation image degradations and to secure the quality of the learned font. Further improvements in the re-recognition stage to take into consideration the font typeface and alignment information are necessary to handle the special issue of certain characters being ”cut off” from further matching. Finally, performance comparisons of the proposed approach against some of the state-of-the-art commercial OCR systems on degraded documents will be insightful for objective assessment.

Chapter 6

Re-recognition with a Hypothesis Testing Algorithm

An hypothesis testing approach for recognizing machine-printed word is presented in this chapter. Based on knowledge of the document font and the candidate set of a word, this approach searches a tree of partial word decisions to generate and test hypotheses for character recognition and segmentation. The search starts from both ends of a word image and proceeds inwards at each sequential character position. The accumulated cost of reaching a certain partial word recognition decision is combined with the estimated potential cost to reach a goal state using a uniform cost search algorithm. Visual constraints that are stable under image degradations are used for cost estimation, resulting in a time-efficient search algorithm for word recognition. The proposed algorithm compensates for local character degradation by relying on global characteristics of a word image. This algorithm is used by the word-level deciphering algorithm discussed in Chapter 5 in the re-recognition stage to recognizing words not solved by the deciphering stage. Tests of the algorithm show reliable recognition performance on scanned documents that include touching characters and character fragmentation caused by image degradation.

6.1 Introduction

Recognition of degraded words has been a difficult problem in OCR. One of the major problems caused by image degradation is the increase in character shape deformation that makes character classification more difficult [124]. Another major problem in recognizing degraded words is caused by the occurrence of touching characters and character fragmentation, which usually pose extreme difficulty to recognition algorithms that are based on character segmentation [113, 26, 22, 163, 39].

Many algorithms had been proposed in the past for recognizing degraded words. The more recent and advanced algorithms in this area are based on integrating character segmentation with character classification and using the character or word recognition results as feedbacks to alter or improve segmentation decisions made earlier [53, 17, 25, 167, 149]. Compared with earlier algorithms that do character segmentation independent of classification based on character pitch estimation or on profile and connected component analysis, these methods have made solid progress in achieving better segmentation and recognition results, but their performances are still not reliable with degraded documents that involve extensive touching characters or character fragmentation.

Another approach to word recognition is the holistic approach which is based on extracting features that represent the whole word and comparing the feature representation of the unknown word with those references stored in a lexicon [40, 75, 68]. Later improvements in holistic recognition restricted the possible match to a reduced neighborhood containing words that are visually similar to the unknown word. For example, the neighborhood for the word *"word"* might also contain the word *"work"* and the word *"ward"*. These algorithms effectively compensate for image degradation by choosing the optimal neighborhood size to guarantee that the correct word is included. The reduction of the neighborhood to a single optimal decision that best matches the word image is performed by a hypothesis testing algorithm [76].

In this chapter, a novel hypothesis testing algorithm is proposed that achieves recognition by reducing the candidate neighborhood of a given word image to a single choice. This method uses the information of the candidate neighborhood of a word to generate a structured series of tests that are executed on the input word image. It is assumed that a partial font base that represents font information of at least part of the character alphabet is available either through an explicit font learning step [87] or from the results of a word-level deciphering algorithm that has recognized a portion of the input text [46]. Figure 5.9 in Chapter 5 illustrates the components involved in the proposed method.

With the image of an unknown word as the input, this hypothesis testing algorithm combines the font information and the word candidate neighborhood to test hypotheses of character segmentation and character recognition at each sequential character position of the word image. This hypothesis testing process starts from both ends of the word image and proceeds inwards, forming a tree-like representation of hypotheses of partial word decisions as it proceeds. It searches through this tree structure that represents the potential word decisions using an uniform cost search algorithm. The uniform cost search algorithm chooses which direction to expand based on a cost value that combines the accumulated cost of the partial word decision already made with an estimated cost of decisions to be made further to reach a final word decision. Visual characteristics of word and character shapes that are relatively stable under image degradation are used by the uniform cost search algorithm to estimate the cost of potential decisions. This results in a search algorithm that uses visual constraints to speed up the search. The algorithm achieves precise word recognition by testing local character segmentation and recognition hypothesis and making the optimal decision of word identity using an uniform cost search algorithm.

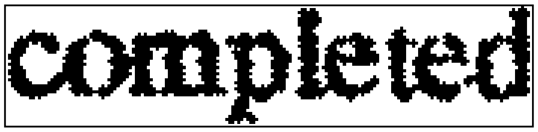
The rest of this chapter contains three sections. Section 2 discusses the major components of the proposed hypothesis testing algorithm. Section 3 presents and discusses the experimental results of applying this algorithm to scanned documents. Conclusions and future directions are pointed out in section 4.

6.2 The Proposed Algorithm

This hypothesis testing algorithm is proposed for the recognition of a word image. The information that is available to the algorithm are the candidate neighborhood of the word image, as well as a partial font base that includes font prototype information of part of the character alphabet. Figure 6.1 shows an example of a degraded word image and its candidate neighborhood provided by a preliminary word classifier, ranked in ascending order of classification confidence. We will use this as an example to illustrate the ideas in the following discussion of the hypothesis testing algorithm. Figure 6.2 shows an example of a partial font base.

The algorithm is a tree searching algorithm that expands on states representing current knowledge of the identity of the input word from a queue structure. The representation of a state includes the reduced partial word image to be matched against character prototypes as well as the updated word neighborhood. The search starts with a single state in the queue that represents the raw image of the input word and its associated word candidate neighborhood.

Starting with the raw word image and its associated candidate neighborhood, as shown in Figure 6.1, the algorithm examines the left-most and right-most character positions in the word image. Identities of the characters are constrained by the word candidate set. For the above example, the left-most character could only be one of the characters from the set {c, s, o}, and the right-most character could only be one of the characters from the set {s, d, l, k}. From the partial font base, the image prototypes for each of the characters in the two sets are then matched against the left-most portion and right-most portion of the word image. Each prototype match returns a value representing the degree of character match. It also returns the location in the word image where the best match was found.



completes	78756
composed	81052
strapped	81413
octagonal	82184
compared	82477
songbook	82964
completed	83449
comprised	83751
congested	83925
snapped	83956

Figure 6.1: Word image and its neighborhood.

A legal combination of potential characters at both ends is one that can find at least one word candidate that has the two characters at the corresponding positions. Combinations that have no word candidate matching them are considered illegal and discarded. Thus for instance, combination [c, l] in the above example will be discarded because it has no word candidate that matches the characters at the corresponding character positions.

For the remaining legal combinations of characters, the algorithm then evaluates the goodness of the combined matches by averaging the two values representing character match at both ends. The various combined matches are then ranked. For the above example, it is likely that the combination [c, d] will be ranked first, combination [o, l] ranked second, [s, d] third and so on. After these steps, the algorithm generates a ranked list of potential combinations of character decisions at both ends, each associated with a value indicating the confidence of

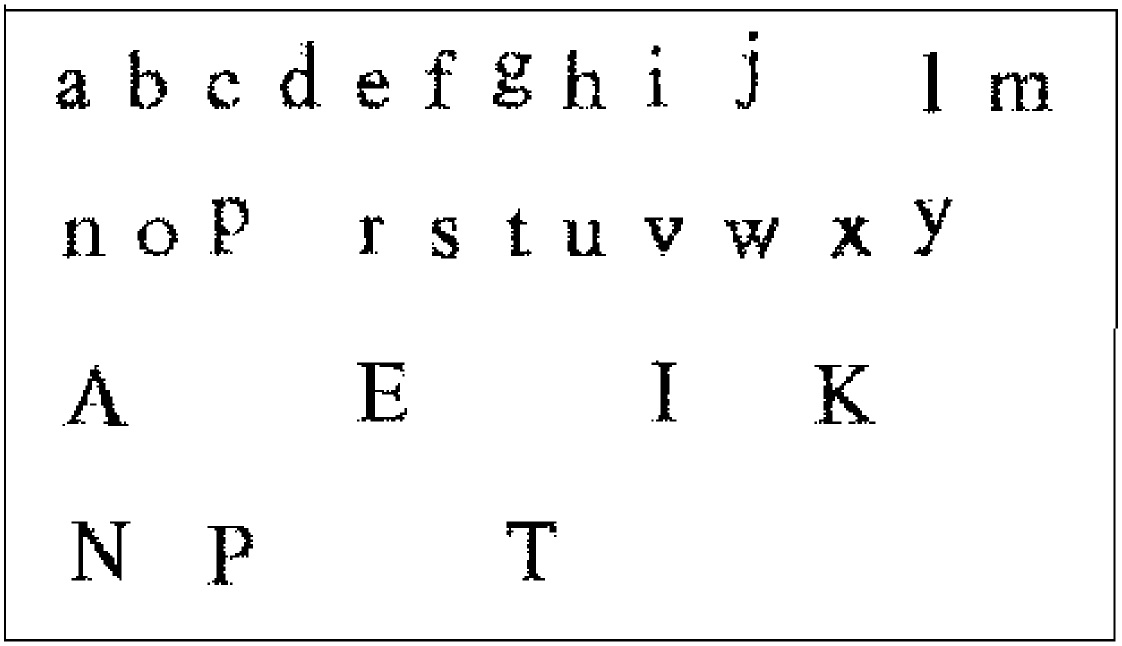


Figure 6.2: An example of a partial font base available to the hypothesis testing algorithm.

that particular combined decision.

In the next step, each potential combination of decisions at both ends is associated with a reduced word candidate set that represents the improved knowledge of the word's identity. In our example, the combination [c, d] will be associated with the reduced candidate set of {*composed, compared, completed, comprised, congested*}, the combination [o, l] will be associated with the reduced candidate set {*octagonal*}, and so on. In addition to the reduced word candidate set, each potential combination of character decisions is also associated with a reduced partial word image. The reduced partial word image is generated by cropping out the left-most and right-most characters with the help of the location information for the best character matches at both ends. Local image modifications are applied to remove possible fragments and white character gaps. Figure 6.3 shows two of the combined decisions, as well as the reduced partial word images and the reduced candidate sets associated with them.

Each of the character decision combinations, with its reduced word candidate set and partial word image, is represented by a new state and placed in the open queue. Each new state is also assigned a cost value that measures the overall goodness of the decisions with this state. Details on the determination of the cost value for each state will be discussed later.

The search algorithm always chooses from the open queue the optimal state with the minimum cost for further expansion. For each selected state, the above process of character matching on both ends is repeated on the reduced word image with its reduced candidate set, generating more states which are saved again into the open queue. This searching process continues until a complete word decision is reached or until the state queue becomes empty. An empty queue of state indicates a failure to reach a word recognition.

Figure 6.4 illustrates how image matching generates new states and how the tree structure representing different potential word decisions is generated and searched.

6.2.1 Tree search

As illustrated in Figure 6.4, the hypothesis testing algorithm that finds the optimal word recognition can be considered as a tree search where each node represents a partial recognition hypothesis. A node at the lower level in the tree corresponds to a combined hypotheses for character matching at an inner position of the word image. The root node corresponds to the raw input word image and its initial candidate neighborhood. Terminal nodes correspond to complete word decisions. A open queue is used to implement the tree search. The algorithm always chooses the node with the minimum cost value for further expansion. The way the cost value is determined is discussed below.

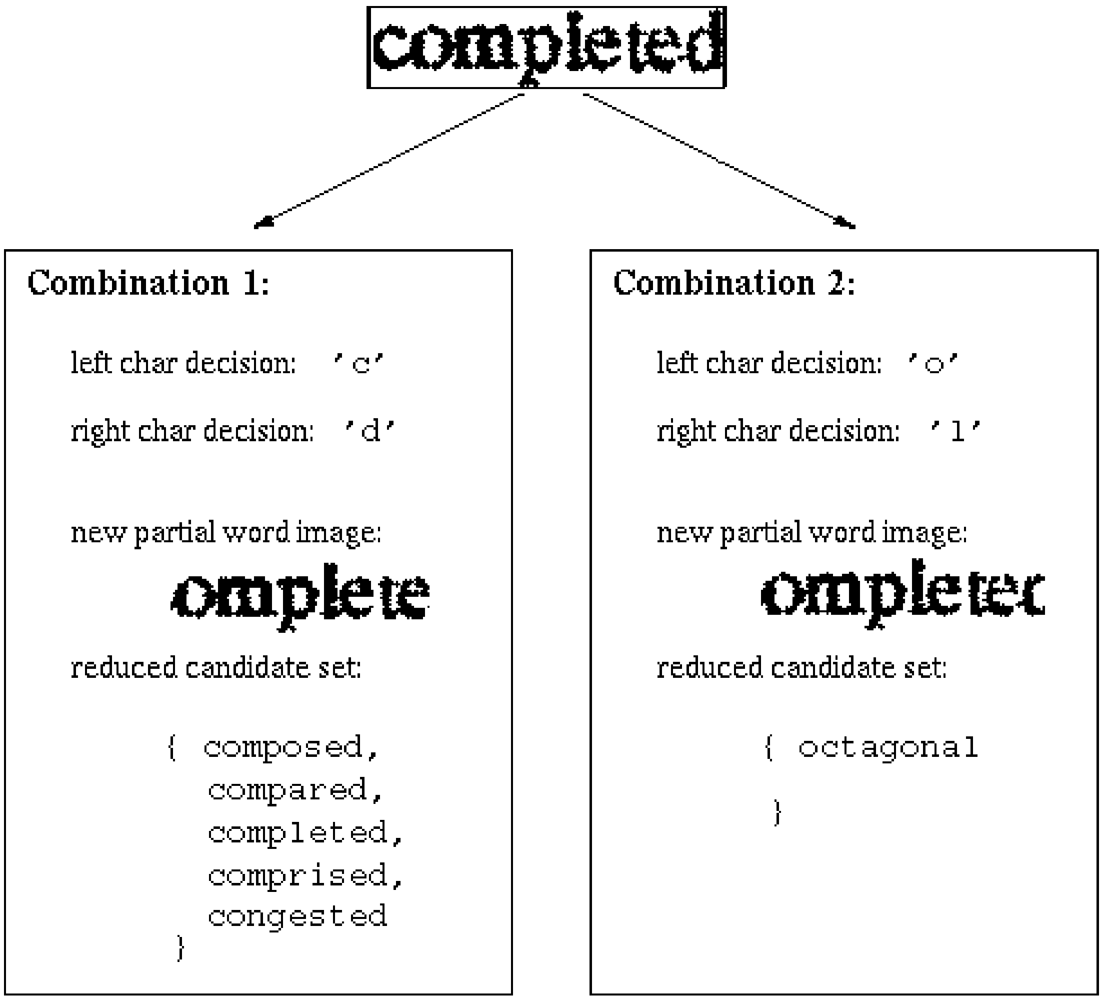


Figure 6.3: Two examples of the combined decisions

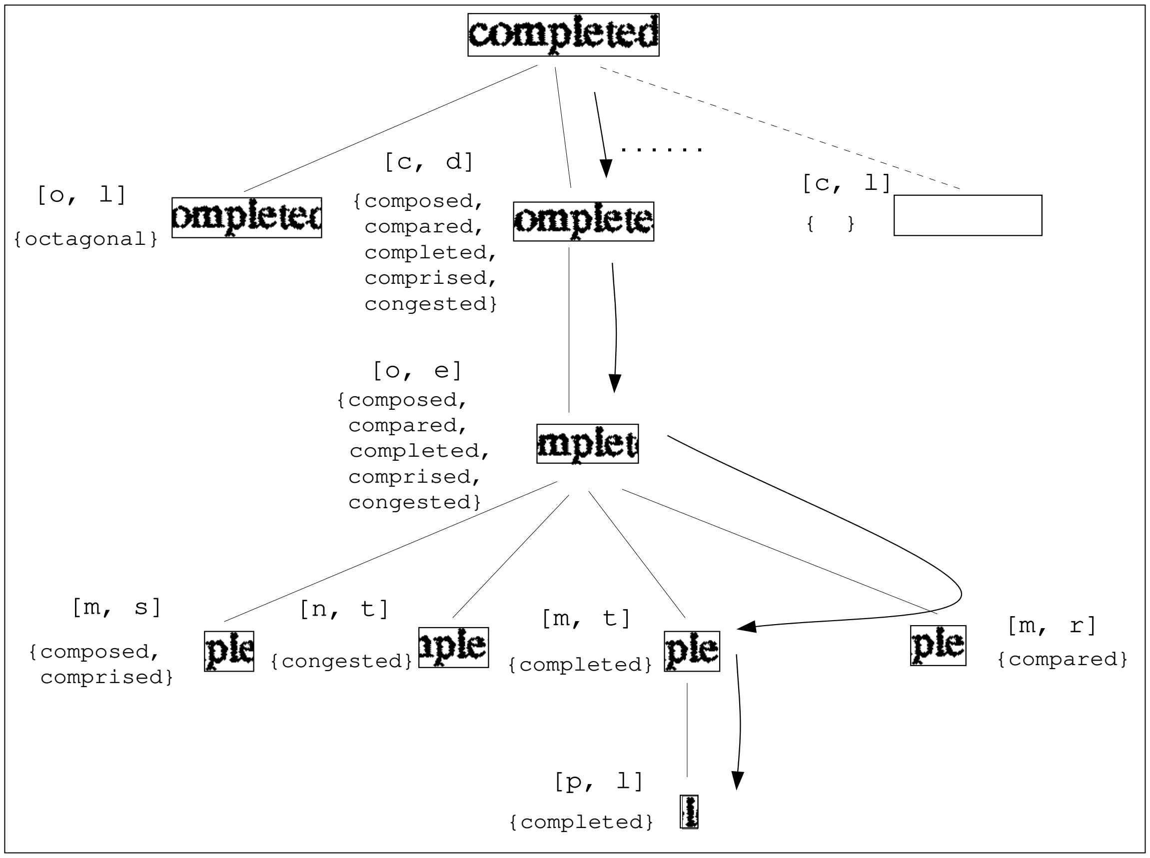


Figure 6.4: Example application of the hypothesis testing algorithm

6.2.2 The cost function

For any node that represents a partial word decision, it will be desirable if we can tell whether the decision made is good or not. It will also be desirable if we can judge relatively how good the decision is in comparison with other competing decisions. These are the important ingredients of a cost function.

As discussed earlier, a partial word decision is formed through a series of character matching at different character positions. Therefore how good each character match was should affect how good the resulted partial word decision is. Also, each partial word decision is associated with a reduced word candidate set and a reduced partial word image. With the help of the font information, we can predict the likelihood the reduced word image matches its reduced word candidate set. Therefore, the cost function of a partial word decision is a combination of two parts. The first part represents the goodness of character matches accumulated so far, while the second part looks ahead to project the match between the reduced word image and its reduced candidate set.

The cost function $f(n)$ of a node n is defined as:

$$f(n) = g(n) + h(n),$$

where $g(n)$ is the accumulated average cost of all the character matchings executed at each successive character position so far and $h(n)$ is an estimate of the potential cost of matching the remaining portion of the word image with its reduced word candidate set.

In our proposed algorithm, $h(n)$ is usually assigned a value of 0 for most cases where there is no obvious indication of mismatch between the remaining portion of the word image and the corresponding characters in the candidates of its reduced candidate set. $h(n)$ is assigned a large value close to 1 if it becomes clear that the remaining portion of the word image doesn't match its associated candidates.

The decision about whether the remaining portion of a word image matches its associated set of candidates is based on matches of the following visual characteristics: the overall size, the existence of descenders and ascenders, and the existence of isolated dots at the top which indicates occurrences of the character 'i' or 'j.'

The rationale of choosing the above characteristics for the estimation of $h(n)$ is that these visual constraints are relatively stable under various image degradation. Care must also be taken in designing the algorithm such that in some rare cases when these constraints do become unstable (like when the top spot of an 'i' is completely wiped off as a result of severe image degradation), the overall decision will not be severely affected. This can be achieved, for instance, by associating each feature dimension with a weight that is application-dependent, so that when severe image degradation happens, the weight associated with the "isolated dots" feature is assigned a relatively smaller value to limit its impact.

The use of multiple visual features for the estimation of $h(n)$ enables the algorithm to fully exploit the heuristic power of visual constraints to speed up the search.

An example of how visual constraints are used to estimate $h(n)$ and to reject a currently promising partial hypothesis is shown in Figure 6.5. The figure shows only a portion of the tree expansion relevant to the discussion. Each node is shown with its current character decisions at both ends, its associated candidate set, as well as the reduced word image. The two branches at the middle of the figure represent two of the competing combined character decisions at the third character position.

As shown in Figure 6.5, because of image degradation, the image prototype for character 'r' happens to match the right-most portion of the partial word image a little better than the prototype for character 't,' which is the correct decision. This makes the right-side node a more favorable partial hypothesis. But the visual compatibility check between the reduced word image and the corresponding characters from the word candidates in the candidate set indicates a gross mismatch with the right-side node. The reduced image, which is the image of string "ple", doesn't match the string "pa" either in overall size or in the ascender area. This results in a large h value being assigned to the node at the right side, making the overall cost of the right-side node much higher than the left-side node. As a result, it becomes much less likely that the partial hypothesis associated with the right-side node will be chosen later for further expansion.

Traditional OCR algorithms that are based on character segmentation and classification have often been criticized as being too prone to image noise. A word recognition result is heavily dependent on the classification results of the individual characters, which usually become very unreliable under severe image degradation. On the other hand, we have also seen in many word-based recognition algorithms the inability to use detailed font information to differentiate words that are visually very similar to each other on a more global scale(e.g. “these” vs. “those”), and thus are hard to tell from each other by using word-level feature representations. As explained in the above paragraphs and illustrated by the example in Figure 6.5, the proposed hypothesis testing algorithm combines the desirable aspects of the above two different recognition approaches while neutralizes the undesirable aspects in each. The proposed hypothesis testing algorithm is able to use the local font information to generate and test hypotheses and estimate the potential cost. But still, the final word recognition results are based on global evaluation of the overall match rather than being sensitively affected by any individual character mismatch caused by local image degradation. The use of visual heuristics also makes the search algorithm time-efficient. These qualities are essential to a word recognition algorithm to produce robust recognition results on scanned documents with image degradation.

6.2.3 Algorithm outline

An outline of the proposed hypothesis testing algorithm is presented in Figure 6.6. It takes as its inputs the word image, the preliminary candidate neighborhood of the word, and the partial font base. The proposed hypothesis testing algorithm uses the input information to generate and test hypotheses of the word’s identity, until a goal state is reached that corresponds to a recognition result with optimal cost as evaluated by the uniform cost search algorithm.

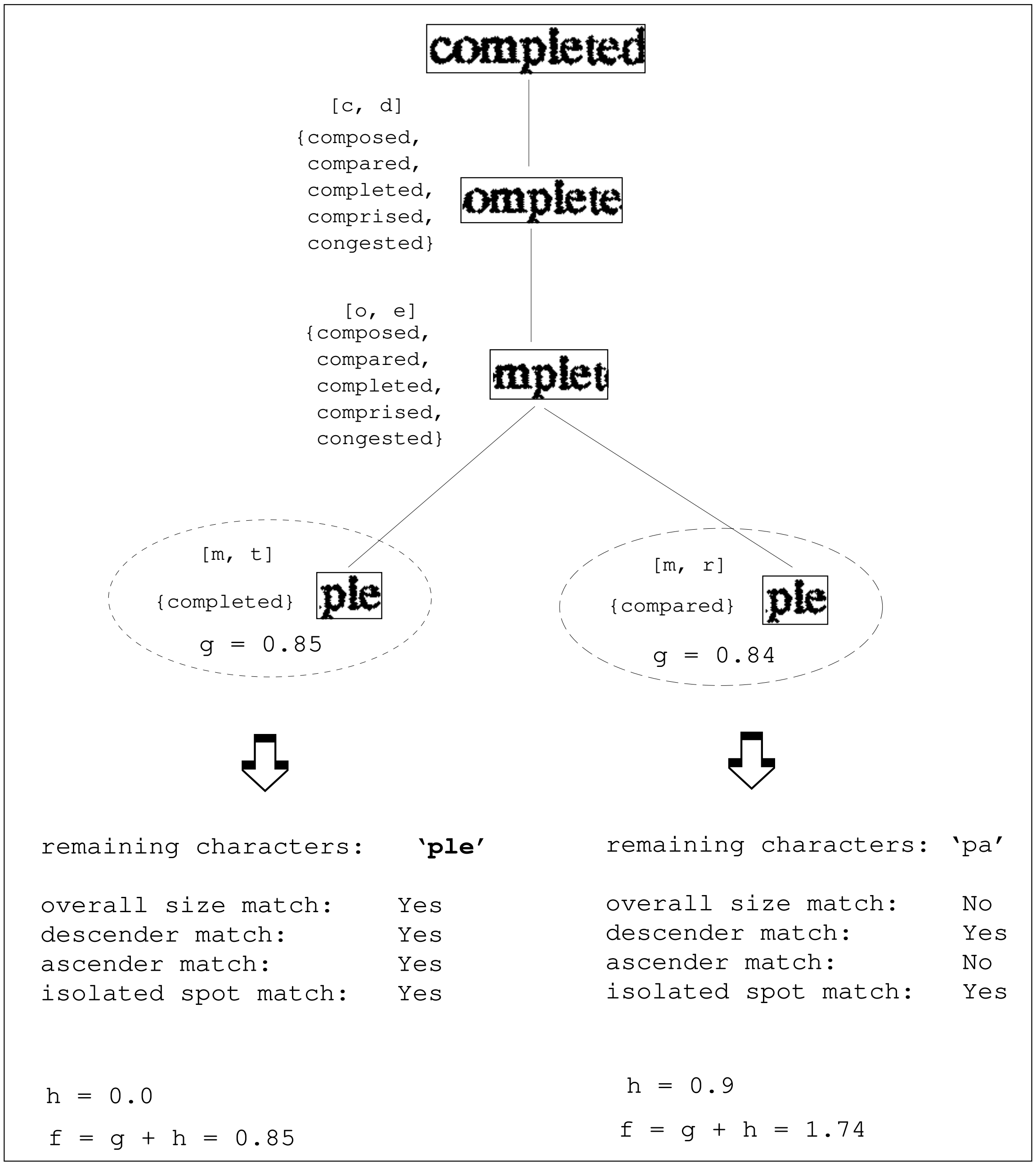


Figure 6.5: Estimation of h^* using visual constraints.

```
Algorithm Hypothesis_Testing(WordImg, FontBase, CandidateSet)
BEGIN
    initialize the OpenQueue of states;
    assign the initial state to be the word image and its associated candidate set;
    put the initial state into the OpenQueue;
    Word_Recognized = FALSE;
    WHILE (OpenQueue not empty and Word_Recognized == FALSE)
        get the optimal state from OpenQueue;
        IF (current state not a goal state)
            get character candidate set at both ends;
            match characters against partial word image at current character position;
            combine the decision at both ends and sort the combined decisions;
            FOR (all the combined decisions)
                IF (combined decision matches with candidate set)
                    update candidate set; generate new partial word image;
                    compute accumulated cost g; estimate potential cost h,
                    f = g + h;
                    generate a new state and put it into OpenQueue;
                END
            END
        ELSE
            get the recognition result;
            Word_Recognized = TRUE;
        END
    END
END
```

Figure 6.6: Outline of the hypothesis testing algorithm

6.3 Experimental Results

The proposed hypothesis testing algorithm for word recognition is used as part of a word-level deciphering algorithm discussed in Chapter 5. The word-level deciphering approach is able to learn the font information of the input text from a decrypted portion of the text. This provides the hypothesis testing algorithm with the partial font information it needs. The preliminary word candidate set for each word in the text also provides the hypothesis testing algorithm with the word candidate constraints.

The word-level deciphering algorithm was applied to both artificially created documents as well as scanned documents. Presentation and discussion of the detail experimental results can be found in Chapter 5. In the following, results from the same experiments as reported in Chapter 5 that are specifically related to the hypothesis testing algorithm are presented.

Table 6.1 summarizes the performance of the proposed hypothesis testing algorithm applied to the 15 scanned documents that were used for experiments in Chapter 5. For more detailed information about the 15 scanned documents used in the test, please refer to Section 5.3.3. in Chapter 5.

Column *# of Words* in Table 6.1 indicates the total number of words in each document to be recognized by the hypothesis testing algorithm. It equals the number of word not decrypted by the deciphering stage and left to be solved by the re-recognition stage. Column *C1* is the number of characters whose font information is available to the hypothesis testing algorithm. This font information is learned after the word deciphering stage. The percentage of those characters whose font information is available among all the characters that appear in the input document indicates the completeness of the font information available to the algorithm. This percentage is listed in Column *% of All Characters*. Notice that for all the characters that appear in a document, the lower case characters usually have much higher occurring frequencies than the upper case characters and digits. And those characters whose font information was learned during the word deciphering stage are usually lower case characters. Therefore, a relatively low percentage value in Column *% of All Characters* might not reveal the fact that a much higher percentage of the frequently occurring characters in the input document are actually covered by the learned font information. For the rest of the items in Table 6.1, Column *C2* stands for the number of characters learned during the re-recognition stage. Column *Word Error* shows the number of words failed to be correctly recognized by the hypothesis testing algorithm. And Column *Correct Rate* represents the overall word recognition rate of the proposed hypothesis testing algorithm. For a comparison with the performance of the preliminary word recognizer used to provide the word candidates, correct word rate of the top choices produced by the preliminary word recognizer is listed in Column *P0*.

Analysis of the recognition results shows that the hypothesis testing algorithm usually produces incorrect recognition results on three groups of words. The first group of words are those that include characters whose font information was not available in the initial font base. Examples of words that belong to this group are "be" and "but", which were recognized as "he" and "hut" because font information of 'b' was not available; the word "next" being recognized as "nest" because font information of 'x' was not available; "$1 billion" being recognized as "81 billion" because '$' was not available in the font base, and also the word "after" being recognized as "alter" because font information of 'f' was not available(Figure 6.7). It is very likely that characters such as 'b,' 'x,' '$' and 'f' in the above examples are not included in the initial font base because of their relatively low occurring frequencies in text documents. Therefore, word recognition errors like those in the above examples account for a large percentage of all the errors produced by the algorithm.

The second group of words are mostly short words that quite often have only two to three characters. One of the important advantages of the proposed algorithm, as mentioned

Doc.	# of Words	C1	% of All Characters	C2	Word Error	P0	Correct Rate
C1(G1)	711	25	71	11	13	77.8%	98.1%
C2(G1)	647	25	69	10	10	76.4%	98.4%
C3(G1)	732	22	64	12	13	81.1%	98.2%
C4(G1)	849	17	59	16	31	75.7%	96.3%
B1(G2)	674	34	87	4	21	81.9%	96.8%
B2(G2)	629	28	80	8	17	77.3%	97.3%
B3(G2)	767	17	61	16	15	74.9%	98.1%
B4(G2)	730	18	63	12	23	79.7%	96.9%
B5(G2)	811	16	54	14	30	82.0%	96.3%
R1(G3)	732	21	74	13	11	77.2%	98.5%
R2(G3)	732	27	84	8	15	74.2%	98.0%
R3(G3)	767	19	60	11	25	78.6%	96.7%
R4(G3)	689	29	80	4	17	82.5%	97.5%
R5(G3)	744	23	74	12	13	80.1%	98.3%
R6(G3)	591	29	82	11	15	75.2%	97.5%

Table 6.1: Experimental results of the hypothesis testing algorithm on the 15 test documents.

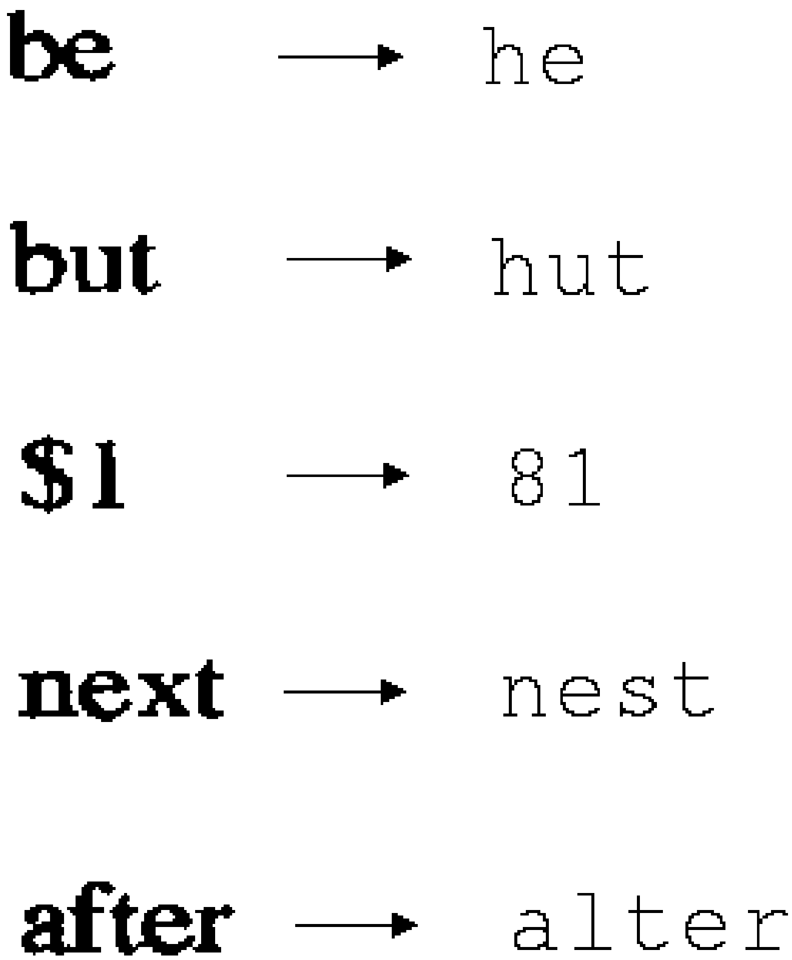


Figure 6.7: Examples of words incorrectly recognized by the hypothesis testing algorithm due to lack of font information. In these cases, font information for characters 'b,' 'x,' '$' and 'f' is not available.

before, is its ability to compensate for individual character deformation by relying on more global constraints in the word to make the recognition decision. The effectiveness of this advantage is eroded when the algorithm is applied to very short words. Using the proposed hypothesis testing algorithm, recognition decisions on short words become largely dependent on local character classification results that are easily affected by image degradation. Mistakes on individual character classifications do not have enough global constraints that otherwise exist in a longer word to correct themselves. Example of these short words are "had" being recognized as "hod," "one" being recognized as "ore," "out" being recognized as "cut," "now" being recognized as "new," and "1" being recognized as "I," where all the characters in these words were included in the initial font base(Figure 6.8). The situation gets worse when short words also include characters whose font information is not included in the font base.

The third group of words are, unsurprisingly, those words that have other words that are visually almost identical to them. Examples are the word "banking" being recognized as "barking," the word "past" being recognized as "post," "fined" being recognized as "fired," "month" being recognized as "mouth," "nationalist" being recognized as "rationalist," and the word "suspicions" being recognized as "suspicious"(Figure 6.9).

Analysis of the experimental results shows that the performance of the proposed hypothesis testing algorithm is dependent on the quality of the font base. A font base that includes font information of most of the lower case characters usually enables the hypothesis testing algorithm to produce good recognition results. On the other hand, a font base with many lower case characters missing produces poor recognition results. The quality of character prototypes in the font base also affects the performance of the algorithm. As previously pointed out in the experimental results analysis in Chapter 5, an inferior font base usually points back to certain characteristics in the input document. These documents are usually relatively short in length, or may include many pronouns or names of places, or are poorly represented by the language base. All these factors lead to poor performance in word-level

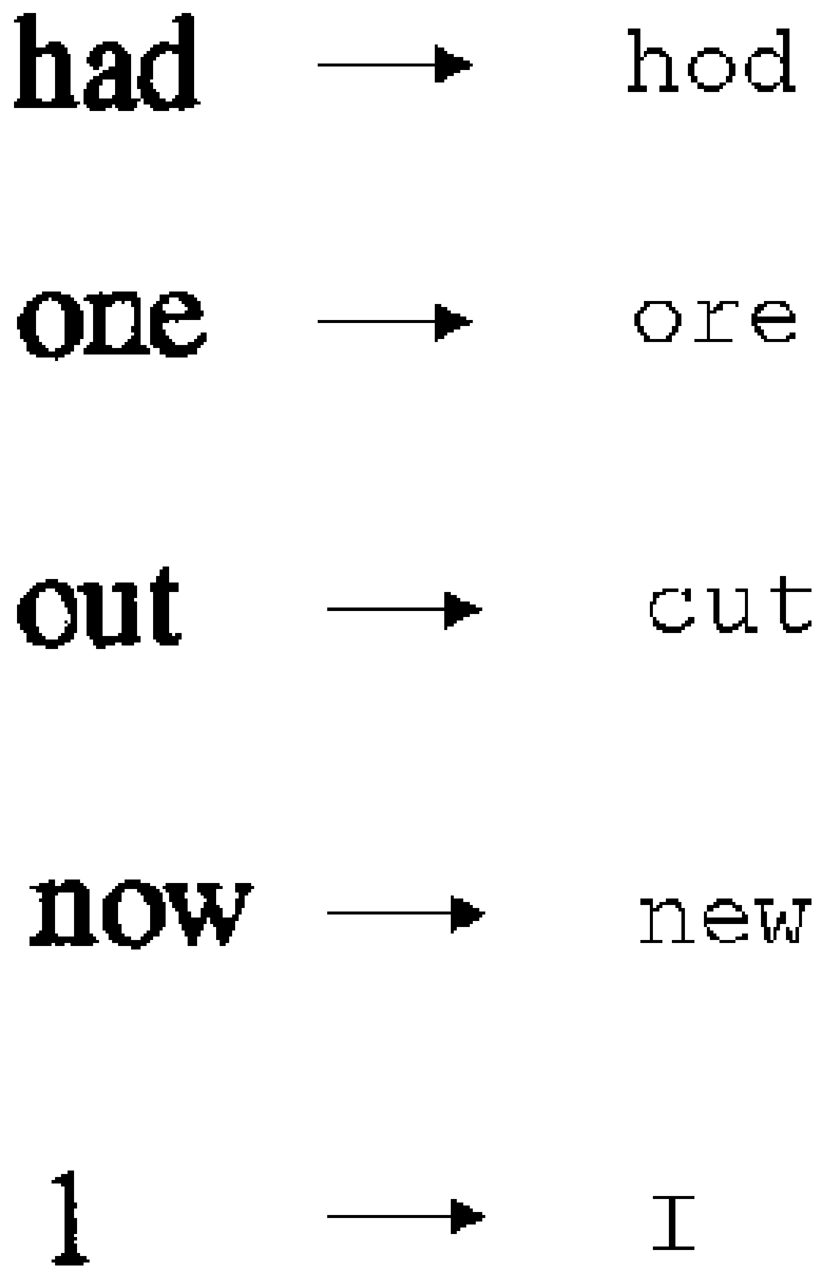


Figure 6.8: Examples of short words that are incorrectly recognized by the hypothesis testing algorithm.

Figure 6.9: Examples of words being incorrectly recognized by the hypothesis testing algorithm as other words that are visually almost identical.

deciphering results, which in turn affects negatively the quality of the learned font base.

For more discussions of the results of the proposed hypothesis testing algorithm, and for evaluations of the performance of the hypothesis testing algorithm as related to the performance of the whole word deciphering algorithm, please refer to Section 3 of Chapter 5.

6.4 Conclusions

A hypothesis testing algorithm for the precise recognition of degraded word has been proposed in this chapter. Based on the knowledge of partial font information and word candidate neighborhoods provided by a preliminary word recognizer, this algorithm generates and tests hypotheses of partial word recognitions until a optimal solution of minimum cost is found. The accumulated cost of reaching a certain partial recognition is combined with a heuristic estimation of potential matching cost by using an uniform cost search framework. Multiple visual constraints that are relatively stable under image degradation are used to estimate the potential matching cost. This enables the algorithm to fully utilize the visual heuristic power to speed up the search. The algorithm is able to compensate for local image degradation in character patterns by relying on global visual and linguistic constraints about a word's identity, and achieve precise recognition of a degraded word.

Tests of the proposed algorithm showed consistently high recognition rate on degraded scanned documents that involve extensive touching characters, character fragmentation, as well as deformations in individual characters.

Chapter 7

Summary and Conclusions

The problem of recognizing degraded document is addressed in this thesis. Traditional approaches to document recognition have been taking the segmentation-based approach in which character images are first segmented and then classified to generate the character recognition results, which are then combined to form the word recognition results. Degradation in document image causes two problems that the segmentation-based approaches usually have difficulty in dealing with. The first problem is character segmentation under image degradation. Touching characters and character fragmentation pose severe difficulties to character segmentation. The second problem is the critical and usually fragile dependence of these approaches on font training. The research presented in this thesis is intended to seek unconventional solutions to these two problems by designing recognition approaches that deliver high recognition performance without the need for character segmentation or font training.

The following three sections summarize the approaches proposed in this thesis and discuss issues that are related to this research. The prospect of implementing word-level deciphering by using a Simulated Annealing algorithm is also discussed.

7.1 The Proposed Approaches

It was proposed in this thesis that the computational task of document recognition being considered as a process of finding the mapping between the visual patterns and their linguistic identities in the input document. Deciphering algorithms were proposed to decrypt the mapping by combining the visual constraints from the input document and the various levels of linguistic constraints from the language base. This deciphering approach to document recognition worked on different levels of language granularities, both on the character level as well as on the word level. Therefore, document recognition was achieved by decoding characters using character level linguistic constraints. It was also achieved by the direct decryption of words using word level linguistic constraints. Two different algorithms were proposed to implement the deciphering.

A modified character-level deciphering algorithm was proposed that used character level linguistic constraints to decrypt the identities of single characters and touching characters. Visual constraints were also used throughout the deciphering process in combination with the linguistic constraints to facilitate decryption. The modified algorithm was able to learn font information of the input text dynamically during the deciphering process and use the learned font information to recognize long touching patterns that were difficult to solve through decryption. Short touching patterns were decrypted along with single characters by matching the cipher-words against the dictionary with the assumption that certain number of characters were involved in each short touching pattern. Unlike single characters or short touching patterns, long touching patterns were difficult to solve through straight forward decryption. Therefore, their processing was postponed towards the end of the deciphering process when font information of most of the single characters and short touching patterns became available, so that long touching patterns could be accurately recognized using the learned font information through partial pattern matching. The modifications also made the algorithm tolerant to clustering mistakes. Potential clustering mistakes were able to be detected by the algorithm. Character confusion heuristics were used to help verify the detections and trace back and reverse the clustering mistakes.

The modified character-level deciphering algorithm was tested on both artificially created documents as well as real scanned documents. The test documents included extensive occurrences of touching characters, some of them extreme long touching patterns. Image degradation in the test documents also created deformations in individual characters. As a result, large number of clustering mistakes of individual characters and their touching patterns happened during the pattern clustering stage. The experimental results showed consistently the algorithm's ability in dealing with touching characters. For the artificially created documents, on average 93.6% of the touching patterns in an input document were correctly recognized either through decryption or by partial pattern matching using the learned font information. For the scanned documents, on average 92.9% of the touching patterns in an input document were correctly recognized. For both artificial and scanned input documents, the majority of the touching patterns were short touching patterns. They were recognized through decryption with consistently high rate. The algorithm's ability in handling clustering mistakes was also demonstrated by the results. Most of the clustering mistakes were detected and corrected by the algorithm, except for cases where visual similarities between single characters and touching patterns were not covered in the character confusion heuristics being used. The algorithm was able to achieve an average word recognition rate of 96.8% on artificially created documents and 96.3% on real scanned documents.

Although the modified character-level deciphering algorithm was able to decrypt touching characters, it did have limitation in handling other kinds of problems caused by image degradation. For example, the way the decryption was implemented on the character level makes this algorithm unable to deal with character fragmentation. A better and more complete solution to character fragmentation can be achieved through the use of word-level deciphering algorithm to directly recognize words. This is where our proposed word-level deciphering approach stands.

A word-level deciphering approach to document recognition was proposed in this thesis. Word-level language constraints were used to first decrypt a selected portion of words from the input text. This selected portion of text has relatively more reliable word n-gram statistics that matches that of a language base. Font information was learned from the decrypted portion of the text and then used to re-recognize the rest of the text unavailable for cipher solution. This approach considered the task of word deciphering as a process of constraint satisfaction and used a probabilistic relaxation algorithm to achieve the decryption by using word bi-gram statistics from a matching language base.

By selectively applying word-level deciphering to a portion of the input text that has relatively more stable and reliable linguistic statistics, the proposed word-level deciphering approach achieved a breakthrough over the difficulty that made substitution cipher decoding above the character level inapplicable to document recognition. This difficulty is due to the fact that for most input documents, there does not seem to exist a stable and well-organized word repetition statistics among the majority of the words that a word-level deciphering algorithm can exploit. However, this stable and well-organized word repetition statistics does exist on smaller scales with a selected portion of the words in the input document. Therefore, although it is close to impossible to implement a word-level deciphering approach generally on an input document, it is possible to execute word-level deciphering on a selected portion of the input text, and use the font information learned from this initial decryption to recognize the rest of the document. Our experimental results have consistently shown that the selected portion of input text which include mostly functional words and subject related key words can be reliably recognized through word-level decryption if a well constructed language base is available. These sets of unique words, although relatively small in number, often comprise a significant portion of the input text and thus provide solid resources for font learning. Our experimental results have also demonstrated this.

The font learning algorithm showed the intelligence of selective font learning by filtering out badly deformed samples so that only sample with good image quality were learned into the font base. It was also able to suppress various character deformations and automatically adapt to the "core image" of character samples. Besides, the font learning algorithm also achieved as a by-product the detection and correction of word clustering mistakes by checking on incoming character samples for gross mismatches with the learned prototypes.

The hypothesis testing algorithm used in the re-recognition stage has shown its ability to use the learned font information to recognize the rest of the words reliably and efficiently. The experimental results have shown that the uniform cost search algorithm made efficient use of the globally stable visual constraints in the input word to estimate potential matching costs and speed up the overall searching. Also, deformations in individual characters were accounted for by the algorithm's ability to find the optimal recognition result.

On average, the proposed word-level relaxation deciphering algorithm was able to achieve a word recognition rate of 98.1% on a group of 15 scanned documents that have extensive occurrences of touching characters, character fragmentation, as well as individual character deformations.

7.2 Discussions of Related Issues

In this thesis, it was proposed that document recognition being considered as a process of finding the mapping between the visual patterns in the input text and their linguistic identities. Deciphering algorithms were proposed to compute the mapping. It will be interesting to look at the issues at a higher level and ask ourself the question of what this research has contributed to the general understanding of document recognition as a computational task. Using the term of the three-level methodology proposed by Marr [107], the contributions of this thesis to the understanding of document recognition as a computational process are mainly on the *representational and algorithm* level.

The uniqueness of considering document recognition as a deciphering process is that visual information and linguistic constraints are treated with equal significance in the recognition task. By adopting this point of view, recognition of an input document lies in finding the optimal mapping between visual patterns and their linguistic identities such that the repetition statistics of visual patterns in the input text best matches the repetition statistics provided by a language model. This is conceptually different from the traditional view in which OCR and document recognition were considered as basically visual tasks and their solutions critically dependent on visual segmentation and classification of characters. The heavy reliance on visual classification in traditional approaches have contributed significantly to the predicament in character segmentation as well as font training, especially in the presence of image degradations. Moreover, the proposed deciphering approaches also differ from some of the recent document recognition approaches that pay increasing attention to the use of linguistic constraints. In those approaches, linguistic constraints are still used as subsidiary aids to visual classifications. In the proposed deciphering approaches, however, linguistic constraints are no longer used only for post-processing character classification results, they are assumed the same importance as the visual constraints in suggesting raw recognition candidates as well as helping post-processing classification results. Adopting a deciphering approach makes the recognition font-independent. It also opens up the potential to a genuine segmentation-free recognition approach because deciphering can be executed above the character level to directly recognize words. We have seen how progress in document recognition has been hindered by the lack of understanding of the basic issues involved. We hope this research illustrates that better understanding of the basic issues in document recognition can promote breakthrough in this field and bring about novel solutions to the difficult problems.

In a broader perspective, the problem of font training in document recognition parallels the problem of information representation in computational vision and pattern recognition. It has to do with the fundamental difficulty of having a generic representation that is concise so that it is computationally feasible, while at the same time is also powerful and sufficient enough to cover the rich varieties of different visual patterns and their variations in the real world [107, 110]. Visual representation has been a challenging issue in vision research.

Gibson's *ecological approach to vision* represents a systematic effort to address the problem of visual representation. Rather than focusing on detecting and responding to sensory *variances*, the function of vision in Gibson's view was to detect and recover the basic "*invariants*" of the external world [59, 58]. In other words, representation of the underlying *invariant properties* of the perceived subject is the core of visual representation and perception.

Gibson's theory is directly relevant to the task of document recognition. The whole issue of font training in document recognition is about the idea of having a representation of *invariant properties* of characters. An ideal implementation of font training would be one that can capture the *invariant properties* of different fonts and their variations under image degradation. But *how* can this be implemented? Gibson failed to provide an answer to this problem in vision. Research in OCR has not been able to find a general and applicable solution to this problem in document recognition. We believe the deciphering approaches proposed in this thesis provide a viable solution to the issue of uncovering the underlying *invariant properties* in the special domain of document recognition. The specific *invariant properties* in this case is the font-invariant correspondence between the repetition statistics of visual patterns in the input text and those of their linguistic identities provided by a language model. As has been shown throughout this thesis, this font-invariant correspondence can be uncovered by the proposed deciphering approaches to achieve reliable recognition of characters and words.

7.3 Prospect of Implementing Word-level Deciphering Using a Simulated Annealing Algorithm

A word-level deciphering approach for document recognition was proposed in Chapter 5 in which the task of recognizing a document was considered a deciphering process of finding the mapping from visual words to their linguistic identities. A word-level probabilistic relaxation algorithm was proposed to decipher the mapping by exploiting word-level language constraints.

The task of word-level deciphering can also be considered from a different point of view as a combinatorial optimization problem. A simulated annealing algorithm can be used to find the optimal solution which corresponds to a mapping from visual words to plain words that has the minimum cost.

7.3.1 Combinatorial Optimization and Simulated Annealing

An instance of a combinatorial optimization problem can be described as a pair$(\mathcal{R}, \mathcal{F})$, where $\mathcal{R}$ the configuration space, is a finite set of configurations, and $\mathcal{F}$ is a cost function that attaches a real number(cost) to each configuration $A \in \mathcal{R}$. The problem is to find a configuration A_{opt} such that $\mathcal{F}(A_{opt})$ is the global minimum. Optimization algorithms that provide a global minimum usually require a prohibitively large amount of time. However the quality of the solution can be traded off against the time required to obtain the solution by using approximation algorithms. Simulated annealing is a approximation algorithm for finding global minimum which is based on randomization techniques.

Simulated annealing combines the simplicity of an iterative improvement algorithm with the power of randomization techniques to search through the configuration space and settle with uniform probability on the set of global minima. In an iterative improvement(or local search) algorithm, there exists a generation mechanism which is a simple means of generating a transition from a given configuration to another one. The generation mechanism defines a neighborhood structure on the set of configurations: the neighborhood of a configuration A is the set of configurations N that are reachable from A in exactly one transition. At each step of the algorithm, an element of the neighborhood is selected but a move occur only if the the cost function decreases. Each output of the generation mechanism selects one element of the neighborhood. Starting from an initial configuration, the local search uses the output of the generation mechanism to consider the transition(the next configuration). The transition is acceptable if the cost function is decreased in the proposed configuration. The algorithm terminates if no neighboring configuration with less cost can be found. The algorithm, although simple and fast, has the disadvantage of finding only a local minimum.

Randomization is introduced in simulated annealing which allows transition to happen occasionally even when the cost function is deteriorating. The acceptance criterion for a transition employs a probabilistic rule called the *Metropolis Criterion*. The Metropolis Criterion, in conjunction with the method of *simulated annealing*, was introduced by Metropolis *et al.* in 1953 as an algorithm for simulating the evolution of a solid in a heat bath. Starting from a configuration A_i with associated cost $F_i = F(A_i)$ and generating configuration A_j with its associated cost F_j, the transition is acceptable with probability $P(A_j acceptable)$,

$$P(A_j acceptable) = \begin{cases} 1 & F_j \leq F_i \\ \exp((F_i - F_j)/T) & F_j > F_i \end{cases}$$

T is the control parameter and is analogous to temperature in the annealing process of metals. For large value of T, large deterioration of the cost function are accepted, and for $T = 0$, the algorithm is reduced to a simple local search algorithm. The algorithm starts from an initial configuration with a relatively large value of T. For a given value of T, a number of transitions are considered and then T is lowered until it reaches zero.

7.3.2 Word-level Deciphering as Simulated Annealing

Simulated annealing algorithm has been applied to OCR by solving a substitution cipher on the character level. The basic idea is to decrypt the optimal mapping between letters and cipher characters such that the character n-gram statistics in the cipher text best matches the letter n-gram statistics in a language model. Nevertheless, no experiment has been reported before of using a simulated annealing algorithm to directly decrypt words by using word-level language constraints.

In the following, we will explain our conception of how a simulated annealing algorithm can be used to implement a word-level deciphering algorithm for document recognition.

Like the probabilistic relaxation algorithm proposed in Chapter 5, a simulated annealing algorithm for word-level deciphering also works on a selected portion of the words in an input document. This portion of the words is selected because they have more stable and reliable word bigram repetition patterns which match those provided by a language base, and thus are better candidates for a cipher solution.

Also as in the probabilistic relaxation algorithm proposed in Chapter 5, it is assumed that each word in the input text is constrained by a candidate neighborhood generated by a preliminary word classifier. This information is available to the proposed word-level simulated annealing algorithm when it starts.

In this simulated annealing algorithm for word decryption, a *configuration* is defined as a specific assignment of all the selected cipher-words to words in their candidate sets. A neighboring configuration is another assignment of all the cipher-words in which only one cipher-word is assigned a different candidate. The *cost* function of a certain configuration A_i which maps the set of selected cipher-words $(w_1, w_2, ..., w_n)$ to the set of plain words $(w'_1, w'_2, ..., w'_n)$ is defined as:

$$F_{A_i} = a_1 * \sum_{j,k} |f^c(w_j, w_k) - f^p(w'_j, w'_k)| + a_2 * \sum_{k=1}^{n} (1 - c_k)$$

where $f^c(w_j, w_k)$ is the relative frequency of cipher-word sequence (w_j, w_k) in the cipher text, and $f^p(w'_j, w'_k)$ is the relative frequency of word sequence (w'_j, w'_k) in the language base, c_k is the confidence value of cipher-word w_k being w'_k which is provided by the preliminary classifier. In the above formulation, the total cost of a configuration is the combination of contributions from two sources. The first is the degree of word bigram match between the selected portion of text and the language base. The second contributing factor is the algorithm's confidence about the current word assignments. The bigram frequency difference is measured on all the possible two-word sequences from the selected portion of input text, and the confidence is measured on all the selected cipher-words $w_k (1 \leq k \leq n)$. a_1 and a_2 are weights that balance the relative contributions by the two factors to the overall cost, where $0 < a_1, a_2 < 1$, and $a_1 + a_2 = 1$.

A proposed simulated annealing algorithm for word-level deciphering using the above cost function is shown in Figure 7.1, where $\mathcal{F}_{start}$, T_{start} are the cost of the start configuration and the start Temperature, A_i is the current configuration, $\mathcal{N}_i$ is the set of neighboring configurations of A_i, $\Delta\mathcal{F}_{ij}$ is the difference in cost between A_i and A_j, and ρ is a constant less than 1.0.

Simulated_Annealing()

BEGIN

$initialize(\mathcal{F}_{start}, T_{start})$;

for $(k = 1; k < max\_iterations; k++)$

for $(l = 0; l < max\_chain\_length; l++)$

BEGIN

$generate\ A_j$ from $\mathcal{N}_i$;

if $(\Delta\mathcal{F}_{ij} \leq 0$ or $random[0,1] \leq exp(-\Delta\mathcal{F}_{ij}/T_{k-1}))$

then $update(A_i)$;

END

$T_k = \rho T_{k-1}$;

if $(stop\_criterion{==}true)$ **then** $exit()$;

END

END

Figure 7.1: A Simulated annealing algorithm for word level deciphering.

Bibliography

[1] E. Aarts and J. Kost. *Simulated Annealing and Bolzmann Machine: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* John Wiley & Sons Ltd., Chichester UK, 1989.

[2] M.R. Anderberg. *Cluster Analysis for Applications.* Academic Press, London, 1973.

[3] S. Ariyoshi. A character segmentation method for japanese printed documents coping with touching character problems. In *11th IAPR Int. Conf. on Pattern Recognition*, volume II, pages 313–316, August 1992.

[4] L. Bahl. A algorithm for solving simple substitution cryptograms. In *Proc. IEEE Int. Symp. Information Theory*, Ithaca, NY, 1977.

[5] Henry S. Baird. Document image defect models and their uses. In *Proceedings of the Second International Conference on Document Analysis and Recognition ICDAR-93*, 1993.

[6] H.S. Baird. Accurate skew estimation and the top-down analysis of document images. In *Proceedings of the 1st Int. Conf. on Computer Vision*, 1987.

[7] H.S. Baird. Document image defect models. In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structural Document Image Analysis*, pages 546–556. Springer Verlag, N.Y., 1992.

[8] H.S. Baird and G. Nagy. A self-correcting 100-font classifier. In *Proceedings of the Conference on Document Recognition of 1994 S&T/SPIE Symposium*, February 6-10 1994.

[9] D.H. Ballard and C.M. Brown. *Computer Vision.* Englewood Cliffs, New Jersey, 1982.

[10] H.B. Barlow. The mechanical mind. *Annual Review of Neuroscience*, 13:15–24, 1990.

[11] J. Beck, B. Hope, and A. Rosenfeld, editors. *Human and Machine Vision.* Academic Press, New York, 1983.

[12] S. Bercu and G. Lorette. On-line handwritten word recognition: an approach based on hidden markov models. In *Proceedings IWFHR III*, page 385, May 1993.

[13] M. Berthod and S. Ahyan. On line cursive script recognition: A structural approach with learning. In *Proceedings of the International Conference on Pattern Recognition*, page 723, 1980.

[14] I. Biederman. Human image understanding: Recent research and a theory. *Comp. Vision Graphics Image Process.*, 32:29–73, 1985.

[15] Mindy Bokser. Omni-document technologies. *Proceedings of the IEEE*, 80(7):1066–1078, 1992.

[16] R. Bozinovic and S.N. Srihari. A string correction algorithm for cursive script recognition. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-14(6):655–663, 1982.

[17] T. Breuel. Design and implementation of a system for recognition of handwritten responses on us census forms. In *Proceedings of IAPR workshop on Document Analysis systems*, 1994.

[18] V. Bruce and P. Green. *Visula Perception: Physiology, Psychology, and Ecology, 2nd ed.* Erlbaum, Hillsdale, NJ, 1990.

[19] C.J.C. Burges, J.I. Be, and C.R. Nohl. Recognition of handwritten cursive postal words using neural networks. In *USPA 5th Advanced Technology Conference*, pages A–177, 1992.

[20] R. Casey and G. Nagy. An autonomous reading machine. *IEEE Transactions on Computers*, C-17:492–503, 1968.

[21] R. Casey and G. Nagy. Advance in pattern recognition. *Scientific America*, 224:56–71, 1971.

[22] R. Casey and K. Wong. Document-analysis system and techniques. In R. Kasturi and M. Trivedi, editors, *Image Analysis and Applications*, pages 1–35. New York, 1990.

[23] R. G. Casey. Text OCR by solving a cryptogram. In *IEEE Proceedings*, pages 349–351, 1986.

[24] R. G. Casey, S.K. Chai, and K.Y. Wong. Un-supervised construction of decision networks for pattern recognition. In *Proceedings of the International Conference on Pattern Recognition*, July 1984.

[25] R. G. Casey and G. Nagy. Recursive segmentation and classification of composite character patterns. In *Proceedings of 6th International Conference on Pattern Recognition*, pages 1023–1026, 1982.

[26] R.G. Casey. Character segmentation in document ocr: Progress and hope. In *Symposium on Document Analysis and Information Retrievals*, pages 13–40, 1995.

[27] R.G. Casey and E. Lecolinet. Strategies in character segmentation: A survey. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR-95*, 1995.

[28] R.G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-18:690–706, 1996.

[29] M. Cesar and R. Shinghal. Algorithm for segmenting handwritten postal codes. *Int. J. Man Machine Study*, 33(1):63–80, 1990.

[30] C. Chen and J. Decurtins. A segmentation-free approach to OCR. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 190–196, 1992.

[31] M.Y. Chen. An alternative to variable duration hmm in handwritten word recognition. In *Proceedings IWFHR III*, page 82, May 1993.

[32] M. Cheriet, Y.S. Huang, and C.Y. Suen. Background region-based algorithm for the segmentation of connected digits. In *11th IAPR Int. Conf. on Pattern Recognition*, volume II, page 619, August 1992.

[33] M.B. Clowes. On seeing things. *Artificial Intelligence*, 2:79–116, 1971.

[34] M. Cooper. *Visual Occlusion and the Interpretation of Ambiguous Pictures.* Ellis Horwood, Chichester, England, 1992.

[35] T.N. Cornsweet. *Visual Perception.* Academic Press, 1970.

[36] R.G. Crowder. *The Psychology of Reading: An Introduction.* Oxford University Press, New York, 1982.

[37] R. Desimone and L.G. Ungerleider. Neural mechanisms of visula processing in monkeys. In F. Boller and J. Grafman, editors, *Handbook of Neuropsychology, vol. 2*, pages 267–299. Elsevier, Amsterdam, 1989.

[38] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[39] C.E. Dunn and P.S.P. Wang. Character segmentation techniques for handwritten text - a survey. In *11th IAPR Int. Conf. on Pattern Recognition*, volume II, page 577, August 1992.

[40] L.D. Earnest. Machine recognition of cursive writing. In C. Cherry, editor, *Information Processing*. Butterworth, London, 1962.

[41] S. Edelman, T. Flash, and S. Ullman. Reading cursive handwriting by alignment of letter prototypes. *Int. J. of Computer Vision*, 5(3):303–331, 1990.

[42] R.W. Ehrich and K.J. Koehler. Experiments in the contextual recognition of cursive script. *IEEE Transactions on Computers*, 24(2):182, 1975.

[43] L.J. Evett et al. Using linguistic information to aid handwriting recognition. In S. Impedovo and J.C. Simon, editors, *From Pixels to Features III: Frontiers in Handwriting Recognition*, pages 339–348. Elsevier Science Publishers, Amsterdam, 1992.

[44] R.J. Evey. Use of a computer to design character recognition logic. In *Proceedings of the Estern Jt. Comp. Conf.*, pages 205–211, 1959.

[45] C. Fang and J.J. Hull. A modified character level deciphering algorithm for OCR in degraded documents. In *Proceedings of the Conference on Document Recognition of 1995 IS&T/SPIE Symposium*, 1995.

[46] C. Fang and J.J. Hull. A word-level deciphering algorithm for degraded document recognition. In *Proceedings of DAIR 95*, 1995.

[47] R. Fenrich. Segmentation of automatically located handwritten numeric strings. In S. Impedovo and J.C. Simon, editors, *From Pixels to Features III: Frontiers in Handwriting Recognition*, page 47. Elsevier Science Publishers, Amsterdam, 1992.

[48] W. Forsyth. *Solving substitution ciphers using the method of simulated annealing.* PhD thesis, Honors Thesis, Dept. of Math, Statistics and Computing Science, The University of New England, March 1992.

[49] W. Forsyth and R. Safavi-Naini. Automated cryptanalysis of substitution ciphers. *Cryptologia*, XVII(4):407–418, 1993.

[50] P.D. Friday and C.G. Leedham. A pre-segmenter for separating characters in unconstrained hand-printed text. In *Proc. Int. Conf. on Image Processing*, Sept. 1989.

[51] V. Fromkin, editor. *Errors in Linguistic Performance: Slips of the Tongue, Ear, Pen and Hand.* Academic Press, New York, 1980.

[52] K-S Fu and T.S. Yu. *Statistical Pattern Classification using Contextual Information.* John Wiley & Sons Ltd., Chichester, England, 1980.

[53] H. Fujisawa and K. Kurino Y. Nakano. Segmentation methods for character recognition: from segmentation to document structure analysis. *Proceedings of the IEEE*, 80(7):1079–1092, 1992.

[54] P. Gader et al. A segmentation-free neural network classifier for machine-printed numeric fields. In *Proceedings of the 5th USPS Advanced Technology Conference*, December 1992.

[55] P. Gader, M. Magdi, and J-H. Chiang. Segmentation-based handwritten word recognition. In *Proceedings of the 5th USPS Advanced Technology Conference*, December 1992.

[56] R. Ganesan and A. Sherman. Statistical techniques for language recognition: An empirical study using real and simulated english. in preparation.

[57] R. Ganesan and A. Sherman. Statistical techniques for language recognition: An introduction and guide for cryptanalysis. *Cryptologia*, XVII(4), 1993.

[58] J.J. Gibson. *The Senses Considered as Perceptual Systems.* Houghton Mifflin, Boston, 1966.

[59] J.J. Gibson. *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston, 1979.

[60] A.M. Gillies. Cursive word recognition using hidden markov models. In *USPS 5th Advanced Technology Conference*, 1992.

[61] M. Gilloux, J.M. Bertille, and M. Leroux. Recognition of handwritten words in a limited dynamic vocabulary. In *Proceedings IWFHR III*, page 417, May 1993.

[62] A. Goshtasby and R. W. Ehrich. Contextual word recognition using probabilistic relaxation labeling. *Pattern Recognition*, 21(5):455–462, 1988.

[63] V.K. Govindan and A.P. Shivaprasad. Character recognition - a review. *Pattern Recognition*, 23(7):671–683, 1990.

[64] L.D. Harmon. Automatic recognition of print and script. *Proceedings of the IEEE*, 60(10):1165–1177, 1972.

[65] K.C. Hayes. Reading handwritten words using hierarchical relaxation. *Computer Graphics and Image Processing*, 14:344–364, 1980.

[66] H.S. Heaps. *Information Retrieval: Computational and Theoretical Aspects*. Academic Press, New York, 1978.

[67] R.B. Hennis. The ibm 1975 optical page reader: system design. *IBM J. of Res. and Dev.*, pages 346–353, Sept. 1968.

[68] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. A word shape analysis approach to lexicon based word recognition. *Pattern Recognition letters*, 13:821–826, 1992.

[69] Hoffman. Segmentation methods for recognition of machine-printed characters. *IBM J. of Res. and Dev.*, pages 153–165, March 1971.

[70] M. Holt, M. Beglou, and S. Datta. Slant-independent letter segmentation for off-line cursive script recognition. In S. Impedovo and J.C. Simon, editors, *From Pixels to Features III: Frontiers in Handwriting Recognition*, page 41. Elsevier Science Publishers, Amsterdam, 1992.

[71] D.H. Hubel. *Eye, Brain, and Vision.* Freeman, New York, 1988.

[72] D.H. Hubel and T.N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J. Physiology, London*, 195:215–243, 1968.

[73] D.A. Huffman. Impossible objects as nonsense sentences. In B. Melzer and D. Michie, editors, *Machine Intelligence*, volume 6, pages 295–323. Edinburgh Univ. Press, 1971.

[74] J. Hull et al. Combination of segmentation-based and holistic handwritten word recognition algorithms. In S. Impedovo and J.C. Simon, editors, *From Pixels to Features III: Frontiers in Handwriting Recognition*, page 261. Elsevier Science Publishers, Amsterdam, 1992.

[75] J.J. Hull. Hypothesis generation in a computational model for visual word recognition. *IEEE Expert*, pages 63–70, 1986.

[76] J.J. Hull. Hypothesis testing in a computational theory of visual word recognition. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 718–722, 1987.

[77] Jonathan J. Hull and Y. Li. Interpreting word recognition decisions with a document database graph. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 488–492, Tsukuba Science City, Japan, October 1993.

[78] Jonathan J. Hull and Y. Li. Word recognition result interpretation using the vector space model for information retrieval. In *Proceedings of the Second Annual Symposium on Document Analysis and Information Retrieval*, pages 147–155, 1993.

[79] D. Hunter and A. McKenzie. Experiments with relaxation algorithms for breaking simple substitution cipher. *The Comput. J.*, 26(1):68–71, 1983.

[80] A.K. Jian and R.C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[81] M.A. Jones, G.A. Story, and B.W. Ballard. Integrating multiple knowledge sources in a bayesian ocr post-processor. In *Proceedings of the First International Conference on Document Analysis and Recognition(ICDAR-91)*, pages 925–933, 1991.

[82] M.A. Just and P.A. Carpenter. *The psychology of reading and language comprehension*. Allyn and Bacon, Newton, MA, 1987.

[83] Fukushima K and T. Imagawa. Recognition and segmentation of connected characters with selective attention. *Neural Networks*, 6(1):33–41, 1993.

[84] S. Kahan and T. Pavlidis. On the recognition of printed characters of any font and size. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-9:274–287, 1987.

[85] D. Kahneman and A. Henik. Perceptual organization and attention. In M. Kubovy, B.R. Pomerantz, and D.R. Davis, editors, *Perceptual Organization*. Erlbaum, Hillsdale, NJ, 1981.

[86] R.L. Kashyap and B.J. Oommen. An effective algorithm for string correction using generalized edit distance. *Inf. Sci.*, 23:123–142, 1981.

[87] S. Khoubyari and J.J. Hull. Font and function word identification in document recognition. *Computer Vision, Graphics, and Image Processing*, 1994.

[88] H.S. Kim and K.H.Park. Shape decomposition based on perceptual structure. In L. Shapiro and A. Rosenfeld, editors, *Computer Vision and Image Processing*, pages 363–383. Academic Press, San Diego, 1992.

[89] F. Kimura, S. Tsuruoka, M. Shridhar, and Z. Chen. Lexicon directed segmentation-recognition procedure for unconstrained handwritten words. In *Proceedings IWFHR III*, page 122, May 1993.

[90] J. King and D. Bahler. An implementation of probabilistic relaxation in the cryptanalysis of simple substitution cipher. *Cryptologia*, XVI(3):215–225, 1992.

[91] R. Kirby. A product rule relaxation method. *Computer Graphics and Image Processing*, 13:158–189, 1982.

[92] V.A. Kovalevsky. *Character Readers and Pattern Recognition*. Spartan Books, Wash. D.C., 1968.

[93] H. Kucera and W. N. Francis. *Computational Analysis of Present-day American English*. Brown University Press, 1967.

[94] K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439, 1992.

[95] Solomon Kullback. *Statistical methods in cryptanalysis*. Aegean Park Press, Laguna Hills, California, 1976.

[96] A. Kundu, Y. He, and P. Bahl. Recognition of handwritten word: First and second order hidden markov model based approach. *Pattern Recognition*, 22(3):283, 1989.

[97] P.T. Van Larrhoven and E.H. Aarts. *Simulated Annealing: Theory and Practice*. Kluwer Acedemic, Dordrecht, 1988.

[98] E. Lecolinet. *Segmentation d'images de mots manuscripts*. PhD thesis, Universite Pierre et Marie Curie, 1990.

[99] E. Lecolinet and J-V. Moreau. A new system for automatic segmentation and recognition of unconstrained zip codes. In *Proceedings of the 6th Scandinavian Conference on Image Analysis*, 1989.

[100] M. Leroux, J-C. Salome, and J. Badard. Recognition of cursive script words in a small lexicon. In *Proc. Int. Conf. Document Analysis and Recognition*, page 774, 1991.

[101] M.D. Levine. *Vision in Man and Machine*. McGraw-Hill, New York, 1985.

[102] S. Liang, M. Ahmadi, and M. Shridhar. Segmentation of touching characters in printed document recognition. In *Proceedings of the Second International Conference on Document Analysis and Recognition(ICDAR-93)*, pages 569–572, 1993.

[103] Y. Lu. On the segmentation of touching characters. In *Proceedings of the Second International Conference on Document Analysis and Recognition ICDAR-93*, pages 440–443, 1993.

[104] M. Lucks. A constraint satisfaction algorithm for the automated decryption of simple substitution ciphers. In *Advance in Cryptology, CRYPTO'88*, pages 132–144, 1988.

[105] M. Maier. Separating character in scripted documents. In *8th IAPR Int. Conf. on Pattern Recognition*, page 1056, 1986.

[106] J. Mantas. An overview of character recognition methodologies. *Pattern Recognition*, 19(6):425–430, 1986.

[107] David Marr. *Vision*. W.H.Freeman, San Francisco, 1982.

[108] J.D. McCafferty. *Human and Machine Vision: Computing Perceptual Organization*. Ellis Horwood, Chichester, England, 1990.

[109] Marvin Minsky. *Perceptrons*. MIT Press, Cambridge, MA, 1968.

[110] Marvin Minsky. A framework for representing knowledge. In P.H Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

[111] G. Nagy. Optical character recognition: theory and practice. In P.R.Krishnaiah and L.N.Kanal, editors, *Handbook of Statistics*, volume 2, pages 621–649. 1982.

[112] G. Nagy. Efficient algorithms to decode substitution cipher with application to OCR. In *IEEE Proceedings*, pages 352–355, 1986.

[113] G. Nagy. At the frontiers of OCR. *Proceedings of the IEEE*, 80(7), 1992.

[114] G. Nagy, S. Seth, and K. Einspahr. Decoding substitution cipher by means of word matching with application to OCR. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-9(5):710–715, 1987.

[115] R. Nagy, K.H. Wong, and F. Fallside. Script recognition using hidden markov models. In *IEEE ICASSP*, pages 2071–2074, 1986.

[116] T.A. Nartker. *1992 Annual Report of ISRI.* University of Nevada, Las Vegas, 1992.

[117] T.A. Nartker. *1993 Annual Report of ISRI.* University of Nevada, Las Vegas, 1993.

[118] T.A. Nartker. *1994 Annual Report of ISRI.* University of Nevada, Las Vegas, 1994.

[119] K. Ohta, I. Kaneko, Y. Itamoto, and Y. Nishijima. Character segmentation of address reading/letter sorting machine. *NEC Research and Development*, 34(2):248–256, 1993.

[120] B. Oommen and J. Zgierski. Breaking substitution ciphers using Stochastic Automata. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-15(2), 1993.

[121] H. Ouladj, G. Lorette, and E. Petit. From primitives to letters: A structural method to automatic cursive handwriting recognition. In *Proceedings of the 6th Scandinavian Conference on Image Analysis*, 1989.

[122] T. Paquet and Y. Lecourtier. Handwriting recognition: Application on bank cheques. In *Proc. Int. Conf. Document Analysis and Recognition*, page 749, 1991.

[123] S.K. Parui, B.B. Chaudhri, and D.D. Majumder. A procedure for recognition of connected handwritten numerals. *Int. J. System Science*, 13(9):1019–1029, 1982.

[124] T. Pavlidis. Problems in the recognition of poorly printed text. In *Symposium on Document Analysis and Information Retrievals*, pages 162–173, 1992.

[125] S. Peleg. Ambiguity reduction in handwriting with ambiguous segmentation and uncertain interpretation. *Computer Graphics and Image Processing*, 10:235–245, July 1979.

[126] S. Peleg. A new probabilistic relaxation scheme. In *IEEE Conf. on Pattern Recognition and Image Processing*, pages 337–343, Chicago, August 1979.

[127] S. Peleg and A. Rosenfeld. Breaking substitution ciphers using a relaxation algorithm. *Communications of the ACM*, 22:598–605, 1979.

[128] A.P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[129] B. Plessis et al. Isolated handwritten word recognition for contextual address reading. In *USPS 5th Advanced Technology Conference*, page 579, 1992.

[130] A. Pollatsek and K. Rayner. Reading. In M.I. Posner, editor, *Foundations of Cognitive Science*. MIT Press, Cambridge, MA, 1989.

[131] W. Prinzmetal. Principles of feature integration in visula perception. *Perception and Psychophysics*, 30:330–340, 1981.

[132] Z.W. Pylyshyn. Computing in cognitive science. In M.I. Posner, editor, *Foundations of Cognitive Science*. MIT Press, Cambridge, MA, 1989.

[133] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, 1986.

[134] R. Ramesh, G. Athithan, and K. Thiruvengadam. An automated approach to solve simple substitution ciphers. *Cryptologia*, XVII(2):202–217, 1993.

[135] J. Raviv. Decision making in markov chains applied to the problem of pattern recognition. *IEEE Transactions on Information Theory*, IT-13(3):536–551, 1967.

[136] J.R. Rhyne and C.G. Wolf. Recognition-based user interfaces. In H.R. Hartson and D. Hix, editors, *Advances in Human-Computer Interaction*. Ablex, Norwood, N.J., 1993.

[137] S. V. Rice, J. Kanai, and T. A. Nartker. A report on the accuracy of OCR devices. ISRI Technical Report TR-92-02, University of Nevada, 1992.

[138] S.V. Rice, F.R. Jenkins, and T.A. Nartker. The fourth annual test of OCR accuracy. In *1995 Annual Report of ISRI, University of Nevada, Las Vegas*, pages 11–50, 1995.

[139] S.V. Rice, J. Kanai, and T.A. Nartker. An evaluation of OCR accuracy. In *1993 Annual Report of ISRI, University of Nevada, Las Vegas*, pages 9–34, 1993.

[140] S.V. Rice, J. Kanai, and T.A. Nartker. The third annual test of OCR accuracy. In *1994 Annual Report of ISRI, University of Nevada, Las Vegas*, pages 11–40, 1994.

[141] J. Rocha and T. Pavlidis. A new method for word recognition without segmentation. In *Proceedings of EI*, 1993.

[142] J. Rocha and T. Pavlidis. A solution to the problem of touching and broken characters. In *Proceedings of the International Conference on Document Analysis and Recognition(ICDAR-93)*, pages 602–605, 1993.

[143] J. Rocha and T. Pavlidis. Character recognition without segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 17(9):903–909, 1995.

[144] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(6):420–433, June 1976.

[145] A. Rosenfeld and A. Kak. *Digital Picture Processing.* Academic Press, New York, 1976.

[146] D.E. Rumelhart and J.L. McClelland. An interactive activation model of context effects in letter perception: Part 2. *Psychological Review*, 89:60–94, 1982.

[147] G. Salton. *Automatic text processing.* Addison Wesley, 1988.

[148] K.M. Sayre. Machine recognition of handwritten words: a project report. *Pattern Recognition*, 5:213–228, 1973.

[149] J. Schurmann et al. From pixels to contents. *Proceedings of the IEEE*, 80(7):1101–1119, 1992.

[150] M.W. Schustack, S.F. Ehrlich, and K. Rayner. The complexity of contextual facilitation in reading: Local and global influences. *Journal of Memory and Language*, 26:322–340, 1987.

[151] A.W. Senior and F. Fallside. An off-line cursive script recognition system using recurrent error propagation networks. In *Proceedings IWFHR III*, page 132, May 1993.

[152] R. Sennhauser. Integration of contextual knowledge sources into a blackboard-based text recognition system. In *Proceedings of IAPR workshop on Document Analysis systems*, pages 211–228, 1994.

[153] M. Shridhar and A. Badreldin. Recognition of connected and simple connected handwritten numerals. *Pattern Recognition*, 19(1):1, 1986.

[154] H.A. Simon and C.A. Kaplan. Foundations of cognitive science. In M.I. Posner, editor, *Foundations of Cognitive Science*. MIT Press, Cambridge, MA, 1989.

[155] Abraham Sinkov. *Elementary Cryptanalysis: A Mathematical Approach*. Random House, New York, 1968.

[156] British Computer Society. *Character Recognition*. British Computer Society, London, England, 1971.

[157] R. Spillman, M. Janssen, B. Nelson, and M. Kepner. Use of a generic algorithm in the cryptanalysis of simple substitution ciphers. *Cryptologia*, XVII(1):31–44, 1993.

[158] R.K. Srihari. Integrating textual and visual information. In *Proceedings of AAAI-91: National Artificial Intelligence Conference*, 1991.

[159] R.K. Srihari. Linguistic context in vision. In *Proceedings of the IEEE Context-Based Vision Workshop*, 1995.

[160] S. Srihari. *Computer Text Recognition And Error Correction.* IEEE Computer Science Press, Silver Spring, MD, 1985.

[161] Sargur N. Srihari, editor. *Computer Text Recognition and Error Correction.* IEEE Computer Software Press, Piscataway, N.J., 1984.

[162] Sargur N. Srihari. From pixels to paragraphs: the use of models in text recognition. In *Symposium on Document Analysis and Information Retrievals*, pages 47–64, 1993.

[163] C.C. Tappert, C.Y. Suen, and T. Wakahara. The state of the art in on-line handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-12:787, 1990.

[164] G.T. Toussaint. The use of context in pattern recognition. *Pattern Recognition*, 10:189–204, 1979.

[165] A. Treisman. Features and objects in visual processing. *Scientific American*, 254:114–124, 1980.

[166] A. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[167] S. Tsujimoto and H. Asada. Resolving ambiguity in segmenting touching characters. In *Proceedings of the First International Conference on Document Analysis and Recognition(ICDAR-91)*, 1991.

[168] S. Tsujimoto and H. Asada. Major components of a complex text reading system. *Proceedings of the IEEE*, 80(7):1133–1149, 1992.

[169] S. Ullman. Visual routines. *Cognition*, 18:97–159, 1984.

[170] S. Ullman. An approach to object recognition: aligning pictorial descriptions. mit ai memo. Technical report, MIT, December 1986.

[171] D.L. Waltz. Generating semantic description from drawings of scenes with shadows. MIT Technical Report AI271 AI271, MIT, 1972.

[172] J. Wang and J. Jean. Segmentation of merged characters by neural networks and shortest path. *Pattern Recognition*, 27(5):649–658, 1994.

[173] R. A. Wilkinson. Comparison of massively parallel segmenters. NIST internal report, U.S. National Institute of Standards and Technology, Gaithersburg, Maryland, 1992.

[174] R. A. Wilkinson et al. The first census optical character recognition system conference. NIST internal report, U.S. National Institute of Standards and Technology, Gaithersburg, Maryland, 1992.

[175] K.Y. Wong, R.G. Casey, and F.M. Wahl. Document analysis systems. *IBM J. of Res. and Dev.*, 26(6):656–757, 1982.

[176] S. W. Zucker, E.V. Krishnammarthy, and R.L. Haar. Relaxation process for scene labeling: convergence, speed and stability. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8:41–48, 1978.