Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report
RR-90-08

A Step Towards

Understanding Paper Documents

Andreas Dengel

June 1990

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Nixdorf, Philips and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Intelligent Communication Networks
- Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

A Step
Towards Understanding Paper Documents

Andreas Dengel

DFKI-RR-90-08

This report is an extented version of papers published in the proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit, MI, Aug. 1989, and in the proceedings of the International Association for Pattern Recognition Workshop on Syntac-tical & Structural Pattern Recognition (SSPR-90), Murray Hill, NJ , June 1990.

A Step Towards Understanding Paper Documents

Andreas Dengel

Author´s Abstract
This report focuses on analysis steps necessary for a paper document processing. It is divided in three major parts: a document image preprocessing, a knowledge-based geometric classification of the image, and a expectation-driven text recognition. It first illustrates the several low level image processing procedures providing the physical document structure of a scanned document image. Furthermore, it describes a knowledge-based approach, developed for the identification of logical objects (e.g., sender or the footnote of a letter) in a document image. The logical identifiers provide a context-restricted consideration of the containing text. While using specific logical dictionaries, an expectation-driven text recognition is possible to identify text parts of specific interest. The system has been implemented for the analysis of single-sided business letters in Common Lisp on a SUN 3/60 Workstation. It is running for a large population of different letters. The report also illustrates and discusses examples of typical results obtained by the system.

CONTENTS:

1 INTRODUCTION

The rapid dissemination of workstations and their networking to each other has led to a change in the way in which information is dealt with. An unavoidable consequence thereof is an increasing amount of data to be processed. Despite the advances in electronic tools, this fact also leads to an increasing paper consumption [Schäfer and Fröschle, 1986]. Therefore, an effective support of humans in information handling is necessary, and moreover tech-nical support for the transformation of printed documents into electronic form has to be provided.

All activities in an organisation require or produce information. Therefore, a document is not only the main information carrier but also the central topic for an integration of office functions [Donner, 1985]. The human stands at the very center of the office work place, with his own creativity for the drafting and design of documents and his capability to evaluate and make decisions regarding incoming information as to how it should further be dealt with.

In a typical office, information arrives in multimedia (paper, electronic, audio and visual) and in mixed-mode (text, graphic, image, speech and handwriting) form. The carrier for this indirect communication is, in an abstract sense, the document. Person A creates a document. At a later time, person B attempts to interprete this document, in order to extract the transmitted information.

Interests in electronic media and dependency on paper make it necessary to develop interfaces, which allow information to be exchanged between paper and electronic media. This task comprises the syntactic as well as the semantic analysis of a document. As a result, it will become possible to manage paper documents as well as electronic ones in a uniform representation by a centralized electronic archive.

Therefore, the intention is to design systems which embody knowledge about the basic structures of different kinds of documents, as well as a set of characteristics of their components, and the special relations among them. The resulting knowledge sources have to be used to analyze and identify the different logical objects within a paper document, like sender, company logo, or date of a letter, and transmit them into an internal, electronical representation.

Relevant work in this field includes the use of classification and segmentation methods to establish a physical representation of the whole document and the different layout objects within it. Different techniques have been proposed and used, to varying degrees and success. The physical representation of the document page resulted from these preprocessing is the input for a highlevel control structure, that attempts to interpret the several layout objects by their logical meaning, thereby using different knowledge sources.

To automatically "read & understand" documents, classical approaches of pattern recognition, concepts for a suitable knowledge representation and several AI-techniques can be fruitfully combined. Many applications using knowledge-based systems have been developed in the last years. [Woehl, 1984] i.e., illustrates the use of relational data bases coupled with a PROLOG expert system. The applications of production systems for a document understanding have been proposed by [Kubota et al, 1984] and by [Niyogy and Srihari, 1986]. For the analysis of business letters, [Bergengrün et al, 1986] used ATN´s (Augmented Transition Networks) in combination with fuzzy relationships. To model syntactical knowledge about paper forms [Domke et al, 1986] proposed the application of Petri-Nets and finally the use of X-Y trees for the representation of information about a document image has been described by [Nagy and Seth, 1984].

This report describes several aspects serving a basis for a knowledge-based document processing system called ANASTASIL which means: *Ana*lysis *S*ystem *t*o Interpret *A*reas in *Si*ngle-sided *L*etters.

First, the several preprocessing steps of the document image are illustrated in Section 2. Thereby, the report concentrates on an approach for skew angle detection within document images. This is a necessary early step in document analysis, because a small skew causes problems to further processing steps. Furthermore, the report describes the entire segmen-tation procedure to obtain a layout representation of documents.

In Section 3, a special kind of page layout description is introduced and used to establish a so called *geometric tree*. This foundamental knowledge source represents knowledge at different layout abstraction levels. The nodes of the tree contain hypotheses for different logical objects, like *date* or *receiver* of the letter. In addition, the design of the statistical data base (SDB) as a second knowledge source is decribed. It moreover illustrates the use of the geometric tree and the SDB to identify several layout objects of a document page as logical objects.

Section 4 proposes a mixed character/word recognition approach that is based on the knowledge of having logical objects, like *receiver*, or *sender* of a letter, thus, using specific logical dictionaries. Finally, experimental results are discussed in Section 5.

2 DOCUMENT PREPROCESSING

As a starting point, we examine a scanned paper document (pixel matrix). For later stages (document classification, graphics coding, etc.) it is an essential early step to apply an automatic segmentation procedure. The goal thereof is to split the binary image into component-like characters, text-lines and text-blocks, as well as graphics- and image-parts. Most bottom-up clustering methods are working in a very time consuming way [Srihari and Zack, 1986]. Top-down segmentation, by contrast, is fast but vulnerable to non-zero skew angles. Most of the methods are able to find composite areas of black pixels, but a significant amount of skew can cause problems. Consequently, the virtual skew scan has to be canceled by rotating the document image after having detected its skew angle.

2.1 SKEW ANGLE DETECTION

The procedure we use to handle the problem of skew angle detection is called *left margin search* (LMS) [Dengel and Schweizer, 1989]. After having scanned a document, we obtain an image of about 1400 * 1750 pixels. Following the approach of [Trincklin, 1984], we create a vector V of size n, with $n \leq 1750$, denoting line numbers. Consequently, we choose a column j of the image which is on the right of the boundary of the scanned document. V contains at position i the distance between column j and the first black pixel at line i. Taking the example of Fig. 1, the vector V contains at positions 1 to 10 the following values:

i	1	2	3	4	5	6	7	8	9	10	...
$V[i]$	0	43	43	45	39	37	32	28	24	21	...

In the next step we try to determine straight line segments which describe the left margin of text lines or blocks. This is different from [Trincklin, 1984], where all possible straight line orientations are taken into account. Trincklin´s strategy causes problems in the case of documents with a complex structure containing many small text blocks. Considering the LMS approach, it is not relevant how much text blocks a document may contain, because the horizontal space between them is not considered. The gradient of the straight line defines its angle with respect to the vertical orientation. By weighting it with the length of the line we get a distribution function over all possible angles. The dominant angle (maximum) defines the skew 𝕷 of the image data (see Fig. 1).

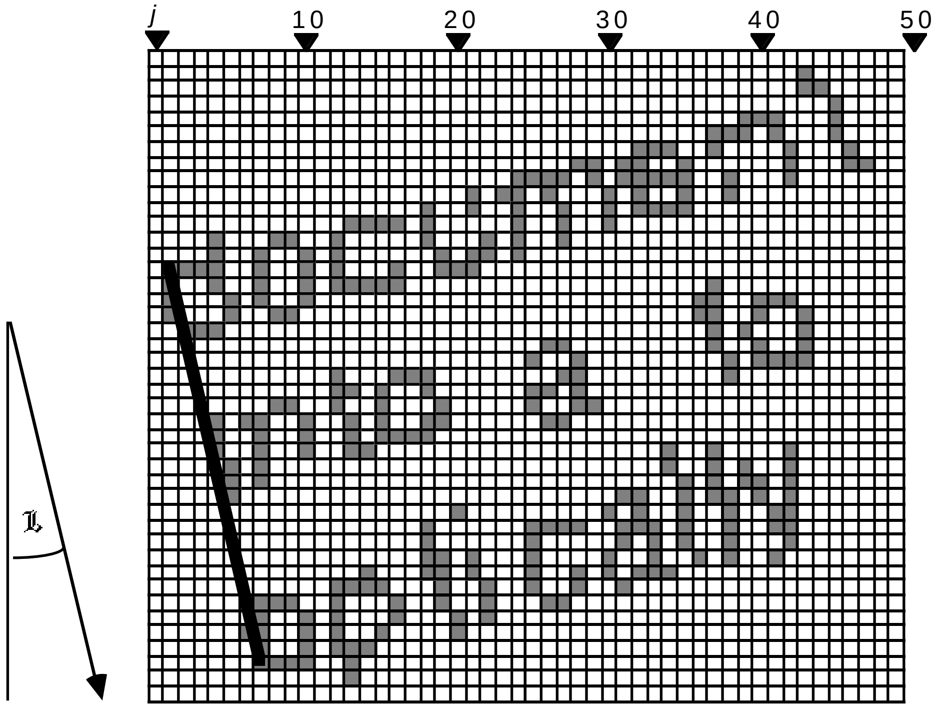


Figure 1: Example for Document Image scanned with 75 dpi.

To realize this goal, we established a filter to eliminate all points *P(i, V[i])* that are not relevant for skew angle detection.

The algorithm starts with an empty set T_0, to which point *(1, V[1])* is added. Continuing with $i = 2$, the procedure tests for every two points *(i-1, V[i-1])* and *(i, V[i])* if they are adjacent, whereby the adjacency is defined along the eight main directions.

If these two points are adjacent, point *(i, V[i])* is added to T_0. Otherwise the procedure tries to find a point, which has an Euclidean distance to point *(i-1, V[i-1])* that is less than one-and-a-half times a factor ϑ, which describes the minimal line height for T_0. ϑ is taking into account line spacing and basically is initialized by a six pixel font height. If no point is found that satisfies the criteria, a new set T_1 is opened and the algorithm iterates.

This filtering procedure creates different sets of points, which describe the left margins of text lines or blocks. Using the method of minimal distances, the procedure determines straight line segments of maximal length, which define the dominant skew within the document image.

In [Dengel and Schweizer, 1989], the LMS approach is compared to the methods of [Trincklin, 1984] and [Postl, 1986] with respect to results, accuracy of results and run-time. Thereby, the resulting skew is defined by the angle with the highest measure of belief. Accuracy of a result means the dominance, in which the resulting angle can be determined. All three approaches have been implemented on a SUN 3/60 in C. The comparison is illustrated giving three examples: a business letter, a news paper which is relatively smudged and an article. All these examples have been considered with two different skew angles. In Figure 2, the grey horizontal lines starting from the right side of the document image, represent the elements of vector *V*.

In Table 1, the appropriate results are listed. Assuming there is a dominant textline orientation, the LMS may be expected to run 3 times faster than the approach of Trincklin and 10 times faster than Postl´s approach. On the average, our approach needs about 0.8 seconds to detect the skew angle in a 2.5 million pixel object (see also Fig. 2).

All three approaches are integrated in the ANASTASIL system [Dengel and Barth, 1989] and can be excellently used for document image analysis.

Figure 2: Skew Angle Detection in Document Images. *V*-Values of Zero are not considered.

Document	Image Size	Exact Scew	Smallest Squares [Trincklin]		Sm. Squares [Trincklin] (mod.)		Left Margin Search		Simulated Skew Scan [Postl]	
			Scew	Time	Scew	Time	Scew	Time	Scew	Time
1	3507x2480	btw. -10 a. -9	-10	6199	-10	4866	-9	2816	-10	13282
1	3507x2480	0	0	5899	0	4383	0	2500	0	13332
2	1727x1423	19	19	1749	20	1299	20	533	19	8066
2	1755x1445	-19	-19	2149	-19	1633	-15	683	-19	8149
3	2333x1650	9	9	3033	9	2299	9	799	9	13332
3	2333x1650	btw. -4 a. -3	-4	3233	-4	2416	-4	900	-4	13349

Table 1: Document Image (in dots), exact skew and recognized skew by the three approaches (in Grad [°]), as well as CPU time (in msec).

2.2 SEGMENTATION

The first step of the segmentation phase is to separate the bilevel (textual- and graphical) information into regions with only textual or only graphical information. To segment a document image into logical text lines or graphical areas, a lot of obstacles have to be removed. First, different criteria have to be determined to separate graphical and textual information. One method to get this distinction is to use projection profiles [Masuda, 1985]. Profiling is a top-down strategy that first attempts to detect layout objects of higher levels, like, in the case of text, sections or paragraphs and consequently decomposes these segments to lower levels e.g. text lines, words or characters.

After skew correction, the horizontal projection profile of text blocks is characterized by having ranges of thick black peaks, separated by white gaps of non-pixel areas. Graphics, by contrast have a relatively uniform profile. Figure 3 shows typical differences in horizontal projection profiles of text and graphical information.

Figure 3: Typical Horizontal Projection Profile of Text and Graphics.

Because our approach will be primarly applied to business letters which are normally established in a single-column representation, we prefer the profiling segmentation technique. We also have tested a smearing approach [Srihari and Zack, 1986], [Schweizer, 1989]. Smearing, by contrast is a bottom-up technique, but because efficiency is the primary goal for our system, we are using the profiling method.

Using projection profile cuts [Nagy et al, 1986], which are obtained by projecting a document image into horizontal and vertical axes, the syntactical structure of the document is established by stepwise refinement. As a result of the segmentation procedure the iden-tified layout components are mapped into a hierarchical data structure, describing the physical representation of a given document. For each component, internal information, e.g., position (left-upper corner), surrounding rectangle, and in the case of characters, chain code and number of holes is stored. In Figure 4, the scheme of physical representation of a document page is illustrated by an example.

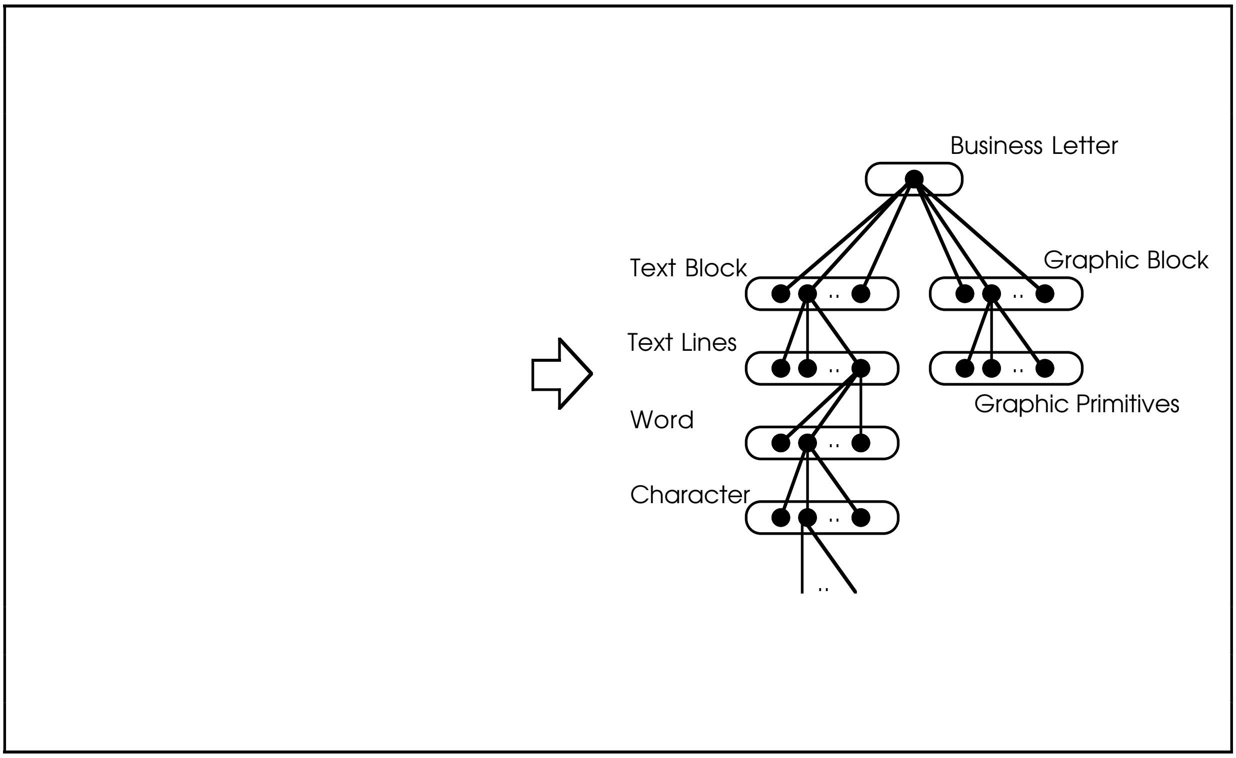


Figure 4: Scheme for a Syntactical Description of a Document after Segmentation.

3 LAYOUT CLASSIFICATION

The result of the preprocessing is the detection and representation of the layout for a given document. The tasks of layout classification comprise the assignment of semantics to objects of the layout structure, so that the logical objects, like a sender or a date of the document, are determined (c.f. Fig. 5).

The task of automatic comprehension of a document is one of the necessary and most important goals of the document analysis. If it will become possible to automatically extract the logical structure of a document, it would be easier to initiate a further analysis of a specific document part. If for example the layout blocks reffering to the addressee are recognized, a procedure for optical character recognition (OCR) enables furthermore the re-cognition of the containing text by matching character-strings with prestored sets of ZIP codes or destination cities [Srihari et al, 1987].

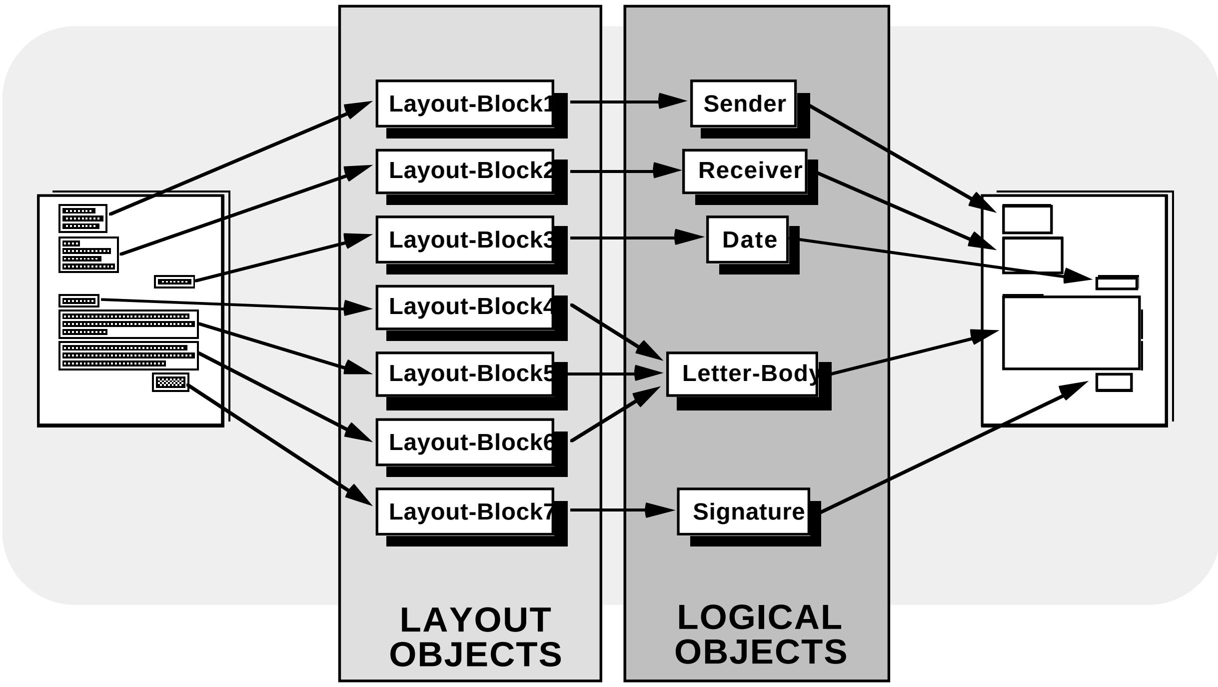


Figure 5: Geometric Structure Analysis.

Shapes of documents differ greatly. Documents represent applications, reports, protocols, letters and other texts, but also plans and drafts. Any document is characterized by its contents and its internal organization. There are documents with a prescribed structure and documents having a more complex and flexible structure. If it would become possible to describe such structures in a homogenious way, it would also be possible to determine the conceptual structure of a document by using AI techniques.

This Section describes the development and use of a knowledge base for document layout classification. For that purpose, a special layout model, a so called geometric tree has been developed to generate working hypotheses about a logical meaning of layout blocks in a document. To verify the hypotheses, an additional statistical data base (SDB) is used. It contains a set of local geometric features of all possible logical objects in business letters. Thus, a hypothesize & test strategy is performed that completely avoids text recognition. Branching in the geometric tree is directed by different measures of similarity. Thus, the system performs a best-first search, which represents a kind of the uniform-cost search, proposed by [Barr and Feigenbaum, 1981].

3.1 DOCUMENT LAYOUT MODELLING

To establish a document layout model, knowledge structure and composition of information in the document is used. Knowledge about document classes which has been obtained by experience as well as from empirical tests serves as a basis for the execution of this rough analysis.

Office documents which are created with today's text systems normally fulfil a minimum amount of formal criteria. This trend towards standardization makes identifying indivi-dual logical parts a lot easier, e.g. receiver or sender as a result of the typical layout of a letter.

Typical images of a printed document have several contiguous regions or blocks, that correspond to logical parts which are significant for a specific document class. Considering business letters, it is possible to state the following characteristics:

- ◊ Some of the letter parts are mandatory, some are optional. Business letters range from simple letters, containing few logical objects (sender, receiver, date, subject and letter-body) to complex letters with several additional parts, like logos, graphics, company specific printings, etc.
- ◊ Logical objects have certain geometric and semantic features.
- ◊ Spatial relationships hold among the different logical objects.

Document image analysis is a search problem, whereby the search space is the entire image. A digitized document page forms a binary two-dimensional space.

The effectiveness of model-based reasoning depends on the certainty and completeness of the underlying model. Any document is characterized by its content and its internal orga-nization. Thus the electronic representation of a document must capture both, the represen-tation of its contents, as well as the document's layout and logical structure. To describe structural knowledge, we have developed a formalism for document page representation, which takes into account the position of information within a document page. The struc-tural elements of a document page, like columns, paragraphs, titles, lines and words of text are generally laid out as rectangular blocks. Orientation of textual information is along horizontal and vertical directions, determined by the rectangular shape of a typical sheet of paper.

Thus, a document page is considered as a rectangle, having a characteristic width and height. To describe its spatial structure, the page is divided into smaller rectangles by vertical and horizontal cuts. Cuts are placed in such a way that they do not intersect with textual or graphical areas. The subrectangles can recursively be divided in the same way, until the layout of the page is described in sufficient detail. To refine annotate the logical structure, different rectangles are assigned a label, which describes their logical meaning.

We therefore use the following definition:

Rule:

For each refinement step in document layout description, choose one of the following possibilities:

1) The rectangle is left unchanged.
2) The rectangle is assigned a semantic label, which represents a hypothesis for the parts it contains.
3) The rectangle is cut along one direction (horizontal, vertical) by one or more cuts and 1) or 2) are executed.

Consequently, most document pages can be partitioned into nested rectangular areas by order, position and orientation of cuts and by assignment with logical labels. Figure 6 shows an example of a partitioned and labeled letter.

Figure 6: Structure Representation of a Letter
- sender (designated as ´F´), receiver (´T´), subject (´S´), date (´D´), body of letter (´B´) and white space (´W´) -.

To transform the definition, we use a special notation with identifiers :H (horizontal), :V (vertical) to the orientation of a cut and :L to name a logical label. Additionally, we use real numbers, so that the letter shown in Figure 2 can be represented by the following list:

```
(:H 0.2 0.3 0.45   (:L  ´F)
                   (:V 0.6  (:L  ´S)
                            (:L  ´D))
                   (:L  ´T)
                   (:L  ´B))
```

To model knowledge about more than one document layout and to obtain a very compact knowledge representation, we use a special form of a decision tree. It describes a global class hierarchy for possible document layouts permitting to describe them in different specifi-cation levels.

The tree consists of a number of nonterminal nodes, including a root node, and a number of terminal nodes. Each nonterminal node in the tree hierarchy represents a different document layout class. Terminal nodes represent results of complete interpretations, their labels indicate the different logical objects and their positions found in this document class. Figure 7 illustrates the principle of the decision tree, which, because of its outlook, we call a geometric tree.

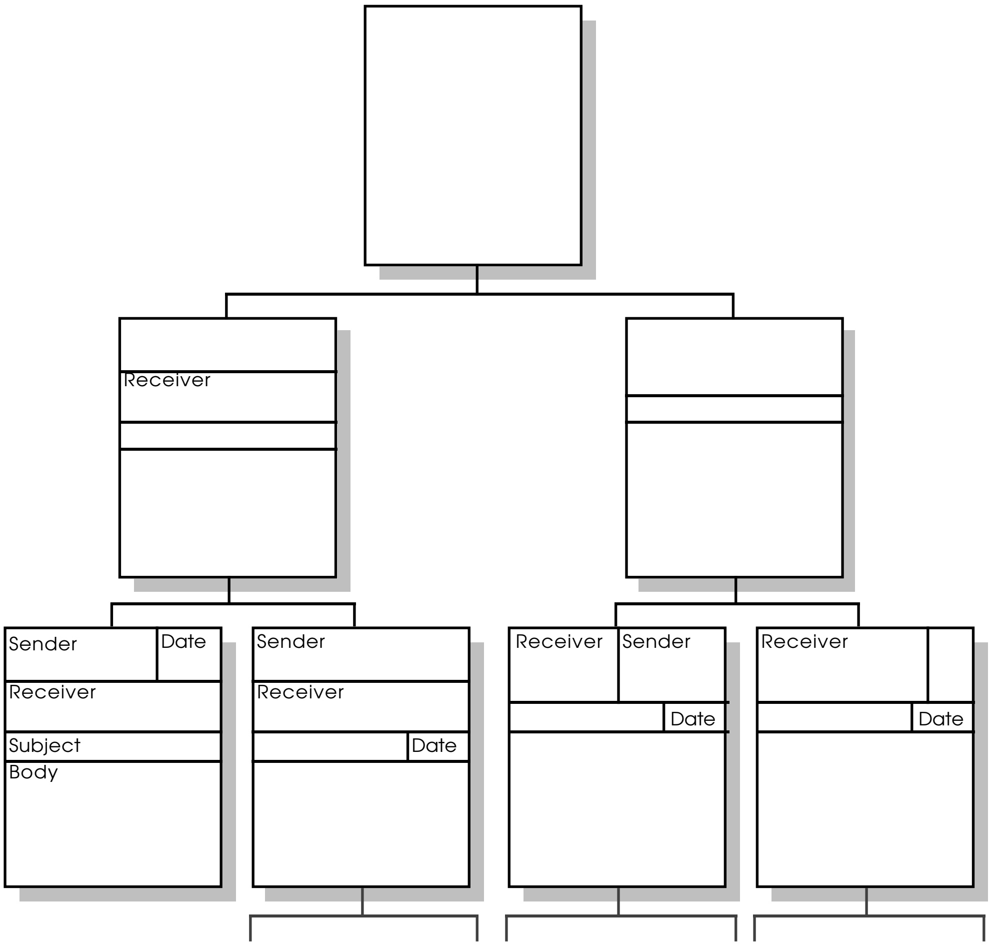


Figure 7: Principle of a Geometric Tree.

The advantages of the tree structure include:

- a guided search from an abstract towards a more concrete layout is possible,
- redundant layout information is avoided,
- a document layout can be described at different levels of specificity, dynamically adapting to the amount of information available.

In a geometric tree, a node is a specialization of its parent node, while at the same time it is a generalization of its child nodes. Thus, the root of the tree represents the most general document layout description. Every document belongs to this class. The internal repre-sentation is organized in a way that parental characteristics, e.g. their layout features, are inherited by children. Moreover, only the layout features which distinguish a child from its siblings are stored at child nodes. Common aspects are stored in the parent node [Dengel and Barth, 1989]. The main advantages of such a tree are due to its straight-forward represen-tation of position, extension, size and spatial relationships of objects within the domain of document page representation and subsets of it. The number of terminal nodes within the tree used in our experiments is about 41.

We take the notation described above and transform this knowledge representation in a combined *list-in-list* implementation. One level represents the different alternatives, whereas at the other level, complete structures are described.

Another major advantage of this model is the fact that at each level no area of the document page is left unaccounted for. As a consequence thereof, the model is fault-tolerant with respect to preprocessing errors. Should the address of a letter, which usually consists of one single block, be erroneously split into different lines by the segmentation procedure, the lines are still contained within the area hypothesized to contain the address. This is quite different from other models that attempt to classify each single block in a document. However, completeness and certainty of model knowledge is responsible for the effective-ness of model based reasoning. We do not want to propose our approach as the universal solution, but it behaves pretty well for most practical applications.

3.2 STATISTICAL DATABASE

Several layout classes of the geometric tree described above only represent possible basic structures of documents within the document class business letter. But there is no modelling of knowledge about different logical parts within business letters, like the receiver, the company logo or company specific printings.

To bridge this gap, we have established a Statistical Database (SDB). It contains results of a statistical evaluation of a few hundred business letters. As a result thereof, we describe the usual logical parts of a business letter by their general intrinsic characteristics. For example, the receiver in a business letter can usually be characterized by the following features:

- the position the receiver is in the first upper third of the page;
- the left margin of the receiver is within the left quarter of the page;
- the horizontal extension is not longer than a third of the page width;
- the receiver is not written in an extremely large or small font;
- the receiver consists of four to six text lines, which are left justified.

Note that there are no features with respect to content information. Up to this processing step, we do not employ any recognition procedure for the analysis of textual information. To limit the search time and to avoid all problems in connection with textual recognition, we concentrate only on geometric features of logical objects. In this sense, we describe the following logical parts:

- sender,
- receiver,
- letter-body,
- date,
- theme (subject, title,..),
- preprintings,
- footnote,
- company-logo,
- company-specific-printings,
- sender-short-form,
- initials
- signature.

The different features we use are transformed into subsets of *if-then* rules, one for each feature. These subsets are organized as **case**-constructs and again form a global set of description rules for an entire logical part. In the action part of a rule, we store measures of belief (MB) or measures of disbelief (MD), corresponding to probability values. Thus, a rule has the form:

if <*feature*> **then** <*measure -of-(dis-)belief*>

The measure-of-(dis-)belief thereby reflects the probability resulting from the examination of a specific logical part while considering a certain feature. The possible values for a measure-of-(dis-)belief range between 0 and 1, whereby the measure-of-disblief is defined as:

measure-of-disbelief = 1 - measure-of-belief.

Lateron during analysis, the rules will be given to the classification procedure as knowledge for understanding labeled-area items of a business letter. To know how a specific value will then be used, we furthermore assign to each rule whether it will be used for confirmation (MB) or refutation (MD) of the assigned label. Figure 8 shows an appropriate set of rules foe locating the receiver:

```
case Y_origin   ≥ 1/7 * page_height and  ≤ 1/4 * page_height
                       then <confirm with 0.90>;
                ≥ 1/4 * page_height and  ≤ 1/3* page_height
                       then <confirm with 0.09>;
                else  <refute with 0.99>.

case X_origin   ≥ 0 and  ≤ 1/4 * page_width
                       then <confirm with 0.98>
                else  <refute with 0.98>.

case block_width      ≥ 0 and  ≤ 1/3 * page_width
                             then <confirm with 0.90>
                      else  <refute with 0.90>.

case  maximal_line_height     normal
                                     then <confirm with 0.99>
                              else  <refute with 0.99>.

case line_count       = 3
                           then <refute with 0.95>;
                      ≥ 4  and  ≤ 6
                           then <confirm with 0.93>
                      else  <refute with 0.98>.

case block_count      > 0  and  ≤ 2
                           then <confirm with 0.90>;
                      = 3
                           then <refute with 0.91>
                      else  <refute with 0.99>.

case left_justified   true
                           then <confirm with 0.99>
                      else  <refute with 0.99>.

case       one_word_first_line    TRUE  then <confirm with 0.99>

case       last_line_separated    TRUE  then <confirm with 0.71>
```

Figure 8: Rules describing the Logical Object *Receiver*.

The whole SDB consists of about 71 such rule packages [Butz, 1988]. This organization of the SDB allows a straightforward testing of intrinsic features of given layout blocks as well as its easy extension by new rules. When new rules are added to the database, the measures of belief and disbelief for existing rules may have not to be altered, because every subset is independent from each other.

3.3 CONTROL STRUCTURE

Knowledge based document classification amounts to the assignment of a logical meaning to distinct layout components within a given document, using the explicit knowledge of the document model.

Within our approach, we have to find a path from the root of the geometric tree to one of its leaves which represents the layout class of a given business letter and describes its layout. With each step on this path, the document at hand has to be matched with document layout classes, represented by respective nodes. While matching a given business letter, several resulting rectangular areas are assigned a logical label. A label thereby indicates a hypo-thesis for the possible constituent within the appropriate area. To verify a hypothesis, the appropriate set of rules in the SDB has to be evaluated. Figure 9 despict the principles of the classification procedure, based on this hypothesize & test strategy.

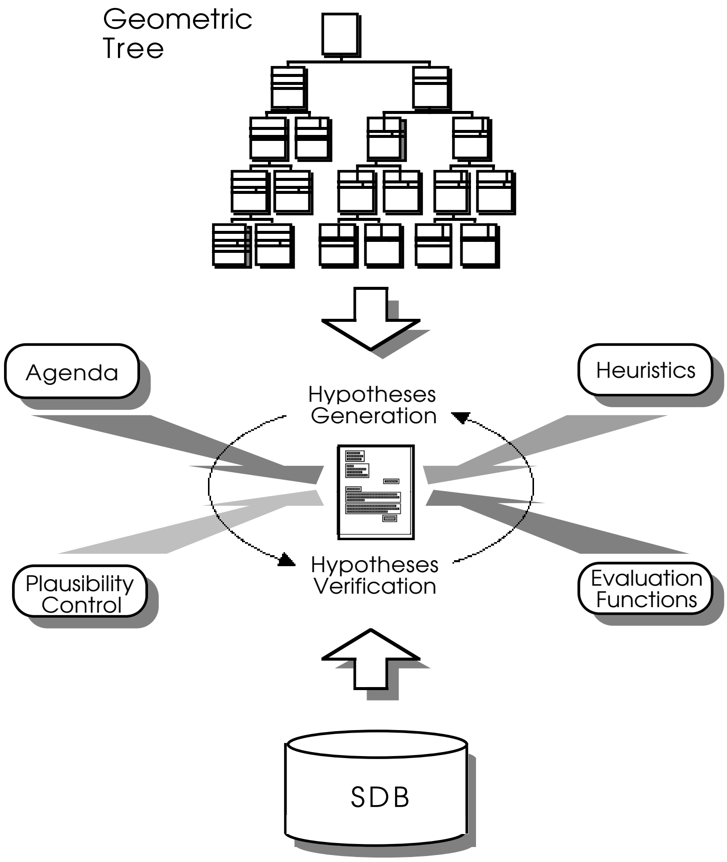


Figure 9: Hypothesize & Test Strategy of the Control

During this procedure, several modules of control use the explicit knowledge about the underlying model, as well as supplementary implicit knowledge. In addition to an agenda and several evaluation functions [Dengel, 1989], implicit knowledge is provided by a plausibility control. It comprises for example, knowledge about possible cut positioning as well as knowledge resulting from a consistency check for plausible cut combinations.

While searching for a path, branching in the tree is directed by the measures of belief (MB) and disbelief (MD). The measures available for a match are obtained by two steps:

- matching the layout of a given business letter with a layout class of the geometric tree and
- evaluation of appropriate rules in the SDB.

Starting at the root, in each step the actual document is matched with the two sublayout-classes of the actual node. The degree of similarity between the layout of the given document and the nodes in the model have to be quantified. Semantic labels in distinct nodes represent hypotheses about the semantic meaning of the contained parts. To verify a hypothesis, the features defined in the SDB have to be compared with the blocks of the area to be examined. Each inspected node gets a measure of belief (MB) for its similarity with the actual document. This measure of belief is composed of a confidence value for its quality of cut matching as well as evidence for hypotheses verification. We use an agenda to store the different intermediate conclusions and conduct a best-first search in the geometric tree. Thus, several "hypothesize & test" processes are performed, and the system reaches a satisfactory conclusion no sooner than a leaf in the geometric tree is reached and all areas of the document are labeled.

While matching a given document with a layout class of the geometric tree, we try to take into account small variations within a document´s layout. Therefore, we allow small shifting of the cuts with respect to their original positions in the layout classes. If a cut position intersects any textual or graphical block of the document, the control mechanism searches for some alternative positions. The validation function for cuts (see Formula 1) works in such a way that the amount of shifting the original position is computed. When looking for a different alternative position *x*, small shifts should not count as much as large shifts. Thus, the quality of each cut is based on a measure of belief *v(x)*, which is calculated by the following validation function.

$$v(x) = \begin{cases} f_1(x) * r_1(x) & l_1 \le x \le c \\ f_2(x) * r_2(x) & c < x \le l_2 \end{cases} \qquad (1)$$

Thereby, c denotes the original position and l_1 and l_2 describe the boundaries of the area to be partitionated. f_i and r_i are defined by the Formulas 2 and 3.

$$f_i(x) = 1 - \left[\frac{(x - c)}{(l_i - c)} \right]^2 \qquad (2)$$

$$r_i(x) = \left| 1 - \frac{(x - c)}{(l_i - c)} \right|^n \qquad (3)$$

$f_i(x)$ defines the entire validation function for cut shifting within the interval $[c, l_i]$, whereas $r_i(x)$ denotes a factor which determines the entire curve of the function. The curve of the function can be altered depending on the degree of layout-standardation of the underlying document class.

Therefore, the variable $\boldsymbol{n}$ is assigned a non-negative value, whereby a lower value indicates less standardization and a higher one more standardation. Figure 10 shows the semantics of the formulas with $n = 0,..,3$.

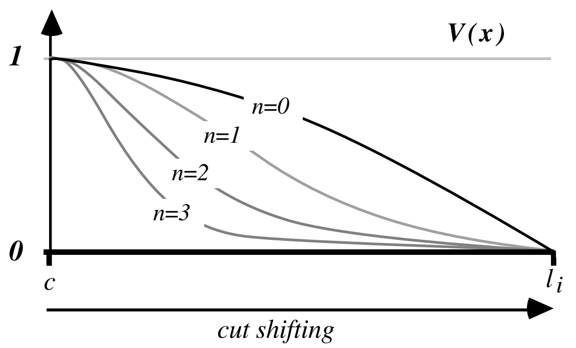


Figure 10: Curve of the function depending on *n*.

In case a cut like the one in Figure 11 intersects a block (shaddowed area), the alternatives a_1 and a_2, are the closest possible positions for a cut with respect to its original position. The measure of belief for the two alternatives is calculated by means of a partially defined function (see Formula 1). The cut has to be placed somewhere between the delimiters l_1 and l_2 of the area under consideration. They confine the domain of the functions. The functions are converging in point *c*. The maximum of *v(x)* is 1, in which case no alternative position must be searched. The minimum 0 is attained at the boundaries l_1 and l_2.

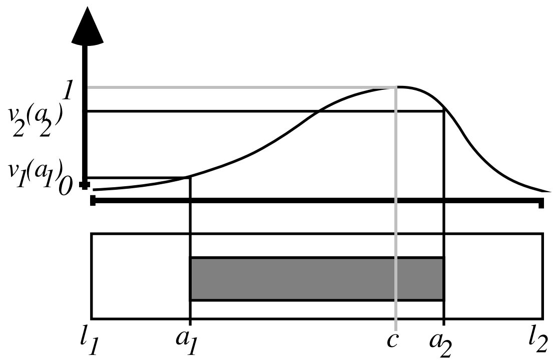


Figure 11: Example for cut-validation.

Usually there is more than one possible cut for each node in the geometric tree. Hence, for every match with a layout class we fold the values of *v* for each of the cuts into a single confidence factor. The weighting of the single cut values is done according to the relative lengths of the cuts. Since we require the function values to be less than 1, we normalize them with respect to the total length of the cuts from the root of the geometric tree to the respective node. That means:

The contribution of each cut to the total measure of belief for the layout equals its share of the total cut-length in the document.

This yields the following expression for V_i:

$$V_i = \frac{1}{C_i}\left(V_{i-1} * C_{i-1} + \sum_{j=1}^{k_i} v_{ij} * c_{ij}\right) \qquad (4)$$

$$V_0 = v_{00} = 1$$

In (4), *i* denotes the level of a tree node (0 being the root), P_{i-1} is the parent node of node P_i. The number of cuts in a node is k_i.

The terms v_{ij} denote measures for cuts *j* of length c_{ij}, $1 \le j \le k_i$. The normalization factor is the total length C_i of all previous cuts:

$$C_i = C_{i-1} + \sum_{j=1}^{k_i} v_{ij} * c_{ij} \qquad (5)$$

$$C_0 = c_{00} = 0$$

The result of the validation process only denotes the quality of cut matching. All patterns obtained as plausible combinations of cut positions are used for further examinations. In other words: all areas getting labels (hypotheses) during this pattern matching step, have to be verified.

The labeling of a specific area amounts to matching the containing layout blocks against the appropriate set of description rules in the SDB. For that reason, all layout components within a labeled-area item are considered as belonging together [Dengel and Barth, 1988]. The corresponding and resulting probabilities expressed by the values for MB and MD within the action-part of the rules have to be combined into a confidence value that reflects the similarity with a certain logical part.

Whereas probabilistic approaches, like the Bayesian formulas, refer to conditional probabilities, the SDB is based upon the combination of completely independent events with their respective probabilities, To this end, we use Dempster-Shafer´s rules of combining probabilities [Shafer, 1976]. Every two pieces of single MB´s and MD´s for specific features f_i are iteratively combined by the following formulas:

$$MB(H, f_1 \text{ and } f_2) := MB(H, f_1) + MB(H, f_2) - MB(H, f_1) * MBH, f_2)$$

$$MD(H, f_1 \text{ and } f_2) := MD(H, f_1) + MD(H, f_2) - MD(H, f_1) * MD(H, f_2)$$

As a result, we obtain two global values for the **confirmation or refutation** of a certain hypothesis *H*, respectively. To combine both values, another formula is used. It expresses for every hypothesis *H* the confidence value $MB_{Logik}(H)$ for its **confirmation and refutation**.

$$MB_{Logik}(H) := MB(H) * (1 - MD(H)) / (1 - MB(H) * MD(H))$$

Combining all resulting measures of belief by a specific formula, we are able to quantify the degree of similarity between the current document layout and the nodes in the model. While handling different intermediate results for document layout class matching, the use of the decision tree allows for easy reduction of the search space. Moreover, the user gets the comfortable feeling that he is controlling the situation such that some classification errors in the last analysis step can be immediately corrected by backtracking.

All intermediate decisions are collected in an agenda [Dengel and Barth, 1988]. In every stage of the analysis, it provides the best intermediate solution for further examination. That means, to perform one step on the path in the geometric tree, the pattern with the highest measure of belief is chosen to generate a more specific layout. Thus we perform a best-first search, which represents a variant of the uniform-cost search, proposed by [Barr and Feigenbaum, 1981].

In [Dengel and Barth, 1989], we have also described an additional knowledge acquisition component, which allows for easy modification and extension of the geometric tree and the SDB. While using a graphics editor, the user himself is able to establish new document layout classes by setting cuts and labels. Subsequently the graphical pattern is automatically converted into the internal notation of the geometric tree. The graphic editor also provides the ability to define new rules about geometric properties of certain logical parts of a business letter. They also are converted automatically into the internal rule syntax. Consequently, a model generator adjusts the existing knowledge sources to the new information. In this sense, our system is capable of virtually classify all considered busi-ness letters. In Figure 12 different examples of classification results obtained by our system are illustrated.

Figure 12: Results of document classification

4 TEXTUAL ANALYSIS

The result of a knowledge-based analysis of the geometrical characteristics of a document page are logical objects that can be considered as document parts describing a restricted logical discourse. Moreover, there exist logical relationships between them. But before detecting such relationships, it is necessary to recognize the containing text. Starting with this task, we can use the knowledge about having restricted logical discourses. In this sense, we are able to establish logical dictionaries, containing specific vocabulary of individual logi-cal objects. For example, the logical object *receiver* contains names of all possible receivers in a company.

To convert textual information of paper documents into a representation which can be handled by computer (e.g. ASCII format), word recognition or a character recognition should be implemented. Therefore, two distinct ways for the recognition process exist (see [Ullman, 1982]).

The first recognition method separates words into isolated characters and then tries to identify them. As long as it is possible to separate the individual characters, even hand writing can be read with high reliability [Schürmann, 1987]. Problems occur when the res-pective system is not trained to deal with the font at hand or characters remain joined. There are also problems caused by an interruption during transmission.

The second, more general, method attempts to recognize entire words using a reference pattern. This method is used predominantly in the case of documents where the type face is bad and words cannot be decomposed into their individual characters, e.g. cursive fonts [Farag, 1979]. This process requires the management of a large number of comparison patterns and only a small range of them is likely to be used during recognition. Nevertheless, the storage of words in dictionaries in the context of a semantic analysis is required anyway. In practice, nearly all approaches are concentrating on the recognition of isolated single characters, only a few applications are considering the word recognition approach [Schürmann, 1978], [Srihari et al, 1987].

The procedure we propose, is a mixed character/word recognition approach, which uses chain code information of basic layout parts (characters) as its essential part. The goal of the approach is to identify basic layout objects as belonging to a specific set of characters (nonterminals). While using information about what basic layout parts form a single word, the procedure provides a set of alternative strings corresponding to word hypotheses. Using additional logic-specific dictionaries and several grammar rules, the nonterminals will be derived to specific characters (terminals).

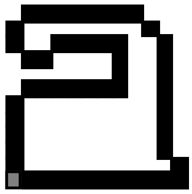
The strategy for this approach bases on the ability to identify areas of a business letter which contain one of its specific constituents and therefore restrict the possible information context. In this sense, the procedure is capable for some of them to recognize the containing textual information, having a restricted word context. This assumption is not global for the whole document, but in the case of e.g. a sender, a receiver or a date, it can be applied with excellent results.

4 .1 CHAIN CODE REDUCTION

A chain code is a boundary description for a certain region. In our case it describes the boundary of a cluster of connected black pixels that correspond to basic layout components of textual information [Scherl et al, 1980]. Starting from the left-lowest point of the cluster, the code is formed by following its boundary in a clockwise direction. Thereby, each of the eight main directions is coded by a special orientation code (numbers

0, 2, 4, 6 e.g. correspond to eastern, northern, western and southern boundaries respectively). In Fig. 13 the cluster of an “ a” and its chain code representation are described.

Starting Point →



Chaincode: $2^5\,10^6 2^3 4^4 543210^7 7^2 6^7 764^{10}$

Figure 13: Chain Code of the character “a”. The exponents denote the number of pixels along a certain direction.

As a result of the segmentation process, for each textual basic layout component, the internal knowledge comprises information about its position (left upper corner), surrounding rectangle, chain code and number of holes. The goal of our approach is to obtain different classes of characters (nonterminals) which have chain code in common sequences.

Thus, step by step we reduce the original chain code description. The consequence thereof is the loss of information, but word recognition has no need for an exact character recog-nition.

To become independent of the font size we first eliminate within the chain code all multiple steps (exponents) in one direction, thus obtaining “2102454321076764” for the example of Fig. 13. Most of the chain codes in practical examples do not equal such ideal descriptions like the one of Fig. 10. In reality, the boundaries of the characters have gaps or are smudged. Fig. 14 shows the image of a properly scanned “a”, as well as a typical example for “a” with a rugged boundary.

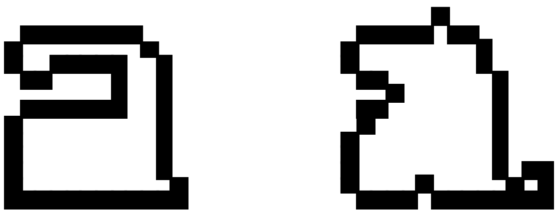


Figure 14: Ideal and Real Boundary Description for the character “a”.

To extract only the general characteristics of such a boundary, we reduce the chain code from an eight-direction to a four-direction boundary code. Only the vertical (north, south) and horizontal (east, west) directions will be further considered. Such a chain code describes an angular hull of a character. A resulting smoothed boundary of the character “a” is shown in Fig. 15. The corresponding reduced chain code (RCC-Code) is “20242064”.

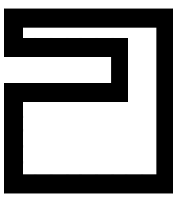


Figure 15: Smoothed Boundary of Character “a” .

To obtain a RCC-Code from the original one, several steps have to be performed.

In the first step, all diagonal orientations are decomposed in their rectangular components. In the second step all single steps in one direction are eliminated, except those cases in which a 180° break would result. In the third step, all 180° breaks within the existing boundary are eliminated. Such breaks are troublesome, because in the case of low scanning density, a thin line would be interpreted in a different sense than a bold line. For example a vertical bar ("I") would probably not have the chain code "22222206666664", but "222222666666". The insertion of an appropriate direction code eliminates these difficul-ties. Fig. 16 illustrates the different smoothing steps for an example of the character "a". The corresponding chain code for the resulting character image is "20242064".

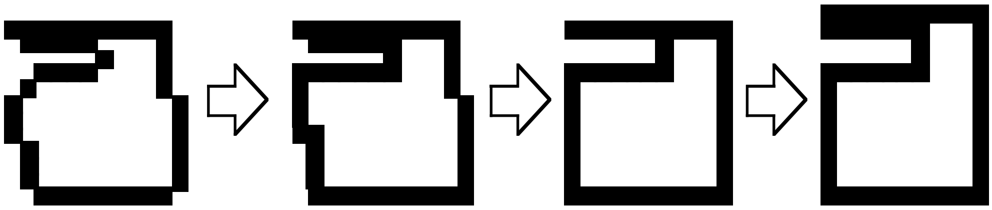


Figure 16: Smoothing Steps for Character "a".

The RCC-Code, resulting from the smoothing operation is size-independent, and with some restrictions font-independent. In general, it also tolerates cursive fonts, as well as incom-plete character images [Hönes, 1989].

4 .2 RECOGNITION OF NONTERMINAL WORDS

As a consequence, we can establish different character classes. To restrict the size of such a class, we also use the information of included holes. For every class, we choose one specific member as its representative.

For efficient character classification, we store the class descriptions that establish the different alternatives of RCC-Codes in a tree-like structure (RCC-Tree). The nodes contain possible directions of the boundary code, whereas all paths belong to RCC-Code descriptions for nonterminals. Fig. 17 shows an example of a RCC-Tree for the nonterminals ~d, ~a, ~c, ~e, ~o and ~I.

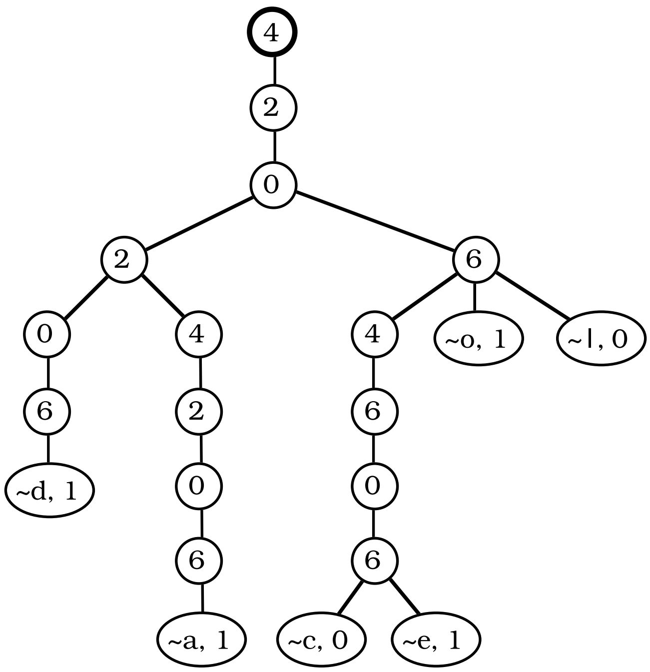


Figure 17: Example for a RCC-Tree.

At the leaves of the RCC-Tree, additional information about the number of holes is stored. Because holes are of very restricted information for classification, an error correction has been developed to handle this problem [Hönes, 1989].

By applying a specific search strategy, which allows the skipping as well as the insertion of code digits, the procedure is capable to generate nonterminal words. Nonterminal words are strings, consisting only of class designators (nonterminals), e.g. “~c~o~d~e”.

4 .3 DERIVATION OF NONTERMINAL WORDS

Every nonterminal can be derived to a set of terminals. Therefore, we use a special cycle-free, context-free grammar that contains derivation rules, like:

~o -> “o”	~U-> “U”	~c -> “c”
~o -> “O”	~U -> “u”	~c -> “C”
~o -> “ø”	~U -> “V”	~c -> “(”
~o -> “D”	~U -> “v”	~c -> ~e
~o -> “8”		
~o -> “B”		

Every derivation rule is assigned a priority value, which directs the order of its use. The priority value depends on statistical tests and a parameter that improves run time of the procedure very much.

To obtain a real word, a nonterminal word is derived from left to right. The derivation is performed while accessing an additional dictionary. Moreover, after each resulting word segment it is checked if such a partial word exists within the dictionary. Thus, a derivation tree is constructed. The root of this tree is a nonterminal word. The children of a certain node represent the string, which is generated by applying only one of the derivation rules. The leaves of the derivation tree represent strings, consisting only of terminals, which means characters. The depth of the tree depends on the number of rules that can be applied to a nonterminal word. The maximal branching factor reflects the maximal size of a character class.

The result obtained, is a single word or a restricted set of words, which can be considered as a solution or possible candidates for its ASCII-Code. For the recognition of textual information within a labeled-area item of business letters, which have a relatively res-tricted information, like the receiver, company-specific printings or the date the procedure behaves very well and the containing words are found with high accuracy. Future work will concentrate on the improvement of this procedure for textual recognition, as well as the restriction of the number of alternatives to one single word, by using complex knowledge about the syntax of sentences, expectations as to the contents of text, knowledge about the partitioning of text on a page as well as common sense.

5 CONCLUSION

Since paper will most probably exist along with electronic media as a carrier for information for some time to come, it is necessary to develop interfaces between paper and electronic information processing. Such flow of information between paper and computer is of great importance in the areas of mass data processing, publishing and information retrieval.

This paper gives an overview of our activities in the field of document analysis. It describes different approaches for document image processing and moreover provides principles for a document understanding. It discusses concepts for the description and recognition of complex document presentations, as well as their symbolic representation. The central aspect of the activities is to develop an approach which is able to identify logical objects within a document image, for expectation-driven partial post-ordered textual analysis. Therefore, we consider higher syntactical information and concentrate on how to represent knowledge about document layouts. The system uses multiple knowledge sources to obtain a multitude of information accessing by electronic media, filing systems and editors [Dengel et al, 1987]. The work we are performing has the intention to capture freely structured documents as they can typically be found in offices. Our future work will also concentrate on techniques for a expectation-driven partial text analysis within document images.

ACKNOWLEDGEMENT

The author gratefully acknowledges the work of Frank Hönes and Eberhard Schweizer, who have been involved in this document analysis project. The author also acknowledges the support and assistance given by Gerhard Barth.

REFERENCES

[Barr and Feigenbaum, 1981]
A. Barr and E. A. Feigenbaum, *The Handbook of Artificial Intelligence*, Vol. 1, The William Kaufmann Inc., Los Angeles 1981

[Bergengrün et al, 1986]
O.Bergengrün, A.Luhn, G. Maderlechner and B. Ueberreiter, *Dokumentanalyse mit Hilfe von ATN´s und unscharfen Relationen*, Proceedings of 9th DAGM-Symposium, Braunschweig 1987, p. 78

[Butz, 1988]
J. Butz, *Untersuchung und Definition von Struktureigenschaften in Bürodokumenten sowie deren Bewertung mit Hilfe von kontinuierlichen Bewertungsmaßen*, Bachelor Thesis, CS Department, University of Stuttgart, 1988

[Dengel et al, 1987]
A. Dengel, A. Luhn and B. Ueberreiter, *Data and Model Representation and Hypothesis Generation in Document Recognition*; Proceedings of the 5th. Scandinavian Conf. on Image Analysis, Stockholm 1987, pp. 57-64

[Dengel and Barth, 1988]
A. Dengel and G. Barth, *High Level Document Analysis Guided by Geometric Aspects*, Internat. Journal on Pattern Recognition and AI, Vol. 2, No. 4, Dec. 1988, pp. 641-656

[Dengel, 1989]
A. Dengel, *A Cut-based Procedure for Document-Layout Modelling and Automatic Document Analysis*, Proceedings of the SPIE/IEEE-Applications of Artificial Intelligence VII, Orlando, FL, März 1989, pp. 1126-1133

[Dengel and Barth, 1989]
A. Dengel and G. Barth, *ANASTASIL: A Hybrid Knowledge-based System for Document Layout Analysis*, Proceedings 11th IJCAI , Detroit, MI, August 1989, pp 1249-1254

[Dengel and Schweizer, 1989]
A. Dengel and E. Schweizer, *Rotationswinkelbestimmung in binären Dokumentbil-dern*, in: H. Burkardt, H. Höhne, B. Neumann (edt.) Mustererkennung 1989 - Pro-ceedings 11. DAGM-Symposium, Hamburg, Oct. 1989, Springer Verlag, pp. 274-278

[Domke et al, 1986]
L. Domke, A. Günther and W. Scherl, *Wissensgesteurte Formularinterpretation mit Hilfe von Petrinetzen*, Proceedings 8th DAGM-Symposium, Paderborn 1986, p. 29

[Donner, 1985]
H. Donner, *Normen schaffen Freiheit im Büro*, *com* Siemens-Magazine for: Computer & Communications, 4 (1985), pp. 8-13

[Farag, 1979]
R. F. H. Farag, *Word-Level Recognition of Cursive Script*, IEEE Trans. Computers, Vol. 28, 1979, pp. 172-175.

[Hönes, 1989]
F. Hönes, *Möglichkeiten der visuellen Erkennung von Worten mit Hilfe von geometrischen Eigenschaften der enthaltenen Zusammenhangskomponenten*, Bachelor Thesis, CS Department, University of Stuttgart, 1988

[Kubota et al, 1984]
K. Kubota, O. Iwata and H. Arakawa, *Document Understanding System*, Proc. 7th Int. Conf. on Pattern Recognition, Montreal 1984, p. 612

[Masuda, 1985]
I. Masuda et al, *Approach to Smart Document Reader System*, ICPR-IEEE-Computer Vision and Pattern Recognition, San Fransisco, 1985, p.551

[Nagy and Seth, 1984]
G. Nagy and S. Seth, *Hierarchical Representation of Optically Scanned Documents*, Proc. of the 7th Int. Conf. on Pattern Recognition, Montreal 1984, p. 347

[Nagy et al, 1986]
G. Nagy, S. Seth and S. Stoddard, *Document Analysis with an Expert System*, Pattern Recognition in Practice II, Elsevier Science Publ. B.V 1986, pp. 149-155

[Niyogy and Srihari, 1986]
D. Niyogy and S. Srihari, *A Rule-based System for Document Understanding*, Proc. AAAI´86, p. 789[

[Postl, 1986]
W. Postl, *Detection of Linear Oblique Structures and Skew Scan in Digitized Documents*, Proceedings of the 8th Intl. Conf.on Pattern Recogntion, Paris 1986, p. 240

[Schäfer and Fröschle, 1986]
M. Schäfer and H. P. Fröschle, *Die Vision vom papierlosen Büro*, Funkschau 19, 1986, p. 45

[Scherl et al, 1980]
W. Scherl, F. Wahl and H. Fuchsberg, *Automatic Separation of Text, Graphic and Picture Segments in Printed Material*, in: Pattern Recognition in Practice, Ed: Gelsema und Kanal, Amsterdam 1980, p. 213

[Schürmann, 1978]

J. Scürmann, *A Multifont Word Recognition System for Postal Address Reading*, IEEE Transactions on Computers, Vol. C-27, No. 8, Aug. 1978

[Schürmann, 1987]
J. Schürmann, *Stand und Entwicklung der optischen Zeichenerkennung*, Münchner Messe- und Austellungsges., Proc. of the Symposa of the Systems ´87, Oct. 1987, p.98

[Schweizer, 1989]
E. Schweizer, *Erfassung, Justierung und Segmentierung von Dokumentstrukturen*, Diploma Thesis, CS Department, University of Stuttgart, 1989

[Shafer, 1976]
G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976

[Srihari and Zack, 1986]
S. N. Srihari and G. W. Zack, *Document Image Analysis*, Proceedings of the 8th ICPR, Paris, 1986, p. 434

[Srihari et al, 1987]
S. N. Srihari, C.-H. Wang, P. W. Palumbo and J. Hull, *Recognizing Address Blocks on Mail Pieces: Specialized Tools and Problem-Solving Architecture*, AI Magazine, Vol. 8 No. 4, Winter 1987, p. 25

[Trincklin, 1984]
J.-P. Trincklin, *Conception d´un Système d´Analyse de Documents: Etude et Réalisation d´un Module d´Extraction de la Structure Physique de Documents à Support visuel*, Ph.D. Dissertation, Universität de France-comte, Besançon, 1984

[Ullman, 1982]
J. R. Ullman, *Advances in Character Recognition*, in Applications of Pattern Recognition, K. S. Fu, ed., CRC Press, Boca Raton, FL, 1982, pp. 197-236

[Woehl, 1984]
K. Woehl, *Automatic Classification of Documents by Coupling Relational Data Bases and Prolog Expert Systems*, Proceedings 2nd Conf. on Very Large Data Bases, Singapore 1984, p. 529

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or per anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-92-43
Christoph Klauck, Jakob Mauss: A Heuristic driven Parser for Attributed Node Labeled Graph Grammars and its Application to Feature Recognition in CIM
17 pages

RR-92-44
Thomas Rist, Elisabeth André: Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP
15 pages

RR-92-45
Elisabeth André, Thomas Rist: The Design of Illustrated Documents as a Planning Task
21 pages

RR-92-46
Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, Wolfgang Wahlster:
WIP: The Automatic Synthesis of Multimodal Presentations
19 pages

RR-92-47
Frank Bomarius: A Multi-Agent Approach towards Modeling Urban Traffic Scenarios
24 pages

RR-92-48
Bernhard Nebel, Jana Koehler:
Plan Modifications versus Plan Generation: A Complexity-Theoretic Perspective
15 pages

RR-92-49
Christoph Klauck, Ralf Legleitner, Ansgar Bernardi:
Heuristic Classification for Automated CAPP
15 pages

RR-92-50
Stephan Busemann:
Generierung natürlicher Sprache
61 Seiten

RR-92-51
Hans-Jürgen Bürckert, Werner Nutt:
On Abduction and Answer Generation through Constrained Resolution
20 pages

RR-92-52
Mathias Bauer, Susanne Biundo, Dietmar Dengler, Jana Koehler, Gabriele Paul: PHI - A Logic-Based Tool for Intelligent Help Systems
14 pages

RR-92-53
Werner Stephan, Susanne Biundo:
A New Logical Framework for Deductive Planning
15 pages

RR-92-54
Harold Boley: A Direkt Semantic Characterization of RELFUN
30 pages

RR-92-55
John Nerbonne, Joachim Laubsch, Abdel Kader Diagne, Stephan Oepen: Natural Language Semantics and Compiler Technology
17 pages

RR-92-56
Armin Laux: Integrating a Modal Logic of Knowledge into Terminological Logics
34 pages

RR-92-58
Franz Baader, Bernhard Hollunder:
How to Prefer More Specific Defaults in Terminological Default Logic
31 pages

RR-92-59
Karl Schlechta and David Makinson: On Principles and Problems of Defeasible Inheritance
13 pages

RR-92-60
Karl Schlechta: Defaults, Preorder Semantics and Circumscription
19 pages

RR-93-02
Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, Thomas Rist:
Plan-based Integration of Natural Language and Graphics Generation
50 pages

RR-93-03
Franz Baader, Berhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, Enrico Franconi:
An Empirical Analysis of Optimization Techniques for Terminological Representation Systems
28 pages

RR-93-04
Christoph Klauck, Johannes Schwagereit:
GGD: Graph Grammar Developer for features in CAD/CAM
13 pages

RR-93-05
Franz Baader, Klaus Schulz: Combination Techniques and Decision Problems for Disunification
29 pages

RR-93-06
Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: On Skolemization in Constrained Logics
40 pages

RR-93-07
Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux: Concept Logics with Function Symbols
36 pages

RR-93-08
Harold Boley, Philipp Hanschke, Knut Hinkelmann, Manfred Meyer: COLAB: A Hybrid Knowledge Representation and Compilation Laboratory
64 pages

RR-93-09
Philipp Hanschke, Jörg Würtz:
Satisfiability of the Smallest Binary Program
8 Seiten

RR-93-10
Martin Buchheit, Francesco M. Donini, Andrea Schaerf: Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

RR-93-11
Bernhard Nebel, Hans-Juergen Buerckert:
Reasoning about Temporal Relations:
A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

RR-93-12
Pierre Sablayrolles: A Two-Level Semantics for French Expressions of Motion
51 pages

RR-93-13
Franz Baader, Karl Schlechta:
A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

RR-93-14
Joachim Niehren, Andreas Podelski,Ralf Treinen:
Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15
Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support
Final Project Report
33 pages

RR-93-16
Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-17
Rolf Backofen:
Regular Path Expressions in Feature Logic
37 pages

RR-93-18
Klaus Schild: Terminological Cycles and the Propositional μ-Calculus
32 pages

RR-93-20
Franz Baader, Bernhard Hollunder:
Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

RR-93-22
Manfred Meyer, Jörg Müller:
Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

RR-93-23
Andreas Dengel, Ottmar Lutzy:
Comparative Study of Connectionist Simulators
20 pages

RR-93-24
Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

RR-93-25
Klaus Fischer, Norbert Kuhn: A DAI Approach to Modeling the Transportation Domain
93 pages

RR-93-26
Jörg P. Müller, Markus Pischel: The Agent Architecture InteRRaP: Concept and Application
99 pages

RR-93-27
Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28
Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker: Feature-Based Allomorphy
8 pages

RR-93-29
Armin Laux: Representing Belief in Multi-Agent Worlds viaTerminological Logics
35 pages

RR-93-33
Bernhard Nebel, Jana Koehler:
Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

RR-93-34
Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35
Harold Boley, François Bry, Ulrich Geske (Eds.):
Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*
150 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

RR-93-36
Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt: Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38
Stephan Baumann: Document Recognition of Printed Scores and Transformation into MIDI
24 pages

RR-93-40
Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf:
Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

RR-93-41
Winfried H. Graf: LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

RR-93-42
Hubert Comon, Ralf Treinen:
The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

RR-93-45
Rainer Hoch: On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

DFKI Technical Memos

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel:
ODA-based modeling for document analysis
14 pages

TM-91-15
Stefan Busemann: Prototypical Concept Formation
An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang: Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

TM-92-02
Achim Schupeta: Organizing Communication and Introspection in a Multi-Agent Blocksworld
32 pages

TM-92-03
Mona Singh:
A Cognitiv Analysis of Event Structure
21 pages

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:
On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben:
The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

TM-92-08
Anne Kilger: Realization of Tree Adjoining Grammars with Unification
27 pages

TM-93-01
Otto Kühn, Andreas Birk: Reconstructive Integrated Explanation of Lathe Production Plans
20 pages

TM-93-02
Pierre Sablayrolles, Achim Schupeta:
Conlfict Resolving Negotiation for COoperative Schedule Management
21 pages

TM-93-03
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
11 pages

DFKI Documents

D-92-19
Stefan Dittrich, Rainer Hoch: Automatische, Deskriptor-basierte Unterstützung der Dokument-analyse zur Fokussierung und Klassifizierung von Geschäftsbriefen
107 Seiten

D-92-21
Anne Schauder: Incremental Syntactic Generation of Natural Language with Tree Adjoining Grammars
57 pages

D-92-22
Werner Stein: Indexing Principles for Relational Languages Applied to PROLOG Code Generation
80 pages

D-92-23
Michael Herfert: Parsen und Generieren der Prolog-artigen Syntax von RELFUN
51 Seiten

D-92-24
Jürgen Müller, Donald Steiner (Hrsg.):
Kooperierende Agenten
78 Seiten

D-92-25
Martin Buchheit: Klassische Kommunikations- und Koordinationsmodelle
31 Seiten

D-92-26
Enno Tolzmann:
Realisierung eines Werkzeugauswahlmoduls mit Hilfe des Constraint-Systems CONTAX
28 Seiten

D-92-27
Martin Harm, Knut Hinkelmann, Thomas Labisch:
Integrating Top-down and Bottom-up Reasoning in COLAB
40 pages

D-92-28
Klaus-Peter Gores, Rainer Bleisinger: Ein Modell zur Repräsentation von Nachrichtentypen
56 Seiten

D-93-01
Philipp Hanschke, Thom Frühwirth: Terminological Reasoning with Constraint Handling Rules
12 pages

D-93-02
Gabriele Schmidt, Frank Peters,
Gernod Laufkötter: User Manual of COKAM+
23 pages

D-93-03
Stephan Busemann, Karin Harbusch(Eds.):
DFKI Workshop on Natural Language Systems: Reusability and Modularity - Proceedings
74 pages

D-93-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1992
194 Seiten

D-93-05
Elisabeth André, Winfried Graf, Jochen Heinsohn, Bernhard Nebel, Hans-Jürgen Profitlich, Thomas Rist, Wolfgang Wahlster:
PPP: Personalized Plan-Based Presenter
70 pages

D-93-06
Jürgen Müller (Hrsg.):
Beiträge zum Gründungsworkshop der Fachgruppe Verteilte Künstliche Intelligenz Saarbrücken 29.-30. April 1993
235 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-07
Klaus-Peter Gores, Rainer Bleisinger:
Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse
53 Seiten

D-93-08
Thomas Kieninger, Rainer Hoch: Ein Generator mit Anfragesystem für strukturierte Wörterbücher zur Unterstützung von Texterkennung und Textanalyse
125 Seiten

D-93-09
Hans-Ulrich Krieger, Ulrich Schäfer:
TDL ExtraLight User's Guide
35 pages

D-93-10
Elizabeth Hinkelman, Markus Vonerden,Christoph Jung: Natural Language Software Registry
(Second Edition)
174 pages

D-93-11
Knut Hinkelmann, Armin Laux (Eds.):
DFKI Workshop on Knowledge Representation Techniques — Proceedings
88 pages

D-93-12
Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein:
RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

D-93-14
Manfred Meyer (Ed.): Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).