

PSEUDOREALISTISCHE RASTERGRAPHIK IM MENSCH-MASCHINE-DIALOG

H. Müller, Karlsruhe

Zusammenfassung: Pseudorealistische Graphik entsteht durch Zusammenwirken von Geometrie und Licht. Sie ermöglicht die informationsreiche und doch verständliche Darstellung von räumlich-geometrischen Daten. Es werden Techniken zu ihrer Generierung sowie Einsatzmöglichkeiten im Mensch-Maschine-Dialog vorgestellt.

1 Überblick

Sinkende Kosten und wachsende Leistung der Hardware erlauben es, hochaufgelöste Rastergraphik einem breiten Anwenderpublikum zugänglich zu machen. Stand der Technik ist hochaufgelöste Bitmapgraphik, d.h. Bilder bestehend aus schwarzen/weißen Bildpunkten. Hier werden Technik und Einsatzmöglichkeiten einer neuen Generation von Rastergraphik vorgestellt, nämlich hochaufgelöster Rastergraphik hoher Farbtiefe. Diese wird insbesondere für Workstations der 5. Generation propagiert [16]. Ein Beispiel dafür sind die Symbolics-Lisp-Maschinen, die teilweise entsprechende Hardwareausstattung anbieten.

Graphiksysteme, die im wesentlichen auf Geometrie basieren, erlauben es nicht, die Fähigkeiten von solchen Geräten voll auszunutzen. Ein Ansatz, der der Gewohnheit des Menschen entgegen kommt, ist es, die Wechselwirkung von Licht und Geometrie zu berücksichtigen. Wohlbekannte Beispiele für diese sogenannte pseudorealistische Graphik sind die "shaded pictures" im CAD der Architektur und des Maschinenbaus, welche jedoch zur Zeit noch meist von geringer Qualität sind. Weitere Einsatzgebiete sind die Naturwissenschaften (Moleküle, Simulation) und die Informatik (Dialogsysteme, Daten- und Prozesszustände). Im nichtwissenschaftlichen Bereich besteht Interesse in der Wirtschaft (Werbung, Businessgraphik) und der Freizeitgestaltung (TV, Videospiele). Bei diesen Anwendungen geht es darum, in Rechnern ge-

speicherte Information in verständlicher, attraktiver Weise dem Benutzer sichtbar zu machen.

Im folgenden werden Grundtechniken der Erstellung pseudorealistischer Graphik vorgestellt, sowie deren Anwendbarkeit im Mensch-Maschine-Dialog untersucht.

2 Graphik hoher Farbtiefe

Experimentelle Untersuchungen zeigen, daß zur digitalen Darstellung von realistischen Bildern (Fotos) ohne sichtbare Defekte eine Quantisierung in 256 Graustufen (1 Byte/Pixel) ausreichend ist. Die Reduktion um 1 Bit (128 Graustufen) kann zu deutlich wahrnehmbaren Effekten, wie ringförmige anstatt kontinuierliche Lichtabfälle, führen. Bei Systemen geringerer Tiefe versucht man, dieses durch Überlagern von Rauschen (Dithering) zu beheben [7], wobei diese Heuristiken nicht immer zufriedenstellend arbeiten. Die Übertragung auf Farbbilder, zusammengesetzt aus Rot-, Grün- und Blaukomponente, ergibt eine Farbtiefe von $2^{24} = 16\text{M}$ Farben, mit der man auf der sicheren Seite ist.

Bezüglich der Auflösung in Bildpunkte sind 512×512 Pixel zur Darstellung auf einem 19 Inch Monitor ausreichend. Die hohe Farbtiefe ermöglicht den Einsatz von Anti-Rasterungs- bzw. Anti-Aliasverfahren [12], welche die bei Rasterschirmen (selbst bei höherer Auflösung) übliche Treppenbildung, etwa bei Linien oder Schriftzeichen, eliminiert, d.h. es besteht eine virtuell höhere Auflösung. Eine höhere Auflösung, etwa 1024×1024 , ist dann wünschenswert, wenn mehr diskrete Information, z.B. Schrift, auf dem Bildschirm untergebracht werden soll.

Kern eines Rastergraphiksystems ist der Bildwiederhol-speicher. Bei $512 \times 512 \times 24$ Bit beträgt seine Größe etwa 750 KByte. Die Eigenentwicklung VIDEO1.3 [15] zeigt, daß eine derartige Rasterbildausgabe schon heute mit geringen Hardwarekosten realisierbar ist. VIDEO1.3 kann an praktisch jeden Rechner mit Parallelschnittstelle angeschlossen werden. Eine Datentransferrate von nahezu 1MByte/sec ermöglicht interaktives Arbeiten.

2.1 Automatische Bildcompilierung

In der Natur treten kontinuierliche Intensitätsübergänge vor allem dort auf, wo räumlich begrenzte Lichtquellen einwirken. Diese rufen optische Effekte wie Schatten, Glanzlichter und Brechung hervor, die es ermöglichen, räumliche Strukturen plastisch zu erkennen (durch ein Glanzlicht wird eine Kreisscheibe zur Kugel). Diese Beobachtung ist Hintergrund des folgenden Verfahrens zur automatischen Bildcompilierung.

Ausgangspunkt der Bildcompilierung sind räumliche Daten. Die Aufgabe von Bildcompilern ist es, aus diesen ein Rasterbild zu erstellen. Räumliche Daten können als Menge von Grundelementen vorliegen (explizite Beschreibung), oder durch einen generierenden Algorithmus (prozedurale Beschreibung) definiert sein. Klassische Grundobjekte sind Flächen im Raum wie Dreiecke und Vierecke, oder Körper wie Kugeln, Quader, Rotationskörper und Translationskörper. Neuer ist die Verwendung von dreidimensional gerasterten Szenen. Das Grundelement hierbei ist das Voxel ("volume element"), die Übertragung des Pixels ("picture element") in den Raum. Eine Szene wird hierbei durch ein reguläres räumliches Würfelraster dargestellt, wobei für die einzelnen Würfel angegeben wird, mit welcher Art von "Material" sie gefüllt sind (z.B. Luft oder "braun"). Derartige Daten werden etwa durch räumliche Digitalisierer wie Computertomographen geliefert. Prozedurale Beschreibungen werden für Freiformflächen (gekrümmte glatte Oberflächen) und Fractals (unregelmäßige natürliche Strukturen wie Gebirge) verwendet.

Ein Verfahren der Bildcompilierung ist in Bild 1 dargestellt. Gegeben ist eine dreidimensionale Szene, in der zusätzlich noch ein oder mehrere Lichtquellen verteilt sind. Diese Lichtquellen werden geometrisch meist durch Punkte beschrieben. Ferner gibt es einen Beobachter (Augenpunkt) und eine Bildebene, in die die Szene abgebildet werden soll. Diese Bildebene ist gerastert, entsprechend der Auflösung des gewünschten Bildes, z.B. 512x512 Pixel. Nun wird vom Augenpunkt aus ein Sehstrahl durch den Mittelpunkt jedes Pixels gezogen. Für einen solchen Strahl wird das erste Objekt in der Szene bestimmt, das getroffen wird. Vom Auftreffpunkt aus sind dann Strahlen zu den verschie-

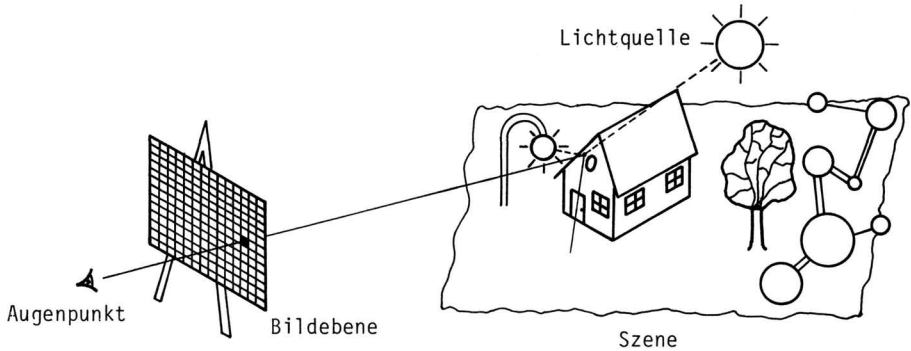


Bild 1: Bildcompilierung durch Strahlverfolgung

denen Lichtquellen zu ziehen, um zu testen, ob sich ein blockierendes Objekt der Szene auf diesem Weg befindet. Ist das Objekt, auf dem sich der Auftreffpunkt befindet, spiegelnd, so wird ein weiterer Strahl verfolgt, nämlich der Reflexionsstrahl zum Sehstrahl, entsprechend dem Reflexionsgesetz der Physik. Für diesen ist wiederum der erste Auftreffpunkt in der Szene zu bestimmen, mit welchem genauso verfahren wird wie zuvor. Ein weiterer Strahl ist zu verfolgen, wenn ein durchsichtiges Objekt getroffen wird. Insgesamt ergibt sich durch diese Strahlverfolgung eine baumartige Struktur. Die Größe des Baumes hängt davon ab, über wieviele Stufen verfolgt wird. Ein Abbruchargument ist die Dämpfung von Licht.

Die Pixelintensität wird nun aus der Materialfarbe der getroffenen Objekte, der jeweils von den Lichtquellen einfallenden Intensität und den Auftreffwinkeln der Strahlen berechnet. Hierfür gibt es verschiedene Beleuchtungsformeln, welche physikalischen Gesetzen nachempfunden sind, z.B.

$$I = p_{\text{diff}} \cdot I_{\text{diff}} + p_{\text{sp}} \cdot I_{\text{sp}}, \quad 0 \leq p_{\text{diff}}, p_{\text{sp}}.$$

I gibt die in Blickrichtung V abgestrahlte Intensität an, vgl. Bild 2. Diese setzt sich zusammen aus einem Diffusanteil und einem Spiegelungsanteil. Der Diffusanteil wird nach dem Lambert-Gesetz der Physik modelliert zu

$$I_{\text{diff}} = \max\{0, N \cdot L\} \cdot M \cdot I_L.$$

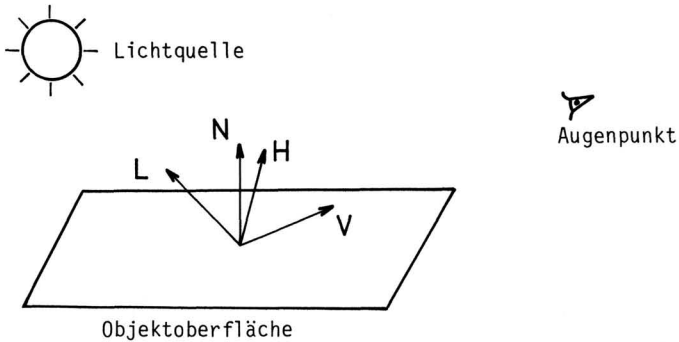


Bild 2: Richtungsvektoren zur Berechnung von Intensitätswerten durch eine Beleuchtungsformel

Dabei ist I_L die aus Richtung L einfallende Lichtintensität, M der Reflexionsgrad der Oberfläche und N der Normalenvektor auf der Objektoberfläche. Bei Rechnung im RGB-Farbmodell müssen diese Formeln auf jede Grundfarbe angewendet werden, wobei dann die drei auftretenden M-Faktoren MR, MG und MB, welches Zahlen zwischen 0 und 1 sind, gerade die Oberflächenfarbe definieren.

Bei der spiegelnden Reflexion ist eine Reflexionskeule zu modellieren, deren Öffnung je nach Stärke der Spiegelung variiert. Das klassische Phong-Modell [5] sieht so aus:

$$I_{sp} = (N \cdot H)^c \cdot I_L.$$

Dabei ist H der mittlere Vektor zwischen V und L, c steuert die Öffnung der Spiegelkeule.

Die Anzahl der Rechenschritte, die bei diesem sogenannten Ray-Tracing-Verfahren zumindest ausgeführt werden, ist von der Größenordnung $m \cdot n + m \cdot l$, wobei m die Anzahl der Pixel, n die Anzahl der Objekte und l die Anzahl der Lichtquellen ist. Ein Rechenschritt umfaßt dabei einen Test auf Schnitt, das Einfügen eines möglichen Schnittpunkts in eine sortierte Liste und gegebenenfalls die Auswertung der Beleuchtungsformel. Setzt man hierfür 1 ms an, was auf einem mittleren bis großen Rechner durchaus realistisch ist, und nimmt man ferner eine Szenengröße von $n = 1000$ Objekten an, sowie $m = 512 \times 512$ und $l = 2$ Lichtquellen, so ergibt das eine Rechenzeit von 70 Stunden.

Im Rahmen eines Projekts am Institut für Informatik I, PICTURE.KA, wurden Bildcompiler zur Verarbeitung verschiedener Geometrietypen entwickelt [9]. Bild 3 zeigt deren Funktion im Gesamtsystem, welches in dem Projekt realisiert wird. Dieses System kann für Dienstleistungen als auch als Testumgebung für weitere Entwicklungen eingesetzt werden. Für diese Bildcompiler

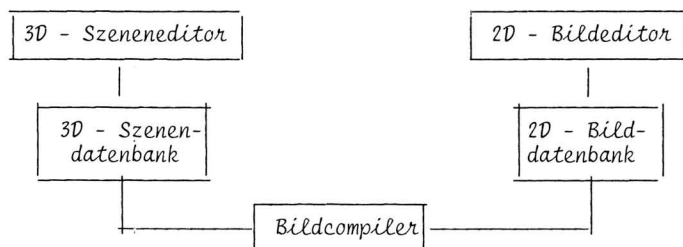


Bild 3: Funktionen des PICTURE.KA-Systems

wurden verschiedene Suchstrategien entwickelt [8]. Der Ansatz dabei ist, die Szene so in eine Datenstruktur vorzuverarbeiten, daß mit deren Hilfe beim anschließenden Ray-Tracing der erste Auftreffpunkt schnell festgestellt werden kann. Im Extremfall wird man für den Augenpunkt und jede Lichtquelle eine vollständige Sichtbarkeitsberechnung mit einem sogenannten objektorientierten "Visible-Surface-Algorithmus" durchführen [11]. Problematisch hierbei wird der u.U. schnell wachsende Speicherverbrauch, wobei aber die Ray-Tracing-Zeit klein wird. Dieser Trade-Off zwischen Rechenzeit und Speicherplatz ist nach Größe des eingesetzten Rechners zu gewichten. Ergebnisse von theoretischen Grundlagenuntersuchungen sind in [1] zu finden.

Zum Editieren von pseudorealistischen 3D-Szenen steht eine Datenbank mit graphischem Editor SDS3/TDS3 zur Verfügung [3]. SDS3/TDS3 ermöglicht neben dem Geometrieentwurf und dem Einschleußen von Fremdgeometriedaten die Definition der optischen Parameter (Farbe, Brechungsindex, Textur, etc.). Letzteres erfordert Expertenkenntnisse, die denen von Photographen nicht unähnlich sind.

2.2 Bildoperatoren

Aufgrund der relativ hohen Kosten der automatischen Generierung pseudorealistischer Bilder ist es nützlich, das gewünschte Bild nicht nur über Szenenmanipulation, sondern direkt durch Bildmanipulation zu erzeugen. Dazu werden Bildoperatoren benötigt, welche es mit wenig Aufwand erlauben, fotoähnliche Bilder interaktiv zu generieren und zu manipulieren.

Zunächst gibt es die Operatoren der klassischen Vektorgrafik, wie sie in GKS [4] definiert sind. Diese müssen in höherer Qualität realisiert werden, als es auf heute gängigen Rastergrafiksystemen geschieht. So ermöglicht eine hohe Farbtiefe, Antialiasverfahren bei der Generierung von Vektoren und Schriftzeichen einzusetzen, welche Treppeneffekte eliminieren. Auf fertige Bilder sind A-posteriori-Antirasterungsverfahren mit gutem Ergebnis anwendbar.

Mächtige Operatoren der interaktiven Grafik sind Füllfunktionen. Zur Konstruktion glatter kontinuierlicher Intensitätsübergänge, welche bei pseudorealistischer Grafik benötigt werden, sind neue Verfahren nötig. Eine Möglichkeit ist das Stützpunktfüllen [2]. Hierbei werden in der zu füllenden Fläche Stützpunkte im Intensitätsgebirge vorgegeben, welche dann mit dem Interpolationsverfahren (Splines) ausgeglichen werden. So können interaktiv einfach Kugeln und Röhren aus Kreisscheiben und Streifen generiert werden.

Ein weiterer Operationstyp ist das Verknüpfen von Bildern durch arithmetische Operationen. Diese bieten erheblich mehr Möglichkeiten als die RasterOps gängiger Bitmapsysteme. Auf Bilder können unter anderem auch Operatoren zur Farbveränderung so definiert werden, wie sie aus der Bildverarbeitung und Mustererkennung bekannt sind.

Schließlich werden stufenlose Bildtransformationen möglich: Verkleinerung, Skalierung, beliebige affine Transformationen, beliebige Verzerrung ("Knittern", Abbildung auf gekrümmte Oberflächen). Diese sind bei der Bildmontage nützlich. Dabei sind auch Segmentierungsverfahren einsetzbar, wie sie im Bereich der Bildverarbeitung und Mustererkennung bekannt sind [12], um etwa Objekte aus Bildern auszuschneiden und woanders einmontieren

zu können.

Im PICTURE.KA-Bilderzeugungssystem erfolgt die Bildmanipulation auf einer Bilddatenbank, welche pseudorealistische Rasterbilder rechteckigen Formats, aber variabler Größe enthält. Dazu gibt es ein Bildeditiersystem, DESIGN1. Die vier Haupttechniken der Bildmanipulation in diesem System sind

- Extraktion und Montage
- Ausfüllen
- Malen
- Schattieren.

Die verschiedenen Entwurfstechniken von DESIGN1, welche auf der Realisierung von zahlreichen der oben beschriebenen Operatoren aufbauen, sind in [2,13] beschrieben. Ferner können über eine Videokamera Bilder, z.B. Fotos, in die Datenbank übernommen werden und beim Bildeditieren verwendet werden.

3 Anwendungen

Im Mensch-Maschine-Dialog ist pseudorealistische Rastergrafik zur Visualisierung abstrakter komplexer räumlich geometrischer Daten einsetzbar, zum Beispiel zur Darstellung von Meßergebnissen, Ergebnissen von Simulationen oder Entwurfsprozessen. Durch Einsatz eines Bildcompilers erhält man verständliche Bilder hohen Informationsgehalts. So wurde etwa mit Hilfe des PICTURE.KA Systems das Verhalten von elektrischem Strom in menschlichem Gewebe, basierend auf Computertomographieschnitten, dreidimensional dargestellt [10]. Eine derartige Datenvisualisierung erfolgt üblicherweise nach einer zeitaufwendigen Rechnung oder einem längeren Entwurfsprozess, bei der die relativ lange Zeit der Bildcompilierung nicht ins Gewicht fällt.

Eine schnellere Rückkopplung besteht bei Systemen, in denen abstrakte Vorgänge oder Dialoge durch fertige Bilder unterstützt werden. Ein Beispiel hierfür ist Btx. Mit Hilfe des in 2. erwähnten einfachen BildwiederholSpeichers VIDEO1.3 ist es möglich, derartige Dialoge mit der Geschwindigkeit von Btx auf einem Arbeitsplatzrechner zu ermöglichen, wobei die Qualität der Darstellung die der üblichen Btx-Seiten natürlich übertrifft.

Eine Einsatzmöglichkeit pseudorealistischer Bilder in diesem Fall ist etwa für Schulungszwecke (Computer-unterstützter Unterricht), bei Diagnoseexpertensystemen [6] und Demonstrationen (Marketing) denkbar.

Ein Einsatz hochaufgelöster Rastergrafik hoher Farbtiefe als Weiterentwicklung der Bitmapdisplays für die hochdynamische Interaktion erfordert komplexe Bildwiederholpeicher bestehend aus mehreren Videoeinheiten zur Parallelarbeit. Solche Videoeinheiten, welche verschiedene Funktionen haben können (Schriftgeneratoren, Vektorgeneratoren, Bildspeicher für Segmente (Windows)) können etwa über eine Pipeline gekoppelt werden, in die sie ihre grafische Information in Pixelform übergeben. Am Ende der Pipeline geschieht die Generierung des Videosignals. Die Videoeinheiten können von ein oder mehreren CPU's über einen Bus mit Information versorgt werden [14]. In solchen intelligenten Systemen können Teilaufgaben der automatischen Compilierung pseudorealistischer Grafik, etwa die Sichtbarkeitsberechnung durch ein Tiefenpufferverfahren, explizit implementiert werden. Ferner ist Animation in hoher Qualität möglich.

Literatur

- [1] Abramowski, S.; Müller, H.: 1-D Queries in 3-D Space. Int. Bericht 9/85, Institut für Informatik I, Uni Karlsruhe 1985.
- [2] Angele, J.; Zimmermann, J.: Realisierung von DESIGN1, Diplomarbeit, Institut für Informatik I, Uni Karlsruhe 1985.
- [3] Bohlen, M.: SDS3 - ein System zum rechnergestützten Entwurf komplexer 3D-Szenen, Diplomarbeit, Institut für Informatik I, Uni Karlsruhe 1985.
- [4] Enderle, G.; Kansy, K., Pfaff, G.: Computer Graphics Programming, Berlin: Springer 1984.
- [5] Foley, J. D.; van Dam, A.: Fundamentals of Interactive Computer Graphics. Reading: Addison-Wesley 1982.
- [6] Hayes-Roth, F.; Waterman, D. A.; Lenat, D. B.: Building Expert Systems. Reading: Addison-Wesley 1983.
- [7] Heckbert, P.: Color Image Quantization for Frame Buffer Display. Computer Graphics 16(3) (1982) 297
- [8] Kadisch, M.: Ray-Tracing komplexer Szenen. Diplomarbeit, Institut für Informatik I, Uni Karlsruhe 1985.

- [9] Müller, H.: Pseudorealistische Rastergrafik: Generieren und Editieren. Velbert: ONLINE GmbH 1985. = KOMMTECH'85, Tagungsband
- [10] Müller, H.; Ernestus, W.; Verhagen-Schönewald, B.: Plastische Darstellung von Voxelszenen durch optische Simulation. erscheint in: Proceedings 7. DAGM-Symposium Mustererkennung, Erlangen, 1985. Berlin: Springer 1985.
- [11] Müller, H.; Schmitt, A.; Abramowski, S.: Visible Surface Calculation for Complex Unstructured Polygonal Scenes, erscheint in: Computing
- [12] Pavlidis, T.: Algorithms for Graphics and Image Processing. Berlin: Springer 1982.
- [13] Schmidt-Lademann, F.-P.; Müller, H.: DESIGN1 - der Arbeitsplatz zur Bildmanipulation des PICTURE.KA-Systems zur Erzeugung pseudorealistischer Rasterbilder. Bonn: GI 1984. = Tagungsband MICROGRAPHICS, Fachgespräch der GI, Nov. 1984
- [14] Schmidt-Lademann, F.-P., Video-Pipeline. Internes Arbeitspapier, Institut für Informatik I, Uni Karlsruhe 1984.
- [15] Schmitt, A.; Schmidt-Lademann, F.-P.; Kebschull, U.: Technische Kurzbeschreibung des Bildwiederholtspeichers VIDEO1.3. Internes Arbeitspapier, Institut für Informatik I, Uni Karlsruhe 1984.
- [16] Willis, P. J.: Colour Graphics and the Fifth Generation Workstation. Computer Graphics Forum 3 (1984) 147-152

Dr. Heinrich Müller
 Institut für Informatik I
 Universität Karlsruhe
 Postfach 6380
 7500 Karlsruhe