

# PEARL – Graphikpaket zur 3-D-Darstellung

## von konkaven und konvexen Polyedern

von Dipl. Ing. Josef Jürka und Dipl. Ing. Uwe Ringstmeyer, Bielefeld

### Zusammenfassung

Im Rahmen einer Diplomarbeit wurde an der Fachhochschule Bielefeld ein 3-D-Graphikpaket mit Eingabe/Ausgabe, Koordinatentransformation und drei aufeinander aufbauenden Visibilitätsalgorithmen in "PEARL" erstellt. Es gestattet die Darstellung von konkaven und konvexen Polyedern mit verdeckten Kanten.

Der Schwerpunkt der Arbeit lag bei den Visibilitätsalgorithmen und der Ausgabe.

Um die Laufzeit des Programmpaketes zu verkürzen, wurde bei der Ausgabe und dem 3. Visibilitätsalgorithmus auf einen Umlaufpuffer gearbeitet.

### Schlüsselwörter:

Aufeinanderaufbauende Visibilitätsalgorithmen, konkave u. konvexe Polyeder, PEARL, Umlaufpufferausgabe.

## 1. Einleitung

In vielen Bereichen der Technik in denen sich die elektronische Datenverarbeitung schon seit längerer Zeit durchgesetzt hat besteht oft der Wunsch die erhaltenen Ergebnisse auch graphisch darzustellen. Andere Bereiche sind nur durch umfassende Graphikmöglichkeiten für die EDV zugänglich geworden.

Im Bereich der kommerziellen DV reicht oft eine Darstellungsmöglichkeit auf zweidimensionaler Ebene aus, etwa für die Darstellung von Balken- oder Kuchendiagrammen. Will man den Informationsgehalt von graphischen Darstellungen steigern, kommt man früher oder später zur dreidimensionalen Darstellung.

Um eine gute Beurteilung der Ergebnisse durch den Menschen zu gewährleisten, sollte die Darstellung dem gewohnten Bild entsprechen; dazu sind Algorithmen notwendig, die es erlauben verdeckte Kanten im Bild auszublenden (Visibilitätsalgorithmen). In vielen Bereichen, wo naturgetreue Darstellungen unbedingt notwendig sind, z.B. beim Karosseriedesign oder neuerdings bei der Produktion von Trick- und Effektfilmen, kommt man um Visibilitätsalgorithmen nicht herum.

## 2. Das Programmpaket

### 2.1 Allgemeines / Funktionen

Um ein voll lauffähiges Programm zur 3-D-Darstellung zu erhalten sind eine Reihe von Programmteilen erforderlich, die den Visibilitätsalgorithmen zuarbeiten, bzw. deren Ergebnisse weiterverwenden.

Es ist eine Eingabemöglichkeit vorzusehen, mit deren Hilfe die notwendigen Datenstrukturen erstellt werden können. Programmteile zur Ausgabe haben dafür zu sorgen, daß die Ergebnisse in gewünschter Form – möglichst auf mehreren Geräten – darstellbar sind. Um eine Manipulation der Körper zu ermöglichen (Rotation, Translation, Dehnung/Stauchung) werden Algorithmen zur Koordinatentransformation benötigt. Die Koordination dieser Funktionen übernimmt eine zentrale Ablaufsteuerung.

Nachdem mit Hilfe der EingabeprozEDUREN die gewünschte geometrische Struktur in Form von verketteten Linien- und Flächenbeschreibungen vorliegt, kann mit Hilfe der Koordinatentransformation der Körper manipuliert werden.

Im Anschluß an die nun folgende Perspektivtransformation kann die Darstellung zur Kontrolle der Manipulationen ohne die Berücksichtigung der verdeckten Kanten auf ein Gerät ausgegeben werden, dazu wird ein Unterprogrammpaket namens GRIBS \*) benutzt, das eine Untermenge zum GKS \*\*) darstellt.

Soll der Gegenstand unter Berücksichtigung der von ihm selbst, oder von anderen Körpern verdeckten Kanten dargestellt werden, schließt sich an die Perspektivtransformation die Bearbeitung durch die zwei/drei Visibilitätsalgorithmen an.

Alle drei Visibilitätsalgorithmen arbeiten nacheinander mit der selben Datenstruktur, in der sie vermerken ob eine Fläche ausreichend bestimmt ist. Somit wird die noch zu bearbeitende Datenmenge zum 3. Algorithmus hin immer geringer.

## 2.1 1. Flächentest

**Aufgabe:** Unterscheidung in unsichtbare und potentiell sichtbare Flächen.

Hierzu wird auf jeder Fläche des zu betrachtenden Körpers die äußere Flächennormale errichtet. Berechnet man nun den Winkel den diese Flächennormale mit dem Vektor der von der Fläche zum Auge des Betrachters gerichtet ist, bildet, so ist der Wert ein Kriterium für die potentielle Sichtbarkeit bzw. Unsichtbarkeit der zugehörigen Fläche. Liegt der Winkel im Bereich zwischen 90 und 270 Grad, so ist die Fläche unsichtbar. Im anderen Fall ist sie potentiell sichtbar.

Die Aussage ist auf unsichtbar/potentiell sichtbar beschränkt, da konkave und konvexe Gebilde zulässig sind, und somit auch Flächen verdeckt werden können, deren Flächennormalen mit dem Sichtvektor einen Winkel zwischen 270 und 90 Grad bilden. Ferner ist es denkbar, daß ein Körper von einem anderen Körper gänzlich verdeckt wird.

Da die Flächennormalen bei der Manipulation der Körper der Koordinatentransformation mit unterzogen werden können, ist deren Bestimmung nur ein einziges Mal notwendig. Deshalb, und weil als Kriterium für die Unsichtbarkeit einer Fläche nur das Vorzeichen des Skalarproduktes aus der Flächennormale und dem Sichtvektor ausreicht, ist dieser Flächentest sehr schnell.

\*) GRaphisches Interaktives Basis Ssystem

\*\*) Graphisches Kern System

## 2.3. 2. Flächentest

**Aufgabe:** Unterscheidung der bisher potentiell sichtbaren Flächen in

- vollkommen unsichtbare Flächen
- teilweise unsichtbare Flächen
- vollkommen sichtbare Flächen

Der Algorithmus untersucht Flächen in der Projektion und testet sie auf Sichtbarkeit. Dazu wählt er sich Testpunkte in der perspektivischen Projektion einer Fläche aus und projiziert sie in das dreidimensionale Koordinatensystem zurück. Dabei testet er, welche Fläche an diesem Punkt sichtbar ist, dieses erfolgt entlang einer Testgeraden, die horizontal durch die Projektion gelegt wird.

Eine Fläche ist unsichtbar wenn bei den Tests alle zu ihr gehörenden Testpunkte hinter den Testpunkten der anderen Fläche/Flächen liegen. Sichtbar ist eine Fläche dann, wenn alle ihre Testpunkte vor den Testpunkten der anderen Fläche/Flächen liegen. Ist für eine Fläche beides nicht der Fall, so wird sie als teilweise sichtbar gekennzeichnet und im folgenden Abschnitt durch den Linientest bearbeitet.

Obwohl nur wenige, ausgesuchte Testgeraden verwendet werden, es wird kein dichtes Raster über die Projektion gelegt, ist dieser Test wegen der umfangreichen Berechnungsvorgänge, die dieses Verfahren verlangt, relativ langsam.

Wie die später angegebenen Zeitmessungen erkennen lassen, ist es sogar günstiger diesen Test zu überspringen, da der anschließend beschriebene Linientest auf die hier gewonnenen Informationen nicht angewiesen ist.

Die längere Laufzeit des Linientestes wird durch das Überspringen des zweiten Flächentestes voll kompensiert.

## 2.4. Linientest

**Aufgabe:** Bestimmung der sichtbaren/unsichtbaren Teile nicht unsichtbarer Flächen.

Bearbeitet werden die Kanten aller nicht als unsichtbar erkannten Flächen. Hierdurch ist es möglich den 2. Flächentest zu umgehen.



Dazu wird eine Kante der zu bearbeitenden Fläche bestimmt und untersucht, ob diese als Bestandteil einer anderen Fläche schon gezeichnet wurde. Wenn das nicht der Fall war, wird ein Endpunkt dieser Kante daraufhin untersucht ob er sichtbar ist.

Die Natur (sichtbar/unsichtbar) dieser Kante kann sich bis zum nächsten Schnittpunkt mit einer anderen Kante (In der perspektivischen Darstellung) nicht ändern. Am Schnittpunkt mit einer anderen Kante wird nun entschieden ob das folgende Linienstück auch sichtbar ist, oder ob es von der Fläche der Schnittpunkte verdeckt wird. Beim nächsten Schnittpunkt mit einer Kante einer Fläche erfolgt nun die gleiche Betrachtung, so daß man für die betrachtete Linie eine Anzahl von sichtbaren und unsichtbaren Linienfragmenten erhält. Für die anderen Kanten der Fläche wird analog verfahren.

Die Menge der dabei anfallenden sichtbaren und unsichtbaren Linienstücke ist vorher nicht bekannt, somit muß, da keine Möglichkeit zur dynamischen Speicherplatzanforderung besteht, das entstandene Linienfragment sofort ausgegeben werden.

Sollen die unsichtbaren Teilstücke als Strichlinien ausgegeben werden ist bei Verwendung einer einzigen Task durch den mehrmaligen Aufruf eines Zeichenunterprogrammes (GPOLYL \*) in GRIBS) und der Ausgabe mit langen Wartezeiten zu rechnen.

2.5. Umlaufpufferfunktionen

Aufgabe: Verkürzung der Laufzeit bei der Ausgabe durch Parallelbetrieb.

Das an Schluß von Punkt 2.4. geschilderte Problem der Wartezeiten läßt sich verringern, wenn man die entstehenden Linienfragmente mit Angabe ihrer Art (Volllinie/Strichlinie) in einen Umlaufpuffer schreibt, und die Ausgabe einer anderen Task überläßt.

3. Laufzeitmessungen

3.1 Die Meßwerte

Um die Laufzeitersparnis durch den Umlaufpuffer und den Anteil

\*) Zeichnen eines Polygonzuges

der drei Visibilitätsalgorithmen an der Gesamtlaufzeit des Programmpaketes zu ermitteln, wurden sechs verschiedene Programme erstellt. Zur Ermittlung der Laufzeit wurde der Zeitpunkt vor Eintritt in die Visibilitätsalgorithmen und der Zeitpunkt des Abschlusses des Zeichenvorganges festgehalten. Die Differenzzeit gibt die Rechenzeit an. Für die einzelnen Kombinationen der drei Algorithmen mit und ohne Umlaufpuffer ergaben sich bei der Ausgabe auf ein Vektorsichtgerät die in Tabelle 1 und in Tabelle 2 angegebenen Zeiten. Die Abkürzungen über den Spalten der Tabellen haben folgende Bedeutung :

- 1.FT -- > 1. Flächentest
- 2.FT -- > 2. Flächentest
- LT -- > Linientest

| Bild      | Mit Umlaufpuffer |         |         | Ohne Umlaufpuffer |         |         |
|-----------|------------------|---------|---------|-------------------|---------|---------|
|           | 1.FT/2.FT/LT     | 1.FT/LT | LT      | 1.FT/2.FT/LT      | 1.FT/LT | LT      |
| Bild 1a   | 3.17             | 2.59    | 2.32    | 3.02              | 2.60    | 2.33    |
| Bild 2a   | 6.79             | 5.00    | 6.51    | 6.70              | 5.05    | 6.44    |
| Bild 3a   | 6.45             | 5.34    | 5.82    | 6.50              | 5.28    | 5.81    |
| Bild 4a   | 11.06            | 9.10    | 12.45   | 11.00             | 9.00    | 12.55   |
| Bild 5a   | 26.18            | 19.36   | 30.14   | 26.21             | 19.97   | 30.16   |
| Bild 6a   | 1:39.98          | 1:11.13 | 2:23.12 | 1:39.97           | 1:11.18 | 2:23.12 |
| Bild 7.1a | 8.68             | 7.12    | 9.16    | 8.74              | 7.12    | 9.15    |
| Bild 7.2a | 10.54            | 8.47    | 9.46    | 10.62             | 8.54    | 9.55    |
| Bild 8a   | 19.71            | 14.24   | 19.35   | 19.89             | 14.56   | 19.39   |

Tabelle 1: Laufzeiten [s] ohne Ausgabe verdeckter Kanten

| Bild     | Mit Umlaufpuffer |         |         | Ohne Umlaufpuffer |         |         |
|----------|------------------|---------|---------|-------------------|---------|---------|
|          | 1.FT/2.FT/LT     | 1.FT/LT | LT      | 1.FT/2.FT/LT      | 1.FT/LT | LT      |
| Bild 1   | 9.06             | 8.88    | 8.15    | 9.05              | 8.74    | 8.24    |
| Bild 2   | 20.71            | 19.15   | 19.37   | 20.99             | 19.14   | 19.49   |
| Bild 3   | 16.58            | 15.49   | 14.44   | 16.61             | 15.63   | 15.36   |
| Bild 4   | 31.95            | 29.96   | 27.00   | 32.47             | 30.27   | 31.34   |
| Bild 5   | 44.25            | 35.94   | 37.98   | 44.86             | 38.17   | 42.58   |
| Bild 6   | 2:12.47          | 1:42.41 | 2:39.59 | 2:16.84           | 1:46.93 | 2:48.77 |
| Bild 7.1 | 43.37            | 41.29   | 39.98   | 43.45             | 40.97   | 41.00   |
| Bild 7.2 | 30.77            | 28.63   | 25.30   | 31.16             | 29.02   | 28.61   |
| Bild 8   | 55.48            | 48.92   | 44.83   | 55.77             | 49.51   | 49.50   |

Tabelle 2: Laufzeiten [s] mit der Ausgabe verdeckter Kanten

### 3.2 Laufzeitunterschiede

#### 3.2.1 Einfluß der verschiedenen Algorithmen

Da der Umlaufpuffer nur bei der Ausgabe der verdeckten Kanten als Strichlinien zeitlich zum Tragen kommt (siehe Tabelle 1 und Tabelle 2), kann für diese Betrachtungen die Messreihe aus Tabelle 1 herangezogen werden. Nachfolgende Aufstellung stellt die Laufzeitverkürzungen dar, die sich beim Überspringen des zweiten Flächentestes ergeben.

| Bild      | $\Delta t[s]$ | %    |
|-----------|---------------|------|
| Bild 1a   | 0.6           | 18.3 |
| Bild 2a   | 1.8           | 26.4 |
| Bild 3a   | 1.1           | 17.2 |
| Bild 4a   | 2.0           | 17.7 |
| Bild 5a   | 6.8           | 26.1 |
| Bild 6a   | 28.9          | 28.9 |
| Bild 7.1a | 1.6           | 18.0 |
| Bild 7.2a | 2.1           | 19.6 |
| Bild 8a   | 5.5           | 27.8 |

Tabelle 3: Verkürzung der Laufzeit durch Überspringen des 2. Flächentestes

Der Einfluß des 1. Flächentestes auf die so erhaltene Programmversion stellt Tabelle 4 dar. Ein negativer Wert ist als Laufzeitverlängerung zu interpretieren.

| Bild      | $\Delta t[s]$ | %     |
|-----------|---------------|-------|
| Bild 1a   | -0.3          | -11.6 |
| Bild 2a   | 1.5           | 23.2  |
| Bild 3a   | 0.5           | 8.2   |
| Bild 4a   | 3.4           | 27.4  |
| Bild 5a   | 10.8          | 35.8  |
| Bild 6a   | 72.0          | 50.3  |
| Bild 7.1a | 2.0           | 22.3  |
| Bild 7.2a | 1.0           | 10.5  |
| Bild 8    | 5.1           | 26.4  |

Tabelle 4: Einfluß des Flächentestes auf ein Programmpaket aus Flächentest mit der Flächennormale und Linientest

#### 3.2.2 Einfluß des Umlaufpuffers

In Tabelle 5 sind die Laufzeitverkürzungen dargestellt, die durch die Verwendung eines Umlaufpuffers bei der Darstellung verdeckter Kanten als Strichlinien erreicht werden konnten. Der Umlaufpuffer ermöglicht es Rechen- und Ausgabevorgänge quasiparallel stattfinden zu lassen. Die Zeitangaben beziehen sich auf den Linientest.

| Bild     | $\Delta t[s]$ | %    |
|----------|---------------|------|
| Bild 1   | 0.1           | 1.1  |
| Bild 2   | 0.1           | 0.6  |
| Bild 3   | 0.9           | 6.0  |
| Bild 4   | 4.3           | 13.8 |
| Bild 5   | 4.6           | 10.9 |
| Bild 6   | 9.2           | 5.4  |
| Bild 7.1 | 1.0           | 2.4  |
| Bild 7.2 | 3.3           | 11.6 |
| Bild 8   | 4.7           | 9.4  |

Tabelle 5: Laufzeitverkürzungen durch Verwendung eines Umlaufpuffers

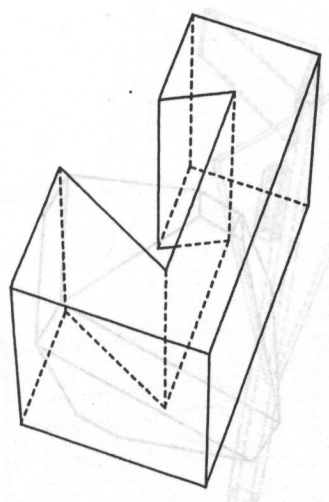


Bild 2

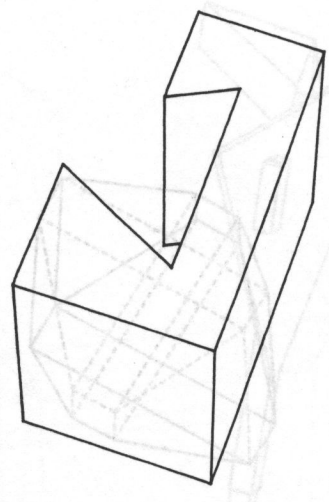


Bild 2a

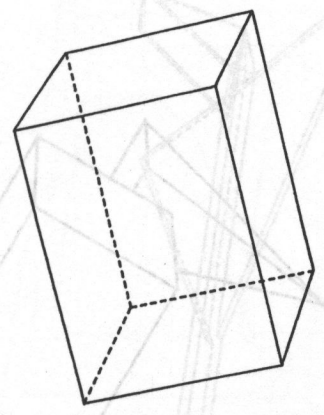


Bild 1

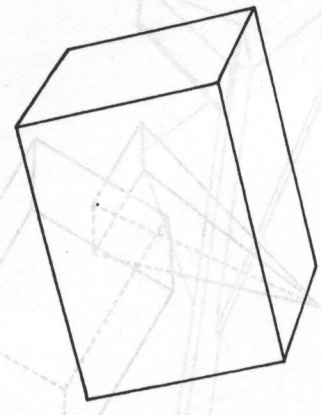


Bild 1a



## 3.2 Laufzeitunterschiede

### 3.2.1 Einfluß der verschiedenen Algorithmen

Da der Unterpunkt nur bei der Ausgabe der verdeckten Kanten als Strichpunkt zeichnen zum Tragen kommt (siehe Tabelle 1 und Tabelle 2), kann bei diesen Berechnungen die Messreihe aus Tabelle 1 herangezogen werden. Nachfolgende Aufstellung zeigt die Laufzeitverkürzungen, die durch das Überprüfen der zweiten Flächenanteile...

| Bild      | $\Delta t(\%)$ | %    |
|-----------|----------------|------|
| Bild 1a   | 1.5            | 20.2 |
| Bild 2a   | 0.5            | 6.2  |
| Bild 4a   | 3.1            | 27.2 |
| Bild 5a   | 10.8           | 54.3 |
| Bild 6a   | 72.3           | 50.3 |
| Bild 7.1a | 1.8            | 22.3 |
| Bild 7.2a | 2.1            | 18.9 |
| Bild 8a   | 5.5            | 27.8 |

Bild 4

### 2. Flächenanteile

Der Einfluß des 1. Flächenanteils auf die so erhaltene Programmschrittzeit ist in Tabelle 4 dargestellt. Ein negativer Wert ist als Laufzeitverlängerung zu interpretieren.

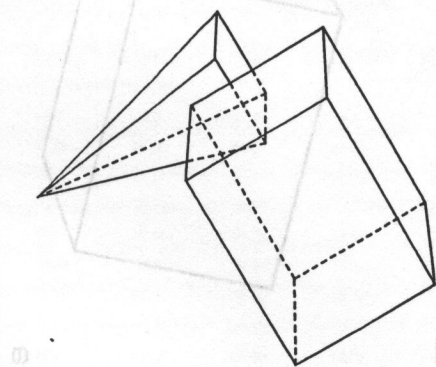


Bild 3

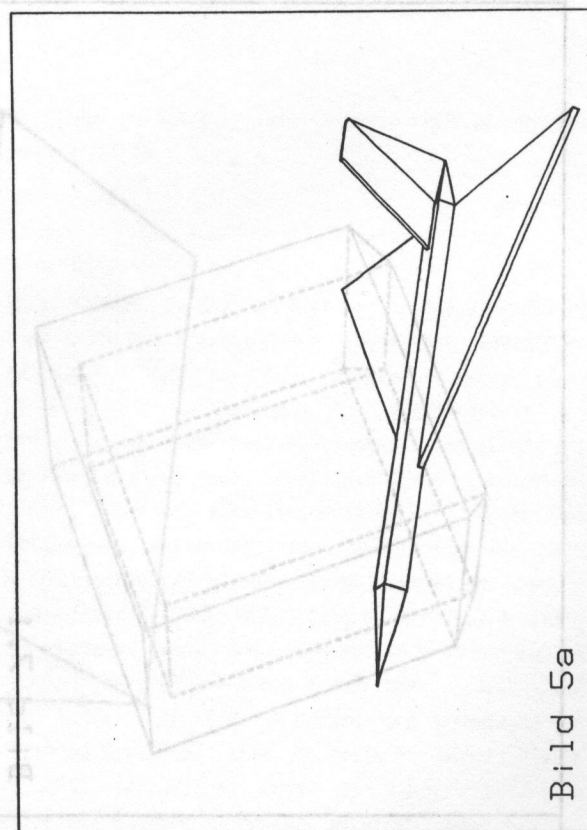
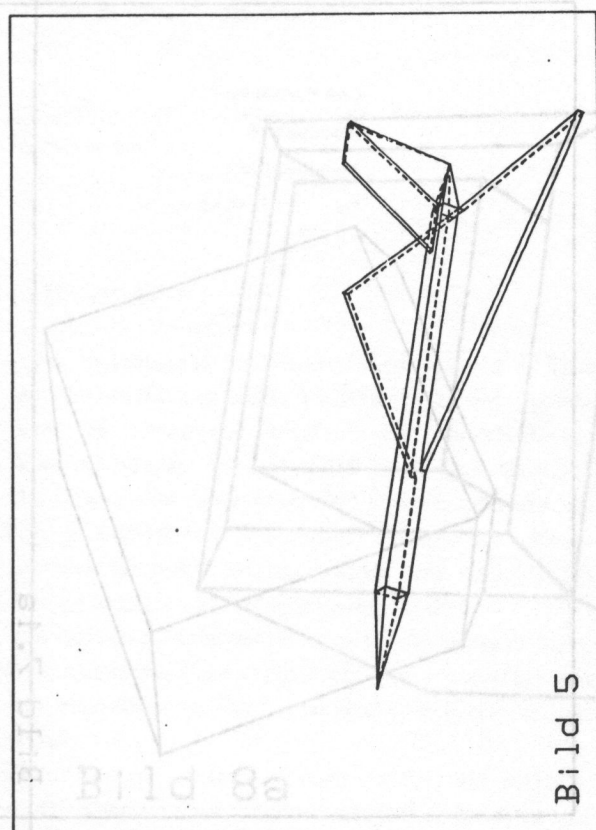
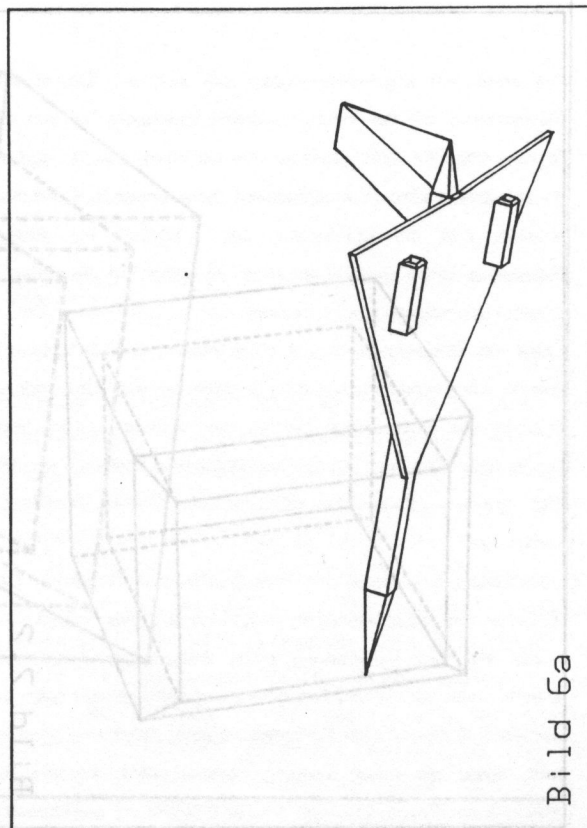
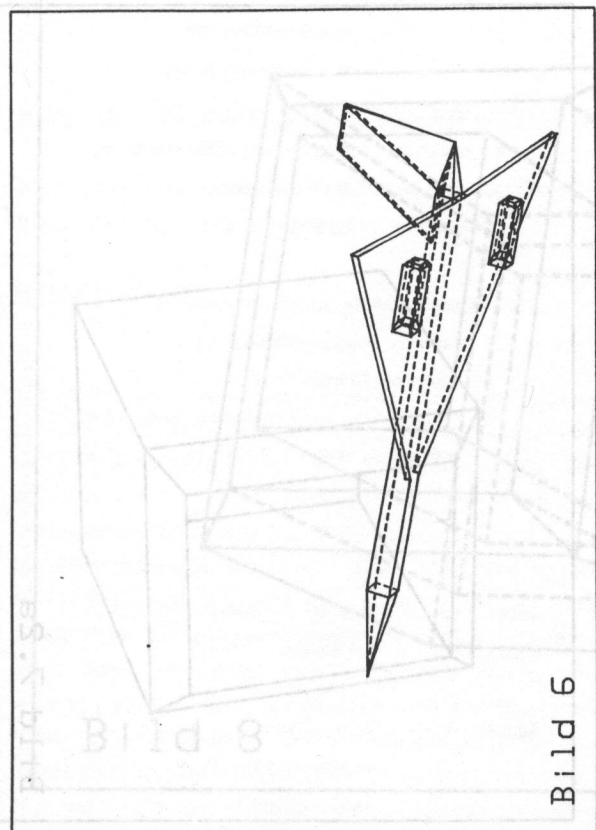
### 3.2.2 Einfluß der verschiedenen Algorithmen

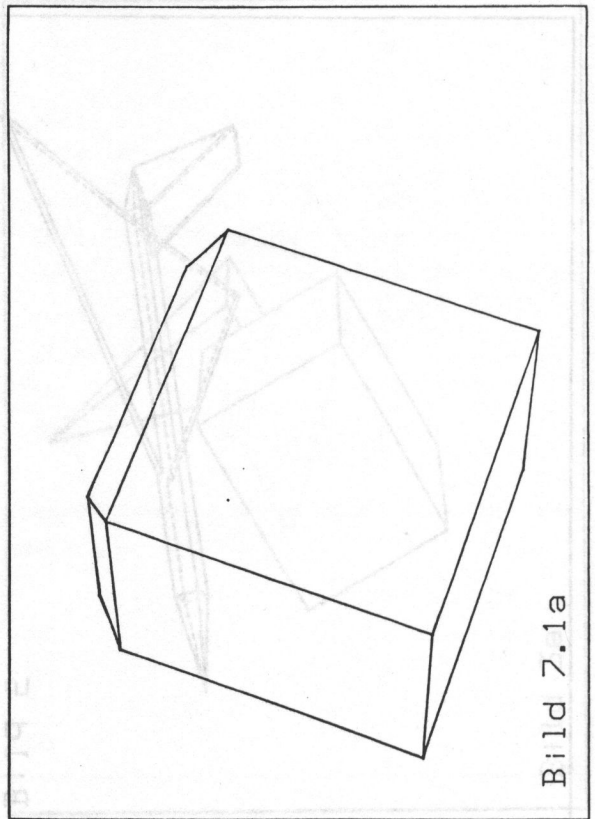
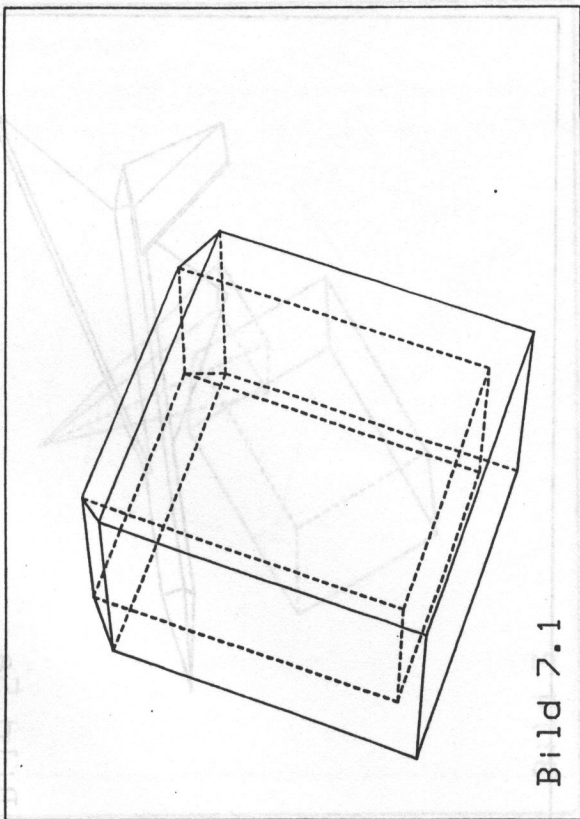
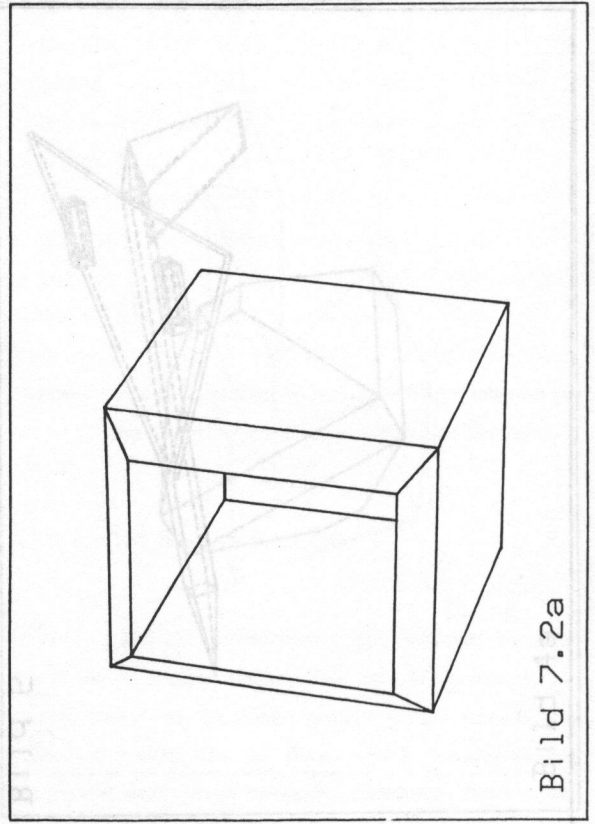
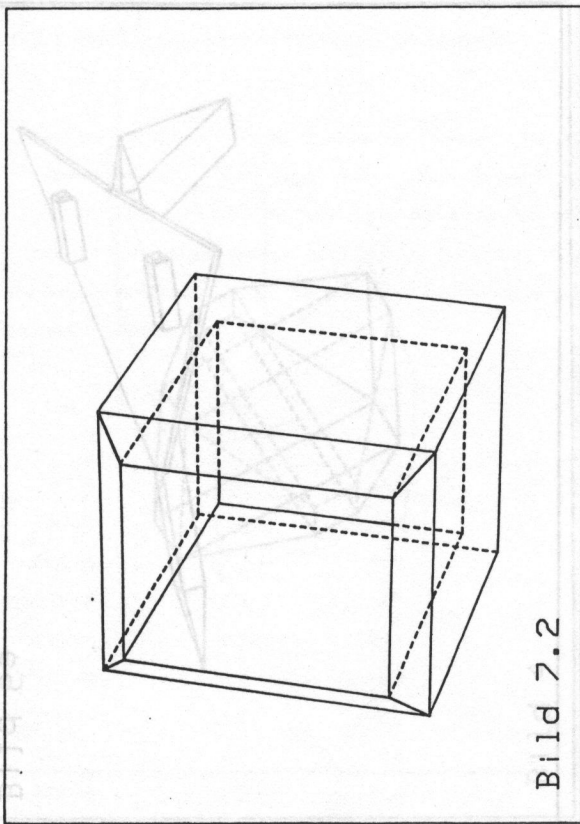
| Bild      | $\Delta t(\%)$ | %    |
|-----------|----------------|------|
| Bild 2a   | 1.5            | 20.2 |
| Bild 3a   | 0.5            | 6.2  |
| Bild 4a   | 3.1            | 27.2 |
| Bild 5a   | 10.8           | 54.3 |
| Bild 6a   | 72.3           | 50.3 |
| Bild 7.1a | 1.8            | 22.3 |
| Bild 7.2a | 2.1            | 18.9 |
| Bild 8a   | 5.5            | 27.8 |

Bild 4a

| Bild     | $\Delta t(\%)$ | %    |
|----------|----------------|------|
| Bild 1   | 0.1            | 1.1  |
| Bild 2   | 0.1            | 0.9  |
| Bild 3   | 6.5            | 3.8  |
| Bild 4   | 1.1            | 13.8 |
| Bild 5   | 10.8           | 5.4  |
| Bild 6   | 1.8            | 1.4  |
| Bild 7.1 | 3.1            | 1.4  |
| Bild 7.2 | 4.2            | 1.4  |

Bild 3a







## 4. Schlußbetrachtungen

Literatur

[1] ENIGMACAD / Computer - Graphics

München/Wien 1975

© Copyright Verlag

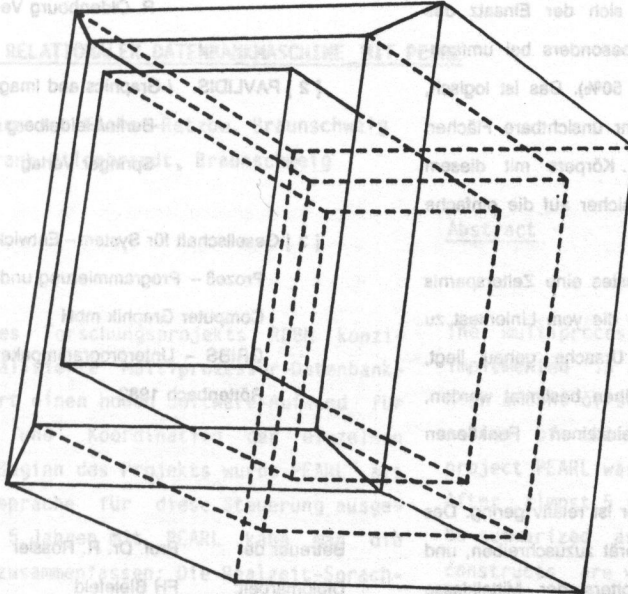


Bild 8

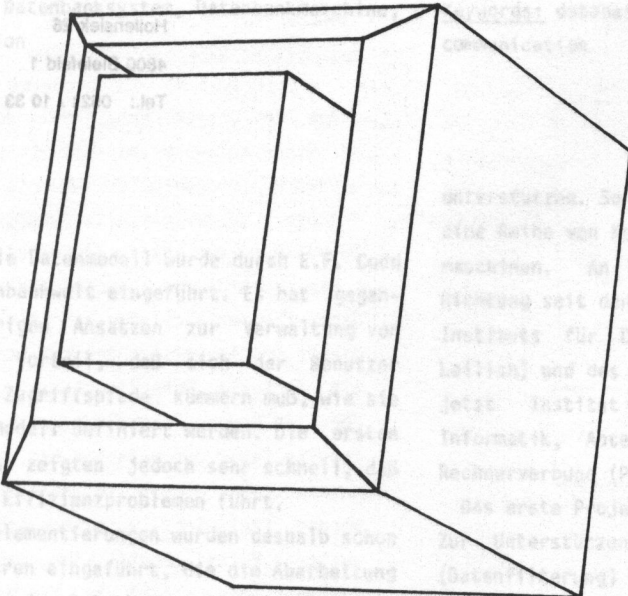


Bild 8a

## 4. Schlußbetrachtungen

Die erhaltenen Ergebnisse zeigen daß sich der Einsatz des Flächentestes mit der Flächennormalen besonders bei umfangreichen Objekten lohnt (Zeitersparnis bis 50%). Das ist logisch, da bei diesen Körpern entsprechend mehr unsichtbare Flächen anfallen. Daß die Bearbeitung des 1. Körpers mit diesem Flächentest länger dauerte als ohne, ist sicher auf die einfache Struktur des Körpers zurückzuführen.

Daß das Überspringen des 2. Flächentestes eine Zeitersparnis bewirkt war unerwartet, reduziert er doch die vom Linientest zu bearbeitende Datenmenge. Worin die Ursache genau liegt, könnte vermutlich nur mit großen Meßreihen bestimmt werden, die genaue Zeitmessungen der einzelnen Funktionen vornehmen.

Die Zeitersparnis durch den Umlaufpuffer ist relativ gering. Das ist sicher dem recht schnellen Ausgabegerät zuzuschreiben, und würde sich bei Verwendung eines Plotters der Mittelklasse ändern. Auch könnte das Programmpaket dahingehend geändert werden, daß nicht bis zur Fertigstellung einer Zeichnung gewartet wird, bevor ein neuer Vorgang gestartet wird, sondern mit Abschluß der Rechenvorgänge mit der Bearbeitung des nächsten Körpers begonnen werden kann.

### Literatur:

[ 1 ] ENCANACAO / Computer - Graphics

München/Wien 1975

R. Oldenbourg Verlag

[ 2 ] PAVLIDIS / Graphics and Image Processing

Berlin/Heidelberg 1981

Springer Verlag

[ 3 ] Gesellschaft für System - Entwicklung

Prozeß - Programmierung und

Computer Graphik mbH

GRIBS - Unterprogrammpaket Version 3.0

Röttenbach 1982

Betreuer der Prof. Dr. R. Rössler

Diplomarbeit: FH Bielefeld

Autoren : Josef Jürka

Wertherstr. 144 / 54

4800 Bielefeld 1

Tel.: priv. 0521 / 10 95 40

dienstl. 0521 / 1 06 23 43

Uwe Ringstmeyer

Hollensiek 26

4800 Bielefeld 1

Tel.: 0521 / 10 33 58