Technischer Bericht Nr. 178
des Heinrich-Hertz-Instituts
- Berlin-Charlottenburg -

Schnelle Kurven- und Flächendarstellung auf grafischen Sichtgeräten

von

Dipl.-Ing. Wolfgang Straßer

Vom Fachbereich 20 (Kybernetik)
der Technischen Universität Berlin
zur Verleihung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

Berlin 1974

D 83

8R
4454
178

Schnelle Kurven- und Flächendarstellung auf grafischen Sichtgeräten

von

Dipl.-Ing. Wolfgang Straßer

Vom Fachbereich 20 (Kybernetik)
der Technischen Universität Berlin
zur Verleihung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

Berlin 1974

D 83

8 R 4454/178

Universitätsbibliothek Techn. Univ. Berlin

Berichter: Prof. Dr.-Ing. W. Giloi
Prof. Dr.-Ing. H. Liebig

Vorsitzener: Prof. Dr.-Ing. P. Rechenberg

Eingereicht am 26. April 1974

Mündliche Prüfung am 5.September 1974

INHALTSVERZEICHNIS

1. Zusammenfassung 1- 1

2. Einleitung 2- 1

3. Mathematische Verfahren zur Flächenbeschreibung 3- 1

3.1 Parameterdarstellung 3- 1

3.2 In Computer Graphics angewandte Flächen 3- 5

3.2.1 Analytisch gegebene Flächen 3- 6

3.2.2 Empirisch gegebene Flächen 3-13

3.3 Interpolation 3-23

3.3.1 Lagrange-Interpolation 3-24

3.3.2 Hermite-Interpolation 3-28

3.3.3 Bikubische Splines 3-33

3.3.4 Gebrochen rationale kubische Funktionen 3-34

3.4 Approximation 3-37

3.4.1 Bézierapproximation 3-37

3.4.2 B-Spline-Approximation 3-50

3.4.2.1 Eigenschaften der B-Splines 3-50

3.4.2.2 B-Spline-Kurven und -Flächen 3-59

3.4.2.3 Interpolation mit B-Splines 3-78

4. Vermittlung eines dreidimensionalen Eindrucks 4- 1

4.1 Perspektive 4- 1

4.2 Rotation 4- 3

4.3 Helligkeitssteuerung 4- 3

4.4 Steuerung der Linienstärke 4- 4

4.5 Schattieren von Flächen 4- 5

5. Hardware 5- 1

5.1 Digitaler Vektorgenerator DVG 5- 1

5.2 Digitaler Kreisgenerator DKG 5- 7

5.3 Digitaler Matrizenmultiplizierer DMM 5-19

5.3.1 (4x4) Matrixspeicher 5-21

5.3.2 16x16 bit Multiplizierer 5-21

5.3.3 Akkumulator und Ergebnisspeicher 5-22

5.4 Digitaler Dividierer 5-23

5.4.1 Division mit zweidimensionaler Schaltkette 5-23

5.4.2 Kombinierter Dividierer-Multiplizierer DMD 5-27

5.4.3 Iteratives Divisionsverfahren 5-31

5.4.4 Modifiziertes iteratives Divisionsverfahren 5-36

5.5 Verbesserter Matrizenmultiplizierer 5-41

5.6 Kurven- und Flächenberechnung 5-42

5.6.1 Differenzenverfahren 5-42

5.6.1.1 Differenzenverfahren ohne DVG 5-47

5.6.1.2 Differenzenverfahren mit DVG 5-48

5.6.2 Matrizenmultiplikation 5-49

5.7 Modulare Display Hardware 5-51

6. Zukünftige Arbeiten 6- 1

7. Literaturverzeichnis 7- 1

Anhang

A Schneller DAU mit Halteverstärker für die Displayablenkung

B Spiralgenerator für variable Linienstärke

Lebenslauf

1. Zusammenfassung

Für die Anwendung beim interaktiven Entwurf von Kurven und Flächen auf grafischen Sichtgeräten werden im ersten Teil der Arbeit bekannte mathematische Verfahren in einer kompakten, einheitlichen Matrizenschreibweise dargestellt. Als neues Verfahren wird die B-Spline-Approximation hinsichtlich ihrer Eigenschaften und Möglichkeiten für den rechnergestützten Entwurf untersucht und an Beispielen erläutert. Die B-Spline-Approximation erweist sich nicht nur als das universellste und am leichtesten zu handhabende mathematische Verfahren, sondern ist auch für eine Hardware-Erzeugung von Kurven und Flächen am besten geeignet. Ein neues Verfahren zur Schattierung von Flächen in Echtzeit wird angegeben und durch Bilder belegt. Im zweiten Teil werden unter Berücksichtigung neuer Bauelemente digitale Komponenten für einen Displayprozessor angegeben: Vektorgenerator, Kreisgenerator, Matrizenmultiplizierer, Dividierer, Kurven- und Flächengenerator.

2. Einleitung

Seit den ersten Anfängen auf dem Gebiet Computer Graphics in den sechziger Jahren bemüht man sich, grafische Sichtgeräte (graphic Displays) im computergestützten Entwurf (CAD) einzusetzen. Besondere Anstrengungen wurden unternommen, um dreidimensionale Gebilde darstellen, interaktiv entwerfen und verändern zu können. Die Grundlage für die notwendigen mathematischen Verfahren wurde von Coons 1964 mit der Veröffentlichung seiner Arbeit "Surfaces for computer-aided design of space figures" /1/ geliefert. Flächen werden danach in Parameterform definiert. Durch Zusammensetzen von Teilflächen - auch patches oder Pflaster genannt - erhält man größere Flächen. Die Darstellung der Flächen erfolgt durch Projektion einiger ihrer Parameterlinien auf den Bildschirm (Bild 2.1).

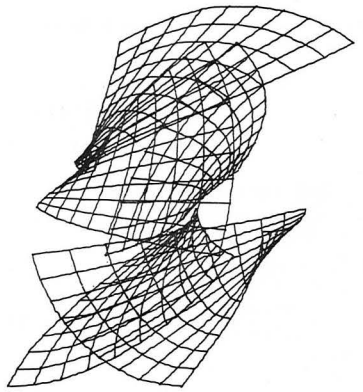


Bild 2.1 Darstellung von 3D Flächen durch Parameterlinien

Die Coons'sche Flächendefinition erfordert vom Anwender erhebliche mathematische Kenntnisse. Zum Beispiel müssen, um ein kontinuierliches Aneinanderfügen von Teilflächen zu ermöglichen, Tangentenvektoren angegeben werden. Normalerweise wird der Anwender aber nur Tangenten, also Steigungen angeben können. Der Einfluß der Länge des Tangentenvektors läßt sich sehr schwer abschätzen.

Ein weiteres Problem ist aus Bild 2.1 erkennbar. Die Darstellungen sind ohne Unterdrückung der verdeckten Kanten unübersichtlich und schwer zu interpretieren. Aus diesem Grund wurden von verschiedenen Autoren /2/ Algorithmen zur Berechnung und Eliminierung der verdeckten Kanten (Hidden Line Algorithmen) entwickelt (Bild 2.2). Diese Verfahren sind alle sehr rechenintensiv und erlauben nicht die Darstellung in Echtzeit, d.h. die Berechnung der verdeckten Kanten mit anschließendem Bildaufbau dauert wesentlich länger als 40 ms (25 Hz Bildwiederholfrequenz).

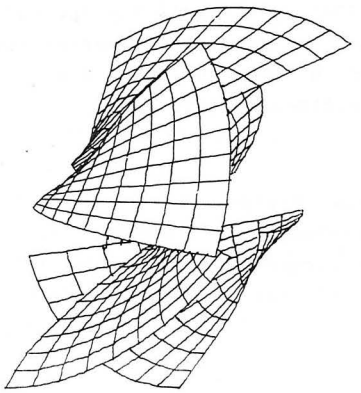


Bild 2.2 Darstellung aus Bild 2.1 ohne verdeckte Kanten /3/

Für viele Anwendungen ist die Darstellung von Körpern durch wenige Parameterlinien unbefriedigend. Eine möglichst realistische Wiedergabe undurchsichtiger Körper verlangt die Schattierung der Oberflächen. Dazu müssen zwei Probleme gelöst werden. Zuerst sind die verdeckten Flächen zu ermitteln und zu eliminieren. Anschließend müssen die Grauwerte für alle Flächenpunkte entsprechend der Auflösung des Sichtgeräts berechnet werden. Die bekannten Verfahren zur Berechnung und Darstellung schattierter Flächen benutzen zur Eliminierung der verdeckten Flächen eine erstmals in /4/ veröffentlichte Methode. Die Flächen werden dabei durch Polygone approximiert. Die Untersuchung auf Verdeckungen wird zeilenweise (line scan method) durchgeführt. Damit reduziert sich die Aufgabe auf ein zweidimensionales Problem (y = Konst.). Trotzdem ist der Rechenaufwand noch sehr erheblich. Um die Darstellung in Bild 2.3 zu erhalten benötigt man auf der PDP-10 einschließlich der Berechnung der Grauwerte entsprechend den lichttechnischen Gesetzen ca. zwei Minuten /5/. Das Gesamtbild ist nur durch zeilenweises Ausgeben auf dem Sichtgerät und fotografisches Zusammensetzen zu erhalten, weil bei einer Speicherzugriffszeit von 1µs nur etwa 20.000 Punkte flackerfrei dargestellt werden können. Dies entspricht bei einer üblichen Auflösung von (10x10) bit nur 1/50 der Sichtgerätekapazität.

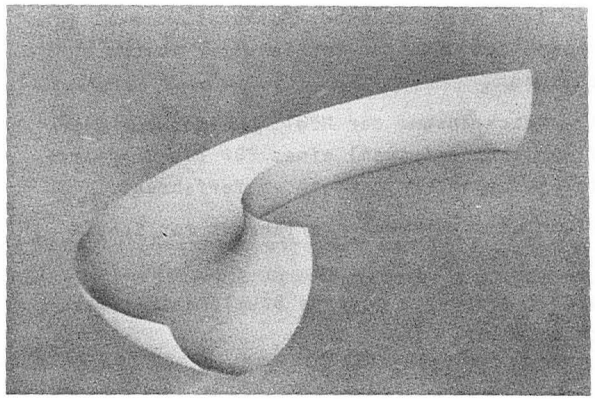


Bild 2.3: Schattierte Fläche /5/

Im CAD Center in Cambridge, England, wurde ein modifiziertes Line scan Verfahren /6/ entwickelt, das etwa 5 mal schneller arbeitet. Unbefriedigend bleibt, daß diese Bilder nicht flackerfrei auf dem Bildschirm zu sehen sind. Interaktives Arbeiten wird dadurch sehr erschwert.

Aus diesen Ausführungen über den jetzigen Stand der Technik der Flächendarstellung ergeben sich folgende Aufgabenstellungen, die in der Gliederung dieser Arbeit ihren Niederschlag finden:

1) Bereitstellung mathematischer Methoden, die auch vom mathematisch ungeschulten Benutzer leicht anzuwenden sind.
2) Sinnvolle Beschränkung der Möglichkeiten zur Flächendefinition und Auswahl eines für die Hardware-Realisierung besonders geeigneten Verfahrens.
3) Entwicklung eines Näherungsverfahrens zur Schattierung von Flächen.
4) Untersuchung der durch moderne Bauelemente gegebenen Möglichkeiten.
5) Vorschlag für eine modular erweiterbare Display-Hardware, mit der die o.g. Probleme gelöst werden.

3. Mathematische Verfahren zur Flächenbeschreibung

3.1 Parameterdarstellung

Eine Fläche läßt sich in einer der folgenden Formen definieren:

1)	Implizite Form	$F(x,y,z) = 0$
2)	Explizite Form	$z = f(x,y)$
3)	Parameterdarstellung	$x = x(u,v)$ $y = y(u,v)$ $z = z(u,v)$

Für Computer Graphics wird meistens die Parameterdarstellung gewählt. Die Gründe hierfür sollen kurz dargelegt werden.

Das Bild auf dem Sichtgerät (Display) entsteht durch Überlagerung der x- und y-Strahlablenkung. Das bedeutet bei Anwendung der impliziten Form, daß nach x und y aufgelöst werden muß, was bereits bei Flächen zweiten Grades erheblichen Rechenaufwand bedeuten kann. Die explizite Form besitzt den Nachteil, daß die Funktion, wie in Bild 3.1 anhand einer Kurve veranschaulicht, mehrdeutig sein kann.

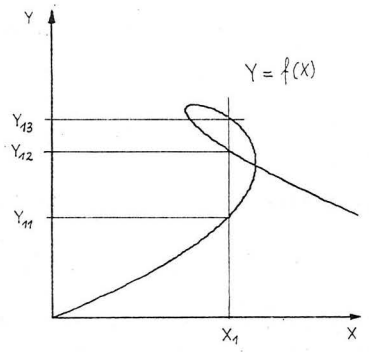


Bild 3.1: Mehrdeutigkeit der expliziten Form

Die explizite wie auch die implizite Form bereiten numerische Schwierigkeiten beim Ableiten von Flächen mit vertikaler Tangente.

Diese Schwierigkeiten treten bei der Parameterdarstellung nicht auf, da sie unabhängig von den Koordinatenachsen ist. Die Koordinaten eines Flächenpunktes werden unabhängig voneinander als Funktion der Parameter berechnet. Damit beschreibt die Parameterdarstellung die

Bewegung des Elektronenstrahls auf der Fläche. Zu jedem Parameterpaar (u_i, v_j) gehört genau ein Flächenpunkt. Bild 3.2 veranschaulicht dies am Beispiel der Kurve aus Bild 3.1.

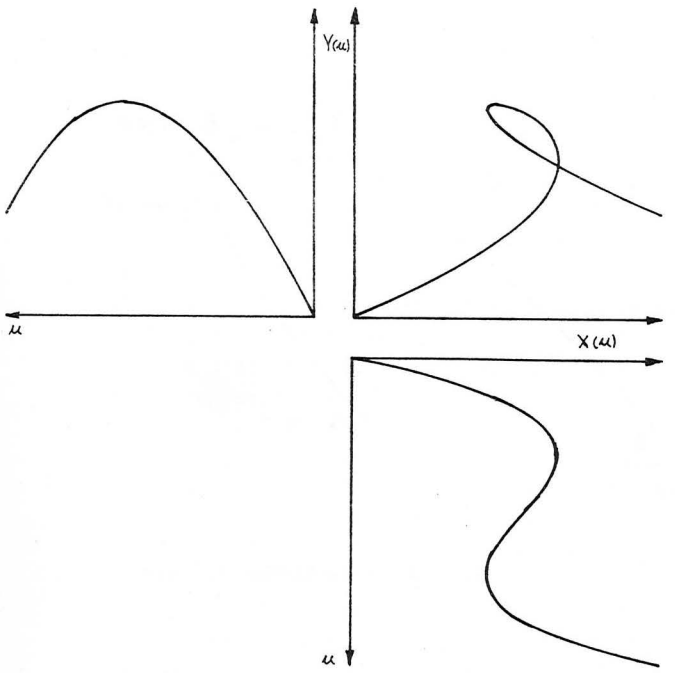


Bild 3.2: Erzeugung der Kurve aus Bild 3.1

Diese Eindeutigkeit erlaubt es, daß sehr einfach durch Angabe eines Parameterbereichs Ausschnitte einer Fläche dargestellt werden können.

Ein weiterer Vorteil der Parameterdarstellung ist die Tatsache, daß bei der Definition einer Fläche durch Angabe von Stützstellen ein gutes "Spacing" erreicht wird. Was darunter verstanden wird, zeigt Bild 3.3 am Beispiel einer Kurve.

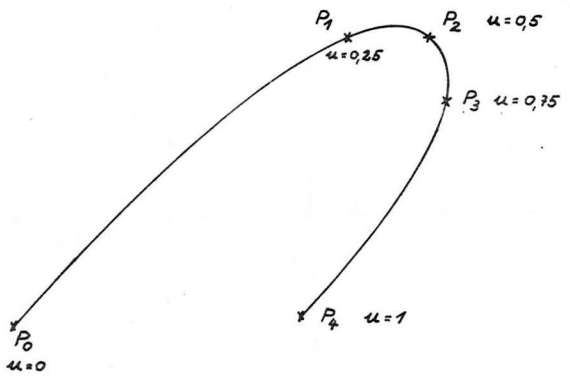


Bild 3.3: Gutes "Spacing" mit Parameterdarstellung

Bei starker Krümmung der Kurve werden zu ihrer Definition mehr Stützstellen angegeben als bei geringer Krümmung. Der Wertebereich des Parameters verteilt sich gleichmäßig über die Stützstellenbereiche. Somit werden an Stellen starker Krümmung viele Kurvenpunkte berechnet. Das bedeutet, daß trotz der linearen Interpolation des Vektorgenerators im Display die beabsichtigte Kurve mit starker Krümmung wiedergegeben wird. Der Wertebereich der Parameter wird üblicherweise auf $o \leq u, v \leq 1$ normiert.

3.2 In Computer Graphics angewandte Flächen

Die in Computer Graphics angewandten Flächen lassen sich in drei Gruppen unterteilen /7/:

1) Analytisch gegebene Flächen wie z.B. Kugelflächen, Paraboloide usw., deren Parameterlinien Kegelschnitte sind. Sie lassen sich als Spezialfälle der allgemeinen gebrochen rationalen kubischen Funktionen darstellen. Ihre Anwendungsgebiete sind vor allem Maschinenbau und Architektur. (In der Architektur beschränkt man sich im wesentlichen auf abwickelbare Flächen.)

2) Empirisch gegebene Flächen, die durch Interpolation zwischen vorgegebenen Stützstellen oder Kurven entstehen. Diese Flächen treten bei experimentellen Optimierungsverfahren, z.B. im Schiffsbau, Flugzeugbau usw. auf. Zur Interpolation werden Lagrange-Polynome, Hermite-Polynome, gebrochen rationale Funktionen und Splines verwandt.

3) Empirisch gegebene Flächen, die durch Approximation von Polygonnetzen entstehen. Diese Flächen werden beim interaktiven Entwurf von Formen, die vor allem ästhetischen Gesichtspunkten genügen müssen, eingesetzt. Die meisten angewandten Verfahren beruhen auf der Bézier-Approximation /8/.

3.2.1 Analytisch gegebene Flächen

Die analytisch gegebenen Flächen werden in dieser Arbeit mit gebrochen rationalen kubischen Funktionen beschrieben, weil diese Funktionen auch bei der Berechnung der empirisch gegebenen Flächen verwendet werden. Damit ist eine einheitliche Matrizendarstellung für alle 3 Gruppen von Flächen möglich. Die gebrochen rationalen kubischen Funktionen treten bei der Verwendung von homogenen Koordinaten /9/, /10/ auf.

Ein Punkt auf der Fläche wird beschrieben durch den Vektor q in homogenen Koordinaten

$$q = [x, y, z, w] \qquad (3.1)$$

Aus (3.1) erhält man die gewöhnlichen Koordinaten durch Division.

$$Q = [X, Y, Z] = \left[\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right] \qquad (3.2)$$

Eine 3D-Fläche ist dann in homogenen Koordinaten gegeben durch

$$q(u,v) = [x(u,v), y(u,v), z(u,v), w(u,v)] = [u^3\ u^2\ u\ 1] \cdot \overline{A} \cdot \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix} = U^T \cdot \overline{A} \cdot V \qquad (3.3)$$

$$0 \leq u, v \leq 1$$

wobei $\overline{A}$ ein (4x4x4)-Tensor ist und die Form der Fläche bestimmt. $\overline{A}$ wird deshalb generierender Tensor genannt.

$$\overline{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} = [A_x, A_y, A_z, A_w] \qquad (3.4)$$

mit $A_x = \begin{bmatrix} a_{11x} & a_{12x} & a_{13x} & a_{14x} \\ a_{21x} & a_{22x} & a_{23x} & a_{24x} \\ a_{31x} & a_{32x} & a_{33x} & a_{34x} \\ a_{41x} & a_{42x} & a_{43x} & a_{44x} \end{bmatrix}$ (3.5)

Parameterlinien beschreiben Kurven auf der Fläche. Man erhält sie durch Festhalten des einen und Variieren des andern Parameters. Insbesondere liefern die Parameterwerte (o,v), (1,v), (u,o) und (u,1) die Randkurven der Fläche. Sei beispielsweise v = o, so wird aus Gl.(3.3) mit Gl.(3.4):

$$q(u) = [u^3 u^2 u\ 1] \cdot \begin{bmatrix} A_{14} \\ A_{24} \\ A_{34} \\ A_{44} \end{bmatrix}$$ und mit Gl.(3.5)

$$= [u^3 u^2 u\ 1] \cdot \begin{bmatrix} a_{14x} & a_{14y} & a_{14z} & a_{14w} \\ a_{24x} & a_{24y} & a_{24z} & a_{24w} \\ a_{34x} & a_{34y} & a_{34z} & a_{34w} \\ a_{44x} & a_{44y} & a_{44z} & a_{44w} \end{bmatrix} \qquad (3.6)$$

Hieraus berechnen sich die Bildschirmkoordinaten zu

$$X = \frac{x}{w} = \frac{a_{14x}\,u^3 + a_{24x}\,u^2 + a_{34x}\,u + a_{44x}}{a_{14w}\,u^3 + a_{24w}\,u^2 + a_{34w}\,u + a_{44w}} = \frac{\sum_{i=1}^{4} a_{i4x}\,u^{4-i}}{\sum_{i=1}^{4} a_{i4w}\,u^{4-i}}$$

$$Y = \frac{y}{w} = \frac{\sum_{i=1}^{4} a_{i4y}\,u^{4-i}}{\sum_{i=1}^{4} a_{i4w}\,u^{4-i}} \qquad (3.7)$$

$$Z = \frac{z}{w} = \frac{\sum_{i=1}^{4} a_{i4z}\,u^{4-i}}{\sum_{i=1}^{4} a_{i4w}\,u^{4-i}}$$

Die Methode zur Parametrisierung der analytisch gegebenen Flächen, d.h. zur Bestimmung des generierenden Tensors $\overline{A}$, soll hier nur angedeutet und mit einigen Beispielen veranschaulicht werden. Eine ausführliche Darstellung ist in /11/, /12/ zu finden.

Ausgehend von einem möglichst einfachen Kegelschnitt in expliziter Darstellung, z.B. der Parabel $x = y^2$, gewinnt man dessen Parameterdarstellung und damit die generierende Matrix. Hieraus erhält man durch lineare Transformationen die generierenden Matrizen für die anderen Kegelschnitte. Die räumlichen Flächen werden dann unter Ausnutzung der Tatsache, daß ihre Parameterlinien Kegelschnitte sind, parametrisiert.

Beispiel:

Die Parabel $X = Y^2$ wird parametrisiert und in den Einheitskreis im Ursprung transformiert.

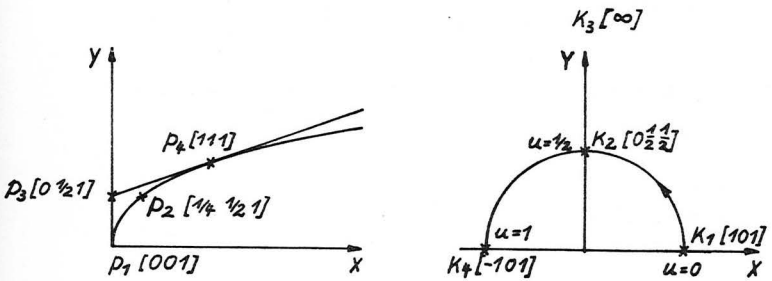


Bild 3.4: Parabel $X = Y^2$ und Einheitskreis

Mit der Parametrisierung $x = u^2$
$y = u$
$w = 1$ erhält man

$$q_{\text{Parabel}}(u) = \left[x(u),\, y(u),\, w(u)\right] = \left[u^2\ u\ 1\right] \cdot \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix} \qquad (3.8)$$

mit $P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ als generierender Matrix.

Die Transformation

$$q_{Parabel} \cdot \begin{bmatrix} 0 & -2 & 2 \\ -2 & 2 & -2 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u^2 & u & 1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 0 & -2 & 2 \\ -2 & 2 & -2 \\ 1 & 0 & 1 \end{bmatrix}}_{k} \quad (3.9)$$

liefert den oberen Halbkreis /13/. Die homogenen Koordinaten sind

$$x = -2u + 1$$
$$y = -2u^2 + 2u$$
$$w = 2u^2 - 2u + 1$$

Einen Zylinder mit der Z-Achse als Rotationsachse erhält man, wenn

$$x = -2u + 1 \longrightarrow A_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$y = -2u^2 + 2u \longrightarrow A_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.10)$$

$$z = v(2u^2 - 2u + 1) \longrightarrow A_z = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$w = (2u^2 - 2u + 1) \longrightarrow A_w = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

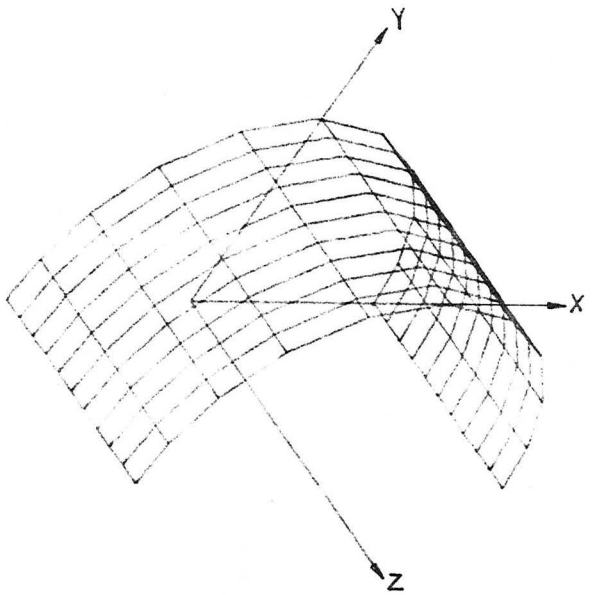


Bild 3.5: Zylinder nach Gl. (3.10)

Hyperbolisches Paraboloid: $Z = X^2 - Y^2$

$$x = -2u+1 \longrightarrow A_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$y = -2v+1 \longrightarrow A_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 \end{bmatrix} \quad (3.11)$$

$$z = 4u^2 - 4u - 4v^2 + 4v \longrightarrow A_z = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & -4 \\ 0 & -4 & 4 & 0 \end{bmatrix}$$

$$w = 1 \longrightarrow A_w = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

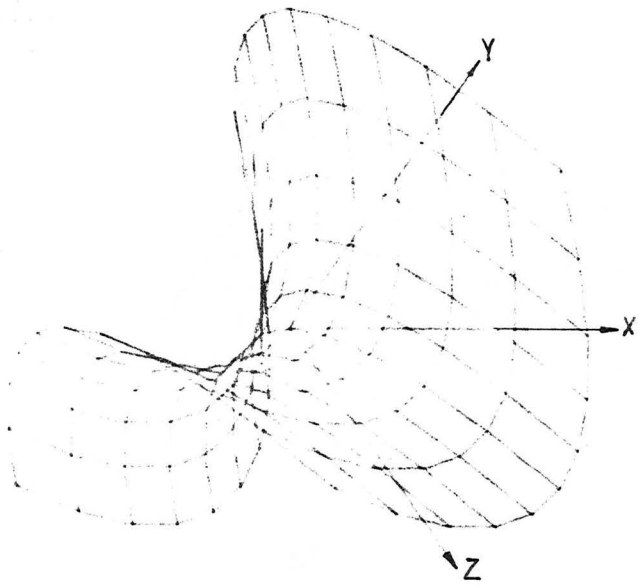


Bild 3.6: Hyperbolisches Paraboloid nach Gl. (3.11)

Kugelfläche: $x^2 + y^2 + z^2 = 1$

$$
\begin{aligned}
x &= (2u-1)\cdot(-2v^2+2v) \longrightarrow A_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -4 & 4 & 0 \\ 0 & 2 & -2 & 0 \end{bmatrix} \\
y &= (-2u^2+2u)\cdot(-2v^2+2v) \longrightarrow A_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -4 & 0 \\ 0 & -4 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
z &= (2u^2-2u+1)\cdot(2v-1) \longrightarrow A_z = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & -2 \\ 0 & 0 & -4 & 2 \\ 0 & 0 & 2 & -1 \end{bmatrix} \\
w &= (2u^2-2u+1)\cdot(2v^2-2v+1) \longrightarrow A_w = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -4 & 2 \\ 0 & -4 & 4 & -2 \\ 0 & 2 & -2 & 1 \end{bmatrix}
\end{aligned}
\qquad (3.12)
$$

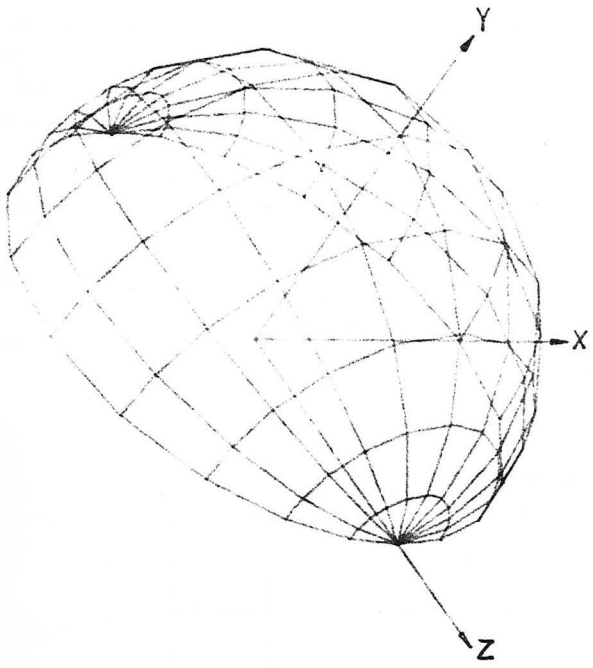


Bild 3.7: Kugelfläche nach Gl. (3.12)

3.2.2 Empirisch gegebene Flächen

Nach Coons /1/ werden Flächen durch Aneinandersetzen von Teilflächen, sog. Patches oder Pflaster beschrieben. Die Gesamtfläche mit den Parametern (u', v') wird so unterteilt, daß gilt:

$$
\begin{aligned}
&o \le u'_i \le u' \le u'_{i+1} \le u'_m \quad , \quad u'_m = 1,2,..,m \\
&o \le v'_j \le v' \le v'_{j+1} \le v'_n \quad , \quad v'_n = 1,2,..,n
\end{aligned}
\qquad (3.13a)
$$

Für das Pflaster mit den Parametern (u,v) soll gelten:

$$o \leq u_i \leq u \leq u_{i+1} \leq 1 \quad , \quad u_i = \frac{i}{m}, \quad i = 0,1,2,..,\ m-1$$
$$o \leq v_j \leq v \leq v_{j+1} \leq 1 \quad , \quad v_j = \frac{j}{n}, \quad j = 0,1,2,..,\ n-1 \qquad (3.13b)$$

Der Zusammenhang zwischen Gesamtfläche und Pflaster ist dann gegeben durch

$$u = u' \ (\text{mod } 1) \quad , \quad u' = i + u$$
$$v = v' \ (\text{mod } 1) \quad , \quad v' = j + v \qquad (3.13c)$$

Bild 3.8 verdeutlicht dies in der u' - v' -Ebene.

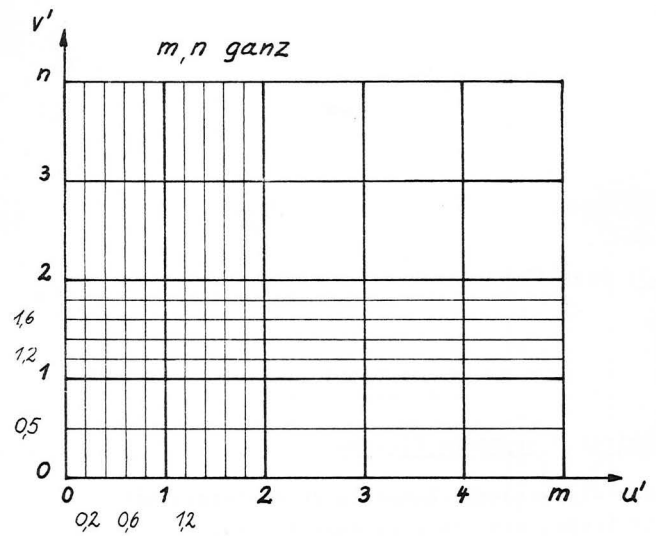


Bild 3.8: Darstellung der Parametrisierung

Sind Pflaster nicht als Funktion zweier Variabler analytisch gegeben, so müssen sie aus einfacheren Daten, d.h. aus Punkten und Tangentenvektoren (Konstanten) oder Kurven (Funktionen einer Variablen) aufgebaut werden. Prinzipiell bestehen dazu drei Möglichkeiten /14/:

1) Definition durch Kartesisches Produkt

Werden mit P(u,v) die Daten zur Definition und mit Q(u,v) die Punkte des berechneten Pflasters bezeichnet, so ist die Fläche durch das Kartesische Produkt folgendermaßen definiert:

$$Q(u,v) = \sum_{i=o}^{m} \sum_{j=o}^{n} P(u_i, v_j) \cdot U_{i,m}(u) \cdot V_{j,n}(v) \qquad (3.14)$$

wobei $U_{i,m}(u)$ und $V_{j,n}(v)$ Interpolations- oder Approximationsfunktionen sind.

In Matrizenform wird Gl. (3.14)

$$Q(u,v) = U^T \cdot \overline{P} \cdot V \qquad (3.15)$$

mit
$$U = \begin{bmatrix} U_{o,m}(u) \\ \vdots \\ U_{i,m}(u) \\ \vdots \\ U_{m,m}(u) \end{bmatrix} \quad \text{bzw.} \quad V = \begin{bmatrix} V_{o,n}(v) \\ \vdots \\ V_{j,n}(v) \\ \vdots \\ V_{n,n}(v) \end{bmatrix} \qquad (3.16)$$

$$\overline{P} = \begin{bmatrix} P(u_o,v_o) & \cdots & P(u_o,v_j) & \cdots & P(u_o,v_n) \\ \vdots & & \vdots & & \vdots \\ P(u_i,v_o) & \cdots & P(u_i,v_j) & \cdots & \\ \vdots & & \vdots & & \vdots \\ P(u_m,v_o) & \cdots & \cdots & \cdots & P(u_m,v_n) \end{bmatrix} \qquad (3.17)$$

und $$P(u_i, v_j) = \left[X(u_i, v_j),\ Y(u_i, v_j),\ Z(u_i, v_j)\right] \qquad (3.18)$$

Gl. (3.14) bzw. Gl. (3.15) beschreibt das Pflaster bei Schrittweiten Δu und Δv durch die Anzahl von Punkten, die der Rasterung $\frac{1+\Delta u}{\Delta u}$, $\frac{1+\Delta v}{\Delta v}$ (konstante Schrittweite wird vorausgesetzt) in der u-v-Ebene entspricht (Bild 3.9).

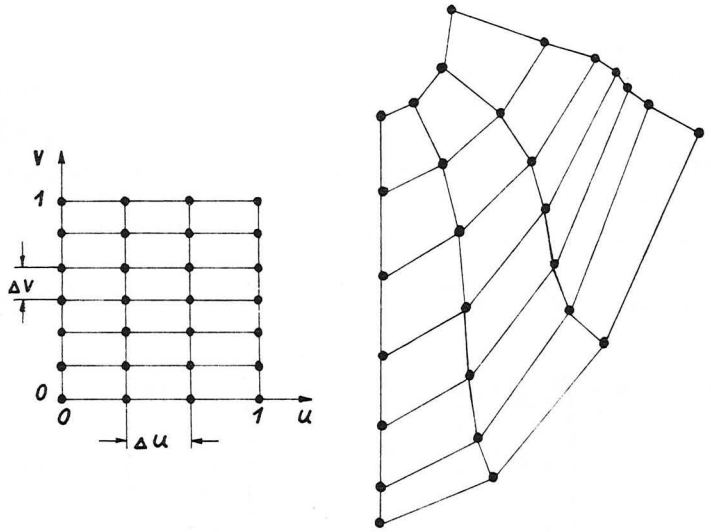


Bild 3.9: Kartesische Produkt Fläche

Die "Kartesische Produkt"-Fläche wird, wie aus Gl. (3.14) zu erkennen ist, durch konstante Werte $P(u_i, v_j)$, also durch Punkte oder Tangentenvektoren bestimmt. Die Form des Pflasters wird zusätzlich durch die Wahl der Interpolations- oder Approximationspolynome beeinflußt.

2) Definition durch eine Kurvenfamilie ("Lofting")

Bei diesem Verfahren sind u- oder v-Kurven die Bestimmungsgrößen des Pflasters. Die Punkte Q(u,v) werden durch Interpolation oder Approximation dieser Kurven berechnet:

$$Q(u,v) = \sum_{i=o}^{m} P(u_i, v) \cdot U_{i,m}(u) \qquad (3.19a)$$

oder $$Q(u,v) = \sum_{j=o}^{n} P(u, v_j) \cdot V_{j,n}(v) \qquad (3.19b)$$

In Matrizenform wird

(3.19a): $$Q(u,v) = U^{T} \cdot \overline{P}(v) \qquad (3.20a)$$

$$= \left[U_{o,m}(u) \ \ldots U_{i,m}(u) \ldots U_{m,m}(u)\right] \cdot \begin{bmatrix} P(u_o, v) \\ \vdots \\ P(u_i, v) \\ \vdots \\ P(u_m, v) \end{bmatrix}$$

(3.19b): $$Q(u,v) = \overline{P}(u) \cdot V \qquad (3.20b)$$

Bei der Anwendung des "Lofting" wird also das Pflaster durch die Anzahl von Kurven dargestellt, die der Rasterung $\frac{1+\Delta u}{\Delta u}$ oder $\frac{1+\Delta v}{\Delta v}$ auf der u- oder v-Achse in der Parameterebene entspricht (Bild 3.9).

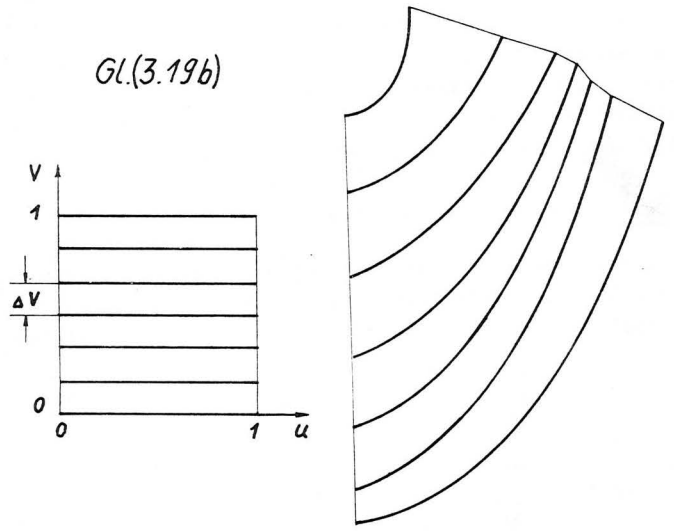


Bild 3.9a: "Lofting"

Werden die Kurven $P(u_i, v)$ bzw. $P(u, v_j)$ durch Interpolation oder Approximation gewonnen, d.h. ist

$$\text{z.B.} \quad P(u_i, v) = \sum_{j=o}^{n} P(u_i, v_j) \cdot V_{j,n}(v), \qquad (3.21)$$

so erhält man durch Einsetzen in Gl. (3.19a) wieder die Kartesische Produkt-Fläche nach Gl. (3.14).

3) Definition durch zwei Kurvenfamilien (Verallgemeinertes Coons'sches Verfahren)

Bei dieser Methode sind u- und v-Kurven die Bestimmungsgrößen des Pflasters. Sie können, abgesehen von der Einschränkung

$$P(u_i, v)\Big|_{v=v_j} \equiv P(u, v_j)\Big|_{u=u_i} \qquad (3.22)$$

wie beim "Lofting" beliebig gewählt werden. Die Punkte $Q(u,v)$ des Pflasters werden durch Interpolation oder Approximation der beiden Kurvenfamilien berechnet. Aus Gl. (3.22) folgt, daß bei additiver Überlagerung der beiden Pflaster nach Gl. (3.19a) und Gl. (3.19b) die Bestimmungsstücke $P(u_i, v_j)$ des Pflasters doppelt eingehen. Um dies zu vermeiden, muß deren Beitrag, der mit dem Kartesischen Produkt nach Gl. (3.14) identisch ist, einmal subtrahiert werden (dies ist besonders einleuchtend, wenn man den Fall der Interpolation $Q(u_i, v_j) \equiv P(u_i, v_j)$ betrachtet):

$$Q(u,v) = \sum_{i=o}^{m} P(u_i, v) \cdot U_{i,m}(u) + \sum_{j=o}^{n} P(u, v_j) \cdot V_{j,n}(v) - \sum_{i=o}^{m} \sum_{j=o}^{n} P(u_i, v_j) \cdot U_{i,m}(u) \cdot V_{j,n}(v) \qquad (3.23)$$

oder in Matrizenform

$$Q(u,v) = \underline{U}^T \cdot \underline{P}(v) + \underline{P}(u) \cdot \underline{V} - \underline{U}^T \cdot \underline{P} \cdot \underline{V} \qquad (3.24)$$

Bei der Definition durch zwei Kurvenfamilien wird das Pflaster durch die Anzahl von u- und v-Kurven dargestellt,

die der Rasterung $\frac{1+\Delta u}{u}$ und $\frac{1+\Delta v}{v}$ auf der u- und v-Achse in der Parameterebene entspricht (Bild 3.10).

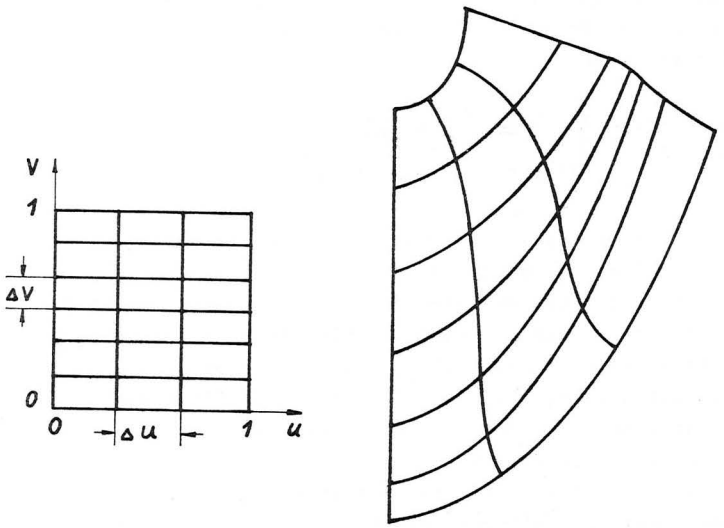


Bild 3.10: Definition durch zwei Kurvenfamilien

Es ist leicht einzusehen, daß die Definitionen durch Kartesisches Produkt und "Lofting" Spezialfälle der Definition durch zwei Kurvenfamilien sind. Mit der Approximation nach

Gl.(3.21): $P(u_i,v) = \sum_{j=o}^{n} P(u_i,v_j)\cdot V_{j,n}(v)$

bzw. $P(u,v_j) = \sum_{i=o}^{m} P(u_i,v_j)\cdot U_{i,m}(u)$

wird aus Gl. (3.23):

$$Q(u,v) = \sum_{i=o}^{m}\left(\sum_{j=o}^{n} P(u_i,v_j)\cdot V_{j,n}(v)\right)\cdot U_{i,m}(u)$$

$$+\sum_{j=o}^{n} P(u,v_j)\cdot V_{j,n}(v)$$

$$-\sum_{i=o}^{m}\sum_{j=o}^{n} P(u_i,v_j)\cdot U_{i,m}(u)\cdot V_{j,n}(v)$$

$$Q(u,v) = \sum_{j=o}^{n} P(u,v_j)\cdot V_{j,n}(v) \quad : \text{"Lofting" Gl. (3.19b)}$$

$$= \sum_{j=o}^{n}\left(\sum_{i=o}^{m} P(u_i,v_j)\cdot U_{i,m}(u)\right)\cdot V_{j,n}(v) : \text{"Kartesisches Produkt" Gl. (3.14)}$$

Es steht also eine Hierarchie von Flächendefinitionen zur Verfügung, wie sie in Bild 3.11 dargestellt ist.

In den weiteren Ausführungen wird die Flächendefinition durch Kartesisches Produkt nach

Gl. (3.15) $\quad Q(u,v) = U^{T}\cdot \overline{P}\cdot V$

angewandt. Dies geschieht aus folgenden Gründen:

1) Die Bestimmungsstücke für diese Flächendefinition sind Punkte und Tangentenvektoren, also konstante Werte. Sie sind für den Benutzer einfacher anzugeben als Kurven (Funktionen einer Variablen), wie sie für die Definitionen durch eine oder zwei Kurvenfamilien benötigt werden.

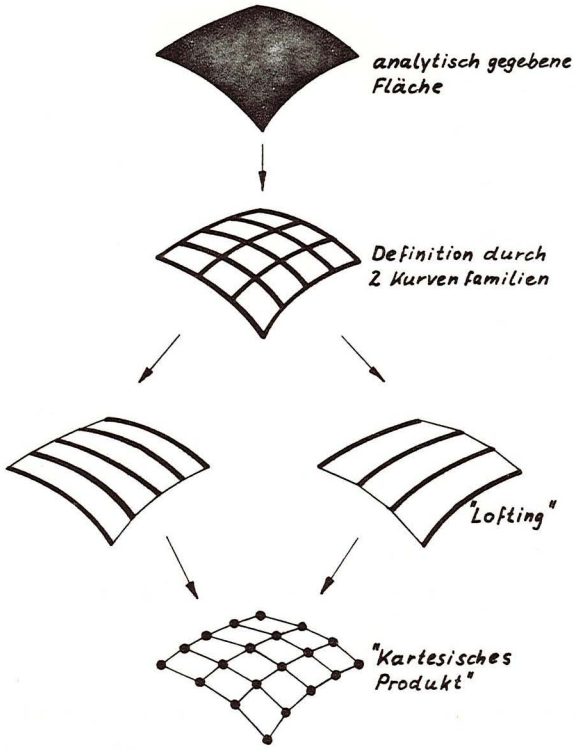


Bild 3.11: Hierarchie der Flächendefinitionen /15/

2) Diese Flächendefinition erfordert den geringsten Rechenaufwand, was in Hinblick auf die angestrebte Echtzeitberechnung mit einer speziellen Hardware wichtig ist.

3) Die gegenüber den beiden anderen Flächendefinitionen geringere Approximationsgenauigkeit /14/ ist aufgrund der beschränkten Auflösung der Sichtgeräte bedeutungslos.

4) Ein Großteil der bekannten, produktiv eingesetzten Entwurfssysteme arbeitet mit der Flächendefinition durch Kartesisches Produkt. Als Beispiele seien genannt: "Sytème UNISURF" bei Renault /8/, "Numerical Master Geometry" bei BAC /16/.

Im folgenden werden nun die Interpolations- und Approximationsverfahren untersucht und danach wiederum eine Auswahl nach den Kriterien "Einfache Handhabung für den Benutzer" und "Geringer Rechenaufwand bei der numerischen Auswertung" getroffen.

3.3 Interpolation

Die Interpolation wird durch die Beziehung

$$Q(u_i, v_j) \equiv P(u_i, v_j) \tag{3.25}$$

charakterisiert. Werden neben Punkten auch Tangentenvektoren zur Bestimmung des Pflasters vorgegeben, so muß zusätzlich gelten

$$\frac{\partial Q(u_i, v_j)}{\partial u} \equiv \frac{\partial P(u_i, v_j)}{\partial u} = P_u(u_i, v_j) \tag{3.26}$$

$$\frac{\partial Q(u_i, v_j)}{\partial v} \equiv \frac{\partial P(u_i, v_j)}{\partial v} = P_v(u_i, v_j) \tag{3.27}$$

$$\frac{\partial^2 Q(u_i, v_j)}{\partial u \partial v} \equiv \frac{\partial^2 P(u_i, v_j)}{\partial u \partial v} = P_{uv}(u_i, v_j) \tag{3.28}$$

Die Interpolationsfunktionen $U_{i,m}(u)$ und $V_{j,n}(v)$ müssen so gewählt werden, daß das Pflaster

$$Q(u,v) = U^T \cdot P \cdot V$$

die o.g. Bedingungen erfüllt.

3.3.1 Lagrange-Interpolation

Werden zur Bestimmung des Pflasters nur Punkte $P(u_i, v_j)$ vorgegeben, so ist die Bedingung

$$Q(u_i, v_j) \equiv P(u_i, v_j)$$

erfüllt, wenn für die Funktionen $U_{i,m}(u)$ und $V_{j,n}(v)$ die Lagrange'schen Interpolationsfunktionen

$$L_{i,m}(p) = \prod_{\substack{k=o \\ k\neq i}}^{m} \frac{(p-p_k)}{(p_i-p_k)} \qquad (3.$$

mit der Eigenschaft

$$L_{i,m}(p_k) = \delta_{ik} = \begin{cases} 1 & \text{für } i = k \\ 0 & \text{sonst} \end{cases} \qquad (3.$$

$$\sum_{i=o}^{m} L_{i,m}(p) \equiv 1 \qquad (3.$$

eingesetzt werden. Werden (m+1) · (n+1) Punkte als Bestimmungsstücke vorgegeben, so berechnet sich das Pflaster aus

$$Q(u,v) = U^T \bar{P} V$$

$$= \begin{bmatrix} L_{o,m}(u) \\ \vdots \\ L_{i,m}(u) \\ \vdots \\ L_{m,m}(u) \end{bmatrix}^T \cdot \begin{bmatrix} P(u_o,v_o) \ldots P(u_o,v_j) \ldots P(u_o,v_n) \\ \vdots \\ P(u_i,v_o) \ldots P(u_i,v_j) \ldots \\ \vdots \\ P(u_m,v_o) \ldots\ldots\ldots\ldots P(u_m,v_n) \end{bmatrix} \cdot \begin{bmatrix} L_{o,n}(v) \\ \vdots \\ L_{j,n}(v) \\ \vdots \\ L_{n,n}(v) \end{bmatrix} \qquad (3$$

Beispiel 1: $m = 3$, $n = 2$, $u_i = \frac{i}{3}$, $v_j = \frac{j}{2}$

Gl. (3.29) für m = 3 bzw. n = 2 ausgewertet und in Gl. (3.32) eingesetzt, ergibt:

$$Q(u,v) = \frac{9}{2} \cdot \begin{bmatrix} -1(u-\frac{1}{3})(u-\frac{2}{3})(u-1) \\ 3u \cdot (u-\frac{2}{3})(u-1) \\ -3u(u-\frac{1}{3}) \cdot (u-1) \\ u(u-\frac{1}{3})(u-\frac{2}{3}) \cdot \end{bmatrix}^T \cdot \begin{bmatrix} P(0,0), P(0,\frac{1}{2}), P(0,1) \\ P(\frac{1}{3},0), P(\frac{1}{3},\frac{1}{2}), P(\frac{1}{3},1) \\ P(\frac{2}{3},0), P(\frac{2}{3},\frac{1}{2}), P(\frac{2}{3},1) \\ P(1,0), P(1,\frac{1}{2}), P(1,1) \end{bmatrix} \cdot \begin{bmatrix} 2(v-\frac{1}{2})(v-1) \\ -4v \cdot (v-1) \\ 2v(v-\frac{1}{2}) \cdot \end{bmatrix} \qquad (3.$$

Bild 3.12 zeigt die Lagrange-Polynome dritten Grades $L_{i,3}(p)$, $0 \leq p \leq 1$. In Bild 3.13 ist das Pflaster nach Gl. (3.34) dargestellt.

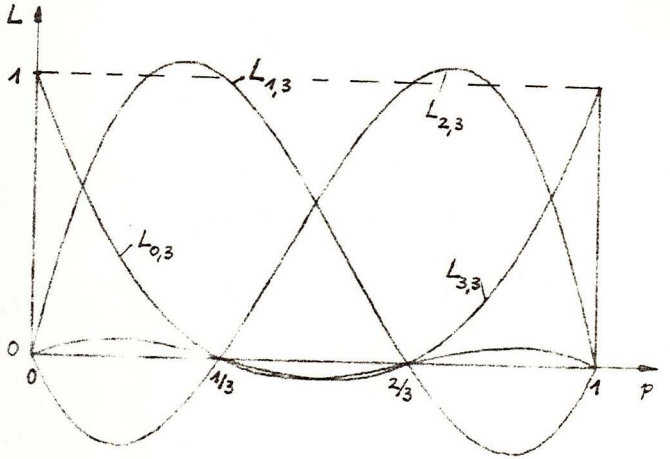


Bild 3.12: Lagrange Polynome für m = 3

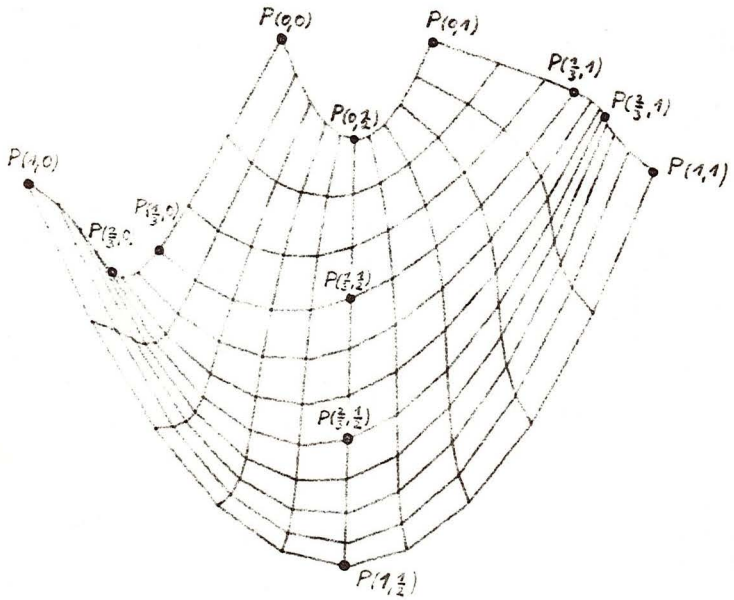


Bild 3.13: Pflaster nach Gl. (3.34)

Beispiel 2: m = n = 3, "16-Punkte-Fläche"

Mit Gl. (3.34) erhält man das Pflaster aus

$$Q(u,v) = \qquad (3.35)$$

$$\frac{9}{2}\cdot\begin{bmatrix} -1(u-\frac{1}{3})(u-\frac{2}{3})(u-1) \\ 3u\ .\ (u-\frac{2}{3})(u-1) \\ -3u(u-\frac{1}{3})\ .\ (u-1) \\ u(u-\frac{1}{3})(u-\frac{2}{3})\ . \end{bmatrix}^T \begin{bmatrix} P(0,0) & P(0,\frac{1}{3}) & P(0,\frac{2}{3}) & P(0,1) \\ P(\frac{1}{3},0) & . & . & \\ P(\frac{2}{3},0) & . & . & \\ P(1,0) & . & . & P(1,1) \end{bmatrix} \cdot \begin{bmatrix} -1(v-\frac{1}{3})(v-\frac{2}{3})(v-1) \\ 3v\ .\ (v-\frac{2}{3})(v-1) \\ -3v(v-\frac{1}{3})\ .\ (v-1) \\ v(v-\frac{1}{3})(v-\frac{2}{3})\ . \end{bmatrix}$$

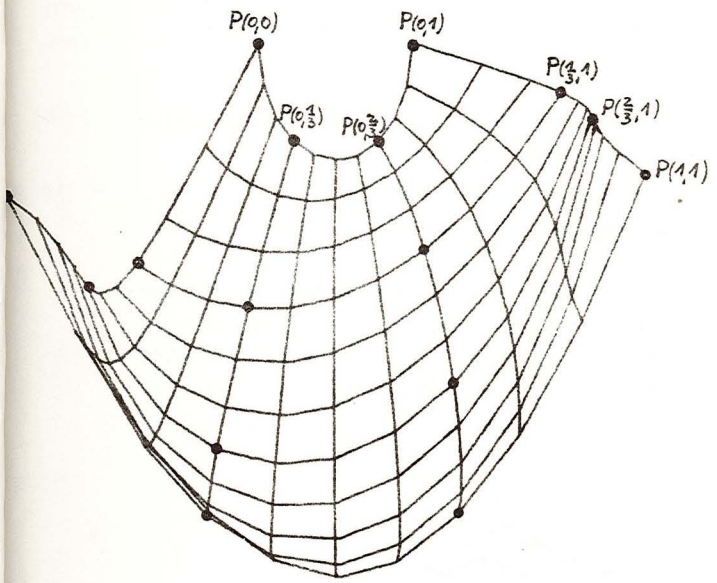


Bild 3.14: "16-Punkte-Fläche"

Die Lagrange-Interpolation hat neben dem Vorteil der einfachen Bestimmung durch Angabe von Punkten viele Nachteile:

1) Die Pflaster können in der Regel nur mit Stetigkeit nullter Ordnung aneinandergefügt werden, d.h. am Übergang zwischen zwei Pflastern tritt eine Kante auf.

2) Der Grad der Interpolationsfunktionen hängt von der Anzahl der Bestimmungsstücke ab. Entsprechendes gilt für den Rechenaufwand zur numerischen Auswertung.

3) Vom fünften Grad aufwärts zeigt die Lagrange-Interpolation eine unerwünschte Welligkeit.

3.3.2 Hermite-Interpolation

Damit Pflaster kantenfrei aneinandergefügt werden können, müssen neben Punkten $P(u_i, v_j)$ auch Tangentenvektoren $P_u(u_i, v_j)$, $P_v(u_i, v_j)$, $P_{uv}(u_i, v_j)$ vorgeschrieben werden, wie Bild 3.15 veranschaulicht.

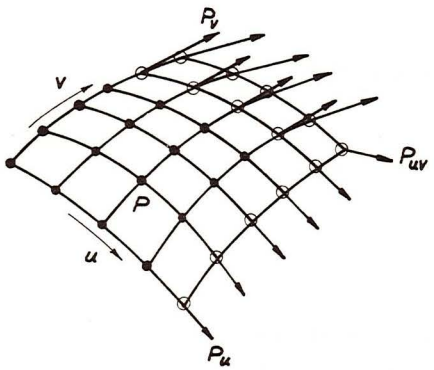


Bild 3.15: Bestimmung des Pflasters durch Punkte und Tangentenvektoren

Das Pflaster muß dann die Bedingungen nach Gl. (3.25) bis Gl. (3.28) erfüllen. Die Vektoren U und V der Interpolationsfunktionen gewinnt man folgendermaßen /17/:

$$Q(u,v) = U^T \cdot \overline{P} \cdot V = (N^{-1}u)^T \cdot \overline{P} \cdot (M^{-1}v) \tag{3.36}$$

$$\text{mit } N = \begin{bmatrix} u_o^m & \dots & u_r^m & , & mu_{r+1}^{m-1} & \dots & mu_m^{m-1} \\ u_o^{m-1} & & u_r^{m-1} & , & (m-1)u_{r+1}^{m-2} & \dots & (m-1)u_m^{m-2} \\ \vdots & & \vdots & & \vdots & & \vdots \\ u_o & \dots & u_r & , & 1 & \dots & 1 \\ 1 & \dots & 1 & & 0 & \dots & 0 \end{bmatrix} \tag{3.37}$$

$$= \left[u_o, \dots\ u_r, \quad u'_{r+1}, \dots\ u'_m\right] \tag{3.38}$$

$$M = \left[v_o, \dots\ v_s, \quad v'_{s+1}, \dots\ v'_n\right] \tag{3.39}$$

N und M sind Matrizen vom Vandermonde-Typ /18/ und nichtsingulär, da alle u_i und u_i' bzw. v_j und v_j' verschieden sind.

$$u = \begin{bmatrix} u^m \\ u^{m-1} \\ \vdots \\ u \\ 1 \end{bmatrix}, \quad v = \begin{bmatrix} v^n \\ v^{n-1} \\ \vdots \\ v \\ 1 \end{bmatrix} \tag{3.40}$$

$$\overline{P} = \begin{bmatrix} P(u_o,v_o) \dots P(u_o,v_s) & P_v(u_o,v_{s+1}) \dots P_v(u_o,v_n) \\ \vdots & \vdots \\ P(u_r,v_o) & \vdots \\ P_u(u_{r+1},v_o) & P_{uv}(u_{r+1},v_{s+1}) \\ \vdots & \vdots \end{bmatrix} \tag{3.41}$$

Beispiel 3: Bikubisches Pflaster : (m = n = 3)

Die 16 Bestimmungsgrößen des bikubischen Pflasters sind mit der Vorgabe der 4 Eckpunkte und den 3 Tangentenvektoren in jedem dieser Eckpunkte gegeben. Gl. (3.41) wird mit $u_o = u_{r+1} = 0$, $u_r = u_m = 1$, $v_o = v_{s+1} = 0$, $v_s = v_n = 1$:

$$\bar{P} = \begin{bmatrix} P(0,0) & P(0,1) & P_v(0,0) & P_v(0,1) \\ P(1,0) & P(1,1) & P_v(1,0) & P_v(1,1) \\ P_u(0,0) & P_u(0,1) & P_{uv}(0,0) & P_{uv}(1,1) \\ P_u(1,0) & P_u(1,1) & P_{uv}(1,0) & P_{uv}(1,1) \end{bmatrix} \tag{3.42}$$

Gl.(3.37): $N = M = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \rightarrow N^{-1} = M^{-1} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$ (3.43)

Mit Gl. (3.43) werden die Vektoren der Interpolationsfunktionen

$U = (N^{-1} u) =$

$$\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} = \begin{bmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{bmatrix} = \begin{bmatrix} H_{0,3}(u) \\ H_{1,3}(u) \\ H_{2,3}(u) \\ H_{3,3}(u) \end{bmatrix} \tag{3.44}$$

$V = (M^{-1} v) = \ldots$. entsprechend Gl. (3.44).

und das bikubische Pflaster

$$Q(u,v) = \left[H_{i,3}(u)\right]^T \cdot \bar{P} \cdot \left[H_{j,3}(v)\right] \tag{3.45}$$

Die Polynome $H_{i,3}(p)$ sind mit den von Coons /1/ benutzten "Blending functions" identisch (Bild 3.16).

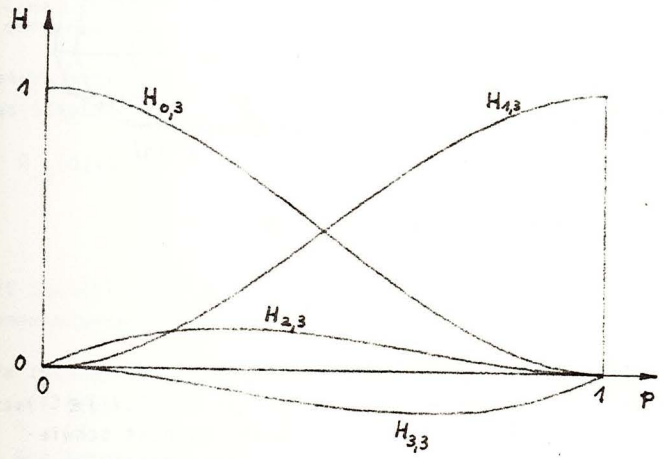


Bild 3.16: Hermitepolynome dritten Grades

Zahlenbeispiel des Tensors $\bar{P}$ für Bild 3.17:

$$\bar{P} = [X|Y] = \left[\begin{array}{cccc|cccc} -6 & 3 & 7 & -3 & 11 & 11 & -2 & 2 \\ 13 & 6 & -1 & 1 & -5 & -6 & -2 & 2 \\ -1 & 2 & 0 & 0 & -2 & -0{,}5 & 0 & 0 \\ 50 & -50 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right]$$

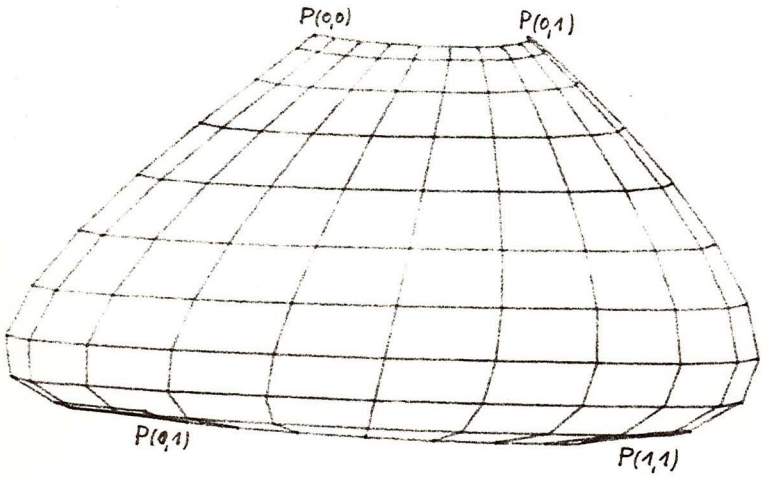


Bild 3.17: Bikubisches Pflaster

Die Hermite-Interpolation erlaubt prinzipiell das kantenfreie Aneinanderfügen von Pflastern ($Q(u',v') \in C^1$). In der Praxis erweist es sich jedoch als äußerst schwierig, die Tangentenvektoren anzugeben. (In /17/ ist eine Möglichkeit zur näherungsweisen Bestimmung der Tangentenvektoren dargestellt.) Die numerische Berechnung wird erschwert, weil die Beträge der Vektoren P, P_u und P_{uv} sich mitunter um Größenordnungen unterscheiden (siehe o.a. Zahlenbeispiel des Tensors P für Bild 3.17).
Die Hermite-Interpolation weist im übrigen die Nachteile der Lagrange-Interpolation auf.

3.3.3 Bikubische Splines /19/

Dieses Verfahren kann als ein Sonderfall der bikubischen Hermite-Interpolation nach Gl. (3.45) angesehen werden, bei dem die Tangentenvektoren nur auf den Randkurven der Gesamtfläche vorzugeben sind. Aus diesen Randbedingungen werden die Tangentenvektoren P_u, P_v, P_{uv} an den 4 Ecken der einzelnen Pflaster mit der Nebenbedingung

$$Q(u',v') \in C^2 \quad , \quad u' \in [u'_o, u'_m] \quad , \quad v' \in [v'_o, v'_n] \qquad (3.46)$$

(2-mal stetig differenzierbar)

rekursiv berechnet. Die Gesamtfläche $Q(u',v')$ wird aus bikubischen Pflastern

$$Q_{ij}(u,v) \quad , \quad u'_{i-1} \leq u' \leq u_i \quad , \quad v'_{j-1} \leq v' \leq v_j \qquad (3.47)$$

mit zweimal stetig differenzierbaren Obergängen zusammengesetzt.

Die Interpolation mit bikubischen Splines hat folgende wesentliche Vorteile:

1) Der Benutzer muß nur die Tangentenvektoren am Rand der Gesamtfläche angeben.

2) Die Gesamtfläche ist überall zweimal stetig differenzierbar.

3) Unabhängig von der Anzahl der vorgegebenen Bestimmungsstücke wird immer mit denselben Polynomen dritten Grades interpoliert.

Als Nachteile sind zu nennen:

1) Bei Veränderung einer Bestimmungsgröße ändert sich die Gesamtfläche.
2) Die Berechnung der Tangentenvektoren erfordert umfangreiche Matrizenoperationen, die beim interaktiven Verändern von großen Flächen die Antwortzeit des Graphics Systems sehr verlängert /19a/.

3.3.4 Gebrochen rationale kubische Funktionen

Wie schon in Abschnitt 3.2.1 dargestellt, werden bei diesem Verfahren homogene Koordinaten angewandt.

Gl. (3.1): $q = [xyzw] = w \cdot [XYZ1]$

$$q = w \cdot [Q\ 1] \tag{3.48}$$

Die üblichen Koordinaten erhält man durch Division nach Gl. (3.2):

$$\frac{q}{w} = [Q\ 1] = \left[\frac{x}{w}\ \frac{y}{w}\ \frac{z}{w}\ 1\right] \tag{3.49}$$

Die bikubischen Flächen werden damit

$$q(u,v) = U^T \cdot \overline{p} \cdot V \quad \text{und} \tag{3.50}$$

$$[Q(u,v)\ 1] = \frac{q(u,v)}{w(u,v)} = \frac{U^T \cdot \overline{p} \cdot V}{U^T \cdot p_w \cdot V}\ , \tag{3.51}$$

wobei U und V die Vektoren mit den entsprechenden Interpolationspolynomen sind. Werden die Hermitepolynome nach Gl.(3.44) gewählt, so erhält man die von Coons/20/ angewandte sog. "Endpoint Derivative Form":

$$q(u,v) = \left[F_0(u),\ F_1(u),\ G_0(u),\ G_1(u)\right] \cdot \overline{p} \cdot \begin{bmatrix} F_0(v) \\ F_1(v) \\ G_0(v) \\ G_1(v) \end{bmatrix} \tag{3.52}$$

mit $F_0(u) = H_{0,3}(u)$ $\quad G_0(u) = H_{2,3}(u)$

$F_1(u) = H_{1,3}(u)$ $\quad G_1(u) = H_{3,3}(u)$ (3.53)

und $\overline{p} = \left[\overline{P}\ p_w\right]$

Da üblicherweise nur $\overline{P}$ angegeben werden kann, müssen die homogenen Koordinaten p_w durch zusätzliche Angaben wie Punkte und Steigungen bestimmt werden. Dies bedeutet, daß gegenüber den anderen Interpolationsverfahren zusätzliche Freiheitsgrade bei der Flächenbestimmung vorhanden sind. Dieser Vorteil ist allerdings nur dem mathematisch versierten Benutzer zugänglich. Eine bemerkenswerte Eigenschaft der gebrochen rationalen Polynome ist ihre Invarianz gegenüber einer Reparametrisierung. Wird der Parameter u bilinear transformiert und durch

$$t = \frac{a \cdot u + b}{c \cdot u + d} \tag{3.54}$$

ersetzt, so bleibt die Form der u-Parameterlinien (v = konst) erhalten. In Bild 3.18 werden die Ergebnisse einer mehrmals parametrisierten Kurve übereinander dargestellt. Punkte gleichen Parameterwertes wurden miteinander verbunden. Man erkennt, daß durch Reparametrisierung ein gutes "Spacing" erreicht werden kann und bei einer Flächendarstellung durch Parameterlinien die Parametrisierung den optischen Eindruck der Fläche wesentlich bestimmt.

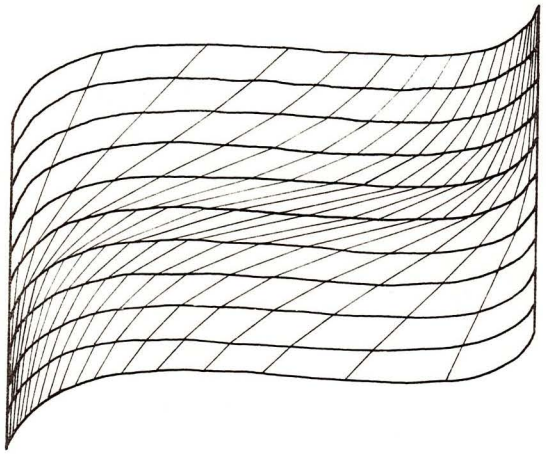


Bild 3.18: Familie invarianter Kurven /21/

Die Darstellung mit homogenen Koordinaten erfordert wegen der nach Gl. (3.51) durchzuführenden Division gegenüber den anderen Verfahren einen höheren Rechenaufwand. Im Hinblick auf eine schnelle Flächenberechnung mit Hardware-Unterstützung wird deshalb diese Darstellung im weiteren nicht mehr betrachtet.

3.4 Approximation

Die meisten in Computer Graphics angewandten Verfahren zur Flächenapproximation beruhen auf der Bézier-Approximation /8/. Zur Bestimmung des Pflasters müssen nur Punkte vorgegeben werden. Verbindet man sie entsprechend ihrer Indizierung geradlinig miteinander, so spannen sie ein räumliches Polygonnetz auf, dem in der u-v-Ebene ein Rechteckraster entspricht. Die das Polygonnetz definierenden Punkte liegen im allgemeinen nicht auf der Approximationsfläche und werden deshalb zur Unterscheidung als Stützstellen S_{ij}, $i = 0,1,..,m$, $j = 0,1,...,n$ bezeichnet. Die dazu korrespondierenden Flächenpunkte werden $P_{ij} = Q(u_i, v_j)$ genannt. Analog Gl.(3.15) wird das approximierende Pflaster nach

$$Q(u,v) = U^T \cdot \bar{S} \cdot V \qquad (3.55)$$

berechnet.

3.4.1 Bézier-Approximation

Bei dem von Bézier angegebenen Approximationsverfahren werden Kurven und Flächen durch Polygonseiten definiert /8/. Um alle Verfahren einheitlich darstellen zu können, wird in dieser Arbeit eine modifizierte Bézier-Approximation /22/ angewandt, die nicht die Polygonseiten sondern die Polygonstützstellen als Bestimmungsstücke enthält und nach Gl. (3.55) zu berechnen ist. Die Approximationsfunktionen

$$B_{i,m}(p) = \binom{m}{i} \cdot p^i (1-p)^{m-i} \quad , \; i = 0,1,...,m. \qquad (3.56)$$

dieser modifizierten Bézierapproximation sind identisch mit den Basisfunktionen der Bernstein-Polynome $B_m(f,p)$,

welche die Funktion f: (0,1)⟶R approximieren:

$$B_m(f;p) = \sum_{i=0}^{m} f(\tfrac{i}{m})\cdot\binom{m}{i}\cdot p^i\cdot(1-p)^{m-i},\quad m \geq 1. \qquad (3.57)$$

Deren Eigenschaften sind ausführlich in /23/ dargestellt. $B_{i,m}(p)$ ist aus der Wahrscheinlichkeitsrechnung als Binominalverteilung bekannt:

Werden m unabhängige Versuche durchgeführt und ist bei jedem dieser Versuche die Wahrscheinlichkeit des Ereignisses E gleich p, $o \leq p \leq 1$, so ist die Wahrscheinlichkeit für i-maliges Auftreten ($o \leq i \leq m$) von E gleich $B_{i,m}(p)$, $o \leq B_{i,m}(p) \leq 1$.

Hieraus folgt die Eigenschaft

$$\sum_{i=0}^{m} B_{i,m}(p) \equiv 1 \qquad (3.58)$$

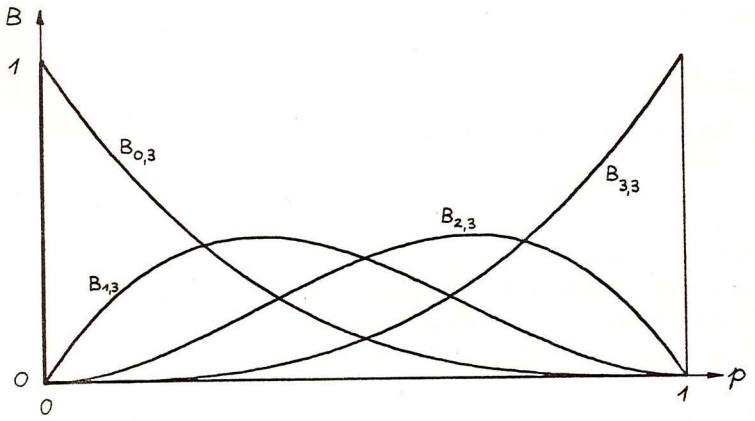


Bild 3.18a: Binomialverteilung $B_{i,3}(p)$

Gl. (3.57) läßt sich leicht auf Funktionen zweier Variabler ausdehnen. Betrachtet man das Polygonnetz $S = \{S_{ij},\ i = 0,1,\ldots m,\ j = 0,1,..n\}$ als Funktion zweier Variabler und schreibt

$$S_{ij} = S(\tfrac{i}{m}, \tfrac{j}{n}), \qquad (3.59)$$

so berechnet sich dessen Bernsteinapproximation

aus $$Q(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} B_{i,m}(u)\cdot S(\tfrac{i}{m},\tfrac{i}{n})\cdot B_{j,n}(v) \qquad (3.60)$$

$$= \left[B_{i,m}(u)\right]^T \cdot \bar{S}\cdot\left[B_{j,n}(v)\right]$$

Die Eigenschaften dieses Pflasters ergeben sich aus den Eigenschaften der Funktionen $B_{i,m}(p)$ /24/:

(3.58): $$\sum_{i=0}^{m} B_{i,m}(p) \equiv 1 \quad \text{für } p\in(0,1)$$

$$0 \leq B_{i,m}(p) \leq 1$$

$$B_{0,m}(0) = 1,\quad B^j_{0,m}(1) = 0$$
$$B_{m,m}(1) = 1,\quad B^j_{m,m}(0) = 0 \qquad 0 \leq j \leq m \qquad (3.61)$$

(Der obere Index gibt die Ableitung nach p an.)

Die Symmetrie bezüglich p = 0,5 ist aus Gl. (3.56) und Bild 3.18a zu erkennen:

$$B_{i,m}(p) = B_{m-1,m}(1-p) \qquad (3.62)$$

$$
\left.
\begin{aligned}
B^{j}_{i,m}(0) &= 0 \quad , \quad 0 \leq j < i \\
B^{i}_{i,m}(0) &= \frac{m!}{(m-i)!} \\
B^{j}_{i,m}(1) &= 0 \quad , \quad 0 \leq j < (m-i) \\
B^{(m-i)}_{i,m}(1) &= (-1)^{m-i}\,\frac{m!}{(m-i)!}
\end{aligned}
\right\} \qquad 1 \leq i \leq m-1 \qquad (3.63)
$$

$B^{1}_{i,m}(\frac{i}{m}) = 0$, $1 \leq i \leq (m-1)$ bedeutet: das Maximum liegt beim Parameterwert $p = \frac{i}{m}$. Die Stützstellen S_{ij} haben demnach bei $Q(\frac{i}{m},\frac{j}{n}) = P_{ij}$ den größten Einfluß auf die Form des Pflasters. Wegen

$$
\begin{aligned}
&B_{o,m}(0) = 1 \quad , \quad B_{o,m}(1) = 0 \quad \text{und} \\
&B_{m,m}(0) = 0 \quad , \quad B_{m,m}(1) = 1 \quad \text{folgt, daß}
\end{aligned} \qquad (3.64)
$$

allgemein nur die 4 Eckpunkte des Polygonnetzes auf dem Pflaster liegen

$$
\begin{aligned}
S(\tfrac{i}{m},\tfrac{j}{n}) &= Q(\tfrac{i}{m},\tfrac{j}{n}) = P_{ij} \quad \text{für} \\
i &= o,m \ , \ j = o,n
\end{aligned} \qquad (3.65)
$$

und die Randkurven des Pflasters nur von den Polygonen am Rand des Netzes abhängen:

$$
\begin{aligned}
Q(u,\tfrac{j}{n}) &= \left[B_{i,m}(u)\right]^{T} \cdot S(\tfrac{i}{m},\tfrac{j}{n}) \quad , \ j = o,n \\
\text{bzw. } Q(\tfrac{i}{m},v) &= S(\tfrac{i}{m},\tfrac{j}{n}) \cdot \left[B_{j,n}(v)\right] \quad , \ i = o,m
\end{aligned} \qquad (3.66)
$$

Für die Ableitungen in den 4 Eckpunkten gilt

$$
\left.\frac{\partial^q}{\partial u^q} Q(u,\tfrac{j}{n})\right|_{u=o} = \frac{m!}{(m-q)!} \cdot \sum_{i=o}^{q} (-1)^{q-i} \cdot \binom{q}{i} \cdot S(\tfrac{i}{m},\tfrac{j}{n}) \ , j=o,n \qquad (3.67)
$$

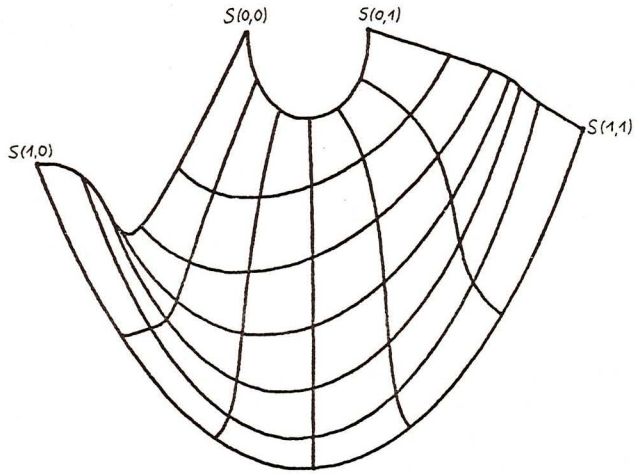
$$
\left.\frac{\partial^q}{\partial u^q} Q(u,\tfrac{j}{n})\right|_{u=1} = \frac{m!}{(m-q)!} \cdot \sum_{i=o}^{q} (-1)^{i} \cdot \binom{q}{i} \cdot S(\tfrac{m-i}{m},\tfrac{j}{n}) \ , j=o,n
$$

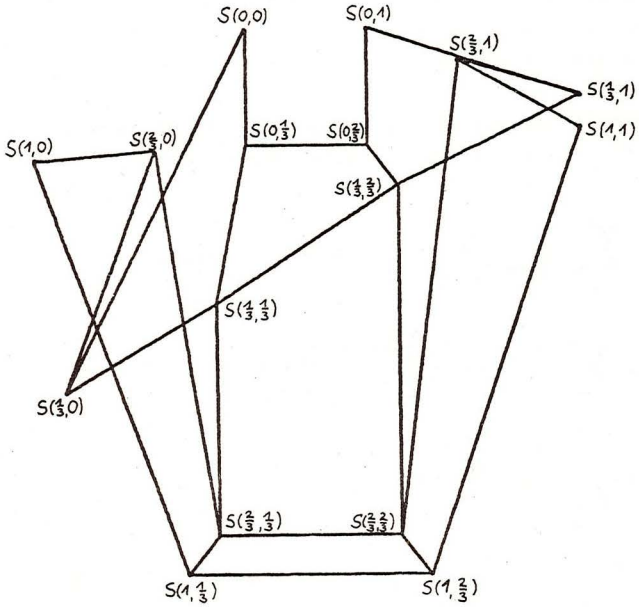
Die Ableitungen nach v werden analog berechnet. Gl.(3.67) besagt, daß die q-te Ableitung im Anfangspunkt $Q(0,\frac{j}{n})$ abhängig ist von den Stützstellen $S(0,\frac{j}{n})$ bis $S(\frac{q}{m},\frac{j}{n})$ und im Endpunkt $Q(1,\frac{j}{n})$ von den Stützstellen $S(\frac{m-i}{m},\frac{j}{n})$ bis $S(\frac{m}{m},\frac{j}{n})$ der Polygone am Rand des Netzes (j = o,n). Damit sind die Polygonseiten in den Ecken $Q(\frac{i}{m},\frac{j}{n})$, i = o,m, j = o,n die Tangenten an die Randkurven des Pflasters.

Für q = 1 erhält man aus Gl. (3.60) mit Gl. (3.67) eine Vorschrift für das kantenfreie Zusammenfügen zweier Pflaster:

$$
\begin{aligned}
Q_u(u,v) &= \sum_{i=o}^{1} \sum_{j=o}^{n} \frac{m!}{(m-1)!} \cdot (-1)^{1-i} \binom{1}{i} \cdot S(\tfrac{i}{m},\tfrac{j}{n}) \cdot B_{j,n}(v) \\
u=o \qquad &= m \cdot \sum_{i=o}^{1} (-1)^{1-i} \cdot \sum_{j=o}^{n} S(\tfrac{i}{m},\tfrac{j}{n}) \cdot B_{j,n}(v) \\
Q_u(o,v) &= m \cdot \sum_{j=o}^{n} \left(S(\tfrac{1}{m},\tfrac{j}{n}) - S(0,\tfrac{j}{n})\right) \cdot B_{j,n}(v) \qquad (3.68) \\
Q_u(1,v) &= m \cdot \sum_{j=o}^{n} \left(S(1,\tfrac{j}{n}) - S(\tfrac{m-1}{m},\tfrac{j}{n})\right) \cdot B_{j,n}(v) \qquad (3.69)
\end{aligned}
$$

Beispiel 5: $Q(u,v) = \left[B_{i,3}(u)\right]^T \cdot \bar{S} \cdot B_{j,3}(v)$





Zwei Pflaster können kantenfrei aneinandergesetzt werden, wenn $\underset{\sim}{Q}_u(o,v) = C \cdot \underset{\sim}{Q}_u(1,v)$, $C > 0$, (3.70)
also wenn die beiden Tangentenvektoren kollinear sind.

Gl.(3.68) und Gl.(3.69) in Gl.(3.70) ergibt

$$m_{II} \sum_{j=o}^{n_{II}} \left(S_{II}\left(\frac{1}{m_{II}}, \frac{j}{n_{II}}\right) - S_{II}\left(0, \frac{j}{n_{II}}\right)\right) \cdot B_{j,n_{II}}(v)$$

$$\overset{!}{=} C \cdot m_I \sum_{j=o}^{n_I} \left(S_I\left(1, \frac{j}{n_I}\right) - S_I\left(\frac{m_I - 1}{m_I}, \frac{j}{n_I}\right)\right) \cdot B_{j,n_I}(v)$$

Wegen der Übereinstimmung der Randkurven ist

$n_I = n_{II} = n$ und $S_{II}(0, \frac{j}{n}) = S_I(1, \frac{j}{n}) = S(\frac{j}{n})$.

Die obige Bedingung ist dann erfüllt, wenn für alle j gilt:

$$S_{II}\left(\frac{1}{m_{II}}, \frac{j}{n}\right) - S\left(\frac{j}{n}\right) = C'\left(S\left(\frac{j}{n}\right) - S_I\left(\frac{m_I - 1}{m_I}, \frac{j}{n}\right)\right) \qquad (3.71)$$

(siehe Bild 3.20).

In Gl.(3.67) wurden die Ableitungen in den Ecken des Pflasters als Funktion der Randstützstellen angegeben. Umgekehrt kann man zu einer Fläche, die sich in der Form

$$Q(u,v) = \sum_{i=o}^{m} \sum_{j=o}^{n} a_{ij} \cdot u^i \cdot v^j \qquad (3.72)$$

darstellen läßt, die Randstützstellen als Funktion der

Ableitungen in den Ecken berechnen:

$$S\left(\frac{i}{m},\frac{j}{n}\right) = \sum_{q=o}^{i} \binom{i}{q} \cdot \frac{(m-q)!}{m!} \cdot \frac{\partial^q}{\partial u^q} \; Q(u,\frac{j}{n})\Big|_{u=0}, \quad j=o,n \tag{3.73}$$

$$= \sum_{q=o}^{m-i} (-1)^q \binom{m-i}{q} \frac{(m-q)!}{m!} \cdot \frac{\partial^q}{\partial u^q} \; Q\,(1,\frac{j}{n})\Big|_{u=1}, j=o,n$$

Die restlichen (m-1)(n-1) Stützstellen des Polygonnetzes lassen sich durch Koeffizientenvergleich zwischen Gl.(3.72) und Gl.(3.60) berechnen.

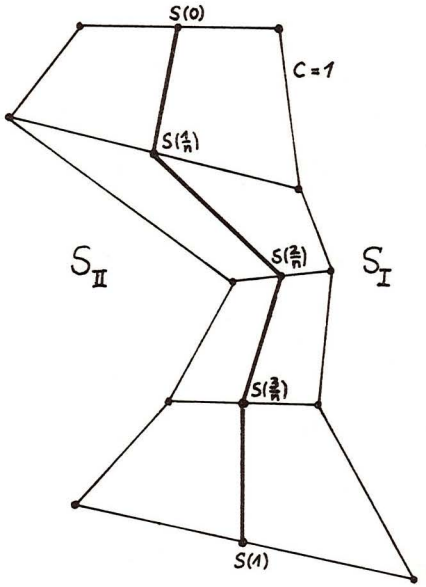


Bild 3.20: Kantenfreies Aneinanderfügen zweier Bézierflächen

Auf Grund der beiden Eigenschaften

$$\sum_{i=o}^{m} B_{i,m}(u) \equiv 1 \quad \text{und} \quad 0 \leq B_{i,m}(u) \leq 1 \;, \; u \in (0,1)$$

der Approximationsfunktionen ist jeder Punkt der Bézierfläche Q(u,v) eine konvexe Linearkombination der Stützstellen $S_{i,j}$. Die Fläche muß deshalb innerhalb der konvexen Hülle der Extrempunkte des Polygonnetzes liegen (siehe Bild 3.21).

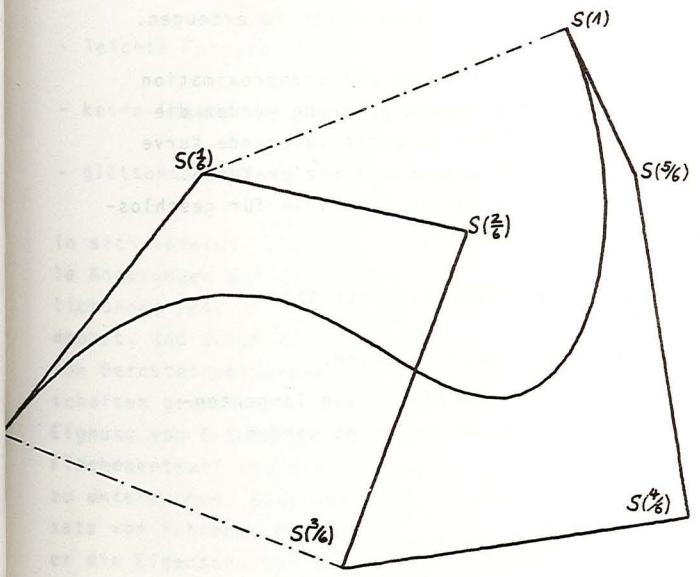


Bild 3.21: Bézierkurve liegt innerhalb der konvexen Hülle der Stützstellen

Die Bézierfläche gibt ungefähr die Form des Polygonnetzes wieder. Unerwünschte Wellen oder Buckel treten wegen der glättenden Eigenschaften der Bernsteinapproximation /25/ nicht auf.

Mit Béziers Methode wird das Konstruieren von Kurven und Flächen einfach. Der Benutzer braucht sich nicht um Ableitungen nach dem Parameter zu kümmern. Mit den von ihm eingegebenen Stützstellen kontrolliert er direkt die Flächenform. Die Implementierung eines Programms zum Entwurf von 3D-Flächen /26/ auf dem AGT 130 der TUB hat gezeigt, daß der Benutzer bereits nach kurzer Übung in der Lage ist, den von ihm gewünschten Kurven- und Flächenverlauf durch Polygoneingabe zu erzeugen.

Bild 3.22 zeigt die schrittweise Bézierapproximation einer Freihandkurve. Zur ersten Näherung werden die Polygonstützstellen auf die zu approximierende Kurve gelegt. Dies liefert die gepunktete Bézier-Kurve.
Die Bilder 3.23 und 3.24 zeigen Beispiele für geschlossene und offene Bézier-Kurven.

Die Vorteile der Bézier-Approximation sind:

1) Einfache Definition durch Polygone.
2) Pflaster können ohne die Angabe von Tangentenvektoren kantenfrei zusammengefügt werden.
3) Glättende Wirkung.

Als Nachteile sind zu nennen:

1) Der Grad der Fläche und somit auch der Rechenaufwand ist abhängig von der Anzahl der vorgegebenen Stützstellen.
2) Bei Veränderung einer Stützstelle ändert sich die gesamte Fläche (ohne Rand). Lokale Änderungen sind nicht möglich.

Die Bewertung der bisher betrachteten Verfahren legt den Gedanken nahe, nach einer Approximation zu suchen, die die Vorteile der Splines

- Segmentweise definiertes Polynom
- gute Stetigkeit an den Segmentübergängen
- Erhöhung des Freiheitsgrades ohne Erhöhung des Polynomgrades

und die der Bezier-Approximation

- leichte Formgebung durch die Poygondefinition
- keine Angabe von Tangentenvektoren notwendig
- glättende Wirkung

in sich vereint. Zusätzlich sollte das Verfahren lokale Änderungen auf der Fläche zulassen. In den Veröffentlichungen /24/,/27/ Mitte des Jahres 1972 wurde angedeutet, daß durch die Anwendung von B-Splines anstelle von Bernsteinpolynomen ein Verfahren mit den o.g. Eigenschaften gewonnen werden könne. Dies gab mir Anlaß, die Eignung von B-Splines für den interaktiven Kurven- und Flächenentwurf und die Hardware-Realisierung eingehend zu untersuchen. Eine wesentliche Hilfe war dabei der Aufsatz von Schoenberg /28/ "On Spline Functions", in dem er die Eigenschaften der B-Splines mit denen der Bernsteinpolynome vergleicht.

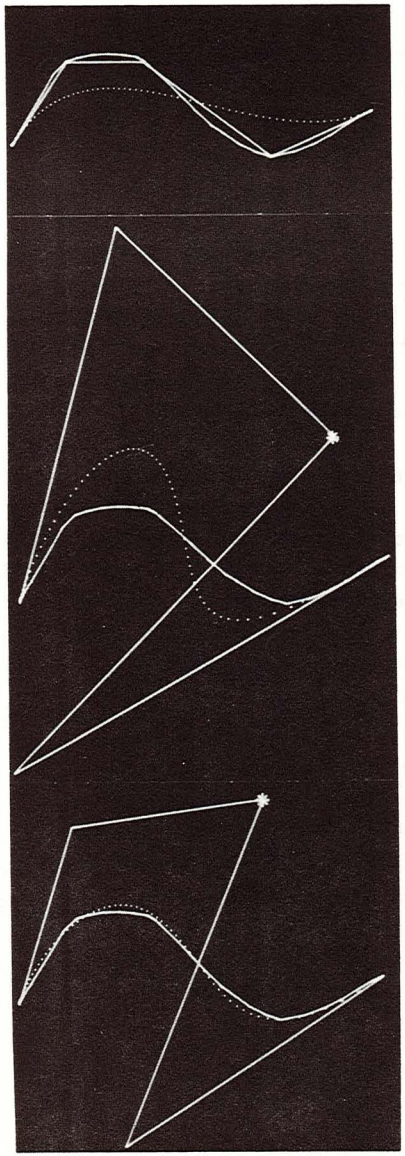


Bild 3.22: Schrittweise Approximation einer Frei-

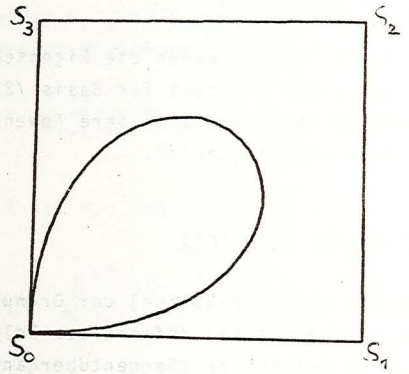


Bild 3.23: Geschlossene Bézier-Kurve

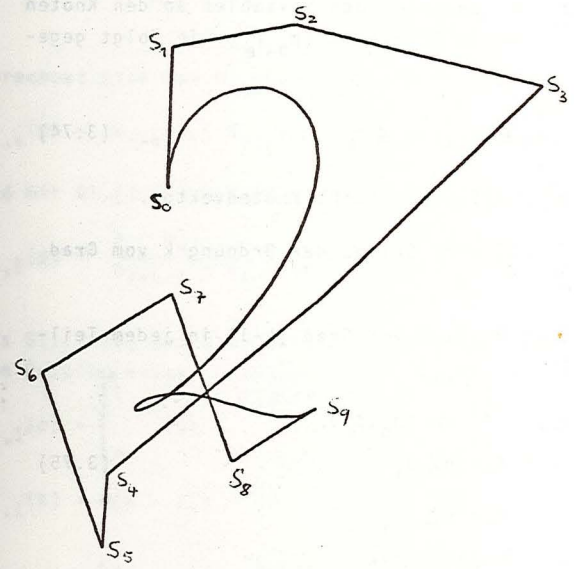


Bild 3.24: Offene Bézier-Kurve

3.4.2 B-Spline-Approximation /29/

Im ersten Teil dieses Kapitels werden die Eigenschaften der B-Spline-Funktionen (B steht für Basis /28/) dargestellt. Im zweiten Teil wird dann ihre Anwendung im Kurven- und Flächenentwurf gezeigt.

3.4.2.1 Eigenschaften der B-Splines

Eine B-Spline-Funktion (kurz: B-Spline) der Ordnung k ist ein stückweise (segmentweise) definiertes Polynom vom Grad (k-1), das an den Knoten (Segmentübergängen) (k-2)-mal stetig differenzierbar ist. k zusammenhängende Segmente sind von Null verschieden(siehe Bild 3.25).

Die Werte p_i der unabhängigen Variablen an den Knoten des Spline seien im Intervall (p_o, p_e) wie folgt gegeben:

$$p_o \leq p_1 \leq \ldots p_i \leq p_{i+1} \ldots \leq p_e \qquad (3.74)$$

$\mathcal{P} = (p_o, p_1, \ldots, p_i \ldots, p_e)$ heißt Knotenvektor.

Dann ist $M_{i,k}(p)$ ein Spline der Ordnung k vom Grad (k-1), wenn gilt

a) $M_{i,k}(p)$ ist Polynom vom Grad (k-1) in jedem Teilintervall (p_i, p_{i+1})

b) $M_{i,k}(p) \in C^{(k-2)}$ in (p_o, p_e).

c) $M_{i,k}(p) = 0$ für $p < p_i$ (3.75)

$M_{i,k}(p) = 0$ für $p \geq p_{i+k}$

oder anders ausgedrückt

$M_{i,k}(p) \neq 0$ für $p_i \leq p < p_{i+k}$ (3.76)

d.h.: der Index i gibt den Knoten an, bis zu welchem der Spline noch den Wert Null besitzt.

Der B-Spline $M_{i,k}(p)$ ist nach de Boor /30/ rekursiv aus folgender Beziehung zu berechnen:

$$M_{i,k}(p) = \frac{p-p_i}{p_{i+k}-p_i} M_{i,k-1}(p) + \frac{p_{i+k}-p}{p_{i+k}-p_i} M_{i+1,k-1}(p) \qquad (3.77)$$

Für k = 1 ist $M_{i,k}(p)$ definiert als

$$M_{i,1}(p) = \begin{cases} (p_{i+1} - p_i)^{-1} & \text{für } p_i \leq p \leq p_{i+1} \\ 0 & \text{sonst} \end{cases} \qquad (3.78)$$

Der normalisierte B-Spline $N_{i,k}(p)$ mit der

Eigenschaft $\sum_i N_{i,k}(p) \equiv 1$ (3.79)

errechnet sich aus $M_{i,k}(p)$ nach der Formel /30/

$$N_{i,k}(p) = (p_{i+k}-p_i)\, M_{i,k}(p) \qquad (3.80)$$

und mit Gl.(3.77) ergibt sich die Rekursionsformel

$$N_{i,k}(p) = \frac{p - p_i}{p_{i+k-1} - p_i} N_{i,k-1}(p) + \frac{p_{i+k} - p}{p_{i+k} - p_{i+1}} N_{i+1,k-1}(p) \qquad (3.81)$$

Aus Gl.(3.78) und Gl.(3.80) folgt für den normalisierten B-Spline erster Ordnung

$$N_{i,1}(p) = \begin{cases} 1 & \text{für } p_i \leq p < p_{i+1} \\ 0 & \text{sonst} \end{cases} \qquad (3.82)$$

$N_{i,1}(p) = H(p - p_i) - H(p - p_{i+1})$, wobei

$$H(p) = \begin{cases} 1 & \text{für } p \geq 0 \\ 0 & p < 0 \end{cases}$$ die Sprungfunktion ist.

In Bild 3.25 sind die Verläufe der B-Splines für k=1 bis k=4 dargestellt.

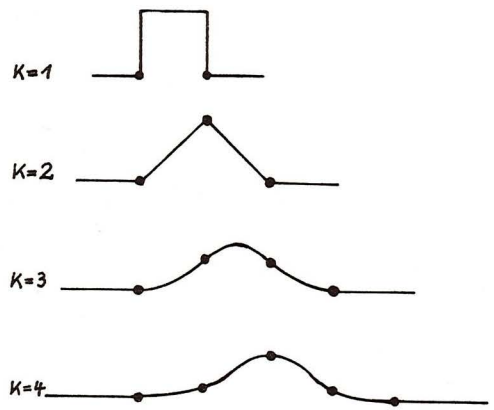


Bild 3.25: B-Splines für k = 1 bis k = 4

Gl.(3.78) bzw. (3.82) impliziert, daß bei zusammenfallenden Knoten $p_i = p_{i+1}$

die Splines $M_{i,1}(p) \equiv N_{i,1}(p) \equiv 0$ sind. (3.83)

Fallen l Knoten zusammen, d.h. ist

$$p_i = p_{i+1} = \ldots = p_{i+(l-1)} \qquad l \in 1,2,\ldots,$$

dann reduziert sich die stetige Differenzierbarkeit der Funktion $M_{i,k}(p)$ an der Stelle p_i von $C^{(k-2)}$ auf $C^{(k-l-1)}$. Ein weiterer Effekt ist, daß der B-Spline nur noch aus k-l+1 Segmenten, die verschieden von Null sind, besteht.

Bild 3.26 zeigt für k = 3 alle Möglichkeiten für das Zusammenfallen von Knoten.

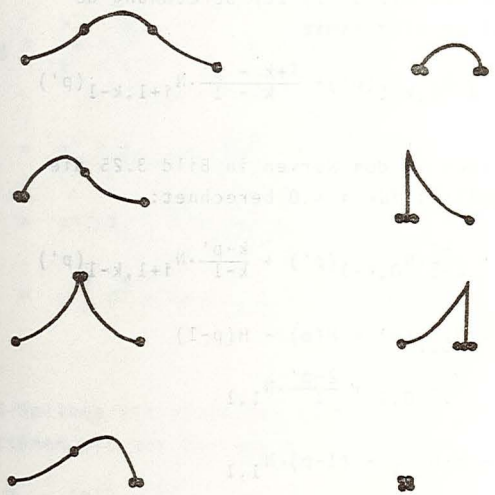


Bild 3.26: k = 3 Zusammenfallen von Knoten

Das Arbeiten mit B-Splines wird wesentlich vereinfacht, wenn die Knoten p_i nach Gl. (3.74) positive ganze Zahlen sind. Zu diesem Zweck werden neue Koordinaten eingeführt:

$$p'_i = i \quad , \quad i \in 0,1,2,\dots,e \qquad (3.84)$$

Zwischen den Knoten soll gelten

$$0 \leq p \leq 1$$

also $\qquad p \equiv p' \pmod 1$

Die Rekursionsformel Gl.(3.81) zur Berechnung de normierten B-Spline wird damit

$$N_{i,k}(p') = \frac{p' - i}{k - 1} \cdot N_{i,k-1}(p') + \frac{i+k - p'}{k - 1} \cdot N_{i+1,k-1}(p') \qquad (3.85)$$

Als Beispiel werden zu den Kurven in Bild 3.25 die zugehörigen B-Splines für i = 0 berechnet:

$$N_{0,k}(p') = \frac{p'}{k-1} \cdot N_{0,k-1}(p') + \frac{k-p'}{k-1} \cdot N_{i+1,k-1}(p') \qquad (3.86)$$

$$k=1 : N_{0,1}(p') = N_{0,1}(p) = H(p) - H(p-1)$$

$$k=2 : N_{0,2}(p') = \frac{p'}{1} \cdot N_{0,1} + \frac{2-p'}{1} \cdot N_{1,1}$$

$$= p \cdot N_{0,1} + (1-p) \cdot N_{1,1}$$

$$k=3 : N_{0,3}(p') = \frac{p'}{2} \cdot N_{0,2} + \frac{3-p'}{2} \cdot N_{1,2}$$

$$= \frac{p'}{2}\left(p \cdot N_{0,1} + (1-p) \cdot N_{1,1}\right)$$

$$+ \frac{3-p'}{2}\left(p \cdot N_{1,1} + (1-p) \cdot N_{2,1}\right)$$

$$N_{0,3}(p') = \frac{p^2}{2} \cdot N_{0,1} + \left(-p^2 + p + \frac{1}{2}\right) \cdot N_{1,1} + \frac{1}{2}(1-p)^2 \cdot N_{2,1}$$

$$k=4 : N_{0,4}(p') = \frac{p^3}{6} \cdot N_{0,1} + \left(-\frac{1}{2} p^3 + \frac{1}{2} p^2 + \frac{1}{2} p + \frac{1}{6}\right) \cdot N_{1,1}$$

Die B-Splines mit zusammenfallenden Knoten berechnet man nach Gl. (3.81)

N' bedeutet 2-fach Knoten

N'' bedeutet 3-fach Knoten etc.

Beispiel: k = 3 , i = 0 mit dem Knotenvektor

$$\mathcal{P} = \left(\underset{0}{p'_0 = p'_1} \; ; \; \underset{1}{p'_2} \; ; \; \underset{2}{p'_3}\right) \quad \text{(siehe Bild 3.26)}$$

$$N'_{0,3} = \frac{p' - p'_0}{p' - p'_0} \cdot N_{0,2} + \frac{p'_3 - p'}{p'_3 - p'_1} \cdot N_{1,2}$$

$$= p' \cdot N_{0,2} + \frac{2-p'}{2} \cdot N_{1,2}$$

$$= p' \cdot (pN_{0,1} + (1-p) \cdot N_{1,1}) + \frac{2-p'}{2} \cdot (pN_{1,1} + (1-p)N_{2,1})$$

$$= \left(-\frac{3}{2} p^2 + 2p\right) \cdot N_{1,1} + \left(\frac{p^2}{2} - p + \frac{1}{2}\right) \cdot N_{2,1}$$

Die B-Splines mit einfachen Knoten sind periodische Funktionen mit der Periode k:

$$N_{i,k}(p') = N_{i+k,k}(p') = N_{(i+k) \bmod e,k}(p') \qquad (3.87)$$

Bild 3.27 zeigt die Familie der periodischen B-Splines für

$$e = 4 \text{ und } k = 3 : N_{i,3} \Big|_{i=0}^{3}$$

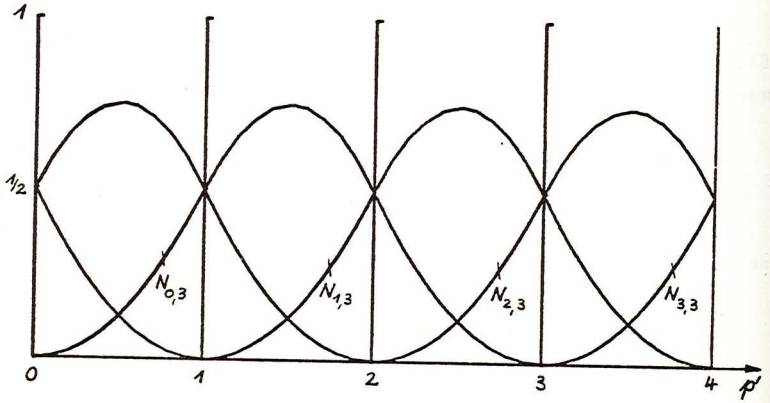


Bild 3.27: Periodische B-Splines $N_{i,3}\Big|_{i=0}^{3}$

Der zugehörige Knotenvektor

$$\mathcal{P} = (p'_0, p'_1, p'_2, p'_3, p'_4 = p'_0) = (P'_{i \bmod 4})$$

läßt sich als geschlossener Kreis darstellen:

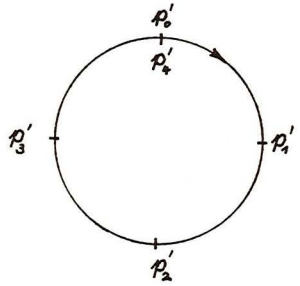


Bild 3.28: "Periodischer" Knotenvektor

B-Splines mit mehrfachen Knoten sind dagegen keine periodischen Funktionen. Dies erkennt man aus Bild 3.29, das die Familie der nichtperiodischen B-Splines mit e = 8, k = 3 und k-fachem Knoten am Anfang und Ende des Intervalls (0, e) zeigt.

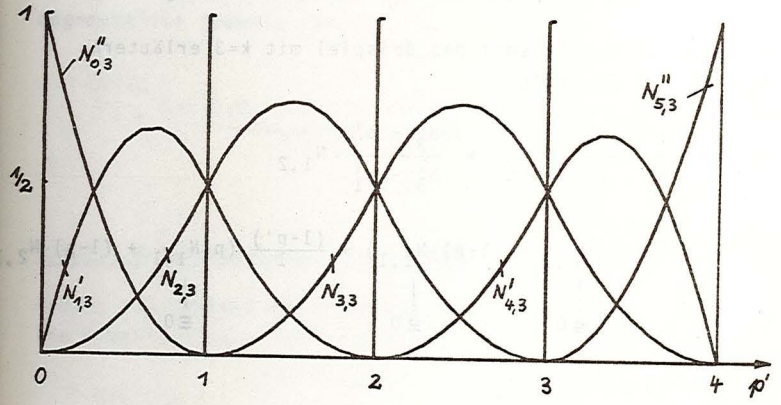


Bild 3.29: Nichtperiodische B-Splines $N''_{i,k}\Big|_{i=0}^{5}$

mit $\mathcal{P} = (p'_0 = p'_1 = p'_2, p'_3, p'_4, p'_5, p'_6 = p'_7 = p'_8)$

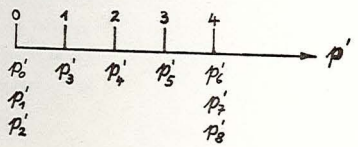
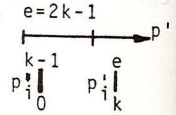


Bild 3.29a:"Nichtperiodischer" Knotenvektor

Ein interessanter Sonderfall entsteht, wenn der Knotenvektor Φ nur aus k-fachem Anfangs- und Endknoten besteht, also

$$\Phi = \underbrace{(p'_o = p'_1 = \ldots p'_{k-1}}_{\text{k-facher Knoten}}, \underbrace{p'_k = \ldots = p'_e)}_{k}, \quad e=2k-1$$



Was dies bedeutet, soll das Beispiel mit k=3 erläutern. Nach Gl. (3.81) gilt:

$$N''_{0,3} = \frac{p' - p_o}{p'_2 - p'_o} \cdot N_{0,2} + \frac{p'_3 - p'}{p'_3 - p'_1} \cdot N_{1,2}$$

$$= \frac{p'}{0} \cdot (p \cdot N_{0,1} + (1-p) \cdot N_{1,1}) + \frac{(1-p')}{1} \cdot (p \cdot N_{1,1} + (1-p) \cdot N_{2,1})$$

$\equiv 0$ $\equiv 0$ $\equiv 0$

$$N''_{0,3} = (1-p)^2 N_{2,1}$$

$$N'_{1,3} = \frac{p' - p'_1}{p'_3 - p'_1} \cdot N_{1,2} + \frac{p'_4 - p'}{p'_4 - p'_2} \cdot N_{2,2}$$

$$= p' \cdot (p \cdot N_{1,1} + (1-p) \cdot N_{2,1}) + \frac{(1-p')}{1} \cdot (p \cdot N_{2,1} + (1-p) \cdot N_{3,1})$$

$\equiv 0$ $\equiv 0$

$$N'_{1,3} = 2p(1-p)N_{2,1}$$

Aus Symmetriegründen ist

$$N''_{2,3}(p) = N'_{0,3}(1-p) = p^2 \cdot N_{2,1}$$

Dieses Ergebnis erhält man auch aus der Formel für die Bernsteinpolynome 2-ten Grades:

$$B_{i,2}(p) = \binom{2}{i} \cdot p^i \cdot (1-p)^{2-i}$$

Satz: B-Splines der Ordnung k sind identisch mit Bernsteinpolynomen vom Grad (k-1), falls die B-Spline aus einem Segment besteht, das von zwei k-fachen Knoten begrenzt ist (Beweis /28/):

$$N^{(k-1)}_{i,k}(p; \underbrace{0,0, \ldots, 0}_{k}, \underbrace{1, 1, \ldots, 1}_{k}) \equiv B_{i,k-1}(p) \qquad (3.88)$$

3.4.2.2 B-Spline-Kurven und -Flächen

Analog zur Definition der Bézierfläche Gl. (3.60) wird die B-Spline-Fläche definiert als

$$Q(u', v') = \sum_{i=o}^{m} \sum_{j=o}^{n} N_{i,k}(u') \cdot S_{ij} \cdot N_{j,l}(v') \qquad (3.89)$$

$$= \left[N_{i,k}(u')\right]^T \cdot \bar{S} \cdot \left[N_{j,l}(v')\right]$$

wobei der Tensor $\bar{S}$ wieder die Koordinaten der Stützstellen des Polygonnetzes enthält.

Im Unterschied zur Bézierapproximation, bei welcher der Grad der Fläche von der Anzahl der vorgegebenen Stützstellen bestimmt wird, kann der Benutzer bei der B-Spline-Approximation den Grad (k-1), (l-1) der Fläche selbst wählen aus

$$2 \leq k \leq m+1 \text{ bzw.}$$
$$2 \leq l \leq n+1 \qquad (3.90)$$

Zusätzlich kann zwischen periodischen und nicht-periodischen B-Splines gewählt werden:

a) Approximation mit nichtperiodischen B-Splines

Durch Vergleich von Bild 3.29 mit Bild 3.18 ist zu erkennen, daß sich die nichtperiodischen B-Splines und die Bernsteinpolynome am Anfang und Ende des Definitionsbereichs gleich verhalten. Insbesondere gilt analog zu Gl. (3.64):

$$\begin{aligned} {}^{(k-1)}N'_{o,k}(0) &= 1 \quad , \quad {}^{(k-1)}N'_{o,k}(1) = 0 \\ {}^{(k-1)}N'_{m,k}(0) &= 0 \quad , \quad {}^{(k-1)}N'_{m,k}(1) = 1 \end{aligned} \tag{3.91}$$

($(k-1)$, bedeutet: k-facher Knoten)

Vorerst werden Kurven betrachtet:

aa) Kurvenapproximation

$$Q(u') = \sum_{i=o}^{m} N_{i,k}(u') \cdot S_i \tag{3.92}$$

$$= \left[N_{i,k}(u')\right]^T \cdot \bar{S}$$

Aus Gl. (3.91) folgt, daß Anfangs- und Endpunkte von Kurve und Polygon zusammenfallen:

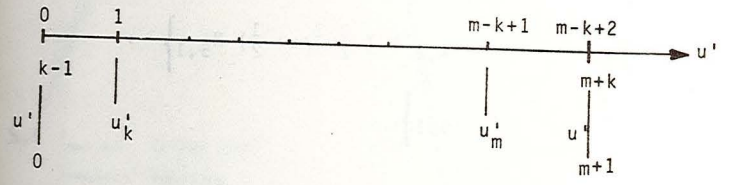
$$S_o = Q(0) \quad , \quad S_m = Q(u'_{m+k}) \tag{3.93}$$

Soll ein Polygon aus (m+1) Stützstellen S_i mit

B-Splines k-ter Ordnung $N_{i,k}(u') \in C^{k-2}$ approximiert werden, so muß der Knotenvektor folgende Form besitzen:

$$\mathcal{U} = (\underbrace{u'_o = u'_1 = \ldots = u'_{k-1}}_{\text{k-fach}}, u'_k, u'_{k+1}, \ldots u'_m, \underbrace{u'_{m+1} = u'_{m+2} = \ldots u'_{m+k}}_{\text{k-fach}}) \tag{3.94}$$

$u'_e = u'_{m+k}$



Die B-Spline-Kurve ist dann überall (k-2) mal stetig differenzierbar. Sie setzt sich aus Kurvensegmenten

$$Q_b(u') = \sum_{i=b-1}^{b+k-2} N_{i,k}(u')\, S_i \quad , \quad \begin{matrix} 1 \leq b \leq m-k+2 \\ b-1 \leq u' \leq b \end{matrix}$$

zusammen.

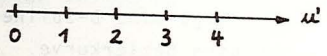
Die Anzahl B der Kurvensegmente ist gegeben durch

$$B = m-k+2 \tag{3.95}$$

Beispiel 6: $m = 5$, $k = 3$ (siehe auch Bild 3.29)

$$Q(u') = \sum_{i=0}^{5} N_{i,3}(u') \cdot S_i$$

$$\mathcal{U} = (u'_o = u'_1 = u'_2, u'_3, u'_4, u'_5, u'_6 = u'_7 = u'_8)$$



$$Q(u') = S_o N''_{0,3} + S_1 N'_{1,3} + S_2 N_{2,3} + S_3 N_{3,3} + S_4 N'_{4,3} + S_5 N''_{5,3}$$

$$\begin{aligned} Q(u') = {} & S_o(u^2 - 2u + 1)\, N_{2,1} + \\ & S_1 \cdot \left\{ (-\tfrac{3}{2}u^2 + 2u) \cdot N_{2,1} + (\tfrac{u^2}{2} - u + \tfrac{1}{2}) \cdot N_{3,1} \right\} + \\ & S_2 \cdot \left\{ (u^2/2) \cdot N_{2,1} + (-u^2 + u + \tfrac{1}{2}) \cdot N_{3,1} + (\tfrac{1}{2}u^2 - u + \tfrac{1}{2}) \right. \\ & S_3 \cdot \left\{ (u^2/2) \cdot N_{3,1} + (-u^2 + u + \tfrac{1}{2}) \cdot N_{4,1} + (\tfrac{1}{2}u^2 - u + \tfrac{1}{2}) \right. \\ & S_4 \cdot \left\{ (u^2/2) \cdot N_{4,1} + (-\tfrac{3}{2}u^2 + u + \tfrac{1}{2}) \cdot N_{5,1} \right\} + \\ & S_5 \cdot \left\{ (u^2) \cdot N_{5,1} \right\} \end{aligned}$$

oder anders geschrieben:

$$\begin{aligned} Q(u') = {} & \left\{ S_o \cdot (u^2 - 2u + 1) + S_1 \cdot (-\tfrac{3}{2}u^2 + 2u) + S_2 \cdot (\tfrac{u^2}{2}) \right\} \cdot N_{2,1} + \\ & \left\{ S_1 \cdot (\tfrac{u^2}{2} - u + \tfrac{1}{2}) + S_2 \cdot (-u^2 + u + \tfrac{1}{2}) + S_3 \cdot (\tfrac{u^2}{2}) \right\} \cdot N_{3,1} + \\ & \left\{ S_2 \cdot (\tfrac{u^2}{2} - u + \tfrac{1}{2}) + S_3 \cdot (-u^2 + u + \tfrac{1}{2}) + S_4 \cdot (\tfrac{u^2}{2}) \right\} \cdot N_{4,1} + \\ & \left\{ S_3 \cdot (\tfrac{u^2}{2} - u + \tfrac{1}{2}) + S_4 \cdot (-\tfrac{3}{2}u^2 + u + \tfrac{1}{2}) + S_5 \cdot (u^2) \right\} \cdot N_{5,1} \end{aligned}$$

Die Kurve besteht aus 4 Segmenten. Eine Stützstelle tritt höchstens in 3 Segmenten auf, woraus sich die Lokalität ihr Einflusses ergibt. Das erste (letzte) Segment bestimmt das Verhalten der Kurve bei S_o (S_m). Für den Sonderfall, daß da Polygon k Stützstellen besitzt, erhält man wieder die Bézie kurve. Bild 3.30 zeigt eine B-Spline-Kurve mit m = 5 , k = und die entsprechende Bézierkurve.

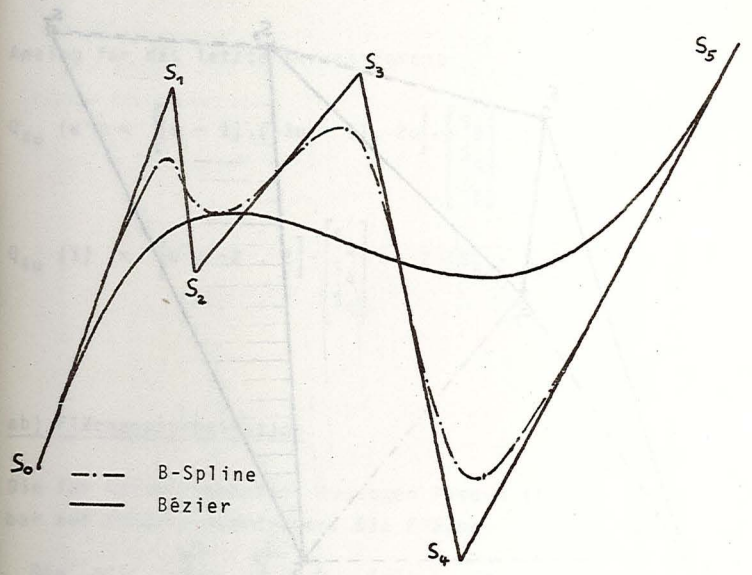


Bild 3.30: Bézier- und B-Spline-Kurve für m = 5, k = 3

Auf Grund der beiden Eigenschaften

$$\sum_i N_{i,k}(u) \equiv 1 \quad \text{und} \quad o \leq N_{i,k}(u) \leq 1 \;,\; u \in (0,1)$$

liegt jedes Kurvensegment Q_b innerhalb der konvexen Hülle der Stützstellen $S_i \Big|_{i=b-1}^{b+k-2}$. Die B-Spline-Kurve gibt damit den Verlauf des Polygons wesentlich exakter wieder als die Bézierapproximation. Bild 3.31 zeigt zu einem Polygon die konvexen Hüllen für Bézier- und B-Spline-Kurve.

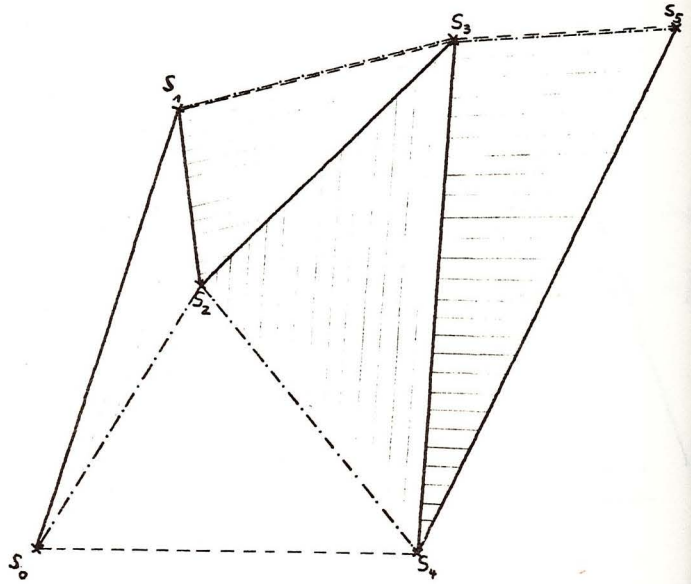


Bild 3.31: Konvexe Hüllen für Bézier- und B-Spline-Kurve (m = 5, k = 3). Für k = m + 1 sind beide identis

Aus Gln. (3.88) und (3.93) ergibt sich, daß die erste und letzte Polygonseite Tangenten im Anfangs- und Endpunkt der Kurve sind. Dies läßt sich durch Differenzieren von Gl. (3.96) verifizieren:

$$\frac{\partial}{\partial u}\cdot Q_1(u') = Q_{1u}(u') = \left[(2u-2)\ ,\ (-3u+2)\ ,\ u\right]\cdot\begin{bmatrix}S_0\\S_1\\S_2\end{bmatrix}$$

$$Q_{1u}(0) = \left[(-2)\ ,\ 2\ ,\ 0\right]\cdot\begin{bmatrix}S_0\\S_1\\S_2\end{bmatrix} = 2\cdot(S_1 - S_0)\ ,$$

Analog für das letzte Kurvensegment:

$$Q_{4u}(u') = \left[(u-1),(-3u+1),\ 2u\right]\cdot\begin{bmatrix}S_3\\S_4\\S_5\end{bmatrix} =$$

$$Q_{4u}(1) = \left[0\ ,\ -2\ ,\ 2\right]\cdot\begin{bmatrix}S_3\\S_4\\S_5\end{bmatrix} = 2\cdot(S_5 - S_4)$$

ab) Flächenapproximation

Die für Kurven gemachten Aussagen lassen sich unmittelbar auf Flächen übertragen. Die Fläche

$Q(u',v') = \sum_{i=o}^{m}\ \sum_{j=o}^{n} N_{i,k}(u')\cdot S_{ij}\cdot N_{j,l}(v')$ setzt sich aus Pflastern

$$Q_{bc}(u',v') = \sum_{i=b-1}^{b+k-2}\ \sum_{j=c-1}^{c+l-2} N_{i,k}(u')\cdot S_{ij}\cdot N_{j,l}(v') \qquad (3.97)$$

$1 \leq b \leq m - k + 2\ ,\ b - 1 \leq u' \leq b$

$1 \leq c \leq n - l + 2\ ,\ c - 1 \leq v' \leq c$ zusammen.

Die Anzahl der Pflaster ist

$$BC = (m - k + 2)\cdot(n - l + 2) \qquad (3.98)$$

Wird eine Stützstelle S_{ij} verändert, ändert sich im Gegensatz zur Bézierapproximation nicht die gesamte Fläche, sondern höchstens k · l Pflaster.

Die Flächenapproximation mit nichtperiodischen B-Splines erfüllt die auf Seite 3-47 gestellten Forderungen. Ein Nachteil ist, daß die Pflaster am Rand der Fläche mit anderen Polynomen (siehe Gl. (3.96)) approximiert werden als die übrige Fläche und deshalb eine einheitliche Darstellung nicht möglich ist. Dieser Mangel tritt bei den periodischen B-Splines nicht auf.

b) Approximation mit periodischen B-Splines

ba) Kurvenapproximation

Ein B-Spline $N_{i,k}(u')$ mit dem Knotenvektor $\mathcal{U} = (u'_i \bmod m+1 = i \bmod m+1)$ hat sein Maximum beim Parameterwert $u' = u'_i + \frac{k}{2} = i + \frac{k}{2}$ (Bild 3.25).

Für die Kurvenapproximation wird Gl. (3.92) so modifiziert, daß für geradzahliges k der mit S_i korrespondierende Kurvenpunkt $P_i = Q(u'_i)$ ist.

$$Q(u') = \sum_{i=o}^{m} S_i \cdot N_{(i-\frac{k}{2}) \bmod (m+1),k}(u') \qquad (3.99)$$

Für ungerades k wird für $\frac{k}{2}$ die nächst kleinere, ganze Zahl eingesetzt:

$$\frac{k}{2} = \begin{cases} \frac{k}{2} & \text{für } k \text{ gerade} \\ \text{entier}\,(\frac{k}{2}) & \text{für } k \text{ ungerade} \end{cases}$$

Approximiert man ein Polygon aus (m+1) Stützstellen nach Gl. (3.99), so erhält man auf Grund der Periodizität der Approximationsfunktionen (Bild 3.27, 3.28) eine geschlossene Kurve, die keine der Stützstellen S_i interpoliert. Die Kurve besteht aus (m+1) Segmenten.

Beispiel 7: m = 3 , k = 3

In Bild 3.27 ist der Verlauf der zugehörigen B-Splines dargestellt.

$$Q(u') = \sum_{i=0}^{3} S_i N_{(i-1) \bmod 4,3}(u')$$

$$= S_o N_{3,3} + S_1 N_{0,3} + S_2 N_{1,3} + S_3 N_{2,3}$$

$$= (S_3 \cdot \tfrac{1}{2}(1-u)^2 + S_o \cdot (-u^2+u+\tfrac{1}{2}) + S_1 \cdot \tfrac{u^2}{2}) N_{0,1}$$

$$+ (S_0 \quad\quad + S_1 \quad\quad + S_2 \quad\quad) N_{1,1}$$

$$+ (S_1 \quad\quad + S_2 \quad\quad + S_3 \quad\quad) N_{2,1}$$

$$+ (S_2 \quad\quad + S_3 \quad\quad + S_0 \quad\quad) N_{3,1}$$

$$= \begin{bmatrix} \frac{1}{2}(1-u)^2 \\ -u^2+u+\frac{1}{2} \\ \frac{u^2}{2} \end{bmatrix}^T \cdot \begin{bmatrix} S_3 & S_0 & S_1 & S_2 \\ S_0 & S_1 & S_2 & S_3 \\ S_1 & S_2 & S_3 & S_0 \end{bmatrix} \qquad (3.100)$$

$$= U^T \cdot \overline{B} \text{ , wobei die erste Spalte von } \overline{B} \text{ das}$$

erste Kurvensegment liefert für $o \leq u \leq 1$, usw. Bild 3.32 zeigt die Approximation eines Rechtecks mit k = 3 und k = 4. Man erkennt, daß die Knoten der Kurve für geradzahliges k gegenüber den korrespondierenden Stützstellen liegen, während sie für ungeradzahliges k in der Nähe der Mitte der Polygonseiten liegen. Für k = 3 liegen sie genau in der Mitte.

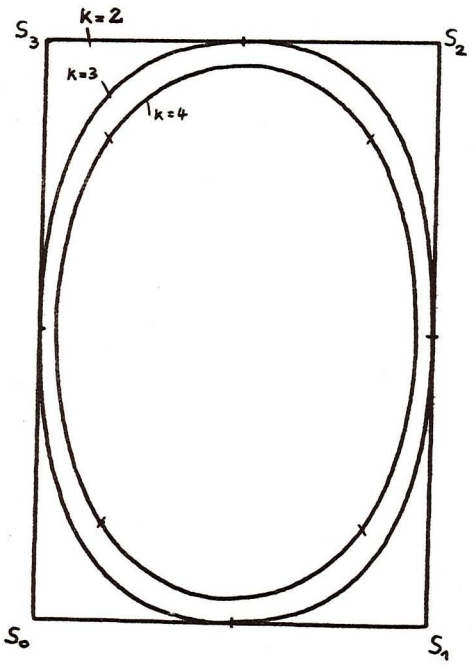


Bild 3.32: m = 3, k = 3 und 4

Soll ein offenes Polygon durch eine offene Kurve so approximiert werden, daß die Endpunkte von Kurve und Polygon zusammenfallen, dann müssen S_0 und S_m (k-1)-fache Stützstellen sein. Die erste und letzte Polygonseite sind dann in S_0 und S_m wieder Tangenten an die Kurve. Die Kurve besteht aus B = (m-k+2) Segmenten, wobei m die (k-1)-fach Stützstellen auch (k-1)-mal enthält. Für die Stützstellen wird folgende Schreibweise eingeführt:

S_i^f wobei der untere Index den Namen
und der obere Index den Wert

der Stützstelle angibt. Ist f = i, wird nur i angegeben!

Beispiel 8: Offenes Polygon mit 4 Stützstellen

$k = 3: \quad S_0^1\, S_1^1\, S_2^2\, S_3^3\, S_4^4\, S_5^4 = S_0^1\, S_1\, S_2\, S_3\, S_4\, S_5^4$

d.h.: m = 5 , B = 4

Aus Gl. (3.100) wird

$$Q(u') = \begin{bmatrix} \frac{1}{2}(1-u)^2 \\ -u^2+u+\frac{1}{2} \\ \frac{u^2}{2} \end{bmatrix}^T \cdot \begin{bmatrix} S_0^1 & S_1 & S_2 & S_3 \\ S_1 & S_2 & S_3 & S_4 \\ S_2 & S_3 & S_4 & S_5^4 \end{bmatrix} \quad \text{und für}$$

$k = 4: \quad S_0^2\, S_1^2\, S_2\, S_3\, S_4\, S_5\, S_6^5\, S_7^5 \longrightarrow m = 7 \longrightarrow B = 5$

$$Q(u') = \begin{bmatrix} \frac{1}{6}(1-u)^3 \\ 3u^3-6u^2+4 \\ -3u^3+3u^2+3u+1 \\ u^3 \end{bmatrix}^T \cdot \begin{bmatrix} S_0^2 & S_1 & S_2 & S_3 & S_4 \\ S_1^2 & S_2 & S_3 & S_4 & S_5 \\ S_2 & S_3 & S_4 & S_5 & S_6^5 \\ S_3 & S_4 & S_5 & S_6 & S_7^5 \end{bmatrix}$$

k wird im Hinblick auf die Hardware-Realisierung auf den Bereich $2 \leq k \leq 4$ beschränkt, so daß nur drei verschiedene Vektoren U_k mit Approximationspolynomen vorkommen können:

$$\begin{aligned} k = 2 &: \quad U_2 = (1-u,\ u) \\ k = 3 &: \quad U_3 = \left(\tfrac{1}{2}(1-u)^2,\ (-u^2+u+\tfrac{1}{2}),\ \tfrac{u^2}{2}\right) \\ k = 4 &: \quad U_4 = \tfrac{1}{6}\left((1-u)^3,\ (3u^3-6u^2+4),\ (-3u^3+3u^2+3u+1),\ u^3\right) \end{aligned} \qquad (3.101)$$

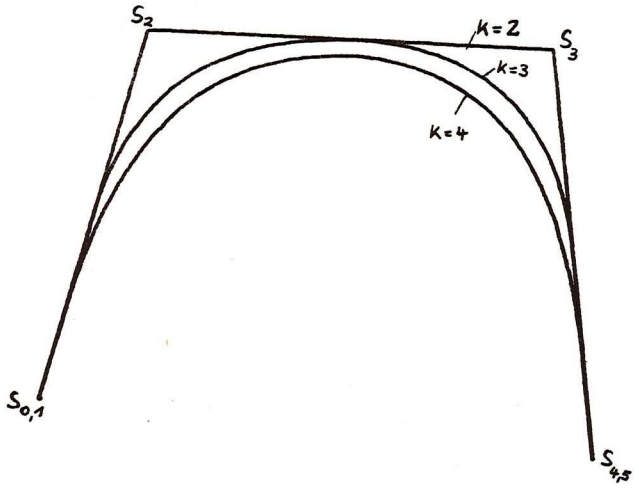


Bild 3.32a: Offenes Polygon mit k = 3, k = 4 approximiert

Damit läßt sich folgende Vorschrift für das Aufstellen der Gleichung einer B-Spline-Kurve angeben:

Seien (m+1) Stützstellen S_i gegeben:

1) Wahl der Ordnung k : $\longrightarrow U_k$

2) Geschlossenes Polygon : $S_o \equiv S_{m+1}$

Das erste Kurvensegment ist bestimmt für

k = 3 durch $S_m\ S_o\ S_1$

k = 4 durch $S_m\ S_o\ S_1\ S_2$

Durch zyklisches Vertauschen der Stützstellen erhält man insgesamt (m+1) Kurvensegmente.

3) Offenes Polygon:

Das erste Kurvensegment ist bestimmt für

k = 3 durch $S_o^1\ S_1\ S_2$

k = 4 durch $S_o^2\ S_1^2\ S_2\ S_3$

Durch Zusammenfassen von jeweils k Stützstellen erhält man (m-k+2) Kurvensegmente.

Für beide Typen der B-Spline-Kurven gilt, daß die Kurve bei k kollinearen Stützstellen ein Geradenstück mit dem Polygon gemeinsam hat. Liegen (k-1) kollineare Stützstellen vor, so ist die entsprechende Polygonseite Tangente der Kurve (siehe Bild 3.32). Die Kurven haben höchstens soviele Wendepunkte wie das Polygon und besitzen nur Schleifen, wenn auch das Polygon Schleifen aufweist.

Oft tritt der Fall auf, daß die Kurve Ecken haben soll, die Steigung an dieser Stelle also unstetig ist. Dies kann erreicht werden, wenn (k-1) Stützstellen zusammengelegt werden. Die Kurve interpoliert diese Mehrfachstützstelle und hat die in diesem Punkt zusammentreffenden Polygonseiten als Tangenten. Bild 3.33 zeigt für den Fall k = 3 zwei aufeinanderfolgende 2-fach Stützstellen. Damit ist auch die Bedingung für k kollineare Stützstellen erfüllt und die Kurve enthält ein Geradenstück!

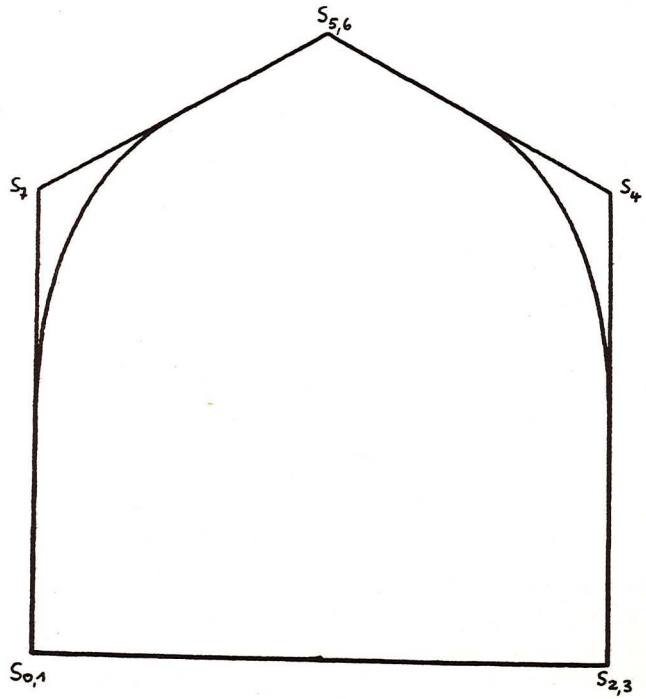


Bild 3.33: k = 3

bb) Flächenapproximation

Die für Kurven gemachten Aussagen lassen sich unmittelbar auf Flächen übertragen. Die Fläche setzt sich aus Pflastern

$$Q_{bc}(u', v') = U_k^T \cdot \bar{S}_{bc} \cdot V_l \qquad (3.102)$$

zusammen. U und V enthalten nur die in Gl. (3.101) angegebenen Polynome. U_k und V_l können von verschiedener Ordnung sein.

Offene Fläche:

Sollen die Eckpunkte von Polygonnetz und Fläche zusammenfallen, dann müssen die Stützstellen der Randpolygone (k-1)-fache bzw. (l-1)-fache Stützstellen sein. Die Fläche setzt sich aus (m-k+2)(n-l+2) Pflastern $Q_{bc}(u',v')$ zusammen, die nach Gl. (3.97) nummeriert werden:

$$\bar{S}_{bc} = \begin{bmatrix} S_{(b-1),(c-1)} & \cdots\cdots & S_{(b-1),(c+l-2)} \\ \vdots & & \vdots \\ S_{(b+k-2),(c-1)} & \cdots\cdots & S_{(b+k-2),(c+l-2)} \end{bmatrix} \qquad (3.103)$$

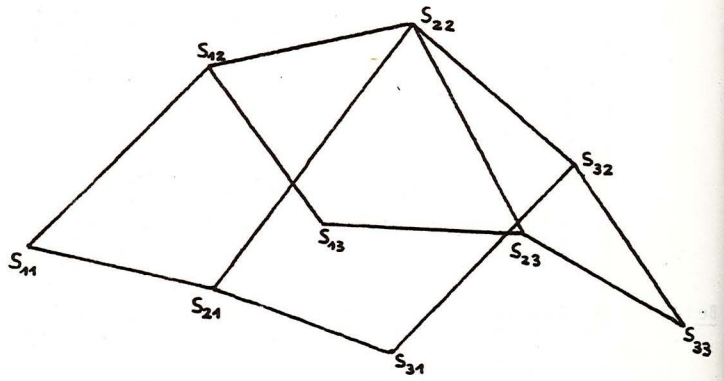
Beispiel 9: 3x3 Sützstellen, k = 3 m = 4
l = 3 n = 4

$$Q(u',v') = U_3^T \cdot \bar{S} \cdot V_3$$

$$\bar{S} = \begin{bmatrix} S_{oo}^{11} & S_{o1}^{11} & S_{o2}^{12} & S_{o3}^{13} & S_{o4}^{13} \\ S_{10}^{11} & S_{11} & S_{12} & S_{13} & S_{14}^{13} \\ S_{20}^{21} & S_{21} & S_{22} & S_{23} & S_{24}^{23} \\ S_{30}^{31} & S_{31} & S_{32} & S_{33} & S_{34}^{33} \\ S_{40}^{31} & S_{41}^{31} & S_{42}^{32} & S_{43}^{33} & S_{44}^{33} \end{bmatrix}$$

$$Q_{11}(u', v') = U_3^T \cdot \bar{S}_{11} \cdot V_3$$

$$\bar{S}_{11} = \begin{bmatrix} S_{00}^{11} & S_{01}^{11} & S_{02}^{12} \\ S_{10}^{11} & S_{11} & S_{12} \\ S_{20}^{21} & S_{21} & S_{22} \end{bmatrix}$$



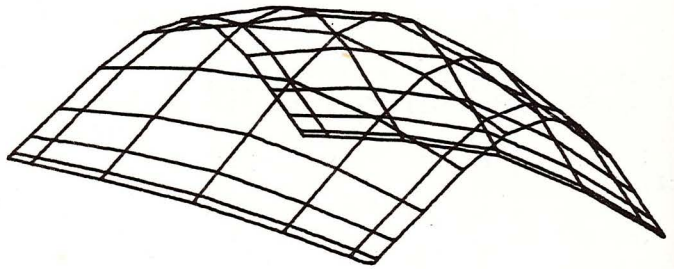


Bild 3.34: Offenes Pflaster. 3x3 Stützstellen
k = l = 3, m = 4, n = 4

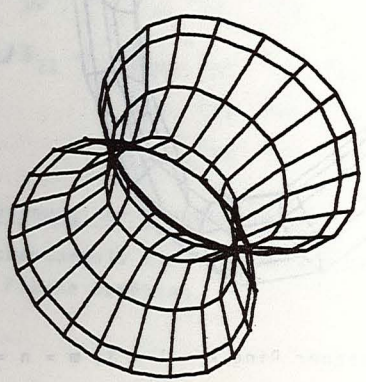
In einer Parameterrichtung geschlossene Fläche:

Das Polygonnetz sei in v-Richtung geschlossen: $S_{i,n+1} \equiv S_{i,o}$

Die Fläche setzt sich dann aus (m-k+2)(n+1) Pflastern zusammen. Der Tensor $\bar{S}$ für die Gesamtfläche wird in u-Richtung nach Gl. (3.103), in v-Richtung durch zyklisches Vertauschen entsprechend der Vorschrift für geschlossene Polygone (S. 3-70) gewonnen.

Beispiel 10: 3x3 Stützstellen: k = l = 3; m = 4 ; n = 2

$$\bar{S} = \begin{bmatrix} S_{00}^{10} & S_{01}^{11} & S_{02}^{12} & S_{00}^{10} & S_{01}^{11} \\ S_{10} & S_{11} & S_{12} & S_{10} & S_{11} \\ S_{20} & S_{21} & S_{22} & S_{20} & S_{21} \\ S_{30} & S_{31} & S_{32} & S_{30} & S_{31} \\ S_{40}^{30} & S_{41}^{31} & S_{42}^{32} & S_{40}^{30} & S_{41}^{31} \end{bmatrix}$$



In beiden Parameterrichtungen geschlossene Fläche:

$$S_{m+1,\ n+1} \equiv S_{oo}$$

Die Fläche setzt sich aus (m+1)·(n+1) Teilflächen zusammen.

Beispiel 11: Ring: 3x3 Stützstellen k = l = 3

$$\overline{S} = \begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{00} & S_{01} \\ S_{10} & S_{11} & S_{12} & S_{10} & S_{11} \\ S_{20} & S_{21} & S_{22} & S_{20} & S_{21} \\ S_{00} & S_{01} & S_{02} & S_{00} & S_{01} \\ S_{10} & S_{11} & S_{12} & S_{10} & S_{11} \end{bmatrix}$$

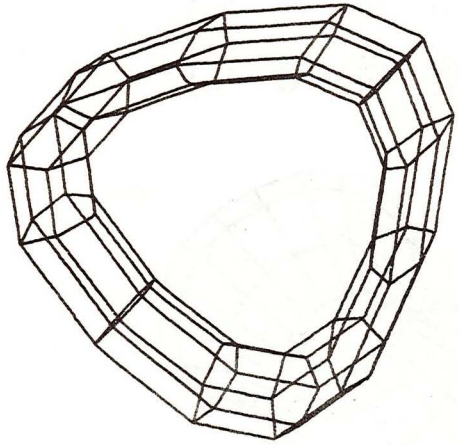


Bild 3.36: Geschlossener Ring k = l = 3, m = n = 2

Aus den angegebenen Beispielen ist zu erkennen, daß das kantenfreie Zusammenfügen von B-Spline-Pflastern kein Problem darstellt. An den Rändern der beiden zusammenzufügenden Pflaster müssen nur die Stützstellen des Randpolygons übereinstimmen. Tangentenbedingungen sind nicht zu beachten. Dies bedeutet, daß der Tensor $\overline{S}$ der neuen Fläche sich durch Zusammensetzen der Einzeltensoren unter Beachtung von eventuellen Mehrfachstützstellen ergibt. Treten innerhalb von $\overline{S}$ keine Mehrfachstützstellen auf, so ist die Fläche überall (k-2) mal stetig differenzierbar.

Beispiel 11: Zusammenfügen zweier Pflaster

$$\overline{S}_I = \begin{bmatrix} S_{00} & S_{01} & S_{02} \\ S_{10} & S_{11} & S_{12} \\ S_{20} & S_{21} & S_{22} \end{bmatrix}, \quad \overline{S}_{II} = \begin{bmatrix} S_{03} & S_{04} & S_{05} & S_{0} \\ S_{13} & S_{14} & S_{15} & S_{1} \\ S_{23} & S_{24} & S_{25} & S_{2} \end{bmatrix}$$

Es soll sein: $Q_I(u,1) = Q_{II}(u,o)$

Vor.: $k_I = k_{II}$; $l_I = l_{II}$

Mit $S_{02} = S_{03}$, $S_{12} = S_{13}$, $S_{22} = S_{23}$ wird

$$\overline{S} = \overline{S}_I \cup \overline{S}_{II} = \begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{04} & S_{05} & S_{06} \\ S_{10} & S_{11} & S_{12} & S_{14} & S_{15} & S_{16} \\ S_{2} & S_{21} & S_{22} & S_{24} & S_{25} & S_{26} \end{bmatrix}$$

bc) Berechnung der Flächennormale

Die Flächennormale wird zur Berechnung der Grauwerte auf der Fläche benötigt.

$$N(u',v') = Q_u(u',v') \times Q_v(u',v') \quad (3.104$$

Aus $Q(u',v') = U_k^T \cdot \overline{S} \cdot V_l$ wird nach /30/

$$Q_u(u',v') = \frac{\partial}{\partial u} \cdot U_k^T \cdot \overline{S} \cdot V_l$$

$$= U_{k-1}^T \cdot \Delta_i \overline{S} \cdot V_l \quad (3.105$$

$$\text{mit } \Delta_i S_{ij} = S_{i,j} - S_{i-1,j} \quad (3.106$$

$$\text{bzw. } Q_v(u',v') = U_k^T \Delta_j \overline{S} \cdot V_{l-1} \quad (3.107$$

$$\text{mit } \Delta_j S_{ij} = S_{i,j} - S_{i,j-1} \quad (3.108$$

Beispiel 12: Berechnung der Ableitung zu Beispiel 8

$$Q_u(u') = \frac{\partial}{\partial u} U_3^T \cdot \begin{bmatrix} S_0^1 & S_1 & S_2 & S_3 \\ S_1 & S_2 & S_3 & S_4 \\ S_2 & S_3 & S_4 & S_5^4 \end{bmatrix} \quad \text{und mit Gl. (3.105}$$

$$Q_u(u') = U_2^T \cdot \begin{bmatrix} S_1-S_0^1 & S_2-S_1 & S_3-S_2 & S_4-S_3 \\ S_2-S_1 & S_3-S_2 & S_4-S_3 & S_5^4-S_4 \end{bmatrix}$$

3.4.2.3 Interpolation mit B-Splines

Beim Entwurf von Formen am Bildschirm tritt oft der Fall auf, daß man von einer bekannten Kurve, die analytisch oder als Punktmenge vorliegt, ausgehen will. Um mit ihr weiter arbeiten zu können, benötigt man deren definierendes Polygon $\overline{S}$. Dies läßt sich unter der Annahme, daß die vorliegende Kurve eine B-Spline-Kurve ist, berechnen. Man muß den Zusammenhang zwischen den Stützstellen S_i, den dazu korrespondierenden Kurvenpunkten P_i und den Knoten u_i' der Spline kennen.

Universitätsbibliothek Techn. Univ. Berlin

Analog zur Bézierapproximation gilt:

$$P_i = Q(\mathcal{S}_i) \quad , \quad (3.109)$$

wobei $\mathcal{S}_i$ den Wert des Parameters u' angibt, für den die Stützstelle S_i ihren größten Einfluß auf die Kurve hat. Sind die $\mathcal{S}_i$ einer Kurve bekannt, so kann folgendes lineares Gleichungssystem zur Bestimmung der Stützstellen S_i aufgestellt werden:

$$\begin{bmatrix} P_0 \\ \vdots \\ P_i \\ \vdots \\ P_m \end{bmatrix} = \begin{bmatrix} Q(\mathcal{S}_0) \\ \vdots \\ Q(\mathcal{S}_i) \\ \vdots \\ Q(\mathcal{S}_m) \end{bmatrix} = \begin{bmatrix} U^T(\mathcal{S}_0) \\ \vdots \\ U^T(\mathcal{S}_i) \\ \vdots \\ U^T(\mathcal{S}_m) \end{bmatrix} \cdot \begin{bmatrix} S_0 \\ \vdots \\ S_i \\ \vdots \\ S_m \end{bmatrix} \quad \text{oder}$$

$$\overline{P} = \overline{U^T} \cdot \overline{S} \quad \text{und hieraus} \quad (3.110)$$

$$\overline{S} = \overline{U^T}^{-1} \cdot \overline{P} = I \cdot \overline{P} \quad (3.111)$$

$\overline{U^T}$ ist nichtsingulär und invertierbar, wenn alle $\mathcal{S}_i$ verschieden sind /23/. Für offene Kurven ist dies nur bei nichtperiodischen B-Splines gegeben.

In /28/ hat Greville nachgewiesen, daß der mit S_i korrespondierende Kurvenpunkt P_i einer B-Spline-Kurve den Parameterwert

$$\mathcal{S}_i = \frac{1}{k-1} \cdot (u_{i+1}' + u_{i+2}' + \ldots + u_{i+k-1}') \quad (3.112)$$

besitzt. Da die Knoten u_i' ganze, positive Zahlen sind, gilt (mit Ausnahme des ersten und letzten Kurvensegments):

Bei diesem Parameterwert hat der B-Spline $N_{i,k}(u')$ sein Maximum. Für geradzahliges k ist ξ_i ein Knoten, d.h. $\xi_i \in u'_i$.

Beispiel 13: Zu der Kurve in Beispiel 6, Bild 3.30 werden die Parameterwerte ξ_i und die Punkte $P_i = Q(\xi_i)$ berechnet. Aus Gl. (3.12) ergibt sich (m=5 , k=3):

$$\xi_i = \frac{1}{2}(u'_{i+1} + u'_{i+2})$$

$$\xi_0 = \frac{1}{2}(u'_1+u'_2) = 0 \;, \quad P_0 = S_0$$

$$\xi_1 = \frac{1}{2}(u'_2+u'_3) = \frac{1}{2} \;, \quad P_1 = \frac{1}{4}S_0+\frac{5}{8}S_1+\frac{1}{8}S_2$$

$$\xi_2 = \frac{1}{2}(u'_3+u'_4) = \frac{3}{2} \;, \quad P_2 = \frac{1}{8}S_1+\frac{3}{4}S_2+\frac{1}{8}S_3$$

$$\xi_3 = \ldots = \frac{5}{2} \;, \quad P_3 = \frac{1}{8}S_2+\frac{3}{4}S_3+\frac{1}{8}S_4$$

$$\xi_4 = \ldots = \frac{7}{2} \;, \quad P_4 = \frac{1}{8}S_3+\frac{5}{8}S_4+\frac{1}{4}S_5$$

$$\xi_5 = \ldots = 4 \;, \quad P_5 = S_5$$

Mit diesen Werten wird aus Gl. (3.111):

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{5}{8} & \frac{1}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 \\ 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & 0 & 0 & \frac{1}{8} & \frac{5}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1}}_{I} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix}$$

In vielen Fällen ist die zu interpolierende Kurve nicht analytisch sondern als Punktfolge entsprechend der Auflösung des Eingabegeräts (z.B. Tablett oder Lightpen) gegeben. Eine gute erste Näherung erhält man, wenn zur Berechnung des zugehörigen Polygons in Gl. (3.111) für die P_i die relativen Maxima und Minima der Kurve gewählt werden. Dieses Ergebnis kann, wenn beide Kurven und das Polygon auf dem Bildschirm vorhanden sind, interaktiv schnell verbessert werden.

Mit der B-Spline-Approximation steht ein Verfahren zur Kurven- und Flächenapproximation zur Verfügung, das für den Benutzer einfach zu handhaben ist /31/ und durch seinen geringen Rechenaufwand eine schnelle Hardware-Realisierung ermöglicht. Neben der Anwendung in Computer Graphics sind B-Splines sicher überall dort von Interesse, wo die Problemstellung stückweise definierte Funktionen verlangt, deren höhere Ableitungen nach dem Parameter (z.B. der Zeit beim Straßenentwurf) stetig sein müssen.

Die Bilder 3.37 und 3.38 demonstrieren den engen Zusammenhang zwischen definierendem Polygon und approximierender B-Spline-Fläche bzw. -Kurve.

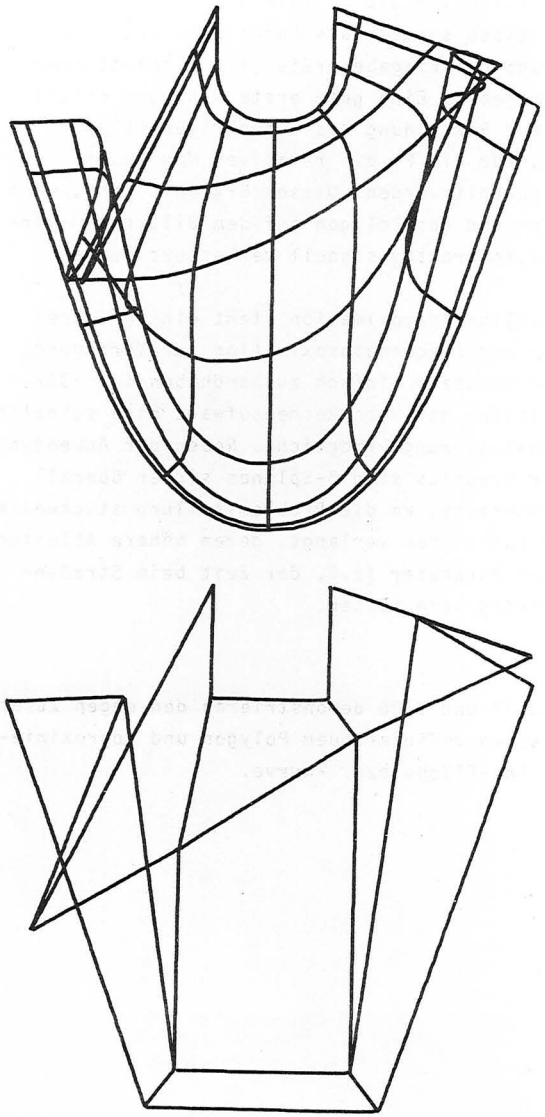
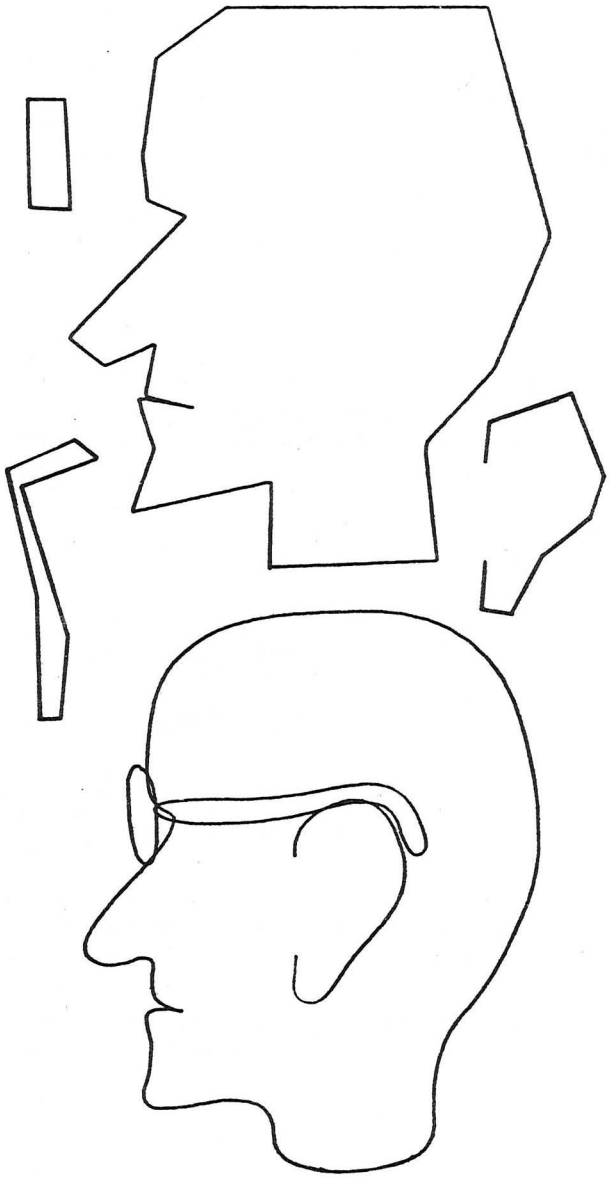


Bild 3.37: B-Spline-Approximation(k=3) des Polygons



4. Vermittlung eines dreidimensionalen Eindrucks

Um dreidimensionale Objekte auf dem ebenen Bildschirm eindeutig interpretieren zu können, müssen für den Betrachter Sehhilfen geschaffen werden, die ihm einen Tiefeneindruck vermitteln. In diesem Kapitel werden nur die Verfahren angegeben, die mit der in Kapitel 5 beschriebenen Hardware realisiert werden können.

4.1 Perspektive

Die perspektivische Projektion eines Punktes $Q = [XYZ]$ wird unter der Voraussetzung, daß die Ebene $Z = 0$ die Bildschirmebene ist und der Blickpunkt die Koordinaten $\left[0, 0, \frac{1}{P}\right]$ besitzt,

nach $$Q_P = \frac{Q}{1-P\cdot Z} \qquad (4.1)$$

berechnt. Ist Q ein Punkt auf einer B-Spline-Fläche,

so gilt $$Q_P\,(u',v') = \frac{U^T\cdot \bar{S}\cdot V}{1-P\cdot U^T\cdot \bar{Z}\cdot V}\ , \qquad (4.2)$$

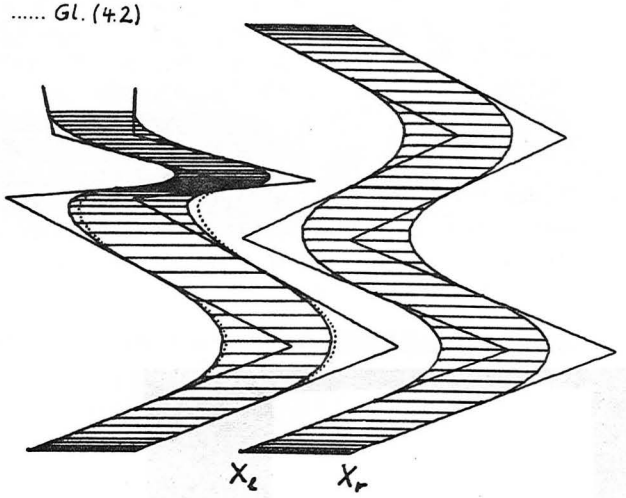
wobei $\bar{Z}$ die Matrix der Z-Komponenten des Tensors $\bar{S}$ der Stützstellenkoordinaten ist. Die Division nach Gl. (4.2) für jeden Flächenpunkt stellt einen erheblichen Rechenaufwand dar. Dies läßt sich vermeiden, wenn man unter Beachtung des engen Zusammenhangs zwischen Polygonnetz und Fläche Gl. (4.2) durch eine "Falsche Perspektive" ersetzt, bei der nicht jeder Kurvenpunkt, sondern nur die Stützstellen S_{ij} perspektivisch transformiert werden.

Mit

$$S_{ijP} = \frac{S_{ij}}{1-PZ_{ij}} \quad \text{und} \quad \bar{S}_P = \left[S_{ijP}\right] \qquad (4.3)$$

wird $$Q_P(u',v') = U^T\cdot \bar{S}_P\cdot V \qquad (4.4)$$

Versuche mit dieser Näherung ergaben gegenüber der Berechnung nach Gl. (4.2) keinen schlechteren Tiefeneindruck.



P = + 30

X_l	-5	-5	7.5	-5	5	-5	-5
X_r	0	0	12.5	0	10	0	0
Y	0	0	5	10	15	20	20
Z	0	0	0	5	-5	10	-5

Bild 4.1: "Falsche Perspektive"

4.2 Rotation

Eine sehr wirksame Sehhilfe ist die Rotation des dreidimensionalen Objekts in Echtzeit auf dem Bildschirm. Für die B-Spline-Fläche müssen nur die Stützstellenkoordinaten transformiert werden:

$$S_{ijR} = S_{ij} \cdot R \;, \quad \overline{S}_R = [S_{ijR}] \qquad (4.5)$$

wobei R die (3x3) Rotationsmatrix ist /32/. Damit

wird $$Q_R(u',v') = U^T \cdot \overline{S}_R \cdot V \qquad (4.6)$$

4.3 Helligkeitssteuerung

Die Intensität des Kathodenstrahls wird als Funktion der Z-Koordinate variiert. Dieses Verfahren ist bei dem eingesetzten Display HP 1310A auf Grund seiner geringen Leuchtdichte nur beschränkt zu verwenden.

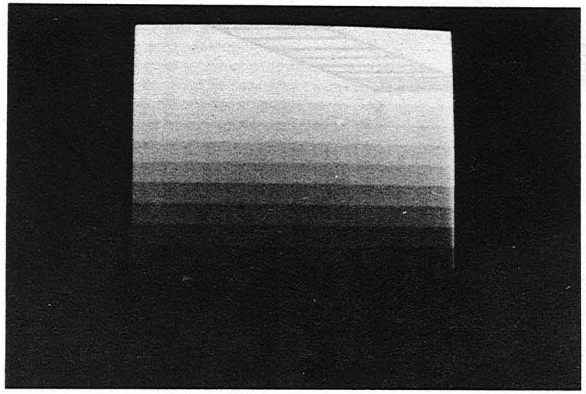
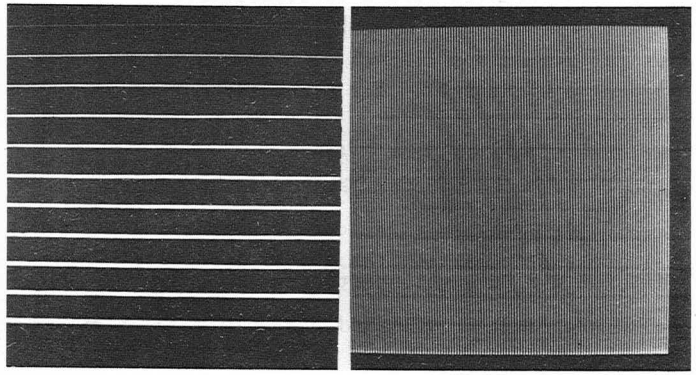


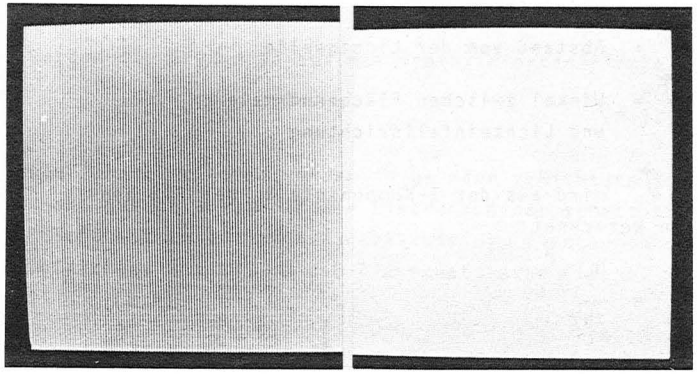
Bild 4.2: Beim Display HP 1310 A erreichbare Helligkeitsstufen

4.4 Steuerung der Linienstärke

Auch die Linienstärke kann mit der Z-Koordinate variiert und mit der Helligkeitssteuerung kombiniert werden. Neben der Verbesserung des Tiefeneindrucks liegt ihre Hauptbedeutung in der Einsparung von Rasterlinien beim Schattieren von Flächen (siehe Kapitel 4.5).



Linienstärken dunkel + schwach



hell + schwach hell + stark

Bild 4.3: Steuerung von Intensität und Linienstärke

4.5 Schattieren von Flächen

Unter Schattieren soll hier das Ausfüllen von Flächen mit Grauwerten verstanden werden. Aus der Lichttechnik ist bekannt, daß die Helligkeit (der Grauwert) eines Gegenstandes durch die Beleuchtungsstärke bestimmt ist. Folgende Einschränkungen werden vorausgesetzt:

1) Diffus reflektierende Flächen
2) Punktförmige Lichtquelle der Lichtstärke I im Blickpunkt
3) Blickrichtung parallel zur Z-Achse
4) Abstand zwischen Körperoberfläche und Lichtquelle ist konstant

Damit gilt für die Beleuchtungsstärke

$$E = \frac{I}{r^2} \cos\vartheta = E_0 \cos\vartheta \qquad (4.7)$$

E = Beleuchtungsstärke

I = Lichtstärke

r^2 = Abstand von der Lichtquelle

ϑ = Winkel zwischen Flächennormale und Lichteinfallsrichtung

Der $\cos\vartheta$ wird aus der Z-Komponente N_z der Flächennormalen berechnet:

$$\cos\vartheta = \frac{N_z}{/N/} \qquad (4.8)$$

Mit Gl. (3.104) wird

$$N_z = \mathcal{M}_z N = \mathcal{M}_z \left(Q_u(u',v') \times Q_v(u',v')\right) \qquad (4.9)$$

$$N_z = \left(X_u(u',v')\, Y_v(u',v') - Y_u(u',v')\, X_v(u',v')\right)$$

Um den Rechenaufwand, der sich aus den Gln. (4.8) und (4.9) für jeden Flächenpunkt ergibt, zu reduzieren, werden die Grauwerte $E(u',v')$ auf der Fläche analog zu den Koordinaten der Flächenpunkte $Q(u',v')$ aus den Grauwerten E_{ij} an den Stützstellen S_{ij} berechnet. Durch Normierung erhält man aus Gl. (4.7):

$$E_{ij} = \cos\vartheta_{ij} \quad \text{und damit}$$

$$E(u',v') = U^T \bar{E} V \qquad (4.10)$$

Diese Berechnungsformel hat den Nachteil, daß die Grauwerte ohne Berücksichtigung der Entfernung vom Betrachter berechnet werden und dadurch Flächenteile mit gleicher Normale unabhängig von der Tiefe des Objekts gleich hell dargestellt werden. Dies gilt insbesondere auch für den approximierten Bereich zwischen Stützstellen gleicher Helligkeit. Deshalb wird zur Bestimmung des Grauwerts $G(u',v')$ die Z-Koordinate des Flächenpunkts berücksichtigt. Mit

$$G_{ij} = E_{ij} \cdot Z_{ij} \qquad (4.11)$$

werden die Grauwerte auf der approximierten Fläche

$$G(u',v') = U^T \bar{G} V \quad . \qquad (4.12)$$

Die Parameterlinien werden in u- oder v-Richtung so dicht geschrieben, daß der Eindruck einer schattierten Fläche entsteht. Der Grauwert kann als vierte Koordinate betrachtet werden, so daß die schattierte Fläche

$$Q_g = [X\ Y\ Z\ G] \qquad (4.13)$$

aus $$Q_g = U^T \bar{S}_g V \qquad (4.14)$$

mit $$S_{gij} = [X_{ij}\ Y_{ij}\ Z_{ij}\ G_{ij}] \qquad (4.15)$$

berechnet wird. (Z_{ij} wird nur für den Fall der Rotation benötigt.) Der Vorteil dieses Verfahrens ist, daß bei Vorhandensein eines genügend großen Bildwiederholungsspeichers mit kleiner Zugriffszeit oder eines schnellen Flächengenerators /34/ die schattierte Fläche unmittelbar auf dem Bildschirm dargestellt werden kann und nicht wie in bisher bekannten Verfahren /3/, /5/, /33/ durch fotografisches Zusammensetzen einzelner Bildzeilen gewonnen werden muß. Da bei diesem Verfahren die verdeckten Flächen nicht eliminiert werden, kommt es im Bereich der Verdeckungen zu einer Mehrfachbeschreibung des Bildschirms. Dies verhindert die Schattierung der sichtbaren Fläche entsprechend Gl. (4.12), was zunächst als Nachteil anzusehen ist. Experimente mit dem grafischen System AGT 130 haben gezeigt, daß durch die Definition der Grauwerte nach Gl. (4.11) der Einfluß durch Doppelbeschreibung des Bildschirms abgeschwächt wird und der Eindruck eines durchscheinenden Körpers entsteht.

Weicht die Form der Fläche wesentlich von der eines Parallelogramms ab, so können wegen unterschiedlicher Dichte der Parameterlinien Helligkeitsschwankungen auftreten. Diesen Nachteil kann man weitgehend beheben, wenn zur Schattierung diejenigen Parameterlinien gewählt werden, die zwei möglichst gleichlange Flächenränder miteinander verbinden. Diese Entscheidung kann durch Überprüfen der Stützstellen getroffen werden.

Die folgenden Bilder sind Photographien vom Bildschirm des Graphics System AGT 130 der TU Berlin. Der Maßstab ist ungefähr 1:2!

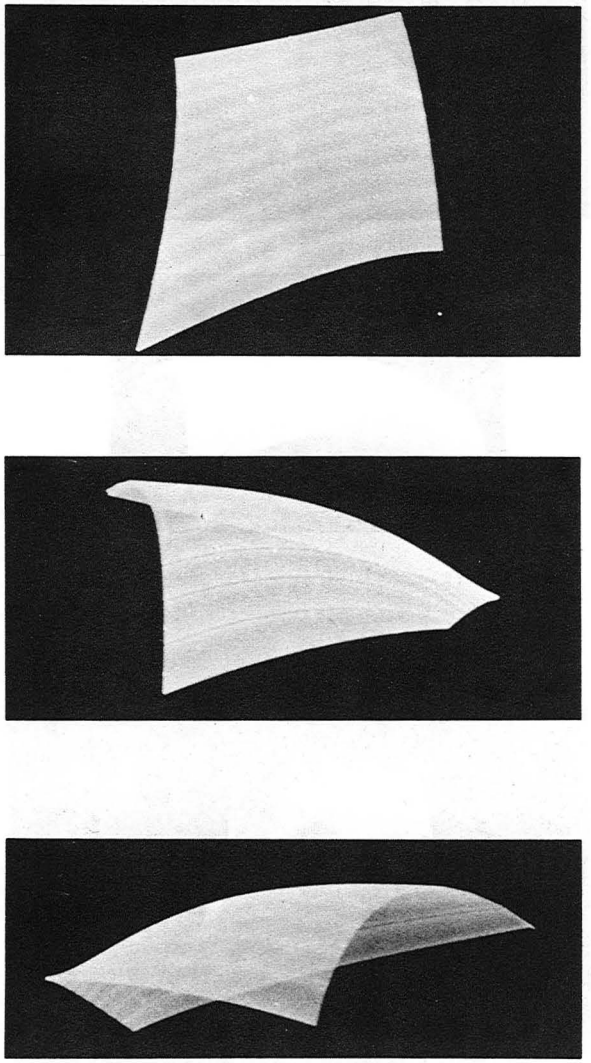
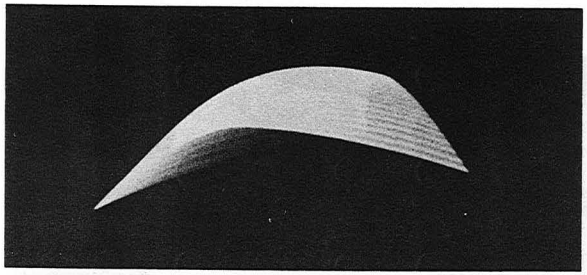


Bild 4.4: Schattierte Flächen mit konstantem Grauwert



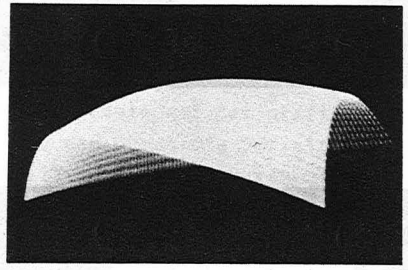
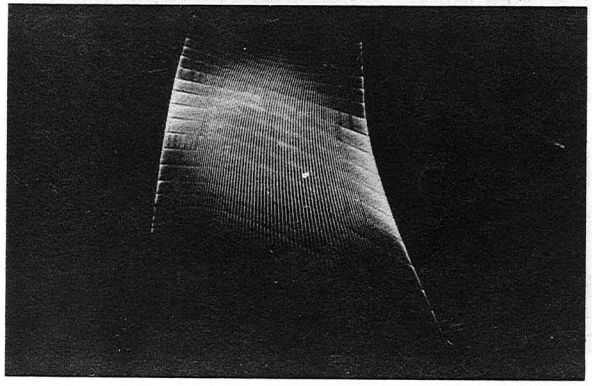


Bild 4.5: Schattierte Flächen mit Grauwert nach Gl.(4.12)



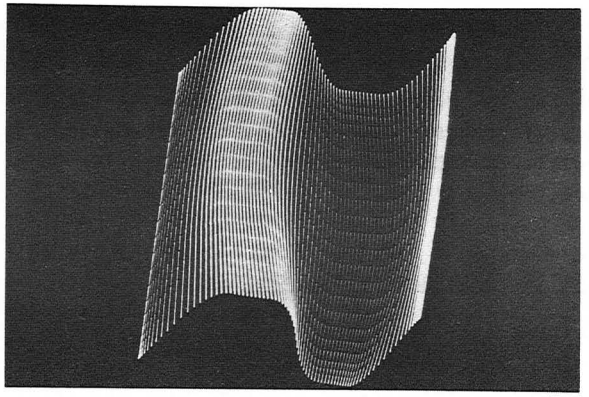


Bild 4.7: Grauwerte nach Gl. (4.12). 80 Parameterlinien à 25 Vektoren

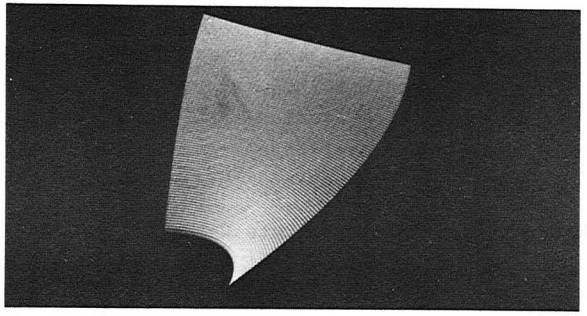


Bild 4.8: 100 Parameterlinien à 100 Vektoren *

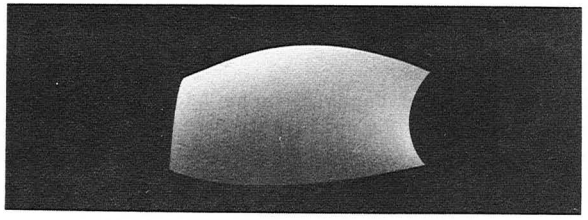


Bild 4.9: 500 Parameterlinien à 50 Vektoren *

* nicht flackerfrei darstellbar auf AGT 130

5. Hardware

In diesem Kapitel werden digitale Komponenten für einen Displayprozessor beschrieben, der die Berechnung, Transformation und Darstellung dreidimensionaler Objekte in Echtzeit erlaubt.

5.1 Digitaler Vektorgenerator (DVG)

Ein Vektorgenerator berechnet die Verbindungsgerade Q(u) zwischen zwei Punkten P_A und P_E. Der hier beschriebene digitale Vektorgenerator arbeitet nach dem Prinzip des digitalen Integriers (DDA). Der Vektor Q(u) berechnet sich aus

$$Q(u) = U^T \cdot \overline{P} = [(1-u),\ u] \cdot \begin{bmatrix} P_A \\ P_E \end{bmatrix} \qquad (5.1)$$

$$= P_A + (P_E - P_A) \cdot u \qquad , o \leq u \leq 1$$

$$Q(u) = P_A + \Delta P \cdot u$$

Der Parameter u kann (k+1) diskrete Werte

$$u_i = \frac{i}{k} \quad , \quad i = 0, 1, \ldots k \qquad (5.2)$$

annehmen. Der Punkt $Q_i = Q(u_i)$ ist dann

$$Q_i = P_A + \Delta P \cdot \frac{i}{k} = P_A + i \cdot \frac{\Delta P}{k}$$

Die Multiplikation wird durch i-fache Addition ersetzt:

$$Q_i = P_A + \underbrace{\frac{\Delta P}{k} + \ldots + \frac{\Delta P}{k}}_{\text{i-fach}}$$

Der Anfangspunkt ist $P_A = Q_o$ und der Endpunkt $P_E = Q_k$. Die einzelnen Geradenpunkte werden durch sukzessive Addition berechnet

$$Q_{i+1} = Q_i + \Delta P/k \qquad (5.3)$$

Die Division $\Delta P/k$ wird mit $k = 2^p$ auf einen p-fachen Schift reduziert. Der Vektor wird dann durch 2^p-faches Addieren des Inkrements $\Delta P/2^p$ auf den Anfangspunkt gewonnen. Damit alle Vektoren auf dem Bildschirm mit gleicher Helligkeit erscheinen, muß die Schreibgeschwindigkeit konstant sein. Da der DVG mit einer festen Taktfrequenz arbeitet, ist diese Bedingung erfüllt, wenn die Zahl der Punkte proportional der in die Bildschirmebene projezierten Vektorlänge ist. Zur Bestimmung der Vektorlänge genügt die Näherung

$$l = \max\left(|\Delta X|, |\Delta Y|\right). \qquad (5.4)$$

Die Anzahl der adressierbaren Punkte des Sichtgeräts wird so gewählt, daß der Punktdurchmesser $D = \sqrt{2}$ RE mit RE gleich der Rastereinheit ist. Damit der Vektor auf dem Bildschirm als geschlossene Linie erscheint, muß das Inkrement $\Delta P/2^p$ kleiner als eine Rastereinheit sein:

$$\begin{aligned} |\Delta X| \,/\, 2^p &< 1 \\ |\Delta Y| \,/\, 2^p &< 1 \end{aligned} \qquad \text{hieraus folgt } 2^p > l \qquad (5.5)$$

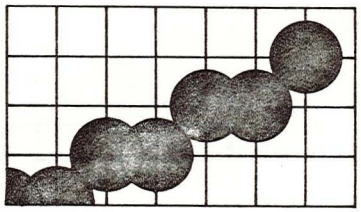
Um die Anzahl der Additionen möglichst niedrig zu halten, wird p so berechnet, daß gilt:

$$2^p > l \geq 2^{p-1} \qquad (5.6)$$

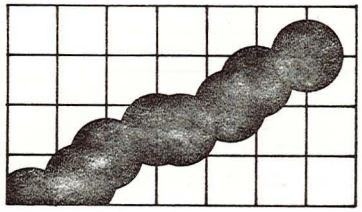
Bei positiven Relativkoordinaten ΔP läßt sich diese Bedingung leicht erfüllen. Es wird solange in Richtung kleinerwertiger Stellen geschiftet, bis in beiden Koor-

dinatenregistern nur noch Nullen vor dem Binärpunkt stehen. Werden negative Relativkoordinaten im 2-Komplement dargestellt, muß der Bestimmung von p die Betragsbildung vorausgehen. Um möglichst viele Vektoren flackerfrei darstellen zu können, werden parallel zum Schreiben eines Vektors die Inkremente $\Delta P/2^p$ für den nächsten Vektor berechnet. Auf diese Weise kann im Akkumulator ohne Verzögerung beim Beginn eines neuen Vektors weiter addiert werden. Damit dieser Betrieb auch bei kurzen Vektoren gewährleistet ist, werden die Inkremente $\Delta P/2^p$ zur Erhöhung der Rechengeschwindigkeit nicht durch Schiften sondern durch Multiplexen erzeugt. Durch Einsatz von Schottky TTL Schaltkreisen läßt sich bei einer Auflösung von (10x10) bit eine Bildschirmdiagonale in 3o µs schreiben. In 30 ms ist jeder Punkt des Displays einmal erreichbar. Die Vektoren können durch Multiplikation der Anzahl 2^p der Additionen mit einem Faktor S skaliert werden, was schaltungstechnisch mit einem Binärratenmultiplizierer zu realisieren ist.

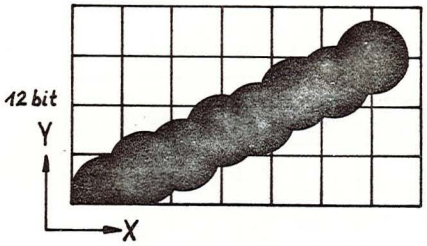
Das Aussehen der Vektoren läßt sich durch Erweiterung der Wortlänge der Digital-Analog-Umsetzer auf Stellen hinter dem Binärpunkt verbessern. Dieser Methode ist allerdings durch die Einschwingzeit der Digital-Analog-Schalter eine Grenze gesetzt. Bild 5.1 zeigt das Aussehen der Vektoren für 3 verschiedene Wortlängen der DAU. In den Bildern 5.2 und 5.3 ist das Ablaufdiagramm und das Blockschaltbild des DVG angegeben. Schaltungstechnische Details, wie die Berechnung der Schiftschritte p, der Multiplexersteuerung und die "Wrap-around"-Unterdrückung sind den Arbeiten /35/,/36/,/37/ zu entnehmen.



X	Y
000.	000.
001.	000.
010.	001.
011.	001.
011.	001.
100.	010.
101.	010.
110.	011.



X	Y
000.1	000.0
001.1	000.1
010.0	001.0
011.0	001.1
011.1	001.1
100.1	010.0
101.0	010.1
110.0	011.0



X	Y
000.11	000.01
001.10	000.11
010.01	001.00
011.00	001.10
011.11	001.11
100.10	010.01
101.01	010.10
110.00	011.00

Bild 5.1: Aussehen der Vektoren bei 3 verschiedenen Wortlängen

Ablaufdiagramm des DVG

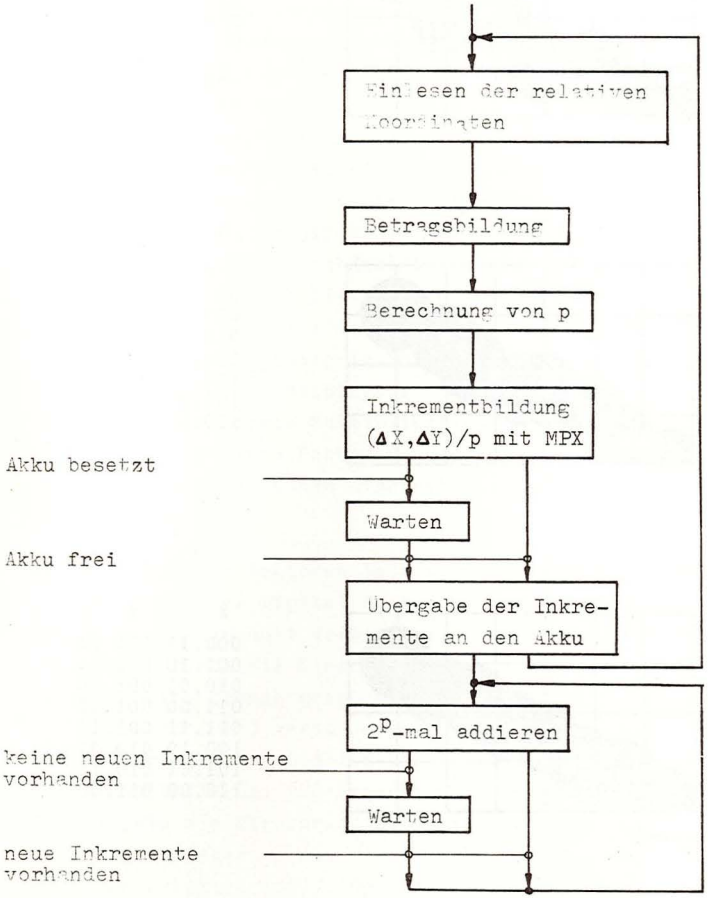


Bild 5.2: Ablaufdiagramm des DVG

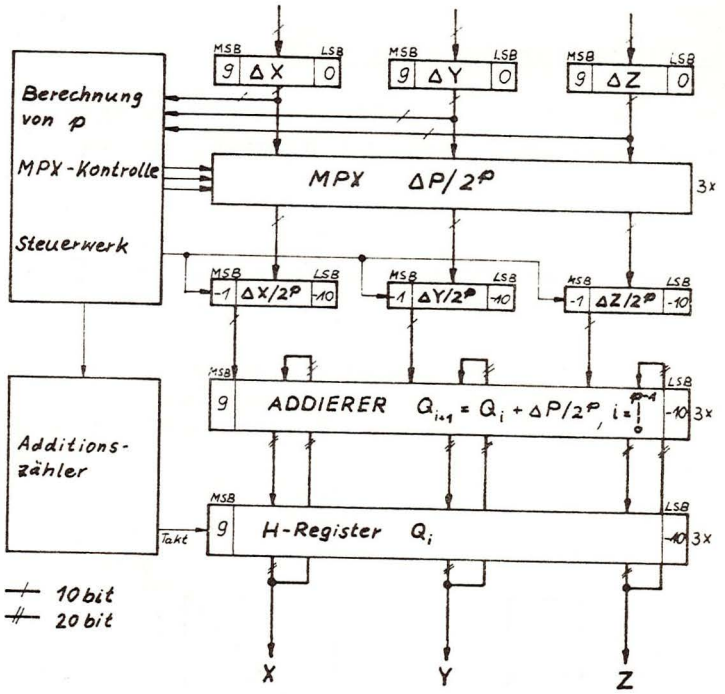
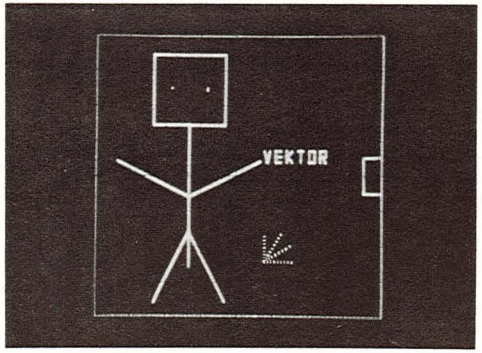


Bild 5.3: Blockschaltbild des DVG



5.2 Digitaler Kreisgenerator (DKG)

Neben dem Vektor ist auch der Kreis ein häufig benötigtes geometrisches Element, dessen Hardware-Realisierung wünschenswert erscheint. In /38/ ist ein Algorithmus zum Zeichnen von Kegelschnitten auf digitalen Plottern beschrieben, der für den Spezialfall des Kreises sehr einfach wird und sich für eine Implementierung in Hardware gut eignet.

Ein digitaler Plotter kann nur acht verschiedene Elementarvektoren zeichnen. Sie werden entsprechend Bild 5.4 nach ihrer Konstruktion in die beiden Gruppen MOVE1 und MOVE2 eingeteilt.

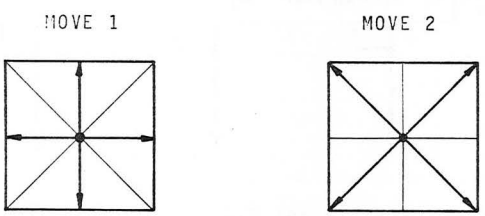
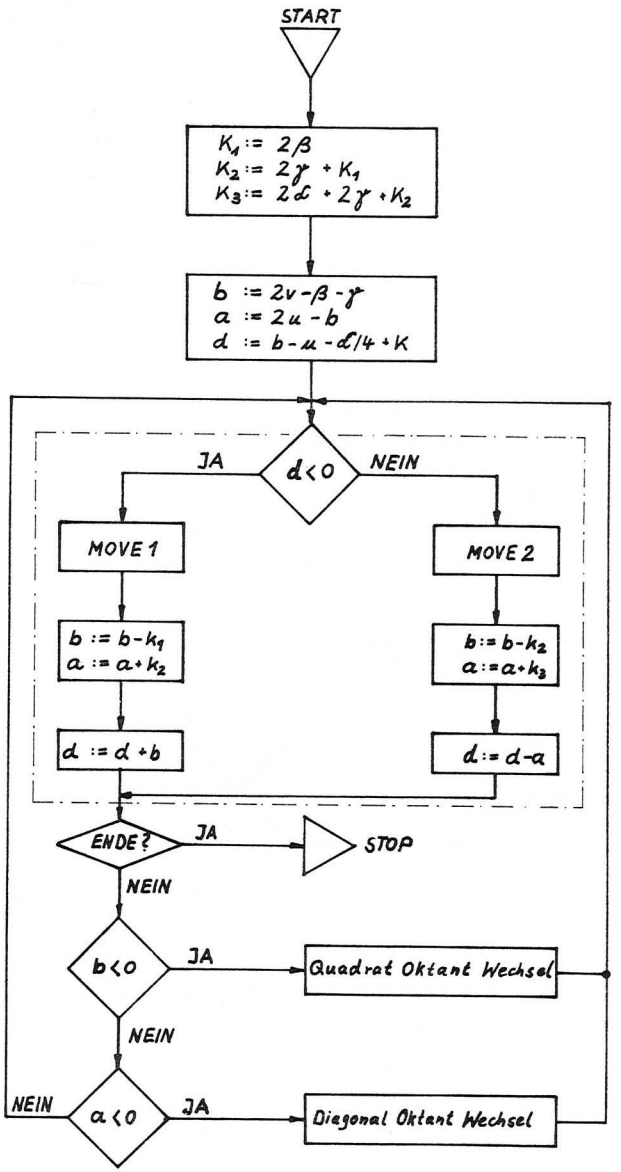


Bild 5.4 Einteilung der Elementarvektoren in MOVE1 und MOVE2.

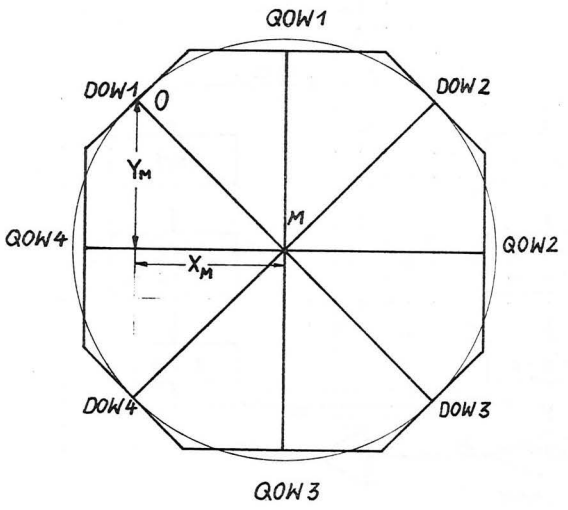
Die allgemeine Gleichung eines Kegelschnitts lautet /38/

$$\alpha Y^2 + \beta X^2 + 2\gamma XY + 2uY - 2vX = k \qquad (5.7)$$

Das Flußdiagramm des Algorithmus zur Darstellung der Kegelschnitte als eine Folge von MOVE1- und MOVE2-Vektoren zeigt Bild 5.5.



Aus dem Flußdiagramm ist zu erkennen, daß die Berechnungszeit für alle Punkte eines Oktanten gleich ist. Diese Eigenschaft bedeutet für einen Kreisgenerator konstante Schreibgeschwindigkeit und damit gleiche Helligkeit für alle darstellbaren Kreise. Bild 5.6 zeigt, daß - wie auf Grund der Symmetrie des Kreises zu erwarten - die Oktanten durch Spiegelung an den Koordinatenachsen und Winkelhalbierenden auseinander gewonnen werden können. Insbesondere sind zwei benachbarte Oktanten bezüglich ihrer Grenze spiegelsymmetrisch.



QOW: Quadrat Oktantenwechsel: MOVE 2 wird geändert
DOW: Diagonal Oktantenwechsel: MOVE 1 wird geändert

Bild 5.6:Zusammensetzen des Kreises aus MOVE 1 und MOVE 2

Man braucht demnach die Folge der MOVE1- und MOVE2-Vektoren nur für einen Oktanten zu berechnen und beim Zeichnen des ganzen Kreises die Richtung dieser Vektoren entsprechend Bild 5.6 zu ändern. Es ist deshalb sinnvoll, die Kreisberechnung in einem Oktantenwechsel, z.B. dem Punkt 0, beginnen zu lassen. Der Punkt 0 wird willkürlich in den Ursprung des Bildschirmkoordinatensystems gelegt. Der Kreismittelpunkt liegt dann bei $X_m = R/\sqrt{2}$ und $Y_m = -R/\sqrt{2}$.

Die Kreisgleichung wird damit

$$(X-R/\sqrt{2})^2 + (Y+R/\sqrt{2})^2 = R^2 \text{ oder}$$

$$X^2+Y^2+\sqrt{2}R\,(Y-X) = 0 \quad \text{und zur Beseitigung}$$

der Wurzel mit $M = \sqrt{2}R$

$$X^2+Y^2+M\,(Y-X) = 0 \qquad (5.8)$$

Durch Koeffizientenvergleich mit Gl. (5.7) werden die Anfangswerte im Flußdiagramm für den Kreis:

$$k_1 = k_2 = 2 \quad ; \quad k_3 = 4 \quad , \quad b = M - 1$$
$$a = 1 \quad ; \quad d = \tfrac{1}{4}(2M-5) \qquad (5.9)$$

Der Zusammenhang zwischen der Anzahl n der Iterationsschritte für einen Oktanten und dem Kreisradius R ist aus Bild 5.6 zu erkennen:

$$X_m = n = R/\sqrt{2} = M/2 \qquad (5.10)$$

Zur Vereinfachung werden neue Parameter so definiert, daß im Kreisalgorithmus nur noch Additionen auszuführen sind. Mit $M = \frac{C}{2}$:

$$K = -8 = -4\,k_1 = -4\,k_2 = -2\,k_3$$
$$B = 4b = 2C-4$$
$$A = -4 \qquad (5.11)$$
$$D = 4d = C-5$$

Bild 5.7 zeigt das Flußdiagramm des Kreisalgorithmus für einen Oktanten. Die Initialisierung entspricht dem Startpunkt 0.

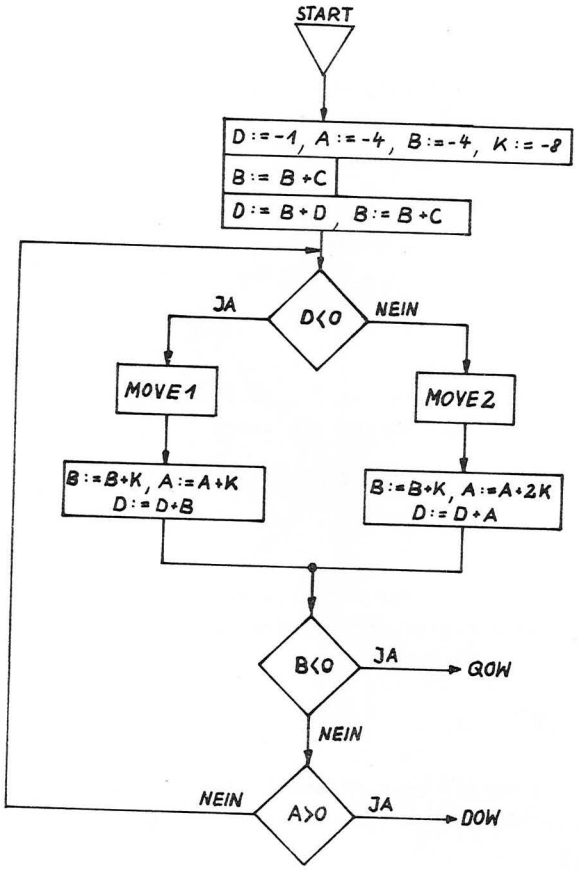


Bild 5.7: Algorithmus für einen Kreisoktanten

Jetzt müssen noch die neuen Parameter beim Oktantenwechsel berechnet werden. Da als Startpunkt 0 ein DOW gewählt wurde, muß der nächste Oktantenwechsel bei einem QOW stattfinden. Aus dem allgemeinen QOW für die Kegelschnitte /38/ wird für den Kreis mit Gl. (5.9):

$$
\begin{aligned}
w &:= k_2 - k_1 &&= 0\\
k_1 &:= -k_1 &&= -2\\
k_2 &:= w + k_1 &&= -2\\
k_3 &:= 4w - k_3 &&= -4\\
b &:= -b - w &&= -b\\
d &:= b - a - d\\
a &:= a - 2b - w &&= a - 2b
\end{aligned}
\qquad (5.12)
$$

Mit M = C/2 erhält man aus Gl. (5.10) C = 4n mit n = 1,2,3,....m. Beim QOW hat B dann den Wert - 4 (siehe Bild 5.7). Damit erhält man aus Gl. (5.12) mit Gl. (5.11) die neuen Parameter beim QOW nach folgendem Algorithmus:

$$
\begin{aligned}
K &:= -K = 8\\
B &:= -B = 4 = B + K\\
D &:= -D + A + B\\
A &:= A + 2B = A + 8 = A + K
\end{aligned}
\qquad (5.13)
$$

Nach dem QOW ist der nächste Oktantenwechsel ein DOW. Da der Algorithmus bei einem DOW gestartet wurde, kann auf Grund der Symmetrieeigenschaften dieselbe Parameterinitialisierung wieder verwandt werden. Der Algorithmus beginnt im Punkt 0 mit einem MOVE2. Auch in den anderen Oktantenwechseln ist die Bewegungsrichtung bekannt und unabhängig vom Radius immer gleich. Deshalb kann man die Abfrage auf $D < 0$, d.h. die Entscheidung ob MOVE1 oder MOVE2 folgt, nach einem Oktantenwechsel sparen. Wird dies bei der Initialisierung berücksichtigt, so ergibt sich das in Bild 5.8 dargestellte Flußdiagramm des Kreisal-

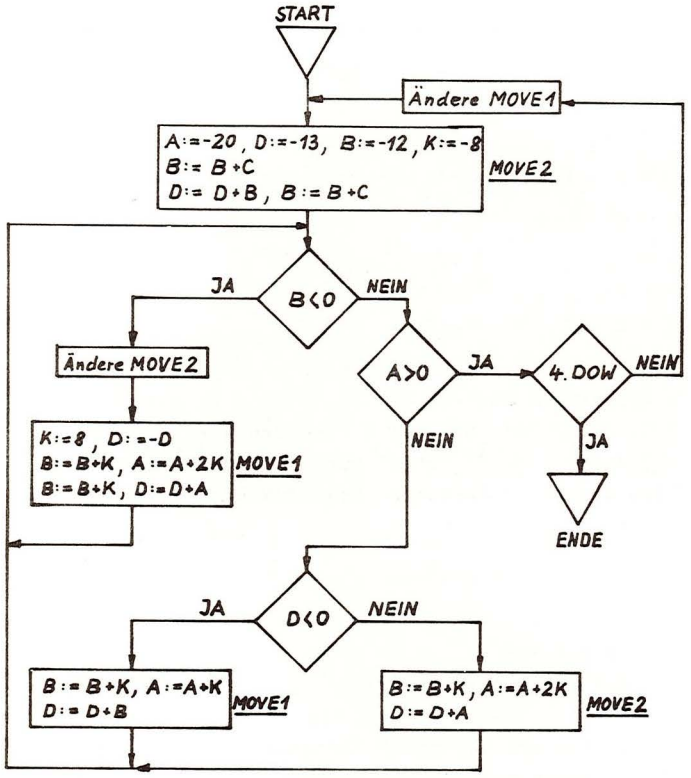


Bild 5.8: Flußdiagramm des Kreisalgorithmus

Der nach diesem Algorithmus realisierte DKC /39/ liefert gute Ergebnisse. Die Kreise sind unabhängig vom Radius gleich hell. Die Schreibdauer t_k ergibt sich aus der Zahl der Iterationsschritte

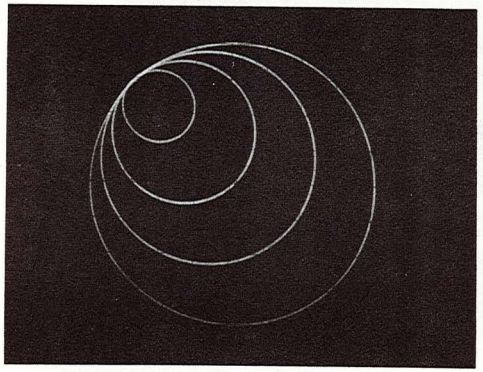


Bild 5.9: Kreise ohne Mittelpunktskorrektur

$N = 8n = 8\ R/\sqrt{2}$ und der Taktfrequenz f_T zu

$$t_k = \frac{24n + 9{,}5}{f_T} \qquad (5.15)$$

Beispiel 14: Bildschirmdurchmesser D = 1000 = 2 R

f_T = 20 MHz

t_k = 1 ms

Der DKG besitzt eine Mittelpunktschaltung /40/, die es erlaubt, bei variablem Radius konzentrische Kreise um einen beliebigen Mittelpunkt zu zeichen (Bild 5.10). Kreissegmente

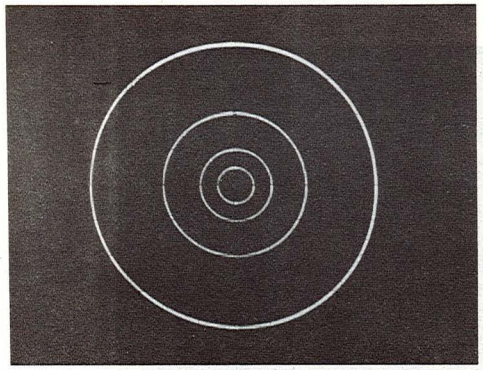


Bild 5.10: Konzentrische Kreise um beliebigen Mittelpunkt

können durch Angabe des Anfangs- und Endwinkels gezeichnet werden (Bild 5.11). Die Bilder 5.12 und 5.13 demonstrieren die Wirkung der eingebauten "Wrap-Around" - Unterdrückung. Mit dem DKG können also Kreisbögen gezeichnet werden, die durch Radius, Mittelpunkt und Anfangs- und Endwinkel spezifiziert sind. Die Kreispunkte werden mit einer Wortlänge von 13 bit berechnet. Durch entsprechende Bewertung mit

den DAU können deshalb ohne wesentliche Qualitätsminderung Kreisbögen gezeichnet werden, deren Radien maximal das Vierfache des Bildschirmdurchmessers betragen. Der kleinste darstellbare Kreis ist durch C = 4 gegeben und besitzt den "Radius" R = $\sqrt{2}$.

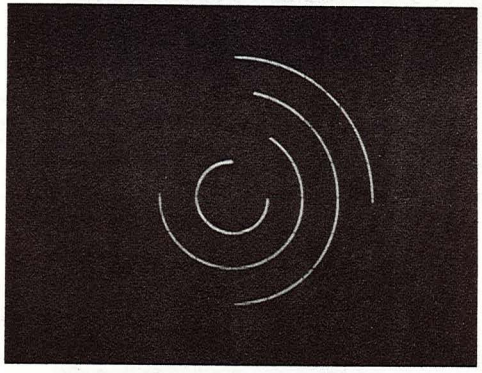


Bild 5.11: Kreissegmente

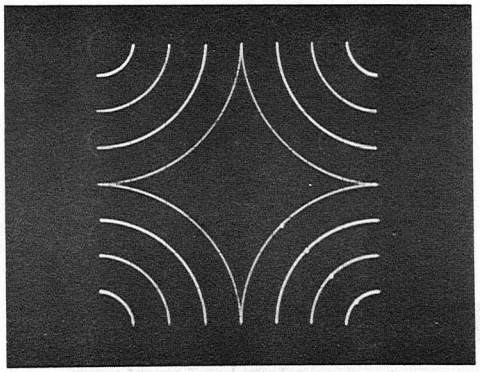


Bild 5.12: Kreise ohne "Wrap-around" Unterdrückung

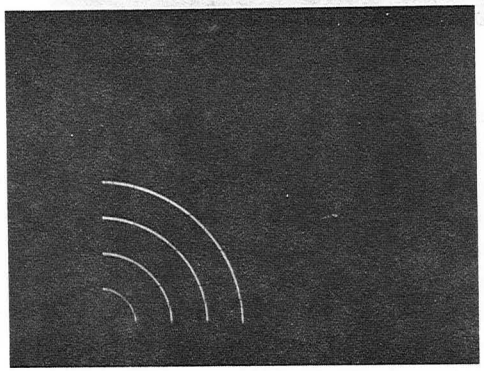


Bild 5.13: "Wrap-around" unterdrückt

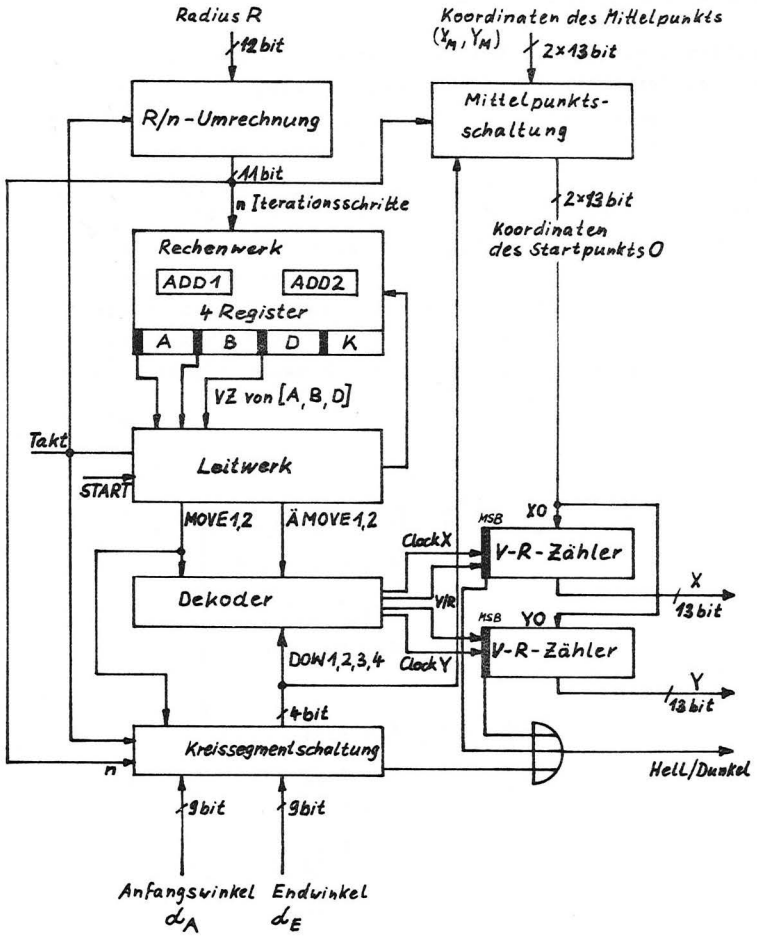


Bild 5.14: Blockschaltbild des DKG

5.3 Digitaler Matrizenmultiplizierer DMM

Hochleistungsdisplays besitzen zur Durchführung der linearen Transformationen in Echtzeit einen Matrizenmultiplizierer. Durch die Verwendung von homogenen Koordinaten lassen sich die Rotation, Skalierung, Translation und Perspektive eines Vektors $[x, y, z, w]$ durch Multiplikation mit einer (4x4)-Matrix berechnen. In /32/ sind die Ergebnisse einer vergleichenden Untersuchung der Möglichkeiten zur Realisierung der (4x4)-Matrix angegeben. Das dort vorgeschlagene Konzept eines digitalen Matrizenmultiplizierers (DMM) soll hier ausgeführt und erweitert werden. Der DMM berechnet in homogenen Koordinaten

$$p' = p \cdot T$$

$$[x',y',z',w'] = [x,y,z,w] \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad (5.16)$$

$$\begin{aligned} = \; & x \, [a_{11} \; a_{12} \; a_{13} \; a_{14}] \\ + \; & y \, [a_{21} \; a_{22} \; a_{23} \; a_{24}] \\ + \; & z \, [a_{31} \; a_{32} \; a_{33} \; a_{34}] \\ + \; & w \, [a_{41} \; a_{42} \; a_{43} \; a_{44}] \end{aligned} \qquad (5.17)$$

Es sind 16 Multiplikationen und Additionen auszuführen. Die Geschwindigkeit, mit der diese Operationen ablaufen müssen, hängt von der mittleren Vektorschreibdauer des Sichtgeräts ab. Sie wurde für den Entwurf des DMM mit 10 µs angenommen. Als Multiplizierbaustein stand der 2-Komplementmultiplizierer AM 2505 zur Verfügung, der die

Multiplikation von zwei 16 bit Wörtern in 135 ns /41/ gestattet. Damit kann man alle Multiplikationen nacheinander ausführen und sich auf den Einsatz eines Multiplizierers beschränken /42/. Die transformierten Koordinaten werden - obwohl die Bildschirmauflösung nur 10 bit verlangt - mit einer Wortlänge von 16 bit berechnet. Dies geschieht aus folgenden Gründen:

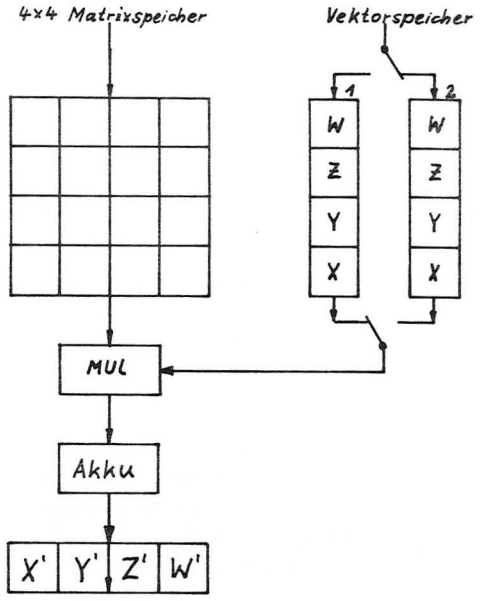


Bild 5.15: Blockschaltbild des DMM

1) Bilder werden oft durch mehrfache Verschachtelung von Unterbildern (subpictures) erzeugt. Durch die Genauigkeit der Matrix-Transformation wird verhindert, daß die Fehlerakkumulation in den Koordinaten

$$p' = p \cdot T_1 \cdot T_2 \cdots T_n$$

die Genauigkeit der Darstellung auf dem Bildschirm herabsetzt.

2) Der Zahlenbereich der Objektkoordinaten in der Datenstruktur (Page-Koordinaten) ist meist um ein Vielfaches größer als der des Bildschirms. Unabhängig von der Bildschirmausgabe sollen auch die Objektkoordinaten ohne Genauigkeitsverlust transformiert werden können.

3) Der DMM ist eine teure spezielle Hardware, deren arithmetische Möglichkeiten aus wirtschaftlichen Erwägungen dem Display-Rechner zur Verfügung stehen und an dessen Wortlänge - meist 16 bit - angepaßt sein sollten.

5.3.1 (4x4)-Matrixspeicher

Die Komponenten des DMM wurden so gewählt, daß die wesentlichen Verzögerungszeiten durch den Multiplizierer verursacht werden und bei eingeschriebener (4x4)-Matrix der Multiplizierer ständig arbeitet. Der Matrixspeicher besteht aus 4 Speicherbausteinen mit jeweils 16 4bit-Wörtern. Durch Parallelschaltung wird die Wortlänge von 16 bit erreicht. Die eingesetzten Halbleiter-RAMs 3101A (Random Access Memory) haben eine maximale Zugriffszeit von 35 ns. Der Speicher wird über einen 4-bit-Dinärzähler adressiert. Durch einen Sprungbefehl ist der Zähler parallel setzbar, so daß zum selektiven Verändern der Matrix jede beliebige Zelle angewählt werden kann /42/.

5.3.2 Multiplizierer für (16x16) bit

Entsprechend den Ausführungen in /41/ wird der Multiplizierer aus 2 Feldern (Arrays, zweidimensionalen Schaltketten) von Multiplizierbausteinen AM 2505 aufgebaut, die parallel die Produkte

$$X_{0\text{-}15} \cdot Y_{0,1,4,5,8,9,12,13} \quad \text{und}$$
$$X_{0\text{-}15} \cdot Y_{2,3,6,7,10,11,14,15}$$

berechnen. Da nur 16 Stellen des Produktes weiterverarbeitet werden, können die Multiplizierer eingespart werden, deren Teilprodukte nur zu den letzten 16 Bits einen Beitrag liefern. Die Ergebnisse der beiden Felder werden in schnellen Addierern AM 9340 mit interner paralleler Übertragsberechnung (carry look ahead) addiert. Das Produkt liegt in 2-Komplementdarstellung nach ca. 200 ns vor. Der aus den Datenblättern ermittelte ungünstigste Fall (worst-case) braucht 260 ns.

5.3.3 Akkumulator und Ergebnisspeicher

Der Akkumulator ist wie der Multiplizierer nur einmal vorhanden. Er besteht aus einem 16 bit Addierer und einem 16 bit Hilfsregister. Die Eingänge des H-Registers sind mit den Ausgängen des Addierers vorbunden. Die Ausgänge des H-Registers belegen die Addiereingänge für einen Summanden. Die Eingänge für den zweiten Summanden sind mit den Ausgängen des Multiplizierers beschaltet. Nach jeweils 4 Rechenschritten steht im H-Register eine Komponente des transformierten Vektors p'. Sie wird in einen der vier 16 bit-Ergebnisspeicher übernommen.

Damit der Multiplizierer bei eingelesener (4x4)-Matrix ständig rechnen kann, werden die zu transformierenden Vektoren p im DMM in einen Pufferspeicher mit kleiner Zugriffszeit übernommen. Der Pufferspeicher besteht aus 2 Teilspeichern, die alternierend geladen und gelesen werden.

Die Rechendauer für die vollständige (4x4) Matrixmultiplikation beträgt im ungünstigsten Fall 8 μs. Im Betrieb des aufgebauten DMM wurden 6 μs gemessen.

5.4 Digitaler Dividierer

Zur Berechnung der perspektivischen Projektion und zur Ermittlung der Bildschirmkoordinaten aus den homogenen Koordinaten müssen Divisionen durchgeführt werden. Es ist wünschenswert, daß die Zahl der darstellbaren Vektoren bei einer Echtzeitberechnung der Perspektive nicht wesentlich reduziert wird. Daraus folgt, daß die Rechenzeit für die Division erheblich kleiner sein muß als für die (4x4) Matrix-Transformation. Diese Forderung ist erfüllt, wenn es gelingt, die Divisionszeit in die Größenordnung der Multiplikationszeit zu bringen. Zu diesem Zweck wurden einige Divisionsverfahren bezüglich Rechenzeit und Hardware-Aufwand untersucht /43/ und zwei schnelle Dividierer entworfen.

5.4.1 Division mit zweidimensionaler Schaltkette

In der Literatur /44/, /45/ sind Realisierungen der "Division ohne Rückstellen des Restes" (nonrestoring division) mit zweidimensionalen Schaltketten (cellular arrays) angegeben. Das einzelne Kettenglied besteht im wesentlichen aus einem kontrollierbaren Addierer-Subtrahierer KAS. An einem Zahlenbeispiel werden die notwendigen Funktionen eines Kettengliedes gezeigt. Da die Bildschirmkoordinaten normiert sind, werden nur Zahlen im Bereich $0 \leq Z \leq 1$ betrachtet. Die Frage des Vorzeichens wird später zusammen mit dem erweiterten DMM behandelt.

Beispiel 15: Dividend $X = 0.\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1$
Divisor $Y = 0.\ 1\ 1\ 0\ 1$
Quotient $Q = Q_0.\ Q_1\ Q_2\ Q_3\ Q_4$
Rest $S: = S \pm Y \cdot 2^{-k}$, $k = 0,1,2,3,4$

0.01110101 : 0.1101 = 0.1001	S: = X
-0.1101	S: = S-Y
(1)10100	$S < 0 \rightarrow Q_0 = 0$
+ 0.1101	S: = S+Y/2
(0)00011	$S > 0 \rightarrow Q_1 = 1$
-0.1101	S: = S-Y/4
(1)01100	$S < 0 \rightarrow Q_2 = 0$
+0.1101	S: = S+Y/8
(1)10011	$S < 0 \rightarrow Q_3 = 0$
+0.1101	S: = S+Y/16
(0)0000	$S \geq 0 \rightarrow Q_4 = 1$

Nach jeder Addition oder Subtraktion wird der Divisor um eine Stelle nach rechts geschiftet. Diese Operation wird durch die Verdrahtung der Kettenglieder festgelegt. Für eine Zeile der Schaltkette müssen alle Glieder mit einer Steuerleitung E auf die vorgeschriebene Arbeitsweise (Addition oder Subtraktion) umschaltbar sein. Der Divisor muß in Schiftrichtung von Zeile zu Zeile durchgeschaltet werden. Damit läßt sich folgendes Kettenglied entwerfen:

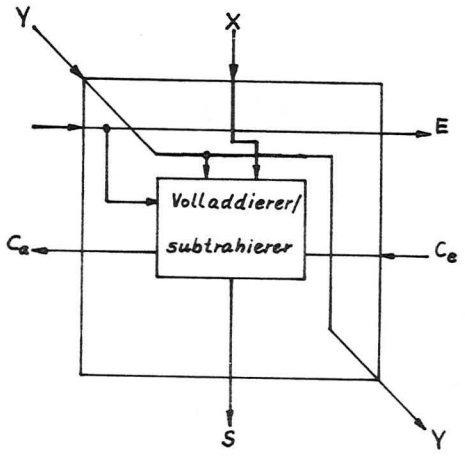
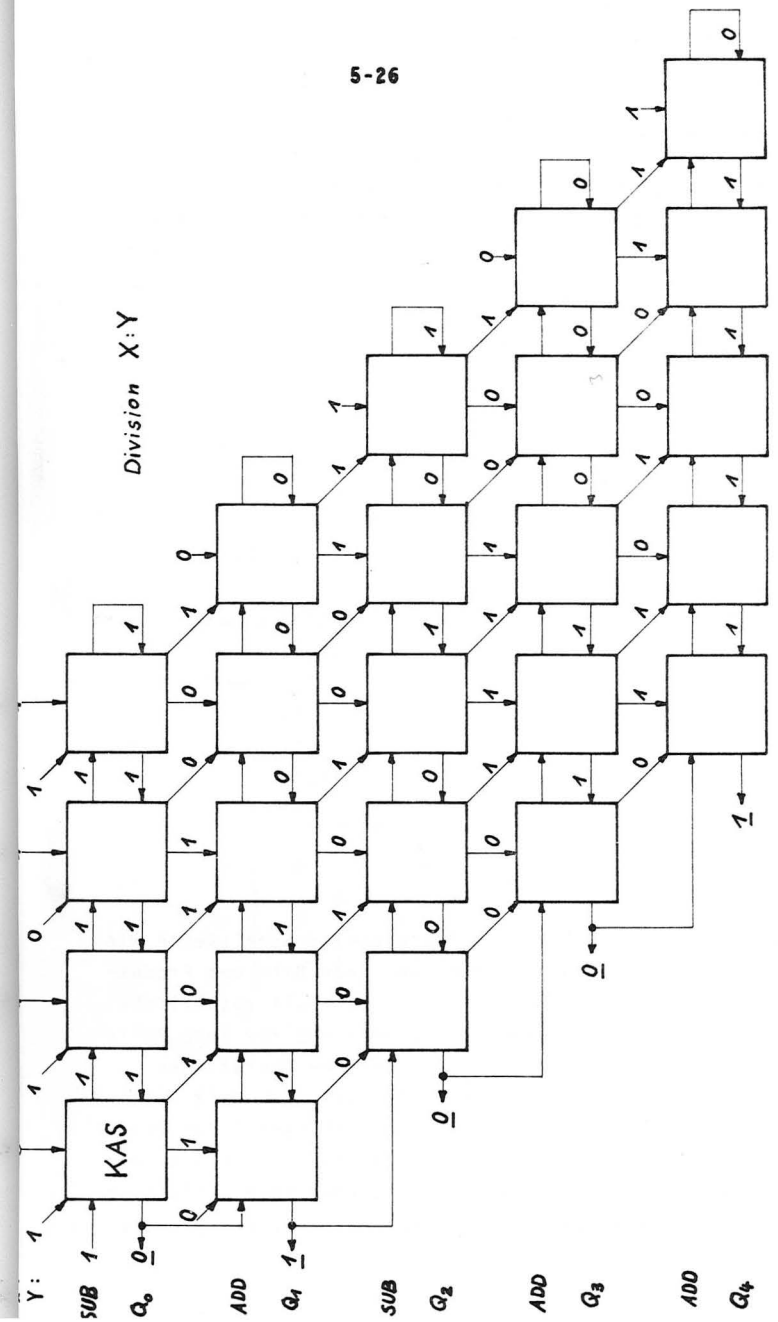


Bild 5.16: Kontrollierbarer Addierer-Subtrahierer KAS

Das Vorzeichen des Restes S bestimmt die Größe des Quotientenbits und die nächste arithmetische Operation. Es wird als letzter Übertrag einer Zeile generiert. Die Subtraktion wird durch Addition des 1-Komplements des Divisors ausgeführt. Zur Korrektur muß dazu im ersten Kettenglied eine 1 addiert werden. **Bild 5.17 zeigt eine zweidimensionale Schaltkette, in die das Zahlenbeispiel 15 eingezeichnet ist.** Die erste auszuführende Operation ist bei der "Nonrestoring Division" die Subtraktion des Divisors. Die Steuerleitung E der ersten Zeile kann deshalb fest auf "1" gelegt werden. Die Anzahl der Zeilen ist gleich der gewünschten Wortlänge N des Quotienten einschließlich der Stelle vor dem Binärpunkt. Die Anzahl der Spalten pro Zeile hängt von der Wortlänge N des Divisors ab. Zur Division sind N·N-1 Kettenglieder notwendig. Ist die Zeit für eine Addition/Subtraktion im KAS gleich T, so berechnet sich die Zeit T_D für die Division näherungsweise nach $T_D \approx N \cdot T$. Ist N=N=16 und wird als arithmetisches Rechenelement der integrierte Schaltkreis 74S181/182 mit einer typischen Verzögerungszeit von 2ns für eine 16 Bit Addition verwendet, so wird T_D = 320 ns. Der Restteil



5.4.2 Kombinierter Dividierer-Multiplizierer DMD

Die im vorangegangenen Abschnitt beschriebene zweidimensionale Schaltkette zur Durchführung der "Division ohne Rückstellen des Restes" beruht auf der Möglichkeit, wahlweise zu addieren oder zu subtrahieren und eine feste Schiftbewegung beizubehalten. Eine Multiplikation läßt sich auf die gleiche Weise durchführen, wie folgendes Zahlbeispiel zeigt:

```
0.01110101 : 0.1101 = 0.1001
-0.1101
(1) 10100
 + 0.1101
   (0)00011
    -0.1101
     (1)01100
      +0.1101
       (1)10011
        +0.1101
         (0)0000
```

```
0.1101 · 0.1001
 0.0000
+0.0000
 0.0000
+ 01101
 0.01101
+ 00000
 0.011010
+ 00000
 0.0110100
+ 01101
 0.01110101
```

Verwendet man zur Multiplikation dieselbe Schaltkette wie zur Division, so gehen die höherwertigen Bits des Produktes verloren. Deshalb müssen für diesen Fall zusätzliche Addierer A vorgesehen werden. Außerdem muß für jede Zeile eine Kontrolleinheit KE die Steuerung der Operationen Addition/Subtraktion übernehmen. Der KAS nach Bild 5.16 muß so erweitert werden, daß der Multiplikant X für den Multiplikator 0 unter Umgehung des Addierer/Subtrahierers zur nächsten Zeile durchgeschaltet werden kann. Bild 5.18 zeigt den modifizierten KAS. In Abhängigkeit der Signale

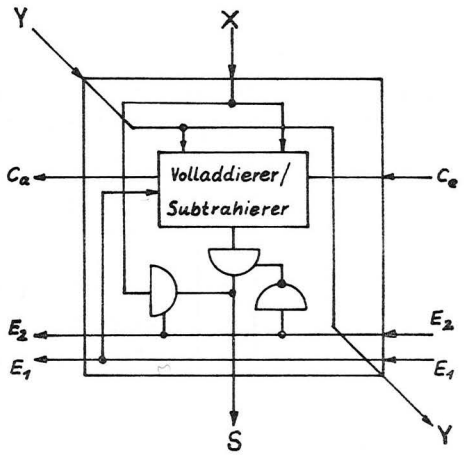


Bild 5.18: Modifizierter KAS zur Division/Multiplikation

E_1, E_2 werden folgende Operationen ausgeführt:

E_1	E_2	S
0	0	$X + Y + C_e$
0	1	X
1	0	$X - Y - C_e$

Die Kontrolleinheit KE steuert die Operation der KAS einer Zeile nach 3 Kriterien:

1) Betrieb Division oder Multiplikation
2) Bei Division: Welches Ergebnis Q lieferte die vorhergehende Zeile?
3) Bei Multiplikation: Welchen Wert besitzt das zugehörige

Dementsprechend läßt sich für die Funktion der Steuerleitungen E_1, E_2 des KAS eine Wahrheitstabelle aufstellen:

Multiplikation: K = 0

P = Y · X

Q	X	S	E_1	E_2
0	0	X	0	1
1	0	X	0	1
0	1	$X+Y+C_e$	0	0
1	1	$X+Y+C_e$	0	0

Division: K = 1

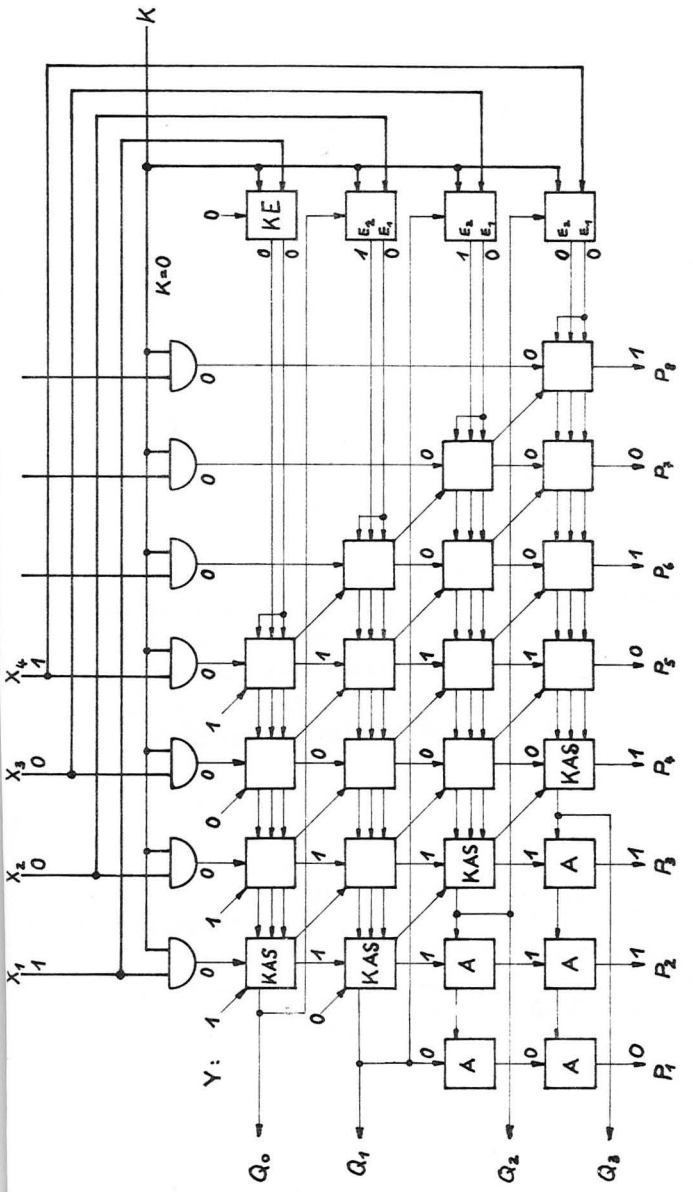
Q = X/Y

0	0	$X+Y+C_e$	0	0
1	0	$X-Y-C_e$	1	0
0	1	$X+Y+C_e$	0	0
1	1	$X-Y-C_e$	1	0

Hieraus erhält man für die Steuerfunktionen

$$E_1 = K\,Q \quad \text{und} \quad E_2 = \overline{K}\,\overline{X}$$

Für den Fall der Multiplikation muß an den X-Eingängen des DMD Null anliegen. Dies wird durch ein vorgeschaltetes UND-Gatter erreicht. Der DMD läßt sich damit als Erweiterung des in Bild 5.17 gezeigten Dividierers darstellen (Bild 5.19). Neben den X-Eingangsgattern und den Kontrolleinheiten KE müssen zur Berechnung des Zahlenbeispiels nur 5 Addierer A zusätzlich aufgewandt werden. Die Anzahl der zur Multiplikation benötigten Kettenglieder ist bei einer Wortlänge K des Multiplikanten bzw. L des Multiplikators einschließlich der Stelle vor dem Binärpunkt (K(L-1)-1) KAS plus L(L-3)/2 A für L > 3.



Multiplikation Y·X = 0.1101 × 0.1001

Die Rechenzeit T_M ist abhängig von der Anzahl E der Einsen im Multiplikator. Die Verzögerung durch die Gatterlaufzeiten T_G beträgt pro Zeile bei Einsatz von Schottky-TTL ca. 6 ns. Zusammen mit der Additionszeit T = 20 ns ergibt sich für die Multiplikationszeit

$$T_M = (L-1)\left(\frac{E}{L-1}\,T + T_G\right) . \qquad (5.18)$$

Für E = 10 z.B. beträgt $T_M \cong 200$ ns und liegt damit über der Multiplikationszeit nach Kapitel 5.3.2. Die Rechenzeit für die 16 bit Division erhöht sich durch die zusätzlichen Gatterlaufzeiten auf

$$T_D \cong M\,(T + T_G) = 420 \text{ ns.} \qquad (5.19)$$

Der vorgeschlagene DMD bietet sich auf Grund seiner zellularen Struktur zur Großintegration an. Eine Realisierung aus einzelnen Elementen auf der Basis der arithmetischen Recheneinheiten 74S181/182 war aus Kostengründen (ca. 10.000 DM im Juli 73) nicht möglich. Deshalb wurde nach einem Divisionsverfahren gesucht, das auf der Basis eines schnellen Multiplizierers arbeitet, wie er im Matrizenmultiplizierer bereits zur Verfügung stand.

5.4.3 Iteratives Divisionsverfahren

Der Quotient $Q = \frac{X}{Y}$ wird durch Multiplikation von X mit dem Reziprokwert von Y gewonnen. Das allgemeine Divisionsproblem wird damit auf die einfachere Reziprokwertbildung reduziert. Wird der Wertebereich des Nenners Y durch Normalisierung auf $1 > Y \geq 0{,}5$ beschränkt, so läßt sich mit

$$Y = 1 - Z \,, \quad Z \leq 0{,}5 \qquad (5.20)$$

folgende Reihenentwicklung angeben:

$$\frac{1}{Y} = \frac{1}{1-Z} = 1 + Z + Z^2 + Z^3 + \ldots .$$

Der Quotient wird damit

$$Q = \lim_{n\to\infty} X \cdot \prod_{i=0}^{n-1} (1+Z^{2^i})$$

$$= X \cdot \lim_{n\to\infty} \prod_{i=0}^{n-1} A_i \quad \text{mit} \quad A_i = (1+Z^{2^i}) \qquad (5.22)$$

Die Zahl n ergibt sich aus der gewünschten Genauigkeit des Ergebnisses. Dies läßt sich abschätzen durch Erweiterung des Nenners mit der Reihenentwicklung nach Gl. (5.21):

$$Y \cdot \prod_{i=0}^{n-1} A_i = (1-Z) \prod_{i=0}^{n-1} (1+Z^{2^i})$$

$$= (1-Z^{2^n}) = 1 - \varepsilon \qquad (5.23)$$

Da $Z < 1$ ist, konvergiert der Ausdruck (5.23) quadratisch gegen 1. In gleicher Weise konvergiert dann (5.22) gegen den gesuchten Quotienten Q. Den maximalen Fehler ε_{max} erhält man bei festem n für

$Z_{max} = 0{,}5$: $\qquad \varepsilon_{max} = 0{,}5^{2^n} = 2^{-2^n}$ (5.24)

Für n = 4 ist $\qquad \varepsilon_{max} = 2^{-16}$, d.h.:

$$Y \cdot \prod_{i=0}^{3} A_i \geq 1 - 2^{-16} . \qquad (5.25)$$

Der Quotient $\qquad Q = X \cdot \prod_{i=0}^{3} A_i = X \cdot A_0 \cdot A_1 \cdot A_2 \cdot A_3$

ist dann mit einem relativen Fehler $\varepsilon_{max} = 2^{-16}$ behaftet.

Die Faktoren A_i werden mit Gl. (5.20) aus dem Nenner Y berechnet nach

$$A_i = (1+Z^{2^i}) = (1+(1-Y)^{2^i}) \qquad (5.26)$$

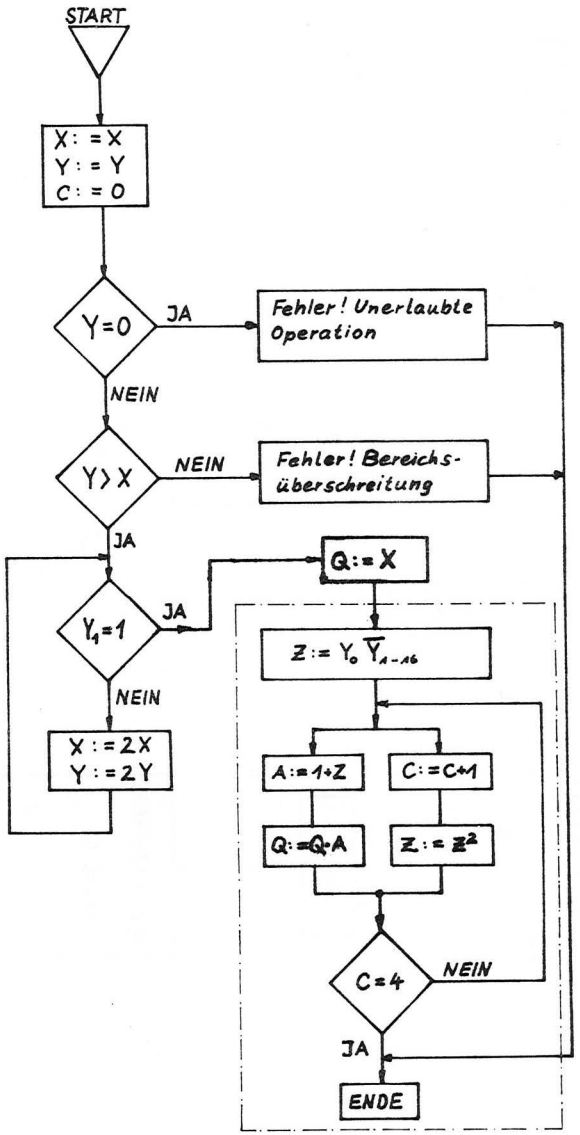
oder mit Hilfe des 2-Komplements rekursiv definiert:

$$
\begin{aligned}
A_o &= (1+(1-Y) = 2 - Y = 2 - Y_o = \tilde{Y}_o \\
Y_1 &= A_o\, Y_o \\
Y_i &= A_{i-1} Y_{i-1} \\
A_i &= 2 - Y_i = \tilde{Y}_i
\end{aligned}
\qquad (5.27)
$$

Für jeden Iterationsschritt sind zwei Multiplikationen und eine Addition oder Komplementbildung durchzuführen. Der vorhandene Multiplizierer liefert nur 16 bit Produkte, was die Konvergenz aber nicht beeinträchtigt /46/. Aus den Gln. (5.25) und (5.26) ist zu erkennen, daß die beiden für eine Iteration notwendigen Multiplikationen unabhängig voneinander durchgeführt werden können. Stehen zwei Multiplizierer zur Verfügung, so kann die Rechenzeit zur Quotientenbildung auf die Hälfte reduziert werden. Diese Möglichkeit wurde bei dem realisierten Dividierer /47/ wahrgenommen. Bild 5.20 zeigt das Flußdiagramm des Divisionsalgorithmus. Zur Vermeidung von Überläufen sind Abfragen auf $Y > X$ und $Y = 0$ eingefügt. Y wird durch Schiften normalisiert. Die Rechenzeit wird im wesentlichen durch die in Gl. (5.25) nacheinander auszuführenden 4 Multiplikationen (4x200 ns) bestimmt. In /47/ wurden die minimale (Y ist bereits normiert) und maximale (Y muß über 14 Stellen geschiftet werden) Rechenzeit mit

$$
\begin{aligned}
T_{min} &= 1120 \text{ ns und} \\
T_{max} &= 1283 \text{ ns angegeben.}
\end{aligned}
$$

Dieser Dividierer läßt sich in 3 Punkten wesentlich verbessern: Prinzipiell durch Verbesserung des Startwerts $A_o = 1 + Z$ der Iteration, technisch durch Verwendung schnellerer Multiplizierer und Umgehung der Schifts durch Einsatz von Multiplexern für die Normalisierung.



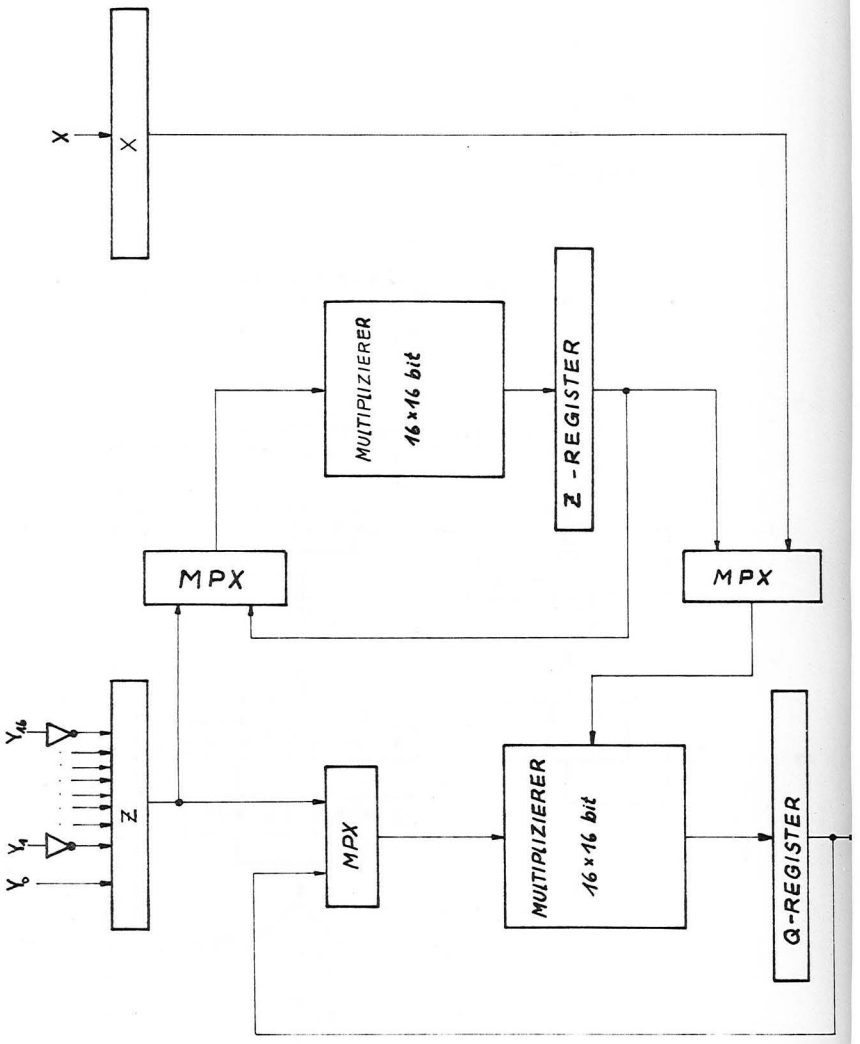


Bild 5.21: Blockschaltbild des Dividierers

5.4.4 Modifiziertes iteratives Divisionsverfahren

Bei dem zuvor beschriebenen iterativen Divisionsverfahren wurde die Zahl der notwendigen Iterationen n unter der Annahme des ungünstigsten Startwerts mit $Z_{max} = 0{,}5$ für einen Fehler $\varepsilon = 2^{-16}$ bestimmt (Gln. (5.24) und (5.25)). Umgekehrt kann man nach dem Startwert als Funktion der Zahl der Iterationen bei vorgegebenem Fehler fragen. Aus Gl. (5.24) erhält man

$$\varepsilon = 2^{-16} = Z^{2^n} \qquad (5.28)$$

Zum Beispiel sei

Z	n = Anzahl der Iterationen
2^{-1}	4
2^{-2}	3
2^{-4}	2
2^{-8}	1
2^{-16}	0

Steht der Reziprokwert 1/Y mit 16 bit Wortlänge in einem Festwertspeicher (ROM) zur Verfügung, so ist keine Iteration notwendig. Die dazu notwendige Speichergröße bei dem gewählten Zahlbenbereich $1 > Y \geq 0{,}5$ umfaßt $15 \cdot 2^{15}$ Werte und ist technisch nicht vernünftig zu realisieren. Dagegen sind ROMs mit einer Kapazität von $8 \cdot 2^8$ bit erhältlich.

Sei $I_o = 1/Y_R$ der Reziprokwert des auf 9 bit aufgerundeten Nenners $Y_R > Y$. Durch Erweitern des Quotienten erhält man

$$Q = \frac{X \cdot I_o}{Y \cdot I_o} = \frac{X_o}{Y_o} \qquad (5.29)$$

mit $1 > Y_o \geq 1 - 2^{-8}$ Damit wird

$$0 < Z = 1 - Y_o \leq 2^{-8} \qquad (5.30)$$

und mit Gl.(5.22) $A_o = 1 + Z \qquad (5.31)$

Der Quotient läßt sich nun mit einer Iteration berechnen Gl. (5.22):

$$\begin{aligned} Q &= X_o \cdot A_o \\ &= X \cdot I_o \cdot (1+Z) \\ &= X \cdot I_o \cdot (1+1-Y_o) \\ Q &= X \cdot I_o \cdot (2-YI_o) \qquad (5.32) \end{aligned}$$

Die beiden Multiplikationen $X \cdot I_o$ und $Y \cdot I_o$ können gleichzeitig ausgeführt werden. Der Wert $(2-YI_o)$ steht als 2-Komplement von $Y \cdot I_o$ schnell zur Verfügung. Die Divisionszeit für eine Genauigkeit von 16 bit ist im wesentlichen durch 2 nacheinander auszuführende Multiplikationen gegeben. Jede weitere Iteration zur Verbesserung der Genauigkeit benötigt ca. eine Multiplikationszeit (2 Multiplikationen parallel). Die Genauigkeit des Startwerts kann unter Beachtung des Wertebereichs $1 > Y \geq 0{,}5$ leicht um eine Stelle verbessert werden. Die erste Stelle nach dem Binärpunkt ist $Y_I = 1$. Es kann deshalb der Reziprokwert $1/0.1Y_{\overline{II-IX}} = 1.I_{\overline{I-VIII}}$ in einem ROM (2^8x8) bit = 2K bit bereit gestellt werden.

Aus Gl. (5.32) ergibt sich der Reziprokwert des Nenners

zu $1/Y = I_1 = I_o\,(1+1-YI_o)$

$= I_o\,(1+Z) = I_o(1+R_1)$

Hieraus läßt sich eine modifizierte Newton'sche Approximation /48/ des Reziprokwertes $I = f(Y) = 1/Y$ ablesen:

$$\begin{aligned} \text{Startwert: } I_o &= f(Y) \\ R_1 &= 1-I_oY \qquad I_1 = (1+R_1)\,I_o \\ R_{i+1} &= 1-I_iY \qquad I_{i+1} = (1+R_{i+1})I_i \end{aligned} \qquad (5.34)$$

Nach der Bestimmung des Reziprokwertes muß noch mit dem Zähler multipliziert werden, um den Quotienten zu erhalten. Dieses Verfahren ist gegenüber dem zuvor beschriebenen langsamer, da die Werte R_{i+1} und I_{i+1} voneinander abhängen und deshalb die beiden notwendigen Multiplikationen nicht gleichzeitig ausgeführt werden können.

Die Division nach Gl. (5.32) wurde in PL/1 programmiert und lieferte unter Berücksichtigung der Wortlänge von 16 bit folgende Ergebnisse für die Reziprokwertbildung ($X=1$, $0{,}5 \leq Y < 1$) /49/:

Maximaler Fehler: $6{,}28 \cdot 10^{-5}$ bei $Y = 0{,}5$

Durchschnittlicher Fehler: $1{,}5 \cdot 10^{-5}$

Minimaler Fehler: 0

Die beiden 16 x 16 bit Multiplizierer werden aus den 4 x 4 bit Multiplizierbausteinen 74284/85 aufgebaut, deren mittlere Rechenzeit für ein 32 bit Produkt mit 100 ns angegeben wird. Diese Multiplizierer können nur Beträge verarbeiten, so daß zur Gesamtrechenzeit noch die Zeit für eventuelles Komplementieren addiert werden muß.

Die Normalisierung von Y kann mit schnellen Multiplexern in 50 ns /47/ durchgeführt werden. Die maximale Rechenzeit für die Division wird damit 350 ns betragen. Die Forderung nach einer Divisionszeit in der Größenordnung der Multiplikationszeit ist damit erfüllt. Die flackerfrei darstellbare Anzahl von Vektoren wird bei einer Echtzeitberechnung der Perspektive nicht wesentlich reduziert. Die Bilder 5.22 und 5.23 zeigen das Flußdiagramm und das Blockschaltbild des Dividierers.

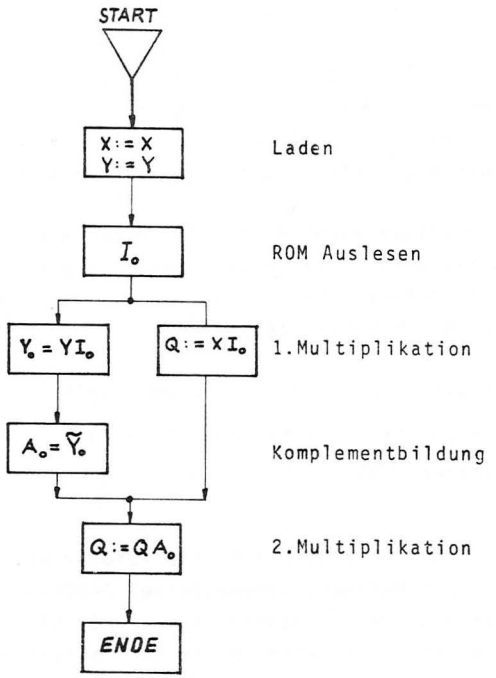


Bild 5.22: Flußdiagramm des Divisionsalgorithmus

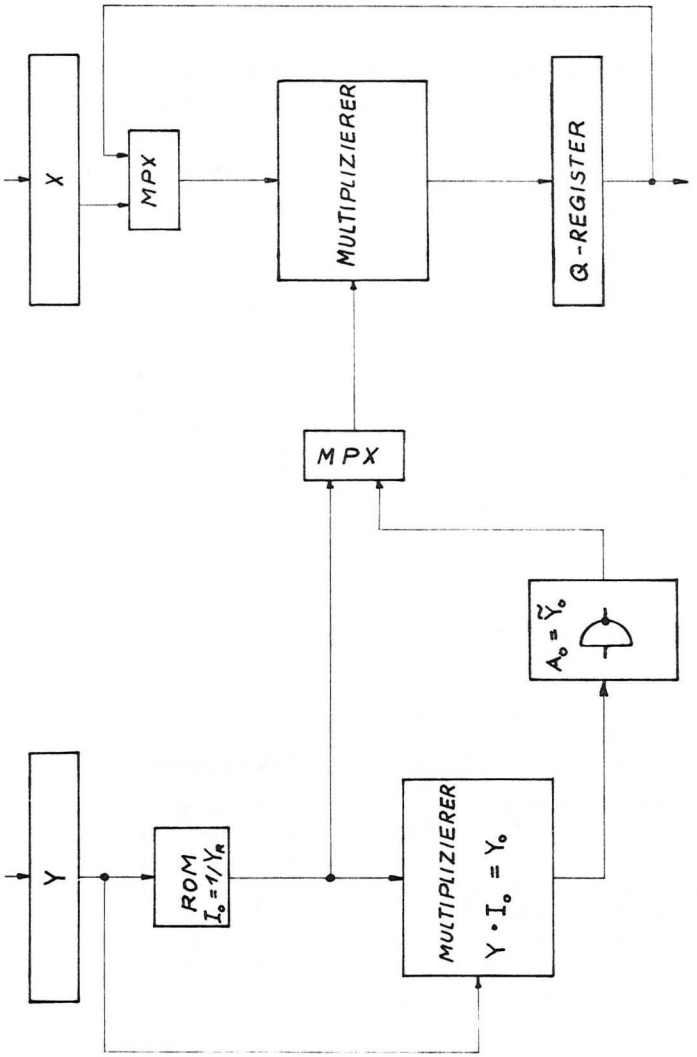


Bild 5.23: Blockschaltbild des Dividierers

5.5 Verbesserter Matrizenmultiplizierer

Für die Matrizenmultiplikation stehen nun zwei schnelle Multiplizierer zur Verfügung. Die Transformation eines 4-Komponentenvektors einschließlich der Division kann in 4 µs durchgeführt werden. Bild 5.24 zeigt das Blockschaltbild des verbesserten Matrizenmultiplizierers.

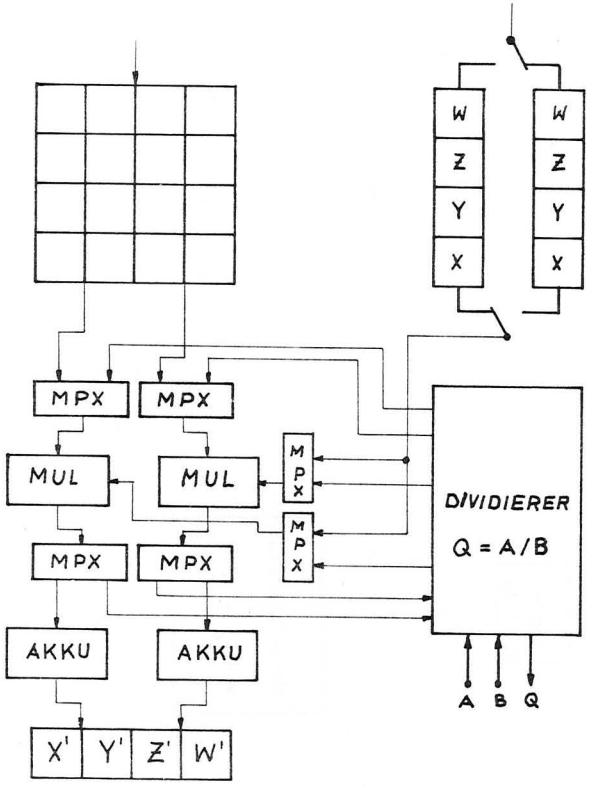


Bild 5.24: Verbesserter Matrizenmultiplizierer

5.6 Kurven- und Flächenberechnung

Die beschriebenen Verfahren zur Kurven- und Flächendarstellung lassen sich alle in die Form der Gl. (3.15)

$$Q(u,v) = U^T \bar{P} V$$

bringen. Diese Tatsache legt den Gedanken nahe, mit einer speziellen Hardware diese Gleichung auszuwerten. Zwei wichtige Vorteile sind damit zu erzielen:

1) Im Bildwiederholungsspeicher muß nur die Flächendefinition gespeichert sein.

2) Die flackerfrei darstellbare Bildinformation wird größer.

Flächen werden auf dem Display durch u- und/oder v-Rasterlinien dargestellt. Diese Kurven sind aus obiger Gleichung prinzipiell auf zwei Arten zu gewinnen:

a) Inkremental nach dem Newton'schen Differenzenverfahren.

b) Durch Multiplikation mit einem Matrizenmultiplizierer.

5.6.1 Differenzenverfahren

Die Vorwärtsdifferenz einer Funktion Q(u) ist bei gleichmäßiger Inkrementierung der unabhängigen Variablen u definiert als

$$\Delta Q(u) = Q(u+\delta) - Q(u) \quad , \quad \delta = \frac{1}{N} \qquad (5.35)$$

$$\Delta(\Delta Q(u)) = \Delta^2 Q = \left[Q(u+2\delta) - Q(u+\delta)\right] - \left[Q(u+\delta) - Q(u)\right]$$

$$\Delta^2 Q(u) = Q(u+2\delta) - 2\,Q(u+\delta) + Q(u)$$

$$\Delta^3 Q(u) = Q(u+3\delta) - 3\,Q(u+2\delta) + 3\,Q(u+\delta) - Q(u)$$

Durch Umschreiben von Gl. (5.35) wird

$Q(u) \longrightarrow Q_n$, $Q(u+\delta) \longrightarrow Q_{n+1}$ erhält man

$$\Delta Q_n = Q_{n+1} - Q_n$$

$$\Delta^2 Q_n = Q_{n+2} - 2\,Q_{n+1} + Q_n$$

$$\Delta^3 Q_n = Q_{n+3} - 3\,Q_{n+2} + 3_{n+1} - Q_n \qquad (5.36)$$

gesucht ist Q_{n+1}: Aus Gl. (5.36) folgt

$$\begin{aligned} Q_{n+1} &= Q_n + \Delta Q_n \\ \Delta Q_{n+1} &= \Delta Q_n + \Delta^2 Q_n \\ \Delta^2 Q_{n+1} &= \Delta^2 Q_n + \Delta^3 Q_n \\ \Delta^3 Q_{n+1} &= \Delta^3 Q_n = \text{konst} \end{aligned} \qquad (5.37)$$

Das Polynom ist also durch Angabe von Anfangswerten Q_o, ΔQ_o, $\Delta^2 Q_o$, $\Delta^3 Q_o$ völlig bestimmt und iterativ ohne Multiplikation berechenbar. Zur Berechnung eines Punktes sind nur 3 Additionen nötig, die gleichzeitig ausgeführt werden können.
Die Hardware-Realisierung des Differenzenverfahrens besteht aus einem Addierer-Register-Modul, der je nach Grad des zu berechnenden Polynoms entsprechend oft hintereinander geschaltet wird (Bild 5.25) /50/.

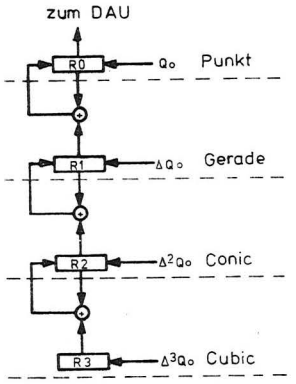


Bild 5.25: Berechnung eines Polynoms 3-ten Grades

Der ARM stellt eine Erweiterung des Vektorgenerators nach dem DDA-Prinzip dar. Bei einer Wortlänge von 16 bit ist die Rechenzeit für einen Iterationsschritt bei Einsatz von Look-ahead-Addierern in Schottky-TTL ca. 30 ns.

Das Differenzenverfahren ist bei der Berechnung von Polynomen exakt. Fehler treten nur durch Akkumulation von Fehlern in den Anfangswerten auf! Ist eine Kurve gegeben durch

$$Q(u) = U^T \overline{P} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} A \\ (4\times4) \end{bmatrix} \overline{P} \,,$$

so erhält man nach n Iterationen

$$\begin{bmatrix} Q_n \\ \Delta Q_n \\ \Delta^2 Q_n \\ \Delta^3 Q_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^n \cdot \begin{bmatrix} Q_o \\ \Delta Q_o \\ \Delta^2 Q_o \\ \Delta^3 Q_o \end{bmatrix} = V^n \cdot \begin{bmatrix} \Delta Q_o \end{bmatrix} \qquad (5.38)$$

mit den Anfangswerten

$$\begin{bmatrix} \Delta Q_o \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 6 & 2 & 0 & 0 \\ 6 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta^3 & 0 & 0 & 0 \\ 0 & \delta^2 & 0 & 0 \\ 0 & 0 & \delta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{C} A \overline{P} \qquad (5.39)$$

Mit der Matrix A werden die Interpolationspolynome bestimmt. Die Produktmatrix C ist dann bei fester Schrittweite δ für alle Kurven gleich, so daß ihre Elemente in einem ROM zur Verfügung gestellt werden können. Die Matrizenmultiplikation

$$\begin{bmatrix} \Delta Q_o \end{bmatrix} = C^* \overline{P}$$

kann im DMM ausgeführt werden. Über grafisches Tablett, Lightpen oder Potentiometer können die Punkte $\overline{P}$ verändert und der gewünschte Kurvenverlauf interaktiv angenähert werden.

Zur Flächendarstellung müssen für jede Rasterlinie neue Anfangswerte geladen werden, die ebenfalls iterativ berechnet werden können. Der Vorgang der Iteration wird mit Bild 5.26 erläutert:

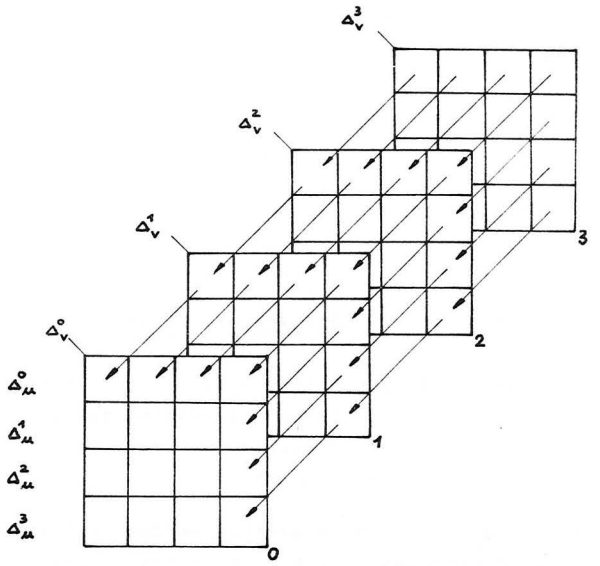


Bild 5.26: Berechnung der Rasterlinien

In Matrix O stehen die Koordinaten des Anfangspunktes $\overline{Q}(o,o)$ und die Differenzen in u-Richtung für v = o. Diese Differenzen sind Polynome 3-ten Grades in v, deren Differenzen in v-Richtung in den Matrizen 1 bis 3 stehen.

Die Fläche wird dann nach folgendem Algorithmus berechnet:

1) Laden der Matrizen M_0 - M_3
2) Lade M_0 in den ARM und berechne eine Rasterlinie
3) Führe die Iteration aus (bis n = N)

$$M_0 := M_0 + M_1$$
$$M_1 := M_1 + M_2$$
$$M_2 := M_2 + M_3$$
$$M_3 := M_3$$

4) Gehe nach 2)

5.6.1.1 Differenzenverfahren ohne Vektorgenerator

Nach dem Differenzenverfahren können Kurven- und Flächenpunkte mit dem ARM so schnell berechnet werden, daß sich ein Vektorgenerator erübrigt. Dieser Vorteil wird aber durch erhebliche Nachteile aufgehoben:

Die Schrittweite $\delta = 1/N$ muß so gewählt werden, daß die Darstellung der berechneten Punkte einen geschlossenen Linienzug ergibt. Dies ist der Fall, wenn N gleich der Länge der Kurve in Rastereinheiten ist. N kann mit einigem Aufwand näherungsweise aus der Länge des in die X-Y-Ebene projizierten Polygons bestimmt werden. Zur Berechnung der Anfangswerte muß ein ROM mit vielen Produktmatrizen C (Gl.(5.39)) für unterschiedliche δ-Werte zur Verfügung gestellt werden.

Auf Grund der Parametrisierung werden Kurven mit starker Änderung der Krümmung und Flächen, deren Form erheblich von der eines Parallelogramms abweicht, unerwünschte Helligkeitsschwankungen aufweisen!

Für ein Polynom 3-ten Grades erhält man aus Gl. (5.39):

$$Q_N = Q_0 + N \Delta Q_0 + \binom{N}{2} \Delta^2 Q_0 + \binom{N}{3} \Delta^3 Q_0 \qquad (5.40)$$

Soll eine Kurve aus N = 1000 Punkten bestehen (entspricht der Bildschirmbreite), so muß der ARM wegen des kubischen Terms in Gl. (5.40) eine Wortlänge von 40 bit aufweisen, damit Q_N 10 bit genau ist.

Da ein Displayprozessor ohnehin über einen Vektorgenerator verfügen muß, ist das im nächsten Kapitel beschriebene Verfahren vorzuziehen.

5.6.1.2 Differenzenverfahren mit Vektorgenerator

Der in Kapitel 5.1 beschriebene digitale Vektorgenerator DVG verarbeitet relative Koordinaten ΔQ (ΔP). Es bietet sich daher an, das Differenzenschema um eine Stufe zu verkürzen und nur ΔQ_n zu berechnen. Aus Gl. (5.38) erhält man

$$\begin{bmatrix} \Delta Q_n \\ \Delta^2 Q_n \\ \Delta^3 Q_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}^n \begin{bmatrix} \Delta Q_0 \\ \Delta^2 Q_0 \\ \Delta^3 Q_0 \end{bmatrix} \qquad (5.41)$$

und mit Gl. (5.40) nach N Iterationen

$$\Delta Q_N = \Delta Q_0 + N \Delta^2 Q_0 + \binom{N}{2} \Delta^3 Q_0 \qquad (5.42)$$

Bei einer Wortlänge des ARM von 24 bit können, wenn für ΔQ_N ein Fehler von 10^{-3} zugelassen wird, bis zu 120 Iterationen durchgeführt werden. Die Praxis hat jedoch gezeigt, daß die meisten Kurven bzw. Rasterlinien einer Fläche durch weniger als 100 Vektoren für das Auge völlig glatt wiedergegeben werden. Der DVG schreibt alle Vektoren mit gleicher Helligkeit, so daß entlang einer Kurve keine

Helligkeitsschwankungen auf Grund der Parametrisierung auftreten können.

Der größte Rechenaufwand des Differenzenverfahrens tritt bei der Bestimmung der Anfangswerte für den Tensorspeicher (Bild 5.26) auf. Diese Werte sind für eine Fläche aber nur einmal zur Verfügung zu stellen und werden deshalb am besten softwaremäßig berechnet. Steht ein Matrizenmultiplizierer zur Verfügung, so ist auch eine Berechnung mit Hardware möglich.

5.6.2 Flächenberechnung durch Matrizenmultiplikation

Der digitale Matrizenmultiplizierer DMM berechnet die Transformation (Gl. (5.16))

$$p' = p \cdot T$$

$$[x',y',z',w'] = [x,y,z,w] \cdot \underset{(4x4)}{T}$$

Wird p durch U^T und T durch $\overline{P}$ ersetzt, so berechnet der DMM die Kurve

$$Q(u') = U^T \cdot \overline{P}$$

Auf diese Weise können höchstens Kurven 3-ten Grades berechnet werden. Die Vektoren U^T werden für das gewählte Interpolations- oder Approximationsverfahren mit einer vorgegebenen Schrittweite für u, $0 \leq u \leq 1$ (z.B. $\delta = 2^{-7}$) in einem ROM zur Verfügung gestellt. Größere Schrittweiten erhält man durch entsprechende ROM-Addressierung. Die Rasterlinien zur Erzeugung der Fläche

$$Q(u',v') = U^T \cdot \overline{P} \cdot V$$

müssen in mehreren Schritten berechnet werden. Zuerst sind die Matrixprodukte $\overline{P}V$ für die v-Rasterlinien bzw. $U^T \overline{P}$ für die u-Rasterlinien zu berechnen. Da der Matrixspeicher des DMM nur 16 Elemente faßt, sind diese Multiplikationen für jede Komponente von $\overline{P}$ getrennt auszuführen und die Ergebnisse in einem zweiten (4x4) Matrixspeicher festzuhalten. Anschließend kann die Rasterlinie berechnet werden. Für das wahlweise Berechnen der u- und/oder v-Rasterlinie wird eine wesentliche Vereinfachung erreicht, wenn U = V ist:

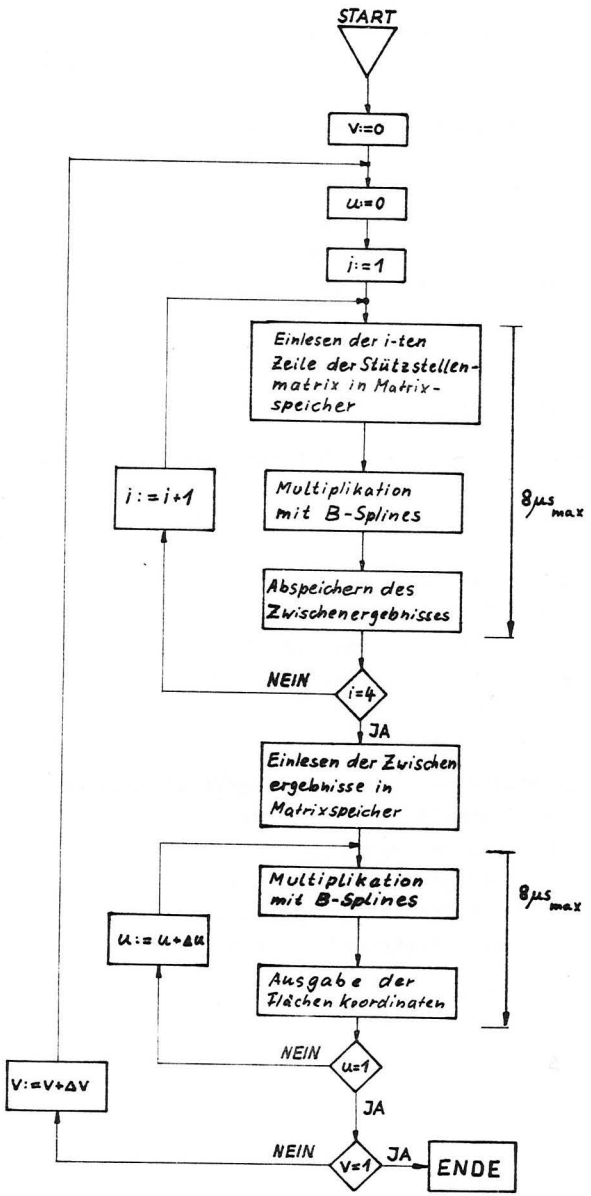
$$Q(u',v') = U^T \cdot \overline{P} \cdot U \qquad (5.43)$$

Man kann dann den Vektor $\overline{P}U$ unter Ausnutzung der Beziehung

$$\overline{P} \cdot U = (U^T \cdot \overline{P}^T)^T \qquad (5.44)$$

berechnen. In der Hardware wird dies durch Umkodieren des Adreßzählers für $\overline{P}$ geleistet /51/. Zusätzlich wird ROM-Kapazität eingespart. Trotzdem können in u- und v-Richtung verschiedene Schrittweiten gewählt werden!

Der Flächengenerator wurde aus Kostengründen nach dem Prinzip der Matrizenmultiplikation realisiert, obwohl das Differenzenverfahren wegen der höheren Rechengeschwindigkeit besser geeignet ist. In Bild 5.27 ist das Flußdiagramm des Rechenalgorithmus dargestellt. Bild 5.28 zeigt das Blockschaltbild des Flächengenerators.



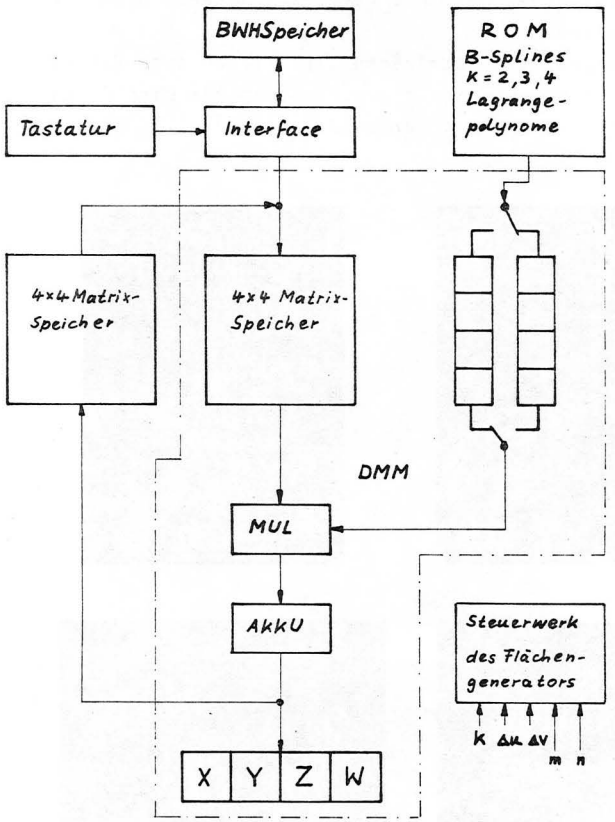


Bild 5.28: Blockschaltbild des Flächengenerators

Die folgenden Photographien zeigen Rasterpunkte Q(u',v') von B-Spline-Flächen, die mit der beschriebenen Hardware berechnet wurden.
Steht der verbesserte Matrizenmultiplizierer nach Kapitel 5.5 zur Verfügung, so kann die Rechenzeit für eine Fläche auf ein Drittel reduziert werden /52/.

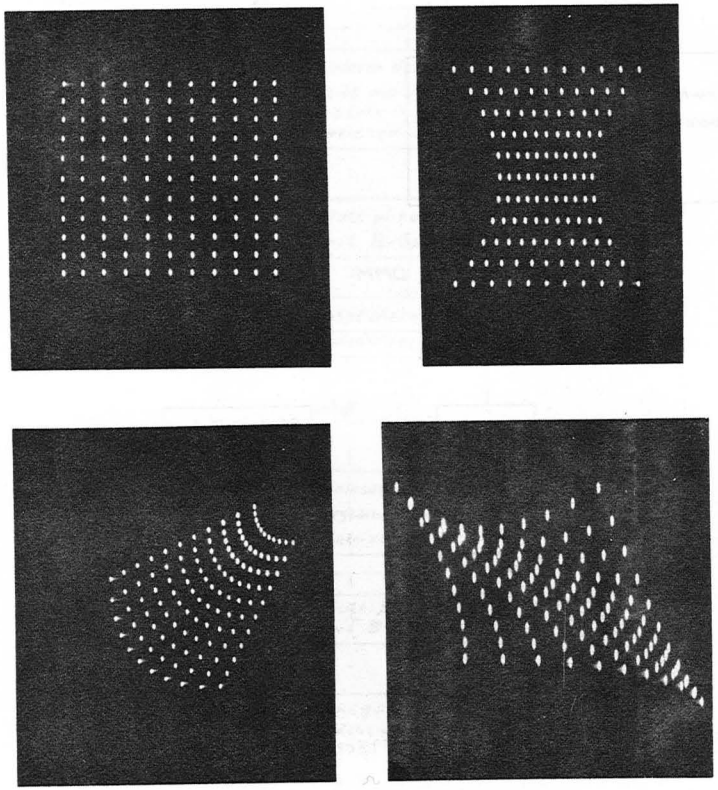


Bild 5.29: Flächen 2-ten Grades ($\Delta u = \Delta v = 0{,}1$)
Rechenzeit 1,2 ms

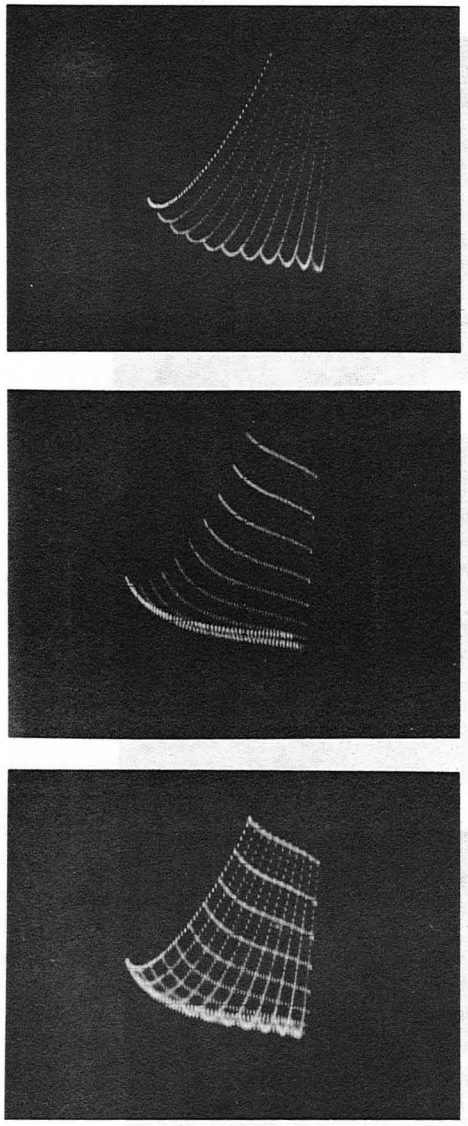
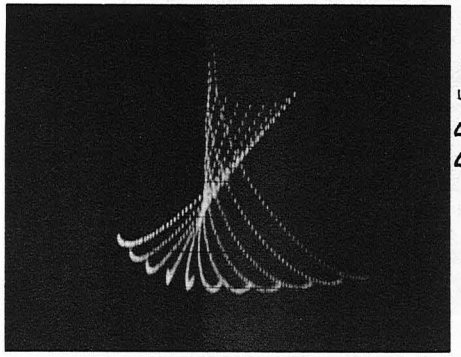
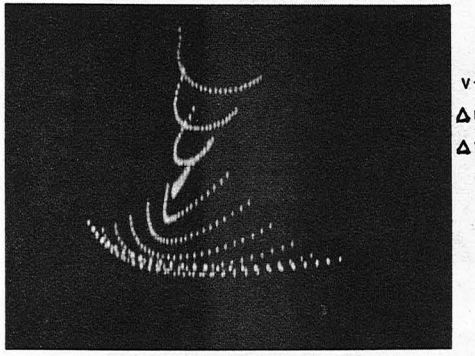


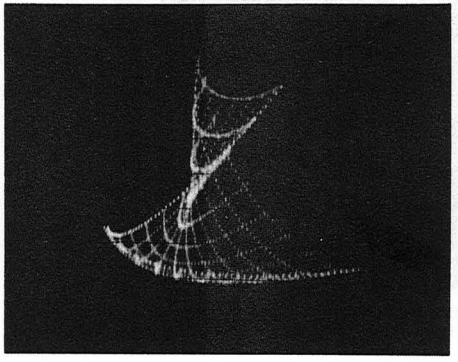
Bild 5.30: Fläche 2-ten Grades
Rechenzeit 4,2 ms



u-Linien
Δu=0,1
Δv=0,033



v-Linien
Δu=0,033
Δv=0,1



u-und
v-Linien

Bild 5.31: Fläche 3-ten Grades

5.7 Modulare Display Hardware

Mit den in Kapitel 5 beschriebenen digitalen Komponenten ist ein modulares Konzept für eine Hardware zur Berechnung, Transformation und Darstellung dreidimensionaler Objekte in Echtzeit möglich. Der Hardwareaufwand ist damit gut an die Erfordernisse der Anwendung anzupassen. Die einzelnen Komponenten seien hier noch einmal aufgezählt:

1) Digitaler Vektorgenerator DVG
2) Digitaler Kreisgenerator DKG
3) Digitaler Matrizenmultiplizierer DMM
4) Verbesserter DMM mit digitalem Dividierer
5) Flächengenerator auf der Basis des DMM
6) Flächengenerator nach dem Differenzenverfahren

Die Numerierung entspricht dem Ausbau vom einfachen Vektor- zum Hochleistungsdisplay.

Zwei wichtige Komponenten fehlen in dieser Aufstellung: eine "Windowing Hardware" und ein "Hidden line processor". Das erste Problem wurde im Rahmen dieser Arbeit nicht berücksichtigt. Eine Hardware-Lösung dafür wurde in /53/ angegeben. Zur Lösung des Verdeckungsproblems in Echtzeit wird in Kapitel 6 ein Vorschlag gemacht.

Mit den vorhandenen Komponenten läßt sich das in Bild 5.32 dargestellte Hochleistungsdisplay bauen. Es kommt dem in /34/ vorgeschlagenen Konzept sehr nahe.

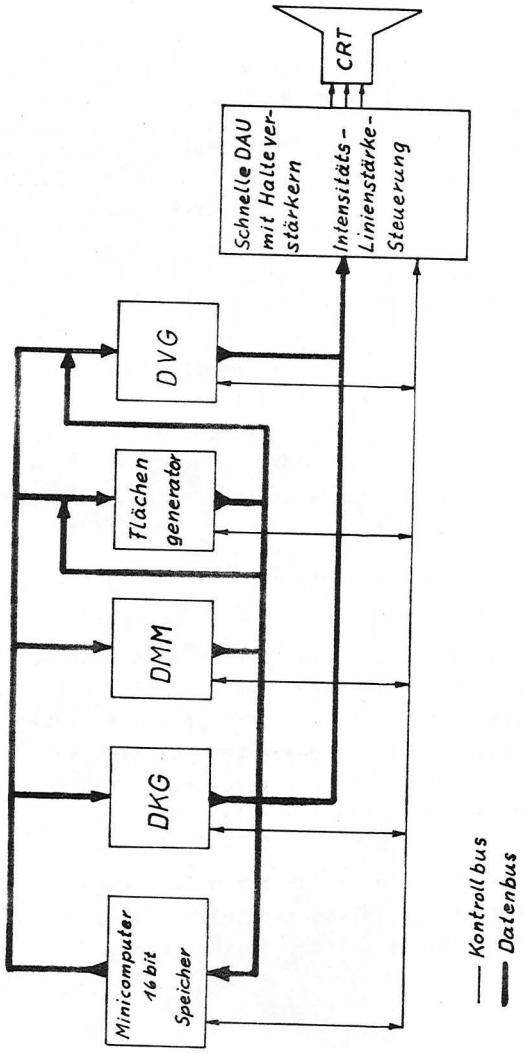


Bild 5.32: Blockschaltbild eines Hochleistungsdisplays mit den beschriebenen Komponenten

6. Zukünftige Arbeiten

Das in dieser Arbeit vorgeschlagene Verfahren zur Erzeugung schattierter räumlicher Objekte auf dem Bildschirm arbeitet ohne Entfernung der verdeckten Flächen. Durch die Intensitätssteuerung wird im Bereich der Verdeckungen der Eindruck eines durchscheinenden Körpers erzeugt (Bild 4.4). Liegen mehrere Flächen sehr dicht hintereinander, so wird trotz Intensitätssteuerung wegen des summierenden Effekts im Phosphor im Bereich der Verdeckungen eine starke Aufhellung eintreten, die nicht den berechneten Lichtverhältnissen auf der Oberfläche entspricht. Hier kann nur die Entfernung der verdeckten Flächen Abhilfe schaffen. Dies ist theoretisch in Echtzeit möglich:

Die Flächenpunkte Q (u',v') werden digital berechnet. Steht ein schneller Halbleiterspeicher zur Verfügung, dessen Zellen den Bildschirmpunkten entsprechen, so kann vor der Helltastung eines neuberechneten Flächenpunktes der Speicher auf "besetzt" oder "frei" abgefragt werden. Im "frei"-Fall wird der Punkt auf dem Bildschirm geschrieben und die korrespondierende Speicherzelle in den "besetzt"-Zustand gebracht. Dieses Verfahren hat den Nachteil, daß zwar Doppelbeschreibungen des Bildschirms vermieden werden, aber nicht notwendigerweise die verdeckten Flächen unterdrückt werden! Dies kann geleistet werden, wenn man für jeden Punkt auch die Z-Koordinate oder den Intensitätswert abspeichert. Nach einer Bildwiederholung (20 ms) steht dann im Speicher das digitalisierte Bild, das durch zeilenweises Auslesen auch auf einem Fernsehmonitor dargestellt werden kann. Beim augenblicklichen Stand der Preise und der Technologie der Halbleiterspeicher ist dieses Verfahren zur Unterdrückung der verdeckten Flächen nicht realisierbar. Abgesehen davon sind aber auch noch Untersuchungen anzustellen, wie Speicherplatz eingespart werden kann, da meist nur 30 % des Bildschirms ausgenutzt werden.

7. Literaturverzeichnis

/1/ COONS, S.A., 'Surfaces for computer-aided design of space figures', M.I.T. ESL 9442-M-139, Jan. 1964

/2/ SUTHERLAND, E.I., SPROULL, R.F., SCHUMACHER, R.A., 'A characterization of 10 hidden-surface algorithms', E&S Computer Corporation, Salt Lake City, 1973

/3/ ECKERT, R. ENCARNACAO, J., MAHNKOPF, P., 'FLAVIS', HHI-Bericht Nr. 148/1973

/4/ WYLIE, C., ROMNEY, G., 'Half-tone perspektive drawings by Computer',AFIPS Proc. FJCC 31, Nov. 1967

/5/ GOURAUD, H., 'Computer Display of curved surfaces', University of Utah, June 1971

/6/ NEWELL, M.E., NEWELL, R.G., SANCHA, T.L., 'A solution to the hidden surface problem', CAD-Center, Cambridge, 1972

/7/ BEZIER, P., 'Numerical Control', John Wiley & Sons, London, New York

/8/ BEZIER, P., 'Système UNISURF', Automatismes, Vol. 13, No 5, May 1968

/9/ MAXWELL, E.A., 'The methods of plane projective geometry based on the use of general homogeneous coordinates', Cambridge University Press 1946

/10/ ROBERTS, L.G., 'Homogeneous matrix representation and manipulation of N-dimensional constructs', The Computer Display Review, C.W. Adams ASSOC., Cambridge, Mass. 1966

/11/ LEE, T.M.P., 'A class of surfaces for computer display', SJCC 1969

/12/ FORREST, A.R., 'The twisted cubic curve', University of Cambridge, CAD Group Document No. 50, Nov. 1970

/13/ AHUJA, D.V., COONS, S.A., 'Interactive graphics in data processing', IBM Systems Journal, 1968

/14/ GORDON, W.J., Distributive lattices and the approximation of multivariate functions', Proc. Symp. Approximation with special emphasis on spline functions, Academic Press, New York 1969

/15/ FORREST, A.R., 'Mathematical principles for curve and surface representation', Proc. Curved Surfaces in engineering, IPC Science and Technology Press Ltd., pp 5-13, 1972

/16/ SABIN, M.A., 'Numerical Master Geometry', Proc. Curved Surfaces in engineering, IPC Science and Technology Press Ltd., 1972

/17/ BÖHM, W., 'Zur Darstellung von Flächen in der Datenverarbeitung', Angewandte Informatik 8/71

/18/ BJÖRCK, A., DAHLQUIST, G., 'Numerische Methoden', Oldenbourg, München, Wien 1972

/19/ SABIN, M., A., 'The BAC Numerical Master Geometry System', Proc. Soc. London, A 321, 1971

/19a/ SEEGER, C., 'Kubische Splines', Diplomarbeit TUB 73, Betreuer: Straßer

/20/ COONS, S.A., 'Surfaces for computer-aided design of space forms', MAC-TR-41, June 1967, M.I.T.

/21/ FORREST, A.R., 'Curves and surfaces for CAD', Dissertation, University of Cambridge, 1968

/22/ FORREST, A.R., 'Interactive interpolation and approximation by Bêzier polynomials', The Computer Journal, Vol 15, No 1

/23/ DAVIS, J.P., 'Interpolation and Approximation', Blaisdell Publishing Co., 1963, pp 108-118

/24/ GORDON, W.J., RIESENFELD, F.R., 'Bernstein-Bêzier methods for the computer aided design of freeform curves and surfaces', GMR-1176, 1972

/25/ SCHOENBERG, I.J., 'On variation diminishing approximation methods', in LANGER, R.E.: On numerical approximation, University of Wisconsin Press, 1959

/26/ CREUTZ, G., 'FLIB'- Ein Programmsystem zum interaktiven Entwurf von Bêzier-Kurven und -Flächen', Diplomarbeit TUB 1973, Betreuer: Straßer

/27/ FORREST, A.R., 'A new curve form for computer aided design', University of Cambridge, CAD Document No 66

/28/ SCHOENBERG, I.J., 'On Spline functions', in Shisha, O.: Inequalities, Academic Press, New York and London, 1967

/29/ STRASSER, W., 'Kurven- und Flächendarstellung', Vorlesungsskript Computer Graphics, 72/73 TUB, Kap. 7

/30/ DE BOOR, C., 'On calculating with B-Splines', Journal of Approximation Theory, Vol 6, No 1, July 72

/31/ BÖLLING, G., 'FABIAN - Ein Programmsystem zum interaktiven Entwurf von B-Spline-Kurven und -Flächen', Diplomarbeit TUB 1973, Betreuer: Straßer

/32/ ENCARNACAO, J., GILOI, W., SANITER, J., STRASSER, W., WALDSCHMIDT, K., 'Programmierungs- und gerätetechnische Realisierung einer 4x4-Matrix für Koordinatentransformationen auf Computer-Bildschirmgeräten', Elektron. Rechenanl. 14, 1972

/33/ GOURAUD, H., 'Continuous shading of curved surfaces', IEEE Transactions on Computers, Vol C-20, No 6, June 71

/34/ GILOI, W., 'Hardware-Aspekte von Computer Graphics', GI-Bericht Nr. 2, 1971

/35/ METZDORF, D., 'Digitaler Vektorgenerator', Diplomarbeit TUB 73, Betreuer: Straßer

/36/ GLAUBERT, W., 'Vektorgeneratoren', Diplomarbeit TUB 1969

/37/ STOCK, K., 'Schneller digitaler Vektorgenerator', Studienarbeit TUB 1974, Betreuer: Straßer

/38/ PITTEWAY, M.L.V., 'Algorithm for drawing ellipses or hyperbolae with a digital plotter', Computer Journal 10 (1968), S. 282-289

/39/ KRAUSE, R., 'Digitaler Kreisgenerator', Diplomarbeit TUB 72, Betreuer: Straßer

/40/ WILHELMI, U., 'Digitaler Kreisgenerator', Diplomarbeit TUB 73, Betreuer: Straßer

/41/ GHEST, R.C., 'New IC multiplies 2-bit by 4-bit number', Electronics, Nov. 1971

/42/ SCHWETER, D., 'Digitaler Matrizenmultiplizierer', Diplomarbeit TUB 72, Betreuer: Straßer

/43/ VOLKMANN, H., 'Digitale Divisionsverfahren', Diplomarbeit TUB 73, Betreuer: Straßer

/44/ GUILD, H.H., 'Some cellular logic arrays for non-restoring binary division', Radio and Electronic Engineer, Vol. 39, No. 6

/45/ MAURUS, C., HAMACHER, C.V., 'An augmented iterative array for high-speed binary division', IEEE Transactions on Computers, VOL C-22, No. 2, Feb. 73

/46/ ANDERSON, S.F. et al, 'The IBM 360/91 floatingpoint execution unit', IBM Journal, Jan.67

/47/ PESY, W., 'Schneller digitaler Dividierer', Diplomarbeit TUB 73, Betreuer: Straßer

/48/ FERRARI, D., 'A division method using a parallel multiplier', IEEE Tr. on C., April 1967

/49/ BORGMANN, D., 'Schneller digitaler Dividierer', Diplomarbeit TUB 73, Betreuer: Straßer

/50/ BUSCHMANN, W., 'Addierer-Register-Modul für das Differenzenverfahren', Studienarbeit TUB 73, Betreuer: Straßer

/51/ SCHIEFEN, G., 'Digitaler Flächengenerator', Diplomarbeit TUB 73, Betreuer: Straßer

/52/ STEIGNER, D., 'Digitaler Flächengenerator', Diplomarbeit TUB 73, Betreuer: Straßer

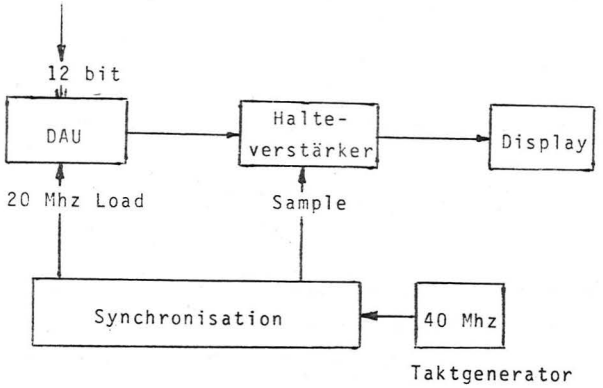
/53/ SPROULL, F.R. and SUTHERLAND, I.E., 'A clipping divider', FJCC 1968

Abschließend möchte ich Herrn Professor Dr.-Ing. Wolfgang Giloi und Herrn Ass.-Professor Dr.-Ing. J. Encarnacao für die Anregung zu dieser Arbeit und wertvolle Diskussionen danken.

Herrn Dipl.-Ing. Wolfgang Kestner danke ich für die Überlassung einer ersten Version seiner grafischen Sprache, mit der ich die theoretischen Überlegungen zur B-Spline-Approximation am Display sehr schnell überprüfen konnte.

Anhang A

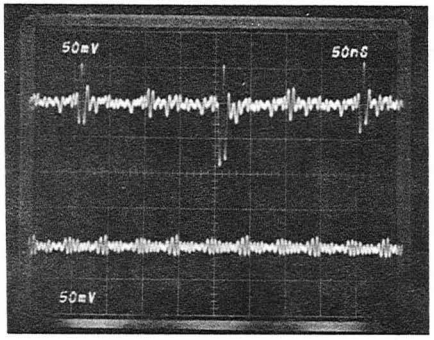
Durch Einsatz eines schnellen Halteverstärkers hinter dem DAU und sorgfältige Synchronisation von Datenübernahme in den DAU und Sample-Impulsgenerierung konnte eine Wortrate von 20 Mhz bei 12 bit Wortlänge ohne störende Spitzen auf dem Display erreicht werden/+/.



vor

dem Halteverstärker

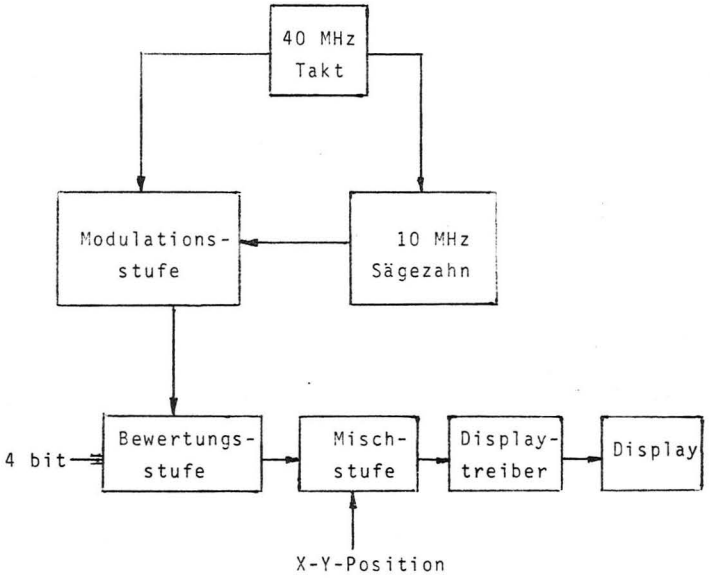
hinter



/+/ Teich,G.,Walf,G., 'Display-Ansteuerung',Studienarbeit TUB 74,Betreuer: Straßer

Anhang B

Die in Kapitel 4.4 beschriebene Steuerung der Linienstärke wird durch Einsatz eines Spiralgenerators erreicht. 16 verschiedene Linienstärken (0,1 mm - 2,4 mm) können eingestellt werden.



Tabellarischer Lebenslauf

10.8.41	geboren in Metzingen, Kreis Reutlingen
März 1960	Abitur am Gymnasium Metzingen
1.4.60-31.3.62	Wehrdienst
1.4.62-31.9.62	Praktikum bei AEG in Bad Cannstatt
Okt. 1962	Immatrikulation an der TU Berlin
Dez. 1969	Diplomhauptprüfung auf dem Gebiet Nachrichtentechnik
20.6.69	Eheschließung mit Dr.Hiltrud Straßer, geb. Zahn
1.12.69-30.9.70	Entwicklungsingenieur bei Nixdorf Computer Gmbh
1.10.70-30.6.73	Wiss.Assistent an der TU Berlin, Informatik Computer Graphics
1.7.73	Stellvertretender Abteilungsleiter am Heinrich-Hertz-Institut Berlin (HHI)
1.8.74	Leiter der Abteilung Informationsverarbeitung am HHI

Berlin, den 4.10.74 Wolfgang Straßer