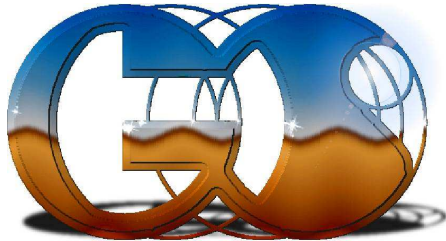


# GAOTENBLATT

Ausgabe: 1. Halbjahr 2002



Gesellschaft für die Anwendung offener Systeme e.V.

## Inhaltsverzeichnis

▶ <b>Editorial</b>	
Gestern war der erste Streich ... ... und der zweite folgt sogleich!	1
▶ <b>Frontnachrichten</b>	
Neuigkeiten aus der freien Welt	2
▶ <b>ΛT<sub>E</sub>X im studentischen Alltag</b>	
Der Versuch einer Motivation	3
▶ <b>Killerapplikation</b>	
Eine Streitschrift wider den technischen Kopierschutz	10
▶ <b>OCR mit statistischen Momenten</b>	
Neue Aspekte der Schriftzeichenerkennung	15
▶ <b>Semantik im Wortschatz</b>	
Wortbedeutungen ohne Weltwissen?	18
▶ <b>PHUTBALL '02</b>	
Programmierwettbewerb Philosophen-Fußball	21
▶ <b>Impressum</b>	
Autorenverzeichnis, Impressum	28

### ▼ Editorial

Gestern war der erste Streich ...  
... und der zweite folgt sogleich!

Liebe Freundinnen und Freunde freier Technologien,

wir haben das *Unmögliche* geschafft! Während sich Politiker und drittklassige Fernsehmoderatoren noch über historische Verantwortung streiten, Parteien sich gegenseitig Wasser auf die (Wind-)Mühlen ihres Wahlkampfes schöpfen und der Kanzler mit Milliarden *Schiefe Türme* kurzfristig begradigen möchte, sind wir schon einen gewaltigen Schritt weiter: Die zweite Ausgabe unserer *Freien Zeitschrift* liegt heute vor Ihnen!

Doch auch Kritik an unserer Auffassung von Freiheit wird langsam lauter. Sie sollte weder überbewertet, noch zum Tabu erklärt werden — begegnen wir ihr lieber mit Humor und Sarkasmus:

Ah, was muß man oft von bösen  
Rechnerprogrammen hören oder lesen!!  
Wie zum Beispiel hier von diesen,  
Welche LUX und Daemon hießen,  
Die, anstatt durch weise Lehren  
Sich zum Geldverdienen zu befehlen,  
Oftmals noch darüber lachten  
Und sich heimlich lustig machten.  
— Ja, zur Konjunkturbremstätigkeit,  
Ja, dazu ist man bereit! —  
Manager necken, Lasten quälen,  
Arbeitsplätze, Kaufkraft, Patente stehlen  
Das ist freilich angenehmer  
Und dazu auch viel bequemer,  
Als auf Messe oder Wirtschaftsschule,  
Festzusitzen auf dem Stuhle.  
— Aber wehe, wehe, wehe!  
Wenn ich auf das Ende sehe!! —  
Ach, das war ein schlimmes Ding,  
Wie es LUX und Daemon gut erging.  
Drum ist hier, was sie getrieben,  
dargestellt und aufgeschrieben.

frei nach WILHELM BUSCH

## ▼ Frontnachrichten

Neuigkeiten aus der freien Welt

### Lobbyarbeit

Das amerikanische Tocqueville-Institut hat kürzlich ein Papier mit dem harmlos erscheinenden Titel »Opening the Open Source Debate« veröffentlicht. Darin werden angebliche Nachteile freier Software dargestellt. Wer Überzeugungsarbeit in Sachen freier Software leisten und sich dazu auf mögliche Gegenargumente vorbereiten möchte, findet hier eine gute Grundlage zum Üben. Ausführliche Kritiken des Papiers liegen inzwischen auch im Netz, Links finden sich bei Heise und bei Slashdot.

### Referenzen

- [1] Kenneth Brown: *Opening the Open Source Debate*  
[http://www.adti.net/html\\_files/defense/opensource\\_whitepaper.pdf](http://www.adti.net/html_files/defense/opensource_whitepaper.pdf)
- [2] Heise-Newsticker: *Open Source als Konjunkturbremse*  
<http://www.heise.de/newsticker/data/ps-11.06.02-000/>
- [3] /.: *ADTI Whitepaper Released*  
<http://slashdot.org/article.pl?sid=02/06/10/1737258>

### Mieten statt kaufen

Der Softwarehersteller Microsoft, bekannt für Produkte wie den Intranetbrowser *Internet Explorer* und Outlook, eine Plattform für mobile Agenten, drängt seine Firmenkunden in ein neues Lizenzmodell. Preisreduzierte Updates auf früher gekaufte Lizenzen soll es künftig nicht mehr geben. Statt dessen haben Kunden die Wahl, jede neue Version voll zu bezahlen, oder aber ein Produkt oder eine Produktgattung für einige Jahre zu mieten. Wobei das Mieten, wie üblich, mit regelmäßigen Zahlungen verbunden ist. Und auch mit der Möglichkeit, daß der Vertrag endet – im schlimmsten Fall kann das bedeuten, daß eine von Mietsoftware abhängige Firma ihre IT-Infrastruktur verliert. Anders als typische Mietobjekte wie Fahrzeuge oder Büroräume läßt sich Ersatz nicht so einfach beschaffen; der Übergang zu anderen Produkten wird durch proprietäre Datenformate mindestens erschwert. Wir

empfehlen Vorbeugung. Wer freie Software einsetzt, macht sich von niemandem abh ngig.

### Referenzen

- [1] Microsoft: *Software Assurance & the 6.0 Volume Licensing Programs*  
<http://www.microsoft.com/licensing/programs/sa/default.asp>

### FreeBSD 5.0-STABLE in Sicht?

Langsam kommt Bewegung in die Teufelsanbeter-Szene: Am 15. Juni wurde der neue 4.6-Zweig der FreeBSD-Distribution als *stabil* bekanntgegeben. Neben den  blichen sicherheitsrelevanten Verbesserungen und einigen Programmupdates wird erstmals XFree86 4.2.0 als Standard-X-Windows-System benutzt. Gegen Ende dieses Jahres soll es dann mit der 5.0-STABLE soweit sein. Abwarten!



### Referenzen

- [1] *FreeBSD 4.6-RELEASE Announcement*  
<http://www.freebsd.org/releases/4.6R/announce.html>

### GAOS e. V. — neuer Vorstand gew hlt

Auf der Mitgliederversammlung vom 07. April 2002 wurde der Vorstand unserer *Gesellschaft f r die Anwendung offener Systeme e. V.* neu gew hlt:

Vorsitzender	Herr MARCUS BAUTZE
Kassierer	Herr SVEN WINDISCH
Schriftf�hrer	Herr ROBERT H�HNDORF

Als Kassenpr fer wurden Herr FRANK-MICHAEL SCHLEIF und Herr ANDREAS ROMEYKE bestellt.

### Referenzen

- [1] *Protokoll Mitgliederversammlung MV070402*  
<http://gaos.org/verein/satzung/>

### ▼ L<sup>A</sup>T<sub>E</sub>X im studentischen Alltag

FRANK-M. SCHLEIF & HEIKO STAMER:  
Der Versuch einer Motivation

Im folgenden Artikel betrachten wir die Anwendungsmöglichkeiten von L<sup>A</sup>T<sub>E</sub>X im studentischen Alltag und versuchen eine Motivation zur sinnvollen Anwendung von L<sup>A</sup>T<sub>E</sub>X zu geben. Es werden verschiedene Anwendungsbereiche betrachtet und an konkreten Beispielen der Nutzen von L<sup>A</sup>T<sub>E</sub>X aufgezeigt, ebenso werden die Vor- und Nachteile von L<sup>A</sup>T<sub>E</sub>X im Vergleich zu WYSIWYG-Editoren erörtert. Dieser Artikel soll kein Tutorial zu L<sup>A</sup>T<sub>E</sub>X sein, sondern vielmehr den Leser dazu motivieren, sich mit diesem sehr nützlichen Textsatzsystem zu beschäftigen.

Warum sollte man sich mit L<sup>A</sup>T<sub>E</sub>X beschäftigen? Sicherlich hat jeder schon einmal das eine oder andere Problem mit dem bisher benutzten Textverarbeitungssystem gehabt und sucht vielleicht nach einer Alternative [12], die weniger Probleme verursacht oder schlicht — den Job besser erledigt.

Wir versuchen in diesem Artikel klar zu machen, wo die Vorteile von L<sup>A</sup>T<sub>E</sub>X im Vergleich zur defacto Standard-Textverarbeitung *a la MS-Word* liegen, wie man L<sup>A</sup>T<sub>E</sub>X verwendet, wo es sich effizient einsetzen läßt und wo vielleicht auch nicht. Dieser Artikel wird allerdings kein Tutorial zu L<sup>A</sup>T<sub>E</sub>X sein, denn derer gibt es viele und die entsprechenden Referenzen sind am Ende in der Literaturliste aufgeführt.

### Satzsystem vs. Wortprozessor

Zunächst betrachten wir einmal den „Normalfall“, wie heutzutage Texte geschrieben werden. Man will einen Text schreiben und diesen dann auf dem Drucker möglichst „schön“ ausdrucken. Da man nicht alles das man tippt gleich gedruckt haben möchte (sonst hätte man ja auch eine einfache Schreibmaschine verwenden können), nutzt man den PC und schreibt den Text per Tastatur und speichert ihn dann digital. Man kann den Text jetzt beliebig ändern, Schnickschnack wie Bilder hinzufügen und dann letztendlich (auch selektiv) drucken. Nun hat der Autor dank moderner Software zwei Möglichkeiten diese Aufgabe zu bewältigen: Er kann ein professionelles Textsatzsystem (z. B. T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X) benutzen oder eine Standard-Textverarbeitung, die i. A. nur ein „dummer Wortprozessor“<sup>1</sup> ist.

<sup>1</sup>So werden diese Systeme in der Literatur und Praxis oft bezeichnet, um sie von „intelligenten“, d. h. den Autor in wesentlichen Arbeitsschritten unterstützenden, (Satz-)systemen zu unterscheiden.

### Textsatzsysteme

Um einen druckbaren Text in einem Textverarbeitungssystem wie *MS-Word* zu erstellen, wird man effektiv genötigt, zwei Dinge auf einmal zu bewältigen:

1. *Das Schreiben des Textes*: Das betrifft den Inhalt und die logische Strukturierung in Einzelteile (Absätze, Abschnitte, Kapitel, Hervorhebungen, Fußnoten usw.).
2. *Die Typographie des Textes*: Das ist z. B. die Wahl der Schriftart und Formatierung für Überschriften, den primären Text, Fußnoten, Bildunterschriften; Gestaltungsmerkmale wie Links- oder Rechtsbündigkeiten, ob und wo Blocksatz oder nicht, ob der Text zweispaltig gesetzt werden soll und vieles andere mehr.

Damit ein Text leicht gelesen und sinnvoll verwendet werden kann, ist neben dem Inhalt auch die Typographie wichtig. Während inhaltliche Fakten die reine „Lesbarkeit“ kaum beeinflussen, hängt eine gute Typographie immer auch vom jeweiligen Ausgabemedium (Papier, Bildschirm, transparente Folien, ...) ab und ist deshalb hinreichend komplex, um Laien große Probleme zu bereiten.

Eigentlich wäre es nur vernünftig, wenn ich mich als Autor ausschließlich um den textuellen Inhalt kümmern müßte — das geht zwar, würde aber zu einem unansehnlichen Text führen, den am Ende vielleicht gar niemand liest. Man stelle sich nur vor, ich würde alle Absätze und Unterteilungen wie Überschriften weglassen.

In Vorcomputerisierter Zeit — und auch heute noch selten — gab es dafür den Beruf des Typographen. Dieser kümmerte sich darum, daß der Text für den Druck korrekt gesetzt wurde. Dafür mußte der Autor nur minimale Zuarbeit, in Form einer logischen Struktur und ggf. Formatbemerkungen, leisten.

Nun haben wir privat oder öffentlich meist keinen Typographen zur Hand, möchten aber dennoch Dokumente adäquat setzen und „schön drucken“. Also benutzen wir den Computer, oder im speziellen ein Textsatzsystem, welches uns diese Arbeit abnimmt. Doch wie geht das nun? — Tja, ich schreibe meinen Text mit einem ASCII-Editor meiner Wahl und setze an den entsprechenden Stellen Formatanweisungen — genau wie (damals) die Bleistiftbemerkungen für den Setzer. Unser Satzprogramm (vorzugsweise L<sup>A</sup>T<sub>E</sub>X/T<sub>E</sub>X) liest diese Befehle und generiert ggf. mit

Standardvorlagen eine „schön gesetzte“ Fassung meines Dokuments.

Text schreiben und Text setzen sind also zwei getrennte Aufgaben und Arbeitsschritte. Dieses Merkmal ist auch der wichtige Unterschied zum folgenden, verbreiteteren Ansatz der WYSIWYG-Texteditierung.

### Wortprozessor

Das klingt ja jetzt doch eher wie ein Rückschritt, denn man macht alles ähnlich wie vor 40ig Jahren?! Nun ja, meißt wird es eben nicht mehr so gemacht. Man benutzt einen „modernen“ Wortprozessor, einen sogenannten WYSIWYG-Editor — sagen wir einfach mal es sei *MS-Word* (genauso-gut-oder-schlecht könnte man auch Wordperfect [4], Star- oder OpenOffice [5] oder KWord [6] verwenden). Dabei wird die Textverarbeitung zu einem Schritt vermischt. Man schreibt den Text und während man fortfährt, erhält der Text am Bildschirm eine konkrete typographische Erscheinung. Diese ist eng an die visuelle Ausgabe während der Bearbeitung angelehnt, wenn man das Dokument ausdruckt, sieht allerdings der Ausdruck dann oft doch nicht so aus wie auf dem Bildschirm. In der Tat wird der Text sofort beim Eintippen gesetzt, wenn man mal von ein paar kleinen automatischen Nachbearbeitungsschritten (Seitennummerierung, etc.) abieht. Auf den ersten Blick scheint das bequem — aber:

1. Der Autor wird von seiner eigentlichen Arbeit — einen Text zu schreiben — abgelenkt und immer wieder vor typographische Entscheidungen gestellt.<sup>2</sup>
2. Die Textsatzalgorithmen in WYSIWYG-Editoren opfern Qualität zu Gunsten von Geschwindigkeit, da alles „live“ umgesetzt wird. Das Endprodukt ist meist (sehr) viel schlechter als das eines guten Textsatzsystems; zumindest bei unzureichender typographischer Vorbildung des Autors bzw. falls man nicht die „versteckten Kniffe“ des Wortprozessors zu verwenden weis.
3. Der Benutzer unterliegt der starken Versuchung die logische Struktur des Textes zu vergessen

<sup>2</sup>Da bricht genau dort die Zeile um, wo man es eigentlich nicht möchte. Wieviele Leerzeilen benutze ich zur Trennung der Absätze? ...

und durch oberflächliche typographische Elemente auszugleichen.<sup>3</sup>

### Die Struktur eines Dokuments

Man nehme zum Beispiel eine Abschnittsüberschrift. Solange die logische Struktur eines Dokuments betrachtet wird, ist lediglich die Auszeichnung als *Abschnittsüberschrift* von Bedeutung. Weder der Schrifttyp, Ausrichtung und Größe, noch die Positionierung im Textfluß spielen vorerst eine Rolle, denn dazu haben sich die Schriftsetzer jahrhundertlang Gedanken gemacht, wie man die Überschrift dann vernünftig auf dem Papier darstellt. Also brauchen Sie — als Textautor — diese erprobten Heuristiken nicht erst neu zu erfinden. Man könnte zum Beispiel nur tippen:

```
\section{Text der Überschrift}
```

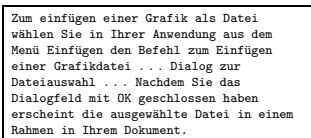
Wie diese Überschrift dann in der fertigen Version aussieht, ist eigentlich eine andere Frage, die nebenbeige sagt ja auch immer vom Ausgabemedium abhängen sollte. Wenn Sie dagegen einen Wortprozessor verwenden, ist das was Sie sehen auch das (und nur das!), was Sie bekommen. Sie werden gezwungen, über die exakt festgelegte typographische Erscheinung Ihrer Überschrift zu entscheiden, sobald Sie sie schreiben.

Mal angenommen Sie entscheiden sich dafür, daß Sie Ihre Überschriften in fetter Schrift und etwas größer als den restlichen Text schreiben wollen. Wie erreichen Sie dieses Aussehen? Es gibt mehrere Möglichkeiten, aber für die meisten Menschen ist der offensichtlichste und intuitivste Weg (mal eine komplette WYSIWYG-Umgebung vorausgesetzt), den Text der Überschrift zu schreiben, ihn zu markieren, auf den Knopf für fette Schrift zu klicken, den Auswahl-dialog für die Schriftgrößen zu öffnen und eine etwas größere Schrift zu wählen. Die Überschrift ist jetzt fett und etwas größer.

Super! Aber wer sagt uns jetzt noch, daß es eine Überschrift ist? Nichts in Ihrem Dokument gibt Auskunft darüber, daß dieses kleine Stück Text, zugegebenermaßen etwas größer und fetter als der Rest, eine Überschrift definiert — Formatänderungen an solchen Überschriften werden jetzt zu einer *Sizifus Arbeit* und man durchforstet jedesmal den ganzen Text.

<sup>3</sup>Mache ich diese Hervorhebung nun nur kursiv und fett, oder unterstreiche ich lieber doppelt? ...

Ein anderes Beispiel ist die Einbindung von Bildern in Dokumenten. Sie möchten eine Grafik aus einer Datei (z. B. .png) in Ihr Dokument einfügen – gängige Wortprozessoren können das, also kein Problem:



Tolle Sache – oder? Mal abgesehen davon, daß Grafiken, die die Seitengröße überschreiten, sonstwo eingefügt werden und kaum noch zu bearbeiten sind, ergeben sich allerlei Probleme: Angenommen Sie haben viele Grafiken eingefügt, dann haben Sie plötzlich eine extrem große Textdatei (da die Bilder mit eingebunden wurden) und schwupps können/sollten Sie diese Datei nicht mehr unkomprimiert per Email verschicken, weil Ihr Provider Nachrichten die größer als 2 MegaByte sind nicht annimmt. Vielleicht wollen Sie aber auch die Farbgrafiken durch die gleichen Grafiken in Graustufen ersetzen, welche fast den selben Dateinamen haben, oder bei jeder Grafik die Größe im Dokument ändern — Tja Pech, jetzt wühlt man sich wieder durch das ganze Dokument (und das vielleicht noch auf einem recht schwachbrüstigen Rechner).

Ergo – da wäre doch eine Anweisung wie

```
\includegraphic{file=bild.png, width=10cm}
```

viel einfacher zu ändern und zu handhaben – Oder?

Nun es gibt einen Weg, die logische Bedeutung von Textstücken in z. B. *MS-Word* festzulegen. So kann man z. B. – wenn man sorgfältig vorgeht<sup>4</sup> erreichen, daß man mit einem einzigen Befehl das Aussehen aller Abschnittsüberschriften ändern kann. Aber nur die wenigsten Wordbenutzer verwenden den kleinen Punkt in der Menüleiste, der ja auch immer auf Standard steht und mit dem solche Formatanweisungen in den Text gesetzt werden. Das ist auch nicht weiter verwunderlich, da der WYSIWYG-Ansatz gerade dazu NICHT verleitet. Man kann/wird einfach Strukturierungen vortäuschen. Auch die Problematik mit den Grafiken läßt sich zumindest in *MS-Word* partiell beheben. Dazu muß die Grafik nur „verknüpft“ ins Dokument eingetragen werden<sup>5</sup>. Auch die vielgepriesene Autoformatierung erweist sich häufig mehr als Fluch: So sagte mir neulich ein Wissenschaftler, den ich auf

<sup>4</sup>das Programm gut kennt oder ein sehr gutes (Hand-)buch hat und dieses auch liest

<sup>5</sup>in vielen Handbüchern ist leider wenig darüber zu lesen

LaTeX ansprach, man könne ganz ordentliche Dokumente mit *MS-Word* erstellen, wenn man sich sehr gut damit auskennt und alle Auto(irgendwas) im Menü *Einstellungen* abstellt. Sonst kann es einem nur zu leicht passieren, daß eine Einrückung plötzlich eine Unterüberschrift wird und nachher im Inhaltsverzeichnis auftaucht (Hoffentlich passiert das nicht zu oft, denn sonst hat man mehr Verzeichnis als Inhalt :-).

Wir wollen an dieser Stelle nicht alle Wortprozessoren „verteufeln“, sondern vielmehr deutlich machen, daß deren Konzept und Qualität für *fastPrint<sup>TM</sup>-Dokumente* ausreichend sein mag, ein professionelles und durchdachtes Textsatzsystem aber nicht erreichen kann.

### Spaßige WYSIWYG-Erlebnisse

Ich habe jetzt also die Wahl. Ich kann einen Texteditor verwenden, darin meinen Text schreiben und wo es nötig ist, füge ich Format und Strukturanweisungen ein (es gibt auch nette Editoren die den Autor dabei unterstützen). Oder ich benutze eine *eierlegende Wollmilchsau* eines Wortprozessors, der mich dazu nötigt, andauernd auf Struktur und Formatierung Zeit zu verschwenden und bei zu exzessiver WYSIWYG-Nutzung mich Stunden damit zubringen läßt, Formatierungen und Strukturierungen von Hand im gesamten Dokument zu ändern. Wem die bisherigen Beispiel zu konstruiert schienen, den bitte ich, sich den Auszug aus unserer Studentenbefragung (große, graue Box auf den folgenden Seiten) näher anzusehen.

### Kurzeinführung LaTeX

Nach dieser, hoffentlich erfolgreichen, Motivation nun zunächst eine kurze Einführung in LaTeX: Das ursprünglich von LESLIE LAMPORT und mittlerweile von vielen Freiwilligen weiterentwickelte LaTeX ist ein leicht zugängliches, hocheffizientes Programm, mit dem Dokumente aller Art in Buchdruckqualität gesetzt werden können. Grob gesprochen ist LaTeX lediglich ein Makroüberbau (Parser, Hilfsprogramme und diverse Makroklassen) für DONALD E. KNUTHs geniales Textsatzsystem TeX und soll dessen Funktionalität auch Laien, d. h. Personen die *Computers & Typesetting* bzw. *The TeXbook* nicht gelesen haben, zugänglich machen. LaTeX und entsprechende TeX-Distributionen existieren mittlerweile für nahezu jede

## Studentenbefragung

### 1. SIMONE M., angehende Diplompsychologin<sup>a</sup>

- Simone benutzt MS-Word 97, da dies in der Fakultät überall installiert war und hat auch zu Hause eine Word97 Installation. Da in der Arbeit viel Statistik vorkommt gibt es auch viele Tabellen – sie hat die Arbeit zu Hause weitestgehend fertig geschrieben und die letzten Tage bis in die frühen Morgenstunden daran gearbeitet, da der Abgabetermin immer näher rückt und sie nicht verlängern will.
- Heute ist der große Tag: Die Arbeit soll 3× (mit Farbgrafiken) ausgedruckt werden – natürlich in der Uni, denn sonst wird es zu teuer. Sie öffnet also Ihr Dokument in Word und schaut nocheinmal durch die Seiten – sieht alles ganz gut aus. Doch was ist das?! Die eine Tabelle paßt plötzlich nicht mehr komplett auf eine Seite und bricht zur nächsten um, dort fehlt dann natürlich die Tabellenüberschrift und niemand versteht was gemeint ist. So ein Miß! Die schöne Übersicht ist zerstückelt. Klarerweise hat sich auch alles andere total verschoben und die Referenzen stimmen auch nicht mehr. Wieso eigentlich? Zu Hause sah alles noch normal aus.
- Versuche der Korrektur: Schriftgröße, Größe der Zellen, Seitenrandformatierung, . . . – nach 3 Stunden sieht es endlich wieder halbwegs erträglich aus. Puhhh – geschafft. Jetzt der Ausdruck. Doch Oh Gott, die Hintergrundfarbe einiger Tabellenspalten ist gar nicht so schön wie auf dem Bildschirm angezeigt – alles viel zu dunkel, da kann man die Werte ja gar nicht mehr lesen. Die kann ich doch jetzt aber unmöglich alle ändern, das sind ja hunderte von Feldern und die müsste man alle einzeln selektieren und dann umfärben, oder die Tabellen gleich komplett neu, geändert einfügen. Aber da gibt es ja noch einen anderen Farbdrucker, mal sehen ob es da besser aussieht – tatsächlich die Werte sind lesbar, zwar sieht das gewählte grün dort wie Grünspan mit etwas Gold gesprenkelt aus, aber immerhin.

### 2. NIELS B., Diplominformatiker<sup>b</sup> – drei Dinge gibt es zu berichten:

- (a) Schreibe nie an einem Dokument auf zwei verschiedenen Rechnern mit einem deutschen und einem amerikanischen Word unterschiedlicher Versionen!<sup>c</sup>  
*Die Folgen: Alle Bilder sind weg, alle Tabellen verschieben sich, die Rechtschreibkontrolle schlägt zurück (funktioniert nicht).*
- (b) Füge niemals zuviele Tabellen in Word ein!  
*Die Folgen:*
  - *Obwohl du eine andere Tabelle bearbeitest, werden andere neu formatiert.*
  - *Die Lösung: Wandeln der vorhandenen Tabellen in Text.*
- (c) Der Fehler, das Word meint es könne jetzt nicht mehr speichern, ist auch in der Version 97 SPxx immer noch vorhanden und tritt bei einer Dateigröße von 1-2 MegaByte auf. Ist diese Hürde genommen passiert es nie wieder . . .  
*Lösung: Speichern im Format RTF, neu laden, speichern im Standardformat DOC und schon geht es wieder.*

### 3. CHRISTOPH M.: *Word (6-97) ist eine echte Herausforderung. Wem Strategiespiele zu langweilig geworden sind, der versucht die Rätsel von Word zu lösen.* Ein Beispiel: Wie bringe ich Word dazu, jede Fußnote auf die richtige Seite zu drucken? Microsoft weiß es nicht – aber es gibt eine Antwort: Man füge nicht-numerierte weitere Fußnoten ein, deren Schriftfarbe dann auf *weiß* gesetzt wird – und schon wird alles gut!

<sup>a</sup>zwei Tage vor der entgeltigen Abgabe der Diplomarbeit

<sup>b</sup>Wurde gezwungen seine Diplomarbeit in Word zu tippen, da dies in der entsprechenden Firma *Standard* ist.

<sup>c</sup>Und glaube das Speichern des Dokumentes in der niedrigeren Version genügt.

Plattform und können somit fast überall für professionellen Satz eingesetzt werden.

### Struktur eines L<sup>A</sup>T<sub>E</sub>X-Dokuments

L<sup>A</sup>T<sub>E</sub>X läßt sich für verschiedenste Zwecke einsetzen, so z. B. für das Schreiben wissenschaftlicher Artikel/-Konferenzbeiträge, Briefen nach DIN-Norm, Bücher u. v. m., wobei für unterschiedliche Dokumentarten verschiedene logische Strukturen/Vorlagen geschaffen wurden. Deshalb sollte man sich zuerst darüber klar werden, welche Dokumentart man erstellen möchte und ob eine geeignete L<sup>A</sup>T<sub>E</sub>X-Klasse dafür existiert:

```
\documentclass{article}
```

Die Dokumentklasse legt die verfügbaren logischen Befehle und Umgebungen sowie deren Standardformatierung fest – hier beispielsweise einen Artikel. Es gibt in L<sup>A</sup>T<sub>E</sub>X verschiedenste Befehle für bestimmte Aufgaben, die in Paketen zusammengefaßt werden. So gibt es z. B. ein Paket zur Einbindung von eps-Grafiken, welches mit

```
\usepackage{epsfig}
```

im Kopf des Dokumentes eingebunden wird. Das eigentliche Haupt-Dokument beginnt dann stets mit

```
\begin{document}
```

und endet mit

```
\end{document}.
```

Hier sieht man auch gleich, wie üblicherweise sogenannte Umgebungen in L<sup>A</sup>T<sub>E</sub>X verwendet werden

```
\begin{Umgebungsname}
... Text in der Umgebung ...
\end{Umgebungsname}
```

Solche Umgebungen definieren bestimmte Formatierungen, wie z. B. die Umgebung *abstract*, die normalerweise dafür sorgt, daß der Text innerhalb dieser Umgebung wie für Zusammenfassungen üblich klein und geblockt vom restlichen Text abgehoben gesetzt wird. Man kann sich beliebig viele eigene Umgebungen z. B. für Definitionen, Sätze, Gedichte, etc. selbst definieren.

Idealerweise sollte man sein Dokument in inhaltlich logische Abschnitte aufteilen. Dies kann man allein auch physisch erreichen in dem der Text in einzelne

Dateien aufgeteilt wird, die dann in das Hauptdokument durch den Befehl *include* eingefügt werden.

Damit haben wir auch schon die grundlegende physische Architektur eines L<sup>A</sup>T<sub>E</sub>X-Dokuments beschrieben — nun noch die logische Struktur.

### Logische Struktur eines L<sup>A</sup>T<sub>E</sub>X-Dokuments

Die L<sup>A</sup>T<sub>E</sub>X-Standardklassen enthalten Befehle, für Umgebungen, mit denen die verschiedenen Elemente der hierarchischen Struktur eines Dokuments definiert werden (Kapitel, Abschnitte, Anhänge). Jeder dieser Befehle definiert eine Gliederungsebene innerhalb der Hierarchie und jedes Strukturelement ist einer dieser Ebenen zugeordnet. Ein typisches Dokument wie z. B. ein Artikel besteht aus einer Überschrift, einigen Abschnitten, die wahrscheinlich in viele weitere Ebenen untergliedert sind, und einem Literaturverzeichnis.

Zur Beschreibung solcher Strukturen wird beispielsweise

```
\maketitle
```

welcher der Dokumententitel erzeugt<sup>6</sup>, Gliederungsbefehle wie

```
\section{Abschnittsüberschrift}
```

oder

```
\subsection{Unterabschnittsüberschrift}
```

und die Umgebung

```
\begin{thebibliography}
```

verwendet.

Im allgemeinen führen die Gliederungsbefehle automatisch eine oder mehrere der folgenden Aktionen aus:

- Erzeugen der Überschriftennummer entsprechend der hierarchischen Ebene
- Speichern der Überschrift als Eintrag in das Inhaltsverzeichnis
- Formatieren der Überschrift (entsprechend der Standardeinstellung oder eigener gewählter Definitionen)

Soviel als Kurzeinführung, wer sich für „alle“ Kleinigkeiten interessiert, dem sei der *Bernhardiner* von Addison-Wesley [10] empfohlen.

<sup>6</sup>Relevante Daten, wie z. B. die Namen der Autoren, das Datum, etc., werden vorher in der *titlepage*-Umgebung festgelegt.

### WYSIWYG - Hilfsmittel

Wenn man sich nun entschlossen hat L<sup>A</sup>T<sub>E</sub>X als Textsatzsystem zu verwenden, so taucht zwangsläufig die Frage auf, wie man denn nun konkret anfängt: Es gibt dafür verschiedene Möglichkeiten. Die simpelste Variante ist sich eine L<sup>A</sup>T<sub>E</sub>X-Distribution aus dem Netz zu laden, zu installieren und mit einem beliebigen Texteditor die Texte zu schreiben, eben entsprechend der obigen Hinweise und Beispiele. Dann übergibt man L<sup>A</sup>T<sub>E</sub>X das Dokument und erhält das druckfertige, typografisch „schön“ gesetzte Dokument. Nun ist es sicher so eine Sache, von einem WYSIWYG-Editor wie *MS-Word* auf ein Textsatzsystem umzusteigen. Zur Eingewöhnung gibt es deshalb allerlei Hilfsmittel – auch WYSIWYG-Editoren – die einem im Umgang mit L<sup>A</sup>T<sub>E</sub>X unterstützen und die Eingewöhnung/Umstieg zum Kinderspiel werden lassen. Einige dieser, als Einstiegswerkzeuge zu verstehenden, Programme wollen wir im folgenden kennenlernen.

### ABI-Word

ABI-Word [8] ist ein Wortprozessor für das X-Window-System und ist für diverse Unix-Derivate verfügbar. Dieser Editor ist in der Lage, Dokumente aus den verschiedensten Dateiformaten zu importieren (darunter auch *MS-Word* DOC) und diese zur Bearbeitung zur Verfügung zu stellen. An sich ist dieses Programm auch nur ein weiterer WYSIWYG-Editor, allerdings einer der ersten, der für Linux verfügbar war. Was diesen Editor interessant macht, ist die Fähigkeit L<sup>A</sup>T<sub>E</sub>X-konforme Dokumente zu erzeugen. Damit hat man die Möglichkeit (mit einigen Einschränkungen) die bereits erstellten Word-Dokumente in die neue Welt „herüberzuretten“ und für den Einstieg in L<sup>A</sup>T<sub>E</sub>X ein Hilfsmittel an der Hand. ABI-Word ist sicherlich noch nicht perfekt, aber es bietet immerhin schon reizvolle Funktionalitäten – und es ist zudem kostenlos erhältlich.

### LYX/GLYX/KLYX

Alle LYX-Derivate [3] egal ob für Konsole, KDE oder GNOME sind ein Überbau für L<sup>A</sup>T<sub>E</sub>X, welche bis auf die Konsolenvariante eine WYSIWYG-Umgebung, einen Wortprozessor darstellen. In vielerlei Hinsicht verhält sich LYX ähnlich wie jeder WYSIWYG-Editor man schreibt seinen Text, die Typographie wird sofort von LYX durchgeführt, man kann Text fett, kursiv, unterstrichen, usw. formatieren, Schriftarten auswählen

und auch sonst hat man ein WYSIWYG-Gefühl. Der interessante Aspekt bei LYX ist, daß der in LYX geschriebene Text wenn er in ein druckbares Dokument umgesetzt wird (Seitenansicht/Druck) in L<sup>A</sup>T<sub>E</sub>X umkonvertiert wird und sogar innerhalb des LYX-Editors in einen L<sup>A</sup>T<sub>E</sub>X-Modus umschaltet werden kann, um L<sup>A</sup>T<sub>E</sub>X bezogen Text einzugeben. Somit stellt LYX ein Bindeglied zwischen Standard WYSIWYG-Editoren wie Word und „Plain-L<sup>A</sup>T<sub>E</sub>X“ her. Es ist der Versuch das Beste aus beiden Welten zu vereinen und zumindest für einen L<sup>A</sup>T<sub>E</sub>X-Einsteiger zu empfehlen – eine Meinung, die offenbar auch die Autoren von [11] teilen. Erst wenn man besondere Anforderungen an die Typsetzung des Dokuments stellt, steigt man dann doch oft zu reinem L<sup>A</sup>T<sub>E</sub>X um. Für den Umstieg von z. B. *MS-Word* zu L<sup>A</sup>T<sub>E</sub>X kann LYX eine brauchbare Brücke schlagen.

### MikT<sub>E</sub>X + Winedt

Die oben angeführten Beispiele gingen alle von einer Unix-Plattform aus und sind meist auch nur mit grossen Mühen (LYX) oder gar nicht unter MS-Windows einsetzbar. Für Windows gibt es zwei bekannte L<sup>A</sup>T<sub>E</sub>X-Distributionen EMT<sub>E</sub>X [1] und MikT<sub>E</sub>X [2]. Hat man eine der beiden installiert, kann man z. B. Winedt einen sehr komfortablen und bekannten Texteditor für Windows einsetzen. Winedt hat besondere Funktionen, um direkt mit L<sup>A</sup>T<sub>E</sub>X umzugehen und bietet ein integrierendes Frontend für L<sup>A</sup>T<sub>E</sub>X unter Windows. So ist es mit Winedt möglich L<sup>A</sup>T<sub>E</sub>X-Dokumente zu erstellen (wie übrigens mit jedem Editor auch – auch mit dem Notepad :-)) und dabei besondere Hilfsfunktionen zu nutzen. Es gibt sehr komfortable L<sup>A</sup>T<sub>E</sub>X-spezifische Menüs z. B. für den L<sup>A</sup>T<sub>E</sub>X-Formelsatz („komplette“ Symbolsätze), die Umgebungserstellung (Tabellen, Aufzählungen, ...) und vieles L<sup>A</sup>T<sub>E</sub>X-spezifische mehr, so daß man sich die L<sup>A</sup>T<sub>E</sub>X-Befehle (meist) nicht merken muß. Sie brauchen eine Aufzählungsumgebung (Wie war doch gleich der Name?) – kein Problem der entsprechende Knopf im Menü ist schnell entdeckt und schon steht die Umgebung auf dem „Papier“ und wartet darauf, mit Inhalt gefüllt zu werden. Für Windowsbenutzer, die L<sup>A</sup>T<sub>E</sub>X nur gelegentlich benutzen wollen, weil sie z. B. eine wichtige Arbeit doch nicht unbedingt in Word schreiben wollen, aber sonst eher *fastPrint*<sup>TM</sup> und/oder *FastKlick*<sup>TM</sup> bevorzugen, ist Winedt sicher eine Al-

## Resümee

Sicherlich sind auch WYSIWYG-Editoren nützlich, allerdings primär, wenn man wirklich sehr viel von der *What-you-see* Eigenschaft benötigt. So z. B. wenn ich wissen will ob die Zeile 35 in Fett, Kursiv, Schriftgröße 36, Font Frankenstein noch auf die Seite paßt oder umgebrochen wird. Auch wenn man eigentlich gar keinen Text sondern mehr ein „optisches Highlight“ entwickeln möchte, daß hier und da einige Textphrasen enthält, kann ein WYSIWYG-Wortprozessor helfen – obwohl man da vermutlich mit einem DTP-Tool besser bedient ist. Auch kann man bei sehr kurzen Dokumenten a la *fastPrint*<sup>TM</sup> sicher mit WYSIWYG ganz glücklich werden.

Sehr große Dokumente mit strukturierter Gliederung wie (Bücher, Diplomarbeiten, Dissertationen, Tagungsbeiträge, Artikel usw.) bringen aber recht schnell die Vorteile von L<sup>A</sup>T<sub>E</sub>X zum Vorschein – zumindest, wenn ich mir ein professionelles Aussehen und Lesbarkeit wünsche. Dort ist die Domäne von L<sup>A</sup>T<sub>E</sub>X und T<sub>E</sub>X, welches schon seit Jahren vorrangig im naturwissenschaftlichen Bereich als Standardtextsatzsystem gilt. Formale Schriftstücke wie DIN-Schreiben lassen sich auch wesentlich einfacher mit L<sup>A</sup>T<sub>E</sub>X setzen, als mit z. B. *MS-Word*, wo dies nur nach intensiver Schulung oder Handbuchstudie mittels „zwischen-den-Zeilenlesen-und-viel-Phantasie-einsetzen“ erfolgreich ist. Nicht zu letzt bleibt die Problematik des Formelsatzes, die bis jetzt in noch keinem WYSIWYG-Editor hinreichend gelöst ist, um L<sup>A</sup>T<sub>E</sub>X damit Konkurrenz zu machen. Man kann ja mal versuchen einige Seiten des Bronstein-Mathematik Nachschlagewerks in Word mit dem Formeleditor zu setzen (mal abgesehen davon, daß das Ergebnis esthetisch nicht zufrieden stellt) – eine sehr mühevollen bis unmögliche Arbeit.

Geht es um Qualität und Portabilität von Texten im großen Stil ist L<sup>A</sup>T<sub>E</sub>X ein Hilfsmittel erster Wahl, sind es nur kleine Dokumente oder Dokumente mit einem hohen Maß an optischer Individualität und Verspieltheit wird man mit WYSIWYG wohl glücklicher sein<sup>a</sup>.

<sup>a</sup>oder auch nicht :-)

ternative.

## Vor-/Nachteile WYSIWYG versus L<sup>A</sup>T<sub>E</sub>X

Welche weiteren praktischen Vorteile ergeben sich aus der Tatsache, daß eine L<sup>A</sup>T<sub>E</sub>X-Datei reiner ASCII-Text (nebst Format und Strukturweisungen) ist?

1. Übertragbarkeit: Jeder kann einen so gesetzten Text auf einem Computersystem einsehen, selbst wenn man den Text nicht in der fertig gesetzten Fassung sieht, sondern nur in der „binär – spricht roh – Form“. Im Gegensatz dazu, kann eine Datei der Textverarbeitung XY vollkommen unverständlich für einen Empfänger sein, der nicht die gleiche Art und Version dieser Textverarbeitung hat.
2. L<sup>A</sup>T<sub>E</sub>X-Dokumente sind meist kompakter, da sie nicht soviel binären Ballast mit sich herumtragen wie die Dateien von Schick XY.
3. Sicherheit: Der Texteditor und Textsatzsystem Ansatz bietet die größtmögliche Sicherheit, daß man niemals irgendwelche Probleme mit
- beschädigten Dokumenten hat. Der Quelltext ist immer da und kann ggf. korrigiert werden. Wer regelmäßig mit Textverarbeitungsprogrammen arbeitet und noch nie Probleme mit kaputten Dateien hatte, sollte sich vorsehen: Es ist dann nämlich auch nicht mehr so unwahrscheinlich, zusätzlich am nächsten Tag vom Blitz erschlagen zu werden, wobei man doch gerade im Lottospiel gewonnen hat.
4. Eine Verbreitung z. B. per email ist wesentlich einfacher, da die L<sup>A</sup>T<sub>E</sub>X-Dokumente kleiner, überall lesbar und sicher sind. Makroviren sind – eine vertrauensvolle L<sup>A</sup>T<sub>E</sub>X-Distribution vorausgesetzt – eigentlich kein Thema, zumal man die Möglichkeit der einfachen Überprüfung hat.
5. Eine Konvertierung in eine andere Markup-Sprache (XML, HTML) ist wesentlich einfacher, da die logische Struktur vom Inhalt weitgehend getrennt ist. Als schlechtes Beispiel seien hier von *MS-Word* erstellte HTML-Dateien angeführt, obgleich auch die Ausgabe von `latex2html` oft Wünsche übrig läßt.

6.  $\LaTeX$ -Dokumente können mit diversen Hilfsprogrammen in nahezu alle anderen Text-Formate konvertiert werden (z. B. `latex2html`, `dvips`, `pdflatex`). Über das Ergebnis läßt sich freilich streiten, aber durch die offengelegte Struktur ist man zumindest nicht nur auf *ein/eine* Produkt/-Firma festgelegt.
7.  $\LaTeX$  und etliche Pakete sind *Freie Software* und kosten damit keinen Pfennig an Lizenzgebühren. Trotzdem gibt es regelmäßige Updates und auch große Mengen an Erweiterungspaketen (natürlich auch kostenfrei) die einfach per Netz von [1] geladen werden können.

### Wofür $\LaTeX$ einsetzen?

Folgende Tabelle fasst nochmal alle wesentlichen Eigenschaften und Einsatzgebiete<sup>7</sup> (X) zusammen:

Was will ich	$\LaTeX$	WYSIWYG
Portabilität	X	
große Dokumente	X	
(natur-)wissenschaftliche Texte	X	
Flyer		eingeschränkt (DTP)
Dokumente ohne Struktur		X
Kunst, Deko		eingeschränkt (DTP)
Normierte Dokumente	X	eingeschränkt
technische Handbücher	x	

Die Folien, zu dem diesem Artikel vorangegangenen Vortrag, sind unter <http://www.gaos.org/vortrag> mit einer Vielzahl von  $\LaTeX$  Beispieldokumenten inklusive Quelltextdateien hinterlegt.

### Referenzen

- [1] Comprehensive  $\TeX$  Archive Network (CTAN)  
<http://www.ctan.org/>
- [2] Mik $\TeX$ -Distribution  
<http://www.miktex.org/>
- [3] WYSIWYG-Editor LYX  
<http://www.lyx.org/>
- [4] WordPerfect  
<http://www.wordperfect.com/>
- [5] StarOffice  
<http://www.stardivision.com/>
- [6] KWord/Koffice  
<http://www.koffice.org/>

<sup>7</sup>sinnvolle — unserer Meinung nach :-/

- [7] Winedt  
<http://www.winedt.com/>
- [8] AbiWord  
<http://www.abisource.com/>
- [9] Printy  
<http://www.printy.de/>
- [10] The  $\LaTeX$  Companion  
<http://www.aw.com/>
- [11] Laura E. Jackson, Herbert Voß: *LYX – Open Source Document Processor*  
DTK 3/2001 S. 24-45 (Teil 1), DTK 4/2001 S. 22-43 (Teil 2), DANTE e. V.
- [12] Volker Kiefel:  *$\LaTeX$  als normales „Schreibwerkzeug“ im Alltag*  
DTK 4/2001 S. 52-60, DANTE e. V.

### ▼ Killerapplikation

**SVEN TÜRPE: Eine Streitschrift wider den technischen Kopierschutz**

In der digitalen Rechteverwaltung sehen Musik- und Filmwirtschaft, Verleger und andere Inhaltsverwerter ihre Zukunft. Als Begründung muß meist die Möglichkeit des Kopierens digitaler Daten und deren angeblicher Mißbrauch herhalten. Die Auswirkungen der digitalen Rechteverwaltung gehen jedoch weit über die Sicherung angemessener Vergütung für Urheber hinaus. Der Artikel gibt einen Überblick, ohne sich um Neutralität zu bemühen.

**D**as digitale Zeitalter ist großartig. Die neue CD von Freund oder Freundin, Bruder, oder Komilitonin wandert in den Computer; heraus kommt eine fast perfekte Kopie. Zwei Menschen glücklich auf einen Schlag und für den Preis einer einzigen CD.

*Copy kills music* sagt dazu jene Branche, welche nach wie vor davon lebt, »Plastik in Lkws durch die Lande zu fahren«<sup>8</sup>. Wie sich Vertreter dieses Wirtschaftszweiges die Zukunft vorstellen, ist inzwischen bekannt [13]. Ob Plastik in Lkws oder Bytes in Kabeln, wer in Zukunft Musik hören möchte, möge bitteschön jedesmal vorher die *Pay*-Taste seiner Fernverum... ähh, -bedienung drücken. Falls sich das technisch durchsetzen läßt, ist die Weitergabe von Kopien dann sogar erwünscht – als kostenlose und effektive Marketingmaßnahme. Ob es überhaupt einen

<sup>8</sup>Gedächtniszitat; Andy Müller-Maguhn auf der Tagung *Digital Rights Management*, November 2000, Berlin

negativen Zusammenhang zwischen der kostenlosen Verfügbarkeit von Musik und den Umsätzen der Musikwirtschaft gibt, ist übrigens, gerade im Lichte solcher Marketing-Ideen, mindestens zweifelhaft [30].

### Zwangsmaßnahmen

Die Vorschläge, wie das Ziel totaler Kontrolle über die Nutzung von Medieninhalten zu erreichen sei, lassen an Skrupellosigkeit wenig zu wünschen übrig. »Kopiergeschützte«, das heißt auf Computern und auch auf manchen Audiogeräten nicht mehr abspielbare<sup>9</sup> CDs sind noch der harmloseste Ansatz. Wenn der Umsatz in Gefahr ist oder scheint, schreckt man auch vor der Forderung nach totaler Überwachung nicht zurück – und übertrifft dabei selbst die kühnsten Vorstellungen der selbsternannten Netzpolizisten, die nur Kinder pornos und Nazis aus unserem Blickfeld verbannen möchten. Für Umsatz und Gewinn darf es gleich eine Firewall [8, 16] um Deutschland sein, die ein Datenpaket nur dann zu durchdringen vermag, wenn es bezahlt ist oder Copperfield heißt<sup>10</sup>. Von dieser speziellen Idee ist man inzwischen zum Glück wieder abgerückt, dafür sucht man jetzt nach Wegen, die Integration von DRM-Systemen am besten gleich in sämtliche elektrisch betriebenen Geräte zu erzwingen [7, 24, 27].

Dabei übersieht die Musikwirtschaft so allerlei, und ignoriert manches mit Absicht. Zuerst die Möglichkeit, daß ein wirksamer Kopierschutz vielleicht überhaupt nicht existiert [20, 38]. Bislang wurde kein wirklich überzeugendes, auf realistischen Annahmen beruhendes Konzept dafür präsentiert, wohl aber eine Reihe unwirksamer Mechanismen. Für das (technische) Grundproblem, zwischen erwünschten, weil verkauften Kopien und unerwünschten zu unterscheiden, gibt es bis jetzt keine Lösung und vieles deutet darauf hin, daß es auch in Zukunft keine geben wird, jedenfalls nicht unter den Randbedingungen *Internet* und *PC als Endgerät*. Dies ist freilich kein Grund, sich beruhigt zurückzulehnen. Eine tatsächliche Wirkung entfalten technische Schutzmechanismen nämlich in einem gänzlich untechnischen Umfeld: im Gerichtssaal. Auch wenn ein Kopierschutzmechanismus aus

<sup>9</sup>Die vorgeschlagenen und teils bereits eingesetzten Verfahren beruhen auf Verletzungen jenes Standards, welcher jede CD auf jedem Gerät abspielbar macht(e). [14]

<sup>10</sup>Das digitale Zeitalter ist wirklich großartig: Jedes Datenpaket kann Copperfield heißen. Aber dies soll Thema eines anderen Artikels sein.

technischer Sicht völlig unbrauchbar, weil leicht zu umgehen ist, kann man ihn immer noch als Argument gegen denjenigen benutzen, der sich diese Designfehler zunutze macht. Diese Menschen werden leicht zu »Crackern« und »Piraten« stilisiert; man unterstellt ihnen, sie hätten kriminelle Energie aufgebracht, um »einen Kopierschutz zu brechen«, wo sie doch in Wirklichkeit lediglich das getan haben, was ihnen die Technik ermöglicht hat [28].

Die Absurdität dieser Argumentation wurde inzwischen deutlich demonstriert. Ein bestimmtes Kopierschutzverfahren für CDs läßt sich mit einem einfachen Filzstift »brechen«, Übermalen einer sichtbaren Spur auf der CD genügt. [12, 29]. Damit wird *jeder* zum potentiellen »Piraten« und ein Alltagsgegenstand zum »Hackertool« und gegenüber der Verwendung von Filzstiften nimmt sich das Ändern einzelner Bits in Dateien [26] in der Tat wie ein ausgewachsener Hack aus.

### Recht und Ordnung?

Noch einen Schritt weiter gehen Versuche, die technisch nicht umsetzbare Idee des Kopierschutzes nicht nur in der Rechtsprechung zu implementieren, sondern gleich in allgemeine Gesetze einzubringen. In den USA können solche Versuche seit Verabschiedung des *Digital Millennium Copyright Act (DMCA)* [23, 32] als erfolgreich gelten. In Europa gibt es Bestrebungen, ähnliche Gesetzeswerke beschließen zu lassen [18, 31]. So etwas wird übrigens schon lange nicht mehr in nationalen Parlamenten ausdiskutiert und beschlossen, sondern in supranationalen Zirkeln, die so klingende Namen wie TRIPS, WIPO, WTO, G8 oder EU-Kommission tragen [31, 33]. Wenn Sie beim Stichwort *Globalisierung* immer nur an das Elend der Dritten Welt gedacht haben, so sollten Sie sich ab sofort persönlich betroffen fühlen.

Der Erfolg in Sachen DMCA hat die *Intellectual Property*-Wirtschaft allerdings keineswegs beruhigt. Im Gegenteil, nun geht es erst richtig los. Als nächstes sollen »Hackertools« verboten werden, darunter jegliche Software, die das Umgehen von Kopierschutzmaßnahmen erlaubt. Wir erinnern uns: ein wirksamer Kopierschutz im technischen Sinne ist bislang unbekannt . . .

Der Kontrollgedanke beschränkt sich nicht auf das Verhindern von Privatkopien. Filmfreunde können ein Lied singen vom Ärger, den die Regionalkodierung von DVDs verursacht. Dort ist die Welt in Zonen einge-

teilt und jeder DVD-Player soll nur Filme abspielen können, die für seine Zone freigegeben sind. Völlig hirnrissig, aber eine Last für jeden, der zum Beispiel des Englischen mächtig ist und sich gern selbst importierte amerikanische Originalfassungen anschaut. Hier finden wir dasselbe Muster wie bei den CDs. Die Technik funktioniert nicht wie gewollt; *Code-free* hat sich zu einem echten Qualitätsmerkmal für DVD-Spieler entwickelt. Dafür wird auf der juristischen Seite kräftig zugeschlagen und schon mal der Warenbestand eines Händlers beschlagnahmt, welcher sich erdreistet hatte, DVDs zu importieren [15]. Langsam setzt sich allerdings die Erkenntnis durch, daß es hier nur um Monopole und Handelshemmnisse geht und nicht um berechnete Interessen von Urhebern [11].

### Dichter und Denker

Ein anderer Ignoranzschwerpunkt wird erst auf den zweiten Blick deutlich. Dann nämlich, wenn wir die *digitalen Güter* genauer betrachten, mit den so wenige so viel Geld verdienen. Es sind Kulturgüter, die eben nicht nur eine wirtschaftliche, sondern auch eine gesellschaftliche Bedeutung haben. Die Qualität mag sich von Mozart zu Britney Spears gewandelt haben, aber die Wichtigkeit hat in unserer Mediengesellschaft noch zugenommen. Wer nie CNN geschaut hat, versteht kaum noch seine Tageszeitung, jedenfalls wenn es eine bessere ist. Und da geht es nur um Nachrichten, um Alltagskultur und -wissen mit Verfallsdatum. Daneben gibt es aber auch Kulturgüter, die es wert sind, Jahrhunderte zu überdauern. Unsere Welt wäre eine andere ohne die allgemeine Zugänglichkeit der Schriften von Goethe und Schopenhauer, der Musik von Bach und Chopin, der Bilder von Rembrandt und van Gogh, aber auch der wissenschaftlichen Abhandlungen von Euler oder Newton. Allein die Aufzählung der bedeutenden Namen könnte Bücher füllen.

Zugang zu Kultur ist Menschenrecht. Wer dieses Recht einschränkt, gefährdet das gesamte Gefüge unserer Gesellschaft. Ein Artikel wie der, den Sie gerade lesen, wird unmöglich ohne den Zugang zu Quellen, nicht nur den im Anhang genannten, sondern auch den vielen wieder vergessenen, die zur Meinungsbildung beigetragen haben. Und nicht nur die explizite Form des Artikels wird unmöglich, sondern auch der Vorgang der Meinungsbildung als solcher. Große Gedanken gehen einfach unter, wenn zeitlich beschränkte Lizenzen auslaufen. Die Lebensqualität sinkt, wenn Musik, Theater und Film fehlen. Zu jener Zeit, da die

Urfassung dieses Artikels entstand, flimmerte gerade ein Beispiel von hoher Symbolkraft über die Fernsehschirme.<sup>11</sup> Nach Jahren der Taliban-Herrschaft drängten sich Menschen in einem afghanischen Kino, um endlich wieder einen Film sehen zu können. Deutlicher läßt sich der Zusammenhang zwischen Kultur und Gesellschaftlichem Sein kaum ausdrücken.

Zum Zugang aber gehört die Möglichkeit der Aufzeichnung und des freien Kopierens von Aufzeichnungen. Das gilt gerade dort, wo die gesellschaftlichen Verhältnisse die Kultur beschränken. Der Autor ist in der DDR aufgewachsen, ja fast erwachsen geworden. Zur Ignoranz der Herrschenden kamen dort banale wirtschaftliche Schwierigkeiten. In ein Schallplatten-geschäft zu gehen und nach irgend einem der auch heute noch recht populären Songs der 80er zu fragen, wäre schlicht naiv gewesen. Nur wenig war überhaupt verfügbar und dieses Wenige war schneller verkauft als geliefert. Ein Eigenimport aber hätte sehr spendable Verwandtschaft oder einen üppigen Devisenvorrat erfordert. So blieb als praktisch einziger Zugang zu westlicher Musik das Radio – zusammen mit dem Kassettenrecorder zur Konservierung. Auf diese Weise wurde allen Schwierigkeiten zum Trotz nahezu die gesamte Jugend der DDR auch westlich sozialisiert, was zweifellos einen großen Beitrag zum Untergang dieses Staates geleistet hat.

Im Umkehrschluß führt diese Beobachtung zu der Vermutung, daß ein von der Wirtschaft etabliertes Kontrollsystem in der Folge auch weniger kulturvollen Herrschern als Unterdrückungsinstrument genutzt werden könnte. Wir sind nicht immun gegen Diktatoren, und wir sind im Begriff, ihnen auch auf diesem Gebiet schon mal die Instrumente zur Verfügung zu stellen, mit denen sie im Ernstfall Unterdrückung implementieren und ihr gar einen Anschein von Legitimität geben könnten. Auch hierfür liefert die friedlich dahingeschiedene DDR Vorbilder. In den Bibliotheken gab es Giftschränke, deren Inhalt nicht jeder lesen durfte. Heute schicken sich Verleger im Zuge der Digitalisierung an, die Archivierung in Bibliotheken gleich gänzlich zu verhindern. Effektiver könnte eine Re-Implementierung kaum sein. »Gefährliche« Werke werden nicht mehr weggeschlossen, sondern machen sich selbst unbrauchbar. Wer Richard Stallmans Horrorvision von einer Zukunft mit allgegenwärtigem Kopierschutz [35] für übertrieben hielt, kann sich jetzt

<sup>11</sup>Und der Autor muß niemanden um Erlaubnis fragen, bevor er Ihnen davon erzählen darf. Bis vor kurzem war das selbstverständlich, aber das ändert sich gerade.

leicht eine noch dramatischere ausdenken. Für Freiheitsrechte interessieren sich die DRM-Advokaten jedenfalls nicht [22].

»Technology can be a liberating influence. Everyman with his own writing or copying machine is freer than the peasant who must turn to a scribe or a priest to express himself beyond the family table.«[5] Kaum aber hat sich herausgestellt, daß die Senderseite dank allgemein verfügbarer Computer und Netzzugänge praktisch nicht mehr kontrollierbar ist, schickt man sich an, Kontrollmechanismen auf Rezipientenseite zu etablieren. Die Hauptrolle spielen dabei nicht etwa politische Extremisten, sondern die Wirtschaft.

Das Muster des Immer-noch-eins-Draufsetzens im Kampf um Kontrolle zeigt sich erneut, wenn wir den Blick von geistigen Kulturgütern ab- und der Natur zuwenden. Mit der Entwicklung der Land- und Forstwirtschaft hat sich auch hier über Jahrtausende eine Kultur herausgebildet, eine Kultur der gestaltenden Naturnutzung. Sie bildet eine viel unmittelbarere Grundlage unseres Seins als es die geistige Kultur je sein kann, denn sie versorgt uns mit dem Nötigsten, mit Nahrungsmitteln. Selbst hier schreckt eine gierige Wirtschaft vor massiven Eingriffen nicht zurück. Über Jahrtausende haben Bauern aus ihrer Ernte Saatgut für das folgende Jahr gewonnen. Ein Vorgang wie er natürlicher nicht sein könnte, ist dies doch der natürliche Vermehrungsmechanismus von Pflanzen. Mit der Gentechnik aber kamen die Biopatente und mit den Biopatenten Zahlungsaufforderungen von Saatgutherstellern. Neuerdings soll einem Bauern sein einmal gekauftes Saatgut nicht mehr gehören. Statt dessen bekommt er eine Lizenz dafür, eine Menge  $X$  des patentierten Saatgutes  $Y$  zu verwenden, unabhängig davon, woher er es bezogen hat [1, 6, 25]. Weitergehende Pläne liegen schon in den Schubladen. Sobald es politisch durchsetzbar ist, wird die Wiederaussaat gentechnisch verhindert – ein technischer Kopierschutz auf Pflanzen [37], willkürlich eingeführt als Gewinngarantie.

### Horrorvisionen?

Wer sich plastisch ausmalt, was sich derzeit noch nur skizzenhaft andeuten, muß zwangsläufig zu dem Schluß kommen, daß er in einer solchen Welt nicht leben möchte. Was in der Folge die Frage aufwirft, wie man denn rechtzeitig etwas dagegen tun kann, um nicht hinterher hilflos vor seinen fragenden Kindern

zu stehen, falls man dann überhaupt noch lizenzfrei welche zeugen, gebären und großziehen darf. An Demonstrationen teilzunehmen und hernach in italienischen Gefängnissen zu verrotten ist eine Möglichkeit, aktives Mitglied in der GAOS e.V.<sup>12</sup> zu werden eine andere. Weitere Vorschläge nimmt der Autor gern per E-Mail entgegen. Ohne Kopierschutz, versteht sich, aber mit der Garantie, daß die Urheber auf Wunsch bei Verwendung namentlich genannt werden.

Übrigens, der eingangs beschriebene Vorgang heißt *Privatkopie* [2] und hätte vor dreißig Jahren mit Schallplatte und Magnetband ebenso funktioniert. Er war damals legal und ist es heute noch [17], und in Form von Geräte- und Leermedienabgaben existiert sogar ein Mechanismus zur Vergütung der Urheber für diese Form der Nutzung ihrer Werke. Und die Musikindustrie karrt immer noch Plastik in Lkws durch die Landschaft; offensichtlich lebt sie ganz gut davon. Kopiermöglichkeiten allein sind also nicht die Killerapplikation, an der ein Wirtschaftszweig zugrunde geht. Kopierschutztechnik aber könnte es werden, denn hier soll nicht etwa das Copyright an den Stand der Technik angepaßt werden, wie die Verfechter gern behaupten, sondern die Technik soll sich einer inakzeptablen Reduzierung von Kulturgütern auf bloße Handelsware mit Umsatzgarantie anpassen.

Wohlgemerkt, dieser Artikel will nicht einer Verpflichtung zur Kostenloskultur das Wort reden. Wer von seinem Urheberrecht dergestalt Gebrauch machen möchte, daß er sich die Nutzung und Vervielfältigung seiner Werke bezahlen läßt, soll das durchaus tun dürfen. Aber nicht mit beliebigen Mitteln und nicht um den Preis, daß der Wirtschaftszweig der Rechteinhaber die Kontrolle über die Weitergabe und Nutzung von Wissen und Kulturgütern erhält. Ein Wirtschaftszweig, der allen Ernstes das laute Vorlesen oder das Verleihen von Elektrobüchern als mögliche Rechteinverletzung ansieht [19], der Sekundärliteratur zu beliebigen Kinderbüchern mit marken- und urheberrechtlichen Argumenten bekämpft [3] und der Wegschauen bei Fernsehwerbung für Diebstahl hält [34]. Und der nicht mit den Urhebern verwechselt werden sollte, obwohl er sich gern auf das Urheberrecht beruft. Ein Verlag schreibt nicht, eine Plattenfirma singt nicht und Hollywood ist weder ein Schauspieler noch ein Drehbuchautor oder Regisseur. Kontrolle der tatsächlichen Urheber über ihre eigenen Werke aber ist nicht vorgesehen in der schönen neuen DRM-Welt.

<sup>12</sup><http://gaos.org>

## Referenzen

- [1] *Biopiraterie bei Sojapflanzen*. Die Tageszeitung, 2001-10-23 (www.taz.de).
- [2] Bogk, A.: *Die Abschaffung der Privatkopie*. <http://www.andreas.org/DieAbschaffungDerPrivatkopie>
- [3] *Die Pottermarke*. Süddeutsche Zeitung, 2002-06-07.
- [4] Emmert, U.: *Hackertools im Visier der Justiz*. c't 02/2002, S. 76
- [5] Felix, C.: *A Short Course in the Secret War*. Madison Books, 2001.
- [6] *Genpatent jetzt auch aufs Feld*. Die Tageszeitung, 2002-02-06 (www.taz.de).
- [7] Godwin, M.: *Hollywood vs. the Internet Why entertainment companies want to hack your computer*. <http://www.reason.com/0205/fe.mg.hollywood.shtml>
- [8] Goltzsch, P.: *Nationales Internet*. Telepolis, 2000-02-24.
- [9] Grötter, R.: *Wer hören will, muß zahlen*. Die Zeit Nr. 24 (6. Juni) 2002.
- [10] Hansen, S.: *Ohren zu und durch*. c't 02/2002, S. 3 <http://www.heise.de/tp/deutsch/inhalt/te/5833/1.html>
- [11] Heise-Newsticker: *Australien nimmt DVD-Regionalcode unter die Lupe*. <http://www.heise.de/newsticker/data/ghi-26.05.01-000/>
- [12] Heise-Newsticker: *CD-Kopierschutz: Dann müssen eben Filzstifte verboten werden*. <http://www.heise.de/newsticker/data/ghi-24.05.02-000/>
- [13] Heise-Newsticker: *Die Medien-Branche setzt auf neue Systeme zum Kopierschutz*. <http://www.heise.de/newsticker/data/jk-10.12.01-000/>
- [14] Heise-Newsticker: *Kopierschutz bei Musik-CDs löst Streit aus*. <http://www.heise.de/newsticker/data/jk-04.03.02-001/>
- [15] Heise-Newsticker: *Razzia bei Region-1-DVD-Händler*. <http://www.heise.de/newsticker/data/nij-22.05.01-001/>
- [16] Heise-Newsticker: *Grenzkontrollen im Internet*. <http://www.heise.de/newsticker/data/fr-24.02.00-002/>
- [17] Himmelein, S.: *Sind wir alle kriminell?* c't 02/2002, S. 80
- [18] Himmelein, S.; Schmitz, P.: *Fast alles ist verboten* c't 02/2002, S. 82
- [19] Kirvin, J.: *Unfair Use*. <http://www.jeffkirvin.com/writingonyourpalm/column010108.htm>
- [20] Kirvin, J.: *Gemstar and the Suicide of the eBook »Industry«*. <http://www.jeffkirvin.com/writingonyourpalm/column001009.htm>
- [21] Landley, R.: *Intellectual Property Is an Oxymoron*. <http://www.fool.com/news/foth/2000/foth000814.htm>
- [22] Mühlbauer, P.: *Content is King! oder die Diktatur des Kleingedruckten*. Telepolis, 2002-02-15. <http://www.heise.de/tp/deutsch/special/copy/11844/1.html>
- [23] Röttgers, J.: *In den Mühlen des Digital Millennium Copyright Act*. Telepolis, 2002-11-29. <http://www.heise.de/tp/deutsch/inhalt/te/11236/1.html>
- [24] Rötzer, F.: *Kopierschutztechnik in alle digitalen Geräte*. Telepolis, 2002-03-22. <http://www.heise.de/tp/deutsch/special/copy/12147/1.html>
- [25] Schmeiser, P.: *Monsanto vs Schmeiser*. <http://www.percyschmeiser.com/>

- [26] /.: *Font Company Wielding DMCA Against Bit-Flipping.*  
<http://slashdot.org/article.pl?sid=02/05/01/2026234>
- [27] /.: *Congress Plans DMCA Sequel: The SSSCA.*  
<http://slashdot.org/article.pl?sid=01/09/08/0238200>
- [28] /.: *Fallout From Def Con: Ebook Hacker Arrested by FBI.*  
<http://slashdot.org/article.pl?sid=01/07/17/130226>
- [29] /.: *Felt Tip Marker Defeats Copy-Protected CDs.*  
<http://slashdot.org/article.pl?sid=02/05/22/1439253>
- [30] /.: *RIAA Almost Down To Pre-Napster Revenues.*  
<http://slashdot.org/article.pl?sid=02/02/27/0213252>
- [31] /.: *The EU's Answer To The DMCA.*  
<http://slashdot.org/article.pl?sid=01/10/27/2326226>
- [32] /.: *Three Years Under the DMCA.*  
<http://slashdot.org/article.pl?sid=02/05/05/1157235>
- [33] /.: *World Copyright Treaty Coming soon.*  
<http://slashdot.org/article.pl?sid=01/12/07/1651252>
- [34] Siegle, J. A.: *Wer pinkeln geht, ist ein Dieb.* Spiegel Online, 2002-06-13  
<http://www.spiegel.de/netzwelt/netzkultur/0,1518,200428,00.html>
- [35] Stallmann, R.: *The Right to Read.* Communications of the ACM Vol 40 No 2, February 1997.  
<http://www.gnu.org/philosophy/right-to-read.html>
- [36] Staun, H.: *Kontrolle rückwärts.* Frankfurter Allgemeine Sonntagszeitung Nr. 13 (31. März) 2002.
- [37] *Über die Ernte entscheidet die WTO.* Die Tageszeitung, 2001-11-12 ([www.taz.de](http://www.taz.de)).
- [38] Zota, Dr. V.: *Klonverbot.* c't 02/2002, S. 90

## ▼ OCR mit statistischen Momenten

FRANK-MICHAEL SCHLEIF: Neue Aspekte der Schriftzeichenerkennung

Im folgenden Artikel betrachten wir die Anwendungsmöglichkeiten von statistischen Momenten für die Erkennung von handschriftlichen Texten. Nach einer kurzen Einführung wird eine einfache Form statistischer Momente vorgestellt und an konkreten Beispielen und Untersuchungen deren Nutzen für ein potentielles Open-Source-OCR-System gezeigt. Der Artikel erhebt nicht den Anspruch einer vollständigen Einführung sondern soll vielmehr einen kleinen Einblick hinter die Kulissen der OCR liefern und eine Motivation zur Auseinandersetzung mit diesem interessanten Gebiet sein.

Der normale Alltag ist heute durch eine Vielzahl von Schriftstücken und Formularen geprägt, die häufig noch vollständig durch Handarbeit erfaßt werden müssen und auch das vor einigen Jahren geweisagte papierlose Büro läßt noch immer auf sich warten.

Eine Lösung für dieses Problem bietet die OCR oder Schriftzeichenerkennung. Dabei werden die in Abbildung dargestellten Arbeitsschritte durchlaufen, von denen hier nur die Merkmalsgewinnung berücksichtigt wird.



Vielfältige Ansätze zur Schriftzeichenerkennung wurden in den letzten Jahren entwickelt. Anschaulich aber naiv könnte man mit Matchingvergleichen (Template-Matching) beginnen.

Dabei wird das zu erkennende Objekt mit allen z. B. in einer Datenbank abgelegten Mustern verglichen (z.B. durch Überlagerung) und der daraus ermittelten, passenden Klasse zugeschlagen.

Nun ist diese Methodik jedoch sehr aufwendig, da bei entsprechend vielen Klassen sehr viele Vergleiche erforderlich werden, ausserdem ist dieser Ansatz nicht sonderlich geeignet, wenn die Muster sehr starke Varianzen aufweisen, da sonst, um das Verfahren funktionsfähig zu halten, für all diese Ausprägungen Templates zum Vergleich vorliegen müßen. Da Schriftzeichen i. A. in verschiedenen Rotationsgraden oder noch schlimmer Schreibstilen auftreten können ist dieser Ansatz eher nicht geeignet.<sup>13</sup> Andere – merkmalsbasierte – Ansätze sind die Anwendung verschiedener Transformationen über den Symbolen, beispielsweise die Fouriertransformation, bei der die ermittelten Ko-

<sup>13</sup>Allerdings ist mit zunehmender Leistungsfähigkeit der Rechner dieses Verfahren zumindest für Druckbuchstaben wieder realistischer geworden.

effizienten als Merkmale genutzt werden. Dabei weisen diese Merkmale eine weit größere Robustheit bzgl. Variationen in der Datenmenge auf und sind vor allem sehr viel einfacher zu berechnen.

### Statistische Momente in der OCR

Das in diesem Artikel vorgestellte Verfahren basiert auf statistischen Momenten, wie sie wohl jedem zumindest in ihrer einfachsten Form bereits einmal begegnet sind. Dabei wird das Symbol als eine 2-dimensionale Dichtefunktion verstanden die auf bestimmte Polynome projiziert wird. Die dabei ermittelbaren Abstandsmaße werden dann als Merkmale genutzt und weisen je nach Verfahrensweise verschiedene Invarianzeigenschaften auf.<sup>14</sup> Zusätzlich dazu erfaßt daß Verfahren eine „beliebig“ genaue Modellierung der darunterliegenden Dichtefunktion (also in unserem Fall des Symbols) und ist damit relativ robust gegenüber individuellen Schreibstilen, was die Eignung zur Erkennung handschriftlicher Texte impliziert. Im folgenden nun zunächst die simpelste Definition von statistischen Momenten.<sup>15</sup>

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (1)$$

Die Merkmale sind dabei die Werte  $m_{pq}$ , wobei das Symbol gegeben durch  $f(x, y)$  auf die monalen Polynome  $x^p, y^q$  abgebildet wird. In einem Satz von HU wird erklärt, daß aus einer über einem Symbol erfaßten unendlichen Mengen dieser  $m_{pq}$  das Bild wieder rekonstruiert werden kann. Somit steckt die Information über die Struktur des Symbols in den  $m_{pq}$ . Es ist damit auch möglich das Symbol aus einer Folge von  $m_{pq}$  zu rekonstruieren, wie es in der Abbildung 1 exemplarisch dargestellt ist.

### Tsirikolias-Mertzios Momente

Die einfachsten Momente sind der Erwartungswert bzgl.  $x$  ( $m_{10}$ ) oder  $y$  ( $m_{01}$ ). Diese Merkmale weisen zunächst keine Invarianzeigenschaften auf, dies wäre aber für die Schrifterkennung ganz nützlich, da es ja immermal kleine Verschiebungen im Schriftbild gibt. Dies kann leicht durch eine Normierung bzgl. eines

<sup>14</sup>So kann man die ermittelten Merkmale z. B. gegenüber Rotation, Verschiebung, Skalierung, Kontrastvarianz, ... invariant gestalten.

<sup>15</sup>Dieses Verfahren ist übrigens keineswegs neu, da die erste Anwendung für die OCR auf Arbeiten von HU um 1960 datiert.



(a) orig.



(b) rec.

Abbildung 1: Beispiel zur Rekonstruktionsfähigkeit mit statistischen Momenten (Symbol aus [5])

unveränderlichen Attributes des Symbols erreicht werden, dies ist z.B. der Erwartungswert oder das Prinzip des Massezentrum, bei dem der Schwerpunkt des Bildes genutzt wird. Im folgenden wählen wir den ersten Ansatz und fügen zusätzlich noch die Skalierungsinvarianz unter Normierung hinsichtlich der Varianz des Bildes hinzu.

$$tm_{pq} = \frac{1}{LM} \sum_{i=1}^L \sum_{j=1}^M \left( \frac{x - \bar{x}}{\sigma_x} \right)^p \left( \frac{y - \bar{y}}{\sigma_y} \right)^q f(x, y) \quad (2)$$

Diese Form von statistischen Momenten wurden von TSIRIKOLIAS und MERTZIOS [3] vorgeschlagen und ist invariant bzgl. Translation und Skalierung. Diese beiden Normierungen sind meist bereits ausreichend. Für andere Invarianzen, beispielsweise bzgl. Rotation, kann man kombinierte Momente berechnen, die auf Basis der Invariantentheorie ermittelt wurden (vgl. auch HU [2, 1]) bzw. kann man auch andere Momentdefinitionen nutzen, die z. B. bereits in ihrer Definition begründet invariant bzgl. Rotation sind (komplexwertige Momente in polaren Koordinaten) - siehe auch [6].

### Anwendung in einem OCR-System

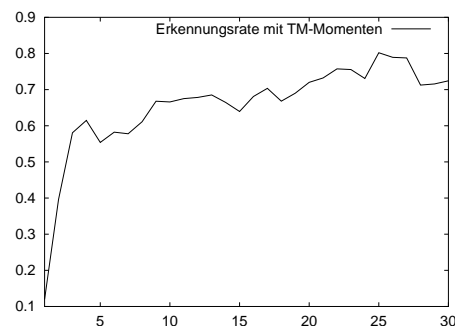
Werden die Symbole bzgl. Translation und Skalierung gestört, so sollten die Werte für den normierten Fall annähernd gleich und für den unnormierten Fall (bei Verwendung der Momente lt. Definition 1) unterschiedlich sein. In der folgenden Tabelle wurden die Werte für beide Fälle für verschiedene Momente ermittelt und verglichen. Dabei wurde das Symbol aus Darstellung 1 mit  $32 \times 25$  bzw.  $14 \times 12$  Pixeln (Skalierung) in einer  $64 \times 64$  Pixel großen Matrix verschoben (Translation).

Typ	Parameter	Translation	Transl. + Skal.
(GM)	(2, 0)	215184	11044
	(0, 2)	194018	9709
	(1, 1)	189333	9775
(GM)	(2, 0)	72414	3939
	(0, 2)	81554	4654
	(1, 1)	61935	3699
(GM)	(2, 0)	65808	3939
	(0, 2)	375004	15157
	(1, 1)	138641	7038
(TM)	(3, 0)	-0.0662	0.07858
	(0, 3)	0.10024	0.100692
	(1, 1)	-0.5178	-0.458401
(TM)	(3, 0)	-0.0662	0.07858
	(0, 3)	0.10024	0.100692
	(1, 1)	-0.5178	-0.458401
(TM)	(3, 0)	-0.0662	0.07858
	(0, 3)	0.10024	0.100692
	(1, 1)	-0.5178	-0.458401

Die mit GM bezeichneten Abschnitte entsprechen den Werten für die geometrischen Momente und die mit TM bezeichneten den Tsirikolias-Mertzios Momente. Dabei sind die Parameter  $p, q$  in der zweiten Spalte angegeben und in der dritten Spalte die berechneten Werte für das verschoben Symbol, in der letzten Spalte folgen noch die Werte für das Symbol '5' bei einmaliger Skalierung auf  $14 \times 12$  ebenfalls dreifach verschoben. Bei einfacher Invarianz (bzgl. Translation oder Skalierung) müssten die Werte eines Momententyps bei gleichen Parametern innerhalb einer Spalte übereinstimmen. Bei Invarianz bzgl. Skalierung und Translation müssten die Werte innerhalb einer Zeile bei gleicher Parameterisierung gleich sein. Wie man erkennt weisen die geometrischen Momente keinerlei Invarianz auf, für die Tsirikolias-Mertzios Momente zeigt sich jedoch, dass sie bzgl. Translation eindeutig invariant sind, da die entsprechenden Werte für jede Verschiebung identisch sind. Auch bzgl. Skalierung sind die Werte sehr ähnlich zu den unskalierten, so daß die Skalierungsinvarianz offenbar wird. Das die Werte nicht völlig gleich sind, ist auf den bei der Skalierung (Verkleinerung) aufgetretenen Diskretisierungsfehler zurückzuführen.

Für die weiteren Untersuchungen wurden ca. 3000 Symbole in 10 Klassen (0-9) aus der frei verfügbaren NIST-Datenbank [4] genutzt, um die Erkennungsraten bei Verwendung der Momente entsprechend Definition 2 zu prüfen. Dazu wurden 2000 der obigen Symbole zum Training des Systems genutzt und 1000 zum Testen. In Abbildung sind die Erkennungsraten bei unterschiedlich vielen genutzten TM-Moment Merkmalen dargestellt.<sup>16</sup>

<sup>16</sup>Als Klassifikation wurde Kreuzkorrelation mit Erwartungswert/Varianz-Normierung verwendet



Dabei ergibt sich bei Verwendung von 25 Merkmalen<sup>17</sup> eine maximale Erkennungsrate von 80 %. Dies ist schon relativ gut und im regelfall ausreichend, da nach der Erkennung in der Nachverarbeitung meist noch mittels Wörterbuchverfahren kleinere Fehler korrigiert werden können.

### Zusammenfassung

Es zeigt sich, daß die statistischen Momente nach MERTZIOS geeignet sind, um eine adäquate Erkennungsrate, auch bei handschriftlichen Daten zu erreichen. Dabei waren die Erkennungszeiten aufgrund der relativ einfachen Berechnungen sehr kurz, so daß der Einsatz dieser Momente in einem OCR-System günstig erscheint. Natürlich war dies nur ein kleiner Teil eines OCR-Systems, denn wie schon angedeutet darf man auch die Segmentierung und die Nachverarbeitung nicht vergessen. Aber wer jetzt Lust hat damit ein wenig zu experimentieren, der sei nocheinmal auf die NIST Datenbank [4] verwiesen, denn dort hat man zumindest schon einmal segmentiertes – freies – Rohmaterial. Wenn man sich nun noch eine einfache Segmentierung (z. B. Histogramm basiert) davor schreibt und vielleicht eine wörterbuchbasierte Korrektur im Anschluss<sup>18</sup> hat man fast schon sein kleines privates OCR System (und das ganz kostenfrei).

### Referenzen

- [1] Jan Flusser. On the independence of rotation moment invariants. *Pattern Recognition*, 33:1405–

<sup>17</sup>Verwendet wurden die mit *Sequential Forward Selection* ermittelten 30 besten Merkmale aus einer Mengen von TM-Momenten bis Ordnung 20 ( $p \leq 20$ ).

<sup>18</sup>Zumindest Wortlisten mit den 10000 häufigsten dt./engl. Wörtern kann man unter <http://wortschatz.uni-leipzig.de/html/wliste.html> herunterladen.

- 1410, 2000.
- [2] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on information theory*, 8:179–187, 1962.
- [3] B.G.Mertzios K. Tsirikolias. Statistical pattern recognition using efficient two-dimensional moments with applications to character recognition. *Pattern Recognition*, 20:877–883, 1993.
- [4] J. L. Blue M. D. Garris. *NIST Form-Based Handprint Recognition System*. National Institute of Standards and Technology, 1995.
- [5] National Institut of Standards and Technology. *FL3 Handwritten Symbol Database - Subset of the NIST Special Database 1*. NIST, ftp://sequoyah.ncsl.nist.gov/pub/databases/data.
- [6] Michael Reed Teague. Image analysis via the general theory of moments. *Journal of Optical Society of America*, 70:920–930, 1980.

### ▼ Semantik im Wortschatz

STEPHAN BORDAG: Wortbedeutungen ohne Weltwissen?

Sprache automatisch zu verarbeiten ist ein seit langem verfolgtes Ziel und eine der ersten Anwendungen von Rechenmaschinen überhaupt. Ging es ursprünglich lediglich darum, kodierte Nachrichten zu dekodieren und damit einen natürlich-sprachlichen Text von einer Darstellungsform in eine andere zu überführen, wachsen die Ansprüche an zu verarbeitenden Aufgaben mindestens proportional mit den Lösungen. Nicht lang dauerte es, bis es etwa die ersten „Bots“ gab, die ein Gespräch simulierten oder automatische Translationsysteme und Dokumentklassifikationssysteme. Bei all den Anwendungen war eines immer klar – das Verstehen des Textes kann lediglich simuliert werden. Es handelte sich stets um rein mechanische Verfahren.

Der Bot zum Beispiel verfügt über eine mehr oder weniger lange Liste von Fragen und Antworten, die er mit der Eingabe und unter Zuhilfenahme von regulären Ausdrücken zu „matchen“ versucht [vgl. Artikel *Regular Expressions*, GÄßTENBLATT, 2/2001]. Das Translationsystem hat eine Liste von Phrasen mit ihren exakten Übersetzungen, plus ein normales Wörterbuch, während Dokumentklassifikationssysteme die

Dokumente nach dem häufigen Auftreten bestimmter Wörter klassifizieren. Und das ist einer der Punkte, an dem gegenwärtig eine Weiterentwicklung zu beobachten ist. Es lohnt nämlich nicht, alle Wörter zu betrachten. Eigentlich, so hat man festgestellt, kann man in der Benutzung der Sprache das *Zipfsche Gesetz* beobachten, welches unter anderem besagt, daß je häufiger ein Wort auftritt, umso weniger Inhalt besitzt es. Und am häufigsten treten natürlich sämtliche Artikel, Pronomen, Zählwörter und ähnliches auf.

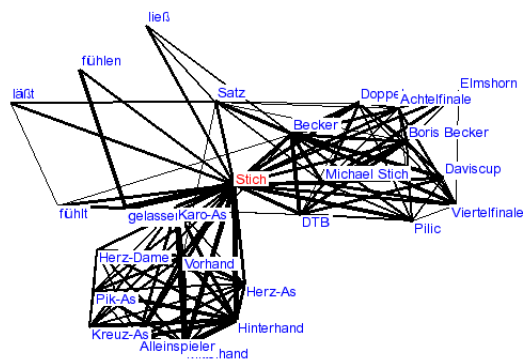
Nimmt man nun also einen Text und filtert alle „häufigen“ Wörter heraus, bekommt man etwas Telegrammstilartiges und was vor allem wichtig ist – der Text ist noch verständlich. Je tiefer die Frequenz der zu filternden Wörter eingestellt wird, umso schwieriger wird es, den genauen Inhalt eines Textes wiederzugeben, aber selbst wenn nur eine Handvoll Wörter übrigbleiben läßt sich immer noch bestimmen, um was es sich ungefähr handelte.

Das wiederum zeigt nur zu deutlich, daß es sinnvoll für die Weiterentwicklung von automatischen Sprachverarbeitungsanwendungen wäre, würde eine größere Ansammlung von Text auf weitere statistische Eigenschaften untersucht werden und dieses Ziel wird mit dem *Wortschatz Leipzig* [1] verfolgt. In einer Datenbank werden täglich Texte aus allen größeren deutschsprachigen Zeitungen in Sätze und Wörter zerlegt und als solche gespeichert, bzw. absolute Anzahlen gemessen. Danach werden diverse Untersuchungen darauf durchgeführt, mit dem Ziel, robuste statistische Effekte zu finden, mit denen das „Verstehen“ von neuem Text noch besser simuliert werden kann, als der bloße Vergleich von einigen seltenen Wörtern zwischen zwei Dokumenten. Ein interessanter Effekt ist die *Satzkookkurrenz*. Es wird beobachtet, welche Wörter statistisch signifikant oft mit welchen anderen in Sätzen auftreten. Das richtige Maß angewendet ergibt diese Methode recht interessante Ergebnisse. So läßt sich zu jedem solcherart gefundenen Wortpaar sagt, daß es „etwas“ miteinander zu tun hat, wie zum Beispiel («rote» «Ampel») oder («Gold» und «Kupfer») usw., allerdings stellt sich hier gleich das nächste Problem: Diese Methode scheint zu grob zu sein, sie mischt ohne weiteres nahezu alle Arten von Beziehungen, die zwischen Wörtern existieren können. Ein Übersicht gibt es in A. LEHR'S Dissertation [4] zu diesem Thema.

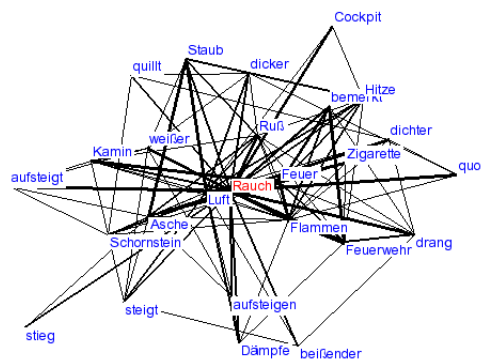
**Beobachtungen im Wortschatz**

Nennt man die Menge aller zu einem Wort gefundenen Kookkurrenzen die Assoziationsmenge des Wortes, so finden sich in dieser Menge stets über Verben, Adjektive bis hin zu Substantiven alle möglichen Klassen von Wörtern. Darüberhinaus ist diese Menge Bedeutungsgemischt. Man nehme das Wort «Stich». Dann treten in seiner Assoziationsmenge sowohl Begriffe aus dem Tennisbereich auf, wegen «Michael Stich», als auch Begriffe wie «Karo-As» oder «Dame» oder «König». Gefragt ist also ein Verfahren, welches diese Assoziationsmengen weiter aufsplitten kann und die Wörter nach ihren Bedeutungen oder Kontexten, in denen dieses Wort benutzt wird, trennen kann. Die Wortklassen dann lassen sich bereits durch auf Silbenebene basierenden Verfahren erkennen und damit splitten. Solch ein Verfahren scheint in der Tat möglich zu sein und wird im weiteren näher erläutert.

Vorher allerdings wird ein Verfahren vorgestellt, welches diese Beziehungen bereits mit einer einfachen Idee grafisch darstellen konnte. Es wird ein Wort genommen und seiner unmittelbaren Nachbarn untersucht. Wenn immer zwei Nachbarn auch miteinander mit großer Signifikanz verbunden sind, werden beide in einen Graphen eingezeichnet, zusammen mit den Verbindungen. Dies ergibt dann für «Stich» den folgenden Graphen:



Es lassen sich hier also mindestens zwei Bedeutungen rein optisch anhand der Worthäufungen erkennen, allerdings war es damit noch nicht möglich, zuverlässig diese Häufungen als Wortlisten zu ermitteln. Ferner kann auch bei der graphischen Darstellung bei den meisten Wörtern kaum etwas erkannt werden, wie etwa bei «Rauch»:



Eine genauere Beschreibung findet sich in F. SCHMIDT'S Diplomarbeit [2].

**Verfahren**

Die weiterführende Idee war nun, immer drei Wörter zu nehmen und die Schnittmenge ihrer Assoziationsmengen zu betrachten. Ist diese Schnittmenge nicht leer, wie sie es in den meisten Fällen sein wird, dann haben die gefundenen Wörter mit allen drei Wörtern etwas zu tun und weil es schwer sein wird, eine Menge von drei Wörtern zu finden, die allesamt noch zweideutig sind, wird auch die gefundene Menge nur noch aus einem Kontext Wörter beinhalten:

Stich Becker Vorhand : Satz Tiebreak Satzball gewonnen Break Match erwandelte spielte Matchball Aufschlag Grundlinie Graf gespielt Return schlug ersten Bälle

Stich verletzte Bauch : stach zog Schulter Messer Oberschenkel

Legt man diese Beobachtung zugrunde, können nun, von einem Wort ausgehend, Paare aus seinen Nachbarn gebildet werden, zusammen ergibt das jeweils Tripel. Entweder das Paar enthielt Wörter aus der einen Bedeutung, dann wird die Schnittmenge nicht leer sein und nur Wörter aus der jeweiligen Bedeutung enthalten. Oder das Paar enthielt Wörter aus beiden Bedeutungen, dann wird die Schnittmenge leer sein.

Wenn nun alle erhaltenen Schnittmengen gruppiert und anschließend zusammengefügt werden, erhält man mit etwas Glück gerade die Mengen von Wörtern, welche die unterschiedlichen Bedeutungskontexte des

Wörter bilden. Mit «Stich», welches zwei solch unterschiedliche Bedeutungen hat, klappt das auch ganz gut (siehe graue Box).

#### Bedeutungsklassen für «Stich»

S1 = { Asse gewinnt verlor Achtelfinale Goran Ivanisevic Pioline Pilic David Prinosil Niki Pilic gewonnen Petr Korda Stefan Edberg Duell Gegner Grand-Slam-Turnier Daviscup Rasen besiegte Haarhuis Wimbledon Sieger Ivanisevic Sampras Turniersieg Jewgeni Kafelnikow Sieger Rosset Krajicek Andrej Tschesnokow French Open spielte siegt Jim Courier Chang Pete Sampras Elmsborn Daviscup-Team Michael Stich Südafrikaner Partie Turniere Agassi Key Biscayne Endspiel Haas Antwerpen Thomas Haas ausgeschieden Carlos Costa auf Niederlage Tennisspieler Guy Forget ... }

S2 = { gestochen schmiert ausspielen Herz- zieht Herz-As gewonnen Karo-Bube Kreuz-Dame Herz- Dame As Pik Pik-Dame Karo-Bube spielte Karo ausgespielte Stiche übernimmt Hinterhand Karo-As stach Siebter Kreuz-Buben Herz-König Herz Alleinspieler zog Karo-König Karo-König Kreuz-As Karo-Dame Herz-Bube wimmelt Gegenspieler Vorhand Pik-Bube }

S3 = { stach zog Schulter Messer Oberschenkel }

Mit anderen Wörtern kann es allerdings durchaus auch nicht so gut aussehen (siehe Bedeutungsklassen für «Rauch»).

#### Bedeutungsklassen für «Rauch»

S1 = { Gebäude Brandherd Mitleidenschaft Stock Explosion Bränden brennt Schaden Qualm rasend Wind Löschwasser Hitze Kamin Treppenhaus Wohnung bläst Anwohner bemerkten Flammen brennenden Feuer Feuerwehr Dämpfe alarmiert bemerkte Brandort brannte ausgebreitet Decke erstickt Fenster giftigen Feuers gemeldet Alarm Zigarette Freie Explosionen Staub erstickten Dach bemerkt Ruß blies plötzlich Feuerwehrleute Brand beschädigt Gasen entdeckt Himmel Asche }

Der Vorteil diese Methode auch bei diesen Wörtern ist allerdings vor allem der, daß auch unpassende Begriffe gefiltert werden und damit dennoch eine Verbesserung der ursprünglichen Kookkurrenzmenge von «Rauch» erreicht wurde.

Damit hat man also mit einer rein statistischen Methode eine bedeutungsorientierte Einteilung der Assoziationsmengen eines Wortes erreicht. Diese Einteilung

ist zwar nach wie vor nach Wortklassen gemischt, doch zumindest im Deutschen lassen sich Wortklassen recht leicht Musterbasiert erkennen: Groß- und Kleinschreibung, Präfixe, Suffixe usw. sind dabei sehr hilfreich.

#### Sachgebiete

Unter Zuhilfenahme dieser „Disambiguierung“ von Wörtern läßt sich auch ein Verfahren konstruieren, welches zu einer Gruppe von Wörtern passende Ergänzungen finden kann. Es sei zum Beispiel ein Sachgebiet «Medizin» mit folgenden Wörtern definiert:

Fettgewebe Gelbsucht Immunsystem Implantation implantieren Infusion injizieren intravenös Katheter Körpergewebe Lymphknoten Melanom

Wenn nun für jedes Wort die verschiedenen Bedeutungsgruppen von Wörtern gefunden werden können und zwischen den Wörtern dann verglichen werden, könnte es geschehen, daß jeweils eine der Gruppen eines Wortes sich einer anderen Gruppe eines anderen Wortes ähnelt. Wenn also von jedem Wort jeweils die zu dem aufgespannten Kontext passende Gruppe ausgewählt wird und alles zusammengemerzt wird, ergeben sich für die oben genannten Wörter folgende neue:

Abwehr Abwehrkräfte Abwehrmoleküle Abwehrreaktion Abwehrsystem Abwehrcellen Aids Aidsvirus aktiviert angreift Antibiotika Antigen Antigene Antikörper Antikörpern Autoimmunkrankheit Bakterien befallen befallenen Behandlung bekämpfen bekämpft bestimmte bilden bildet Blut Blutkörperchen Blutzellen Diabetiker DNA Eindringlinge Eiweiß Eizellen Embryo Empfängers entwickeln Entzündungen Enzyme Erbgut erkannt erkennen erkranken Erkrankung Erkrankungen Ernährung Erreger Erregern Forscher Forschern fremde fremden gebildet Gehirn Gene genetisch gentechnisch geschwächtem geschwächten Gewebe gezielt Haut Hautkrebs hemmen HI-Viren HIV Immunabwehr Immunzellen Impfstoff Impfung Infektion Infektionen ...

Wichtig ist dabei allerdings zu beachten, daß Wortklassen außer Substantiven meist eher ungünstig für wortinhaltsorientierte Verfahren funktionieren, weil

diese wie erwähnt kaum Inhalt tragen, bzw. in vielen Kontexten korrekt benutzt werden können. «angreift» in dem oben genannten Beispiel etwa kann nicht nur in «Medizin» sinnvoll sein, wo es die angegriffene Gesundheit oder das Antibiotikum, welches die Krankheitserreger angreift meint, sondern auch beim «Militär», wo wohl eher ein Manöver gemeint ist, bei «Autos», wo der Rost den Lack angreift, bei Politik, wo der eine Politiker einen anderen verbal angreift. Es ist dann meist auch nicht mehr sinnvoll, von einer echten Ambiguität des Wortes zu sprechen, wie etwa beim «Steuer» des Autos im Gegensatz zur «Steuer», die der Staat erhebt.

Daß derartige Wortarten dennoch bei diesem Verfahren gefunden werden, hat trotz der genannten Gründe seinen Nutzen. Es stimmt zwar schon, daß «angreift» nicht spezifisch für «Medizin» ist, allerdings wird es in diesem Kontext offenbar häufiger genannt, als andere Verben. Diese Information kann dann in einer Anwendung dieses Verfahrens weitergenutzt werden.

Unter Zuhilfenahme der beschriebenen Methode ließen sich nun diverse Verbesserungen von herkömmlichen IR Verfahren einführen, Klassifizierungen von Emails nach Inhalt könnten damit auch ein Stück verbessert werden und vor allem der im GAOS e. V. viel diskutierte Spamfilter könnte damit vernünftig funktionieren, indem einzelne Gruppen von Wörtern vorgegeben werden und dazu eine Wertung von zum Beispiel «erwünscht», «ist egal» und «unerwünscht» gegeben wird. Diese Gruppen würden dann um sämtliche passende Wörter automatisch ergänzt und dann in den Emails gemessen werden.

## Referenzen

- [1] Wortschatz Leipzig:  
<http://wortschatz.uni-leipzig.de>
- [2] F. Schmidt: *Automatische Ermittlung semantischer Zusammenhänge lexikalischer Einheiten und deren graphische Darstellung*, Diplomarbeit, Universität Leipzig, 1999
- [3] G. Ruske: *Automatische Spracherkennung, Methoden der Klassifikation und Merkmalsextraktion*, 2. Auflage, Oldenbourg, 1993
- [4] A. Lehr: *Kollokationen und maschinenlesbare Korpora: Ein operationales Analysemodell zum Auf-*

*bau lexikalischer Netze*, Niemeyer, Tübingen, 1996

## ▼ PHUTBALL '02

JOHANNES WALDMANN: Programmierwettbewerb Philosophen-Fußball

Wieder einmal veranstaltete das Institut für Informatik der Universität Leipzig einen Programmierwettbewerb für ein Strategiespiel. Passend zur Weltmeisterschaft in Korea und Japan geht es um — Fußball! Ich berichte hier über das Spiel und den Wettbewerbsverlauf und schließe allgemeine Bemerkungen zur Rolle von Spielen in der Informatik an.

### Das Spiel Phutball

Der Name des Spiels steht für “Philosophen-Fußball”. Sein Erfinder ist der Gruppentheoretiker, (zelluläre) Automaten-Forscher, Spieltheoretiker (u. v. a. m.) JOHN H. CONWAY. Er beschrieb es in Band 3 von *Winning Ways* [BCG01].

Das Spielfeld ist ein rechteckiges Gitter (meist 15 breit und 19 hoch), auf dem sich genau ein *Ball* (schwarzer Stein) und beliebig viele *Männer* (weiße Steine) befinden.

Das Ziel jedes Spielers ist es, wie im “wirklichen” Leben, den Ball auf oder hinter die gegnerische Torlinie zu befördern.

Dazu sind zwei Arten von Spielzügen erlaubt:

- man setzt einen Mann auf einen beliebigen freien Gitterpunkt, oder
- man läßt den Ball eine lückenlose Folge von Sprüngen ausführen. Dabei führt jeder einzelne Sprung über eine Reihe von Männern, die (vom Ball aus in eine der acht Himmelsrichtungen gesehen) direkt hintereinander stehen. Sofort nach dem Sprung werden die übersprungenen Männer entfernt.

Andere Bewegungen gibt es nicht, insbesondere ist es nicht gestattet, den Ball oder die Männer zu verschieben.

### Ein Tor nach 34 Zügen

Sehen wir uns ein Phutball-Bei-Spiel an. Die Diagramme sind so zu lesen: Der Schiedsrichter legt den Ball



mus gibt, der für einen beliebige Situation entscheiden kann, ob es (sofort) einen Sprung ins Tor gibt. Selbst diese Frage ist  $\mathcal{NP}$ -vollständig! [DDE]

Ich finde Phutball deswegen so interessant (und passend für den Wettbewerb), weil (nach wenigen Eröffnungszügen) meist viele recht unmittelbare Sprünge bis dicht an die Tore drohen. Das ist spannend, und auch für Zuschauer nachvollziehbar, die nur die Regeln, aber keine spezifische Strategie und Taktik kennen.

Damit unterscheidet sich Phutball beispielsweise von Go oder Amazons. In diesen Spielen ist meiner Meinung nach die Strategie bei Eröffnung und Mittelspiel spielentscheidend, was sich aber dem unbedarften Zuschauer oft erst im Endspiel zeigt (wenn schon alles zu spät ist). Das soll nicht heißen, daß diese Spiele weniger interessant sind, sondern nur, daß sich ihr Reiz wohl nicht so unmittelbar erschließt.

Es kann allerdings gut sein, daß es auch für Phutball langfristige erfolversprechende Strategien gibt. Vielleicht trägt der Wettbewerb dazu bei, solche zu finden.

### Der Wettbewerb

Am Ende des Wintersemesters (Januar 2002) entschieden die Hörer meiner Vorlesung *Kombinatorische Spieltheorie* über das Wettbewerbs-Spiel. Die oben genannten Argumente gaben wohl den Ausschlag für Phutball. (Ernsthafte Gegenkandidaten waren Amazons, Konane, Lotus. Vielleicht schlägt deren Stunde beim nächsten mal.)

Im Februar implementierten MIRKO RAHN und ich den Phutball-Schiedsrichter für den (bereits im Vorjahr beim Connections-Wettbewerb benutzen) generischen Spiel-Server, einen einfachen Fußballspieler *Flankengott* sowie ein passendes Java-Applet.

Damit konnten ab März Interessenten zunächst mit dem Applet gegen Flankengott spielen und sich so mit den Spielregeln vertraut machen; und andererseits die Programmierung des Socket-Protokolls ausprobieren.

Zu Semesterbeginn (April) eröffnete ich "offiziell" den Wettbewerb bei einer gemeinsamen Veranstaltung mit dem GAOS e. V.

Während des Studieninformationstages (Anfang Mai) stellten wir das Spiel einigen Interessenten vor, und spielten nebenbei einige Testrunden (der damals vorliegenden, vorläufigen Programmversionen).

Die Endrunde fand, wie im vorigen Jahr, während der Universitätsmesse *Campus 2002* (Anfang Juni)

statt.

Der Wettkampf wurde unterstützt von Sponsoren; besonders bedanke ich mich bei Herrn LEHMANN von VUB/Printmedia.de.

Einige der studentischen Programme werden als Prüfungs(vor)leistung anerkannt, beispielsweise zur Funktionalen Programmierung (Herr PROF. GERBER).

### Der Flankengott

Der Sparring-Partner für die studentischen Programme ist der Flankengott. Dieser Phutball-Spieler benutzt eine generische Alpha/Beta-Spielbaum-Suche. Nachdem diese einmal implementiert ist, kann man mit wenig Aufwand (durch Angabe von Zuggenerator und Bewertungsfunktion) einen passabel spielenden Gegner erhalten.

Die drei Bestandteile der Bewertungsfunktion sind

- Höhe (Ordinate, relativ zur Mittellinie) des Balls
- durchschnittliche Höhe aller Spieler
- Mittelwert der Höhen des tiefsten und höchsten (sofort erreichbaren) Sprungzieles

Diese werden mit Gewichten versehen (der letzte am schwersten), und das war es schon. Eine Schwäche wird deutlich: Die Abszisse des Balles oder der Spieler wird nie benutzt.

Der Zuggenerator liefert

- alle sofort ausführbaren Sprünge
- alle Plazierungen von Männern *in* und *direkt neben* alle (Zwischen-)Landeplätze des Balles bei allen sofort ausführbaren Sprüngen.

Auch hier wird deutlich: das ist einerseits nicht weit-sichtig genug (es gibt sicher Situationen, da wäre es günstig, einen Mann zu setzen, der erst nach einem oder zwei Sprüngen erreichbar ist), andererseits bereits zuviel (wenn zuviele Sprünge möglich sind, dann dauert die Zuggenerierung zu lange, und der Suchbaum wird zu breit).

Den zweiten Fehler versuche ich durch eine zweistufige Bewertung zu mildern: zunächst werden *alle* beschriebenen Züge generiert, und die erreichte Stellung bewertet. Dann werden nur die fünf besten Züge tatsächlich für die Baumsuche (über vier Halbzüge) benutzt.

Der Flankengott ist also mit Absicht kein allzu starkes Programm: einerseits konnte ich nicht sehr viel Zeit investieren, andererseits sollten Interessenten auch nicht durch ein bereits sehr starkes Programm von der Teilnahme abgeschreckt werden.

### Ergebnisse der Finalrunde

Die Endspiele des Programmierwettbewerbss liefen am 8. Juni während der Universitätsmesse *Campus 2002*.

Die Finalisten spielten "jeder gegen jeden" (hin und rück) auf dem offiziellen 15 × 19-Feld bei einer Minute Bedenkzeit pro Zug (die jedoch nie ausgenutzt wurde). Das ergab schließlich diese Plazierungen für die Programme (Programmierer):

1. DioSok (HEIKO STAMER),
2. Steilpaß (JÖRG ENDRULLIS),
3. Weißwurst (THOMAS RABE),
4. Diogenes (UDO STENZEL).

### Dreihirn-Protokoll

Nach allgemeine Erfahrung ist das Ganze mehr als die Summe seiner Teile. PROF. INGO ALTHÖFER (Jena) schlägt unter dem Slogan *Dreihirn* vor, damit die Leistung von Mensch-Maschinen-Systemen für den Bereich der Spielprogramme im Besonderen (und der Diskreten Optimierung im allgemeinen) zu verbessern. Dabei soll jeweils ein Hirn (der (menschliche) Koordinator) aus den Vorschlägen zweier andere Hirne (Programme) auswählen. (Für Schachprogramme hat Herr Althöfer nachgewiesen [Alt98], daß ein Dreihirn mit einem erfahrenen Koordinator stärker spielt als jedes der Einzelhirne.)

Ich möchte hier darstellen, wie man die Dreihirn-Idee mit wenig Aufwand in das bestehende Serverkonzept einbauen kann. Das Protokoll ist auch dafür völlig generisch, also nicht festgelegt auf ein bestimmtes Spiel oder bestimmte Clients.

Ein übliches Spiel-Server-Protokoll besteht im wesentlichen daraus, daß der Server den Spielern gegenseitig die Züge schickt. Das sollte man zunächst trennen, in die Nachricht: *Der\_andere\_zieht ...*, und die Frage: *Wo\_ziehst\_du?*.

Dann gestattet man, daß der Server nur die Frage stellt *Wo\_willst\_du?*, auf die der Client antwor-

ten soll *Ich\_will ...*, aber den Zug noch *nicht* ausführt.

Außerdem müssen die Clients den Server-Befehl *Du\_ziehst ...* implementieren (für den Fall, daß jemand anderes (der Koordinator) das Programm überstimmt).

Dieses *Du\_ziehst ...* benutzen wir auch zum Betrachten von Spielen: wer einfach nur zugucken will, der verbindet sich mit dem Server und bekommt dann ein bereits laufendes Spiel als Folge von *Der\_andere\_zieht .../Du\_ziehst ...* übertragen (wird also nie nach einem eigenen Zug gefragt). Dieses passive Zuschauen ist derzeit bereits implementiert und wird zum Spiel-Betrachten benutzt.

Damit kann man die Dreihirn-Idee so umsetzen: Der Koordinator und der Gegner starten wie üblich ein Spiel. Die zwei "Hirne" sind Zuschauer (wie im vorigen Abschnitt beschrieben). Wenn der Koordinator dran ist, darf er auf die Server-Frage *Wo\_ziehst\_du* mit der Gegenfrage *Frage\_Zuschauer* (an den Server) antworten. Daraufhin schickt der Server an die Zuschauer die Frage *Wo\_willst\_du*, und nachdem der Server beide Antworten (*Ich\_will ...*) hat, schickt er das dem Koordinator (*Zuschauer\_sagen [...], [...]*). dann antwortet der Koordinator mit *Ich\_ziehe ....*

Zusammengefaßt (1 = Koordinator; 2, 3: Hirne)

```
1<< Wo_ziehst_du
1>> Frage_Zuschauer
2<< Wo_willst_du
2>> Ich_will z2
3<< Wo_willst_du
3>> Ich_will z3
1<< Zuschauer_sagen [z2,z3]
1>> Ich_ziehe z0
2<< Du_ziehst z0
3<< Du_ziehst z0
```

### Spiele in der Informatik

Neben der ganz allgemein richtigen Beobachtung, daß Leistung und Leistungsvergleiche Spaß machen; sowie die Studenten dabei erworbenes Wissen (über Algorithmen-Entwurf und -Implementierung) in der Anwendung erproben, kann man fragen, warum ich mich als "theoretischer" Informatiker überhaupt mit (diesem und anderen) Spielen beschäftige.

Es geht besonders um endliche, deterministische Zweipersonen-Nullsummenspiele mit vollständiger Information. (Phutball ist nicht endlich — es gibt Spielabläufe mit Stellungswiederholungen — wobei nicht klar ist, ob diese Spielweise optimal ist. Das Spiel Connections vom vorigen Jahr war jedoch endlich.)

Man kann, für ein gegebenes Spiel, die Menge aller Positionen untersuchen, für die Spieler A eine Gewinnstrategie besitzt. (Im Idealfall möchte man entscheiden, ob die Startposition dazugehört.) Diese Menge ist eine formale Sprache (nicht, wie im Grundstudium, von Wörtern, sondern in diesem Fall von markierten Graphen), und es stellt sich die Frage nach ihrer Komplexität.

Eine Motivation für die Betrachtung von Spielen ist nun die Beobachtung, daß sich viele "praktisch wichtige" Komplexitätsklassen (d. h. solche, in denen praktisch bedeutsame Probleme liegen) auch durch bestimmte Spiele charakterisieren lassen. Methoden zur Untersuchung von Spielen liefern damit Methoden für (augenscheinlich) ganz andersartige Aufgaben.

Eine Anwendung ist die Programm-Analyse, für die ich seit einiger Zeit einen Zugang über Term-Ersetzungs-Spiele untersuche: Spielpositionen sind dabei markierte Bäume (Terme), und Spielregeln sind Baum-Umformungs- (Term-Ersetzungs-)Regeln. Term-Ersetzungs-Systeme sind aber andererseits ein Modell für die Semantik (vorwiegend, aber nicht nur) funktionaler Programme. Die Idee ist nun, Gewinnmengen von Ersetzungsspielen zum Beispiel durch endliche Automaten oder Set Constraints zu beschreiben, dabei diese Beschreibungsverfahren zu erweitern, und diese Erweiterungen schließlich bei der Programmanalyse anzuwenden.

### Auf ein Neues

Nach dem Leipziger Phutball-Finale plane ich, den oder die Sieger in größerem Rahmen zu präsentieren. Warum bescheiden sein — die Teilnehmer haben viel Arbeit investiert, und können damit ruhig ein bißchen angeben, und sich mit französischen oder US-amerikanischen Phutballern und -Programmierern vergleichen.

Ich denke auch an einen erneuten Programmierwettbewerb im Sommersemester 2003. Dessen konkrete Durchführung möchte ich jedoch gern an (z. B.) eine Arbeitsgruppe der Informatik-Fachschaft abgeben. Ich rufe deswegen schon hier die Teilnehmer dieses und des vorigen Jahres auf, sich dabei zu engagie-

ren.

Es ist klar, daß es wieder um ein Spiel gehen sollte. Welches? Denkbar ist einerseits eine stärkere Bindung an Themen aus dem Studium, z. B. Graphen-Probleme wie im `challenger/autotool` [MLK<sup>+</sup>02], oder Term-Ersetzungs-Spiele [Wal02b], oder – kann sicher sehr interessant werden – ein Mehr-Personen-Spiel.<sup>19</sup>

### Referenzen

- [Alt98] Ingo Althöfer. *13 Jahre 3-Hirn*. 1998. <http://www.minet.uni-jena.de/www/fakultaet/iam/personen/althofer.html>.
- [BCG01] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for your Mathematical Plays*. A K Peters, 2001. <http://www.akpeters.com/book.asp?bID=136>.
- [DDE] Erik D. Demaine, Martin L. Demaine, and David Eppstein. Phutball endgames are hard. In Richard J. Nowakowski, editor, *More Games of No Chance*, MSRI Publications. Cambridge Univ. Press. To appear.
- [GN] J. P. Grossman and R. J. Nowakowski. One-dimensional Phutball. In Richard J. Nowakowski, editor, *More Games of No Chance*, MSRI Publications. Cambridge Univ. Press. To appear.
- [MLK<sup>+</sup>02] Georg Martius, Manja Lindemeyer, Alexander Kiel, Jörg Endrullis, and Markus Kreuz. `autotool/challenger` web pages. <http://theopc.informatik.uni-leipzig.de/~challenger/challenger/>, 2002.
- [Wal02a] Johannes Waldmann. Programmierwettbewerb Phutball. <http://www.informatik.uni-leipzig.de/~joe/wettbewerb/phutball/>, 2002.
- [Wal02b] Johannes Waldmann. Rewrite games online. <http://theopc.informatik.uni-leipzig.de/~joe/trs/>, 2002.

<sup>19</sup>Anm. des Editors: Wie wäre es z. B. mit KPEMJIB?

- [Wil00] John Williams. Philosopher's Phutball. *Abstract Games Magazine*, (3):10–14, 2000. <http://www.abstractgamesmagazine.com>.



## Autorenverzeichnis

V. i. S. d. P. sind folgende Personen/Autoren:

- ▼ **Frank-Michael Schleif**  
 schleif@informatik.uni-leipzig.de  
 L<sup>A</sup>T<sub>E</sub>X im studentischen Alltag 3  
 OCR mit statistischen Momenten 15
- ▼ **Heiko Stamer**  
 stamer@informatik.uni-leipzig.de  
 L<sup>A</sup>T<sub>E</sub>X im studentischen Alltag 3
- ▼ **Johannes Waldmann**  
 joe@informatik.uni-leipzig.de  
 PHUTBALL '02 21
- ▼ **Stephan Bordag**  
 mai97eme@studserv.uni-leipzig.de  
 Semantik im Wortschatz 18
- ▼ **Sven Türpe**  
 sven@gaos.org  
 Killerapplikation 10

## Impressum

GAÖTENBLATT

ISSN: 1619-0114

### ▼ Herausgeber

GAOS e.V., c/o Marcus Bautze  
 Rosenowstraße 71, 04357 Leipzig

*eMail*: gaos@gaos.org  
*WWW*: http://gaos.org

### ▼ T<sub>E</sub>Xnischer Editor

Heiko Stamer  
*eMail*: stamer@gaos.org

### ▼ Copyright & Lizenz

Artikel dieser Zeitschrift unterliegen der GNU FREE DOCUMENTATION LICENSE und sind nach deren Maßgabe zu behandeln. Für weitere Informationen sowie den genauen Text dieser Lizenz sei auf <http://www.fsf.org/copyleft/fdl.html> verwiesen. Die Quellen der Dokumentklasse gaosjournal.cls unterliegen der GPL und können gemäß der darin enthaltenen Bestimmungen für eigene Zeitschriften angepasst werden. Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen.



This work was supported by  
 FREEBSD, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, METAFONT, METAPOST  
 BSD gives you the power of an Open Source OS without the hype.