

Liniengraphiken

Vortrag im Rahmen der Ringvorlesung

Graphik in der Mensch-Computer-Interaktion

Tobias Isenberg

e-mail: *isenberg@cs.uni-magdeburg.de*

Magdeburg, 1. Juni 1997

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung – photorealistische Bilder und Liniengraphiken | 2 |
| 2 | Konturgraphiken und Linienstile | 3 |
| 2.1 | Linienmodell nach JUTTA SCHUMANN | 3 |
| 2.2 | Linienmodell nach LARS SCHUMANN | 5 |
| 3 | Texturgraphiken und Stroke Textures | 9 |
| 3.1 | Realisierung von Texturgraphiken mittels <i>Stroke Textures</i> | 9 |
| 3.2 | Anwendungen | 10 |
| 4 | Liniengraphiken in der MCI | 12 |
| | Abbildungsverzeichnis | 14 |
| | Tabellenverzeichnis | 14 |
| | Literatur | 14 |

1 Einleitung – photorealistische Bilder und Liniengraphiken

In den letzten Jahren ist die Qualität der photorealistisch gerenderten Bilder immer besser und perfekter geworden. Das sieht man nicht zuletzt daran, daß inzwischen sogar ein vollständig im Computer erstellter Spielfilm *gedreht* wurde: „Toy Story“ von Walt Disney.

Wenn man sich jedoch Illustrationen in Büchern anschaut stellt man fest, daß dort in den meisten Fällen keine Photos oder photorealistischen Graphiken benutzt werden, sondern Liniengraphiken. Sie werden benutzt, weil man mit ihrer Hilfe wichtige Dinge betonen und weniger wichtige in der Hintergrund rücken kann. Liniengraphiken bieten außerdem ein viel höheres Abstraktionspotential als photorealistischen Graphiken.

Es erscheint also für solche Anwendungsgebiete sinnvoller, einen linienbasierten Ansatz zum Generieren von Computerbildern zu benutzen.

Als Liniengraphiken sollen im folgenden solche Graphiken betrachtet werden, die eines der folgenden Stilelemente oder auch beide enthalten:

- Konturlinien, die Objekte voneinander und vom Hintergrund abgrenzen und die Grobform der Objekte (inklusive einer Binnenstruktur) darstellen und/oder
- Schraffur- bzw. Texturlinien, die die Helligkeits-, Beleuchtungs- und Strukturinformationen darstellen und auch z. B. Rundungen andeuten und damit den Anschein von Räumlichkeit erwecken können.

Liniengraphiken sind in den Bereich der darstellenden Bilder einzuordnen, obwohl sie sehr stark von der sichtbaren Natur abstrahieren.

Im folgenden sollen zuerst zwei Ansätze betrachtet werden, die die von realen Zeichenwerkzeugen erzeugten Linien auf dem Rechner simulieren sollen. Dabei wird ein besonderes Augenmerk auf das unterliegende Linienmodell gelegt werden. In einem weiteren Abschnitt wird dann ein Ansatz zur Generierung von Schraffuren bzw. Texturen vorgestellt.

2 Konturgraphiken und Linienstile

2.1 Linienmodell nach Jutta Schumann

Herkömmliche, mit Hilfe eines Computers (z. B. durch Nutzung eines 2D-Graphik-Systems) erzeugte Liniengraphiken erwecken oft den Eindruck von Statik, Vollständigkeit und Absolutheit. Dies ist jedoch nicht immer erwünscht. JUTTA SCHUMANN griff dieses Problem in ihrer Diplomarbeit [Sch92] auf, indem sie die Idee der Verfremdung von Liniengraphiken verfolgte. Ziel ihrer Arbeit war es, computergenerierte Liniengraphiken „lebendiger“ zu machen, ähnlich handgezeichneten Illustrationen. Um diesen Effekt zu erreichen, werden sogenannte Linienstile genutzt. Diese transformieren eine gegebene Eingabelinie in eine oder mehrere ungenaue Ausgabelinien.

Sie analysierte zuerst herkömmliche Linienzeichnungen und erhielt als Ergebnis eine Reihe von Linienattributen, die Tabelle 1 zu entnehmen sind.

| einzelne Linien | |
|--|--|
| Attribut | Wertevorrat |
| Linienstärke | fein ... breit |
| Farbsättigung | schwach ... stark |
| Sättigungsverlauf | gleichbleibend, zunehmend, verblassend |
| Linienrand | durchgehend, verdickend, verjüngend, flatternd |
| Linienkrümmung | schwach ... stark |
| Linienkontur | scharf, verschwommen |
| Zusammensetzen von Linienstrichen | |
| Strichlänge | kurz ... lang |
| Linienabweichung | schwach ... stark |
| Strichverbindung | überlappend ... offen |

Tabelle 1: Entwurf für Linienattribute. Nach [Sch92].

Anhand dieser Attribute wurde ein Linienmodell entwickelt, das die Erzeugung ungenauer Linien ermöglicht. JUTTA SCHUMANN charakterisiert dazu einen Linienstil durch dessen Geometrie (Kontrollpunkte einer Kurve), den benutzten Kuventyp, die Angabe über Strichelung und deren konkrete Attribute, den Wert der Farbsättigung und der Liniendicke. Für die letzten beiden Werte können noch Angaben über deren Verlauf gemacht werden (anstreigend, abfallend, konstant bleibend, etc.).

Auf jeden Eingabepfad (Geradensegment), der z. B. Ausgabe eines Rendering-Systems oder auch eines 2D-Graphik-Systems sein kann, wird nun ein Stil angewendet. Dieser ist in einer Konfigurationsdatei für das betrachtete Objekt spezifiziert.

Zuerst erfolgt die geometrische Transformation, wie in Abbildung 1 zu sehen ist. Sowohl Stil (Modellinie M) als auch Pfad (Ziellinie Z) werden in den Koordinatenursprung verschoben (M' und Z') und der Stil anschließend durch Rotation und Skalierung auf das Bild des Pfades abgebildet. Abschließend wird der transformierte Stil zu den Ausgangskordinaten des Pfades transliert.

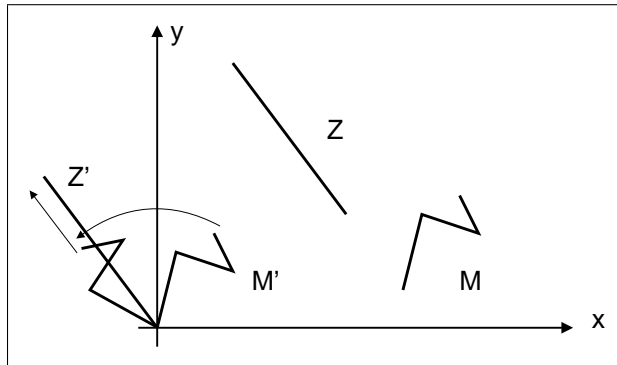


Abbildung 1: Transformation der Modellinie in die Ziellinie. Nach [Sch92]

Die Attribute Liniendicke und Farbverlauf werden auf den fertig transformierten Stil proportional angewendet (lineare Interpolation).

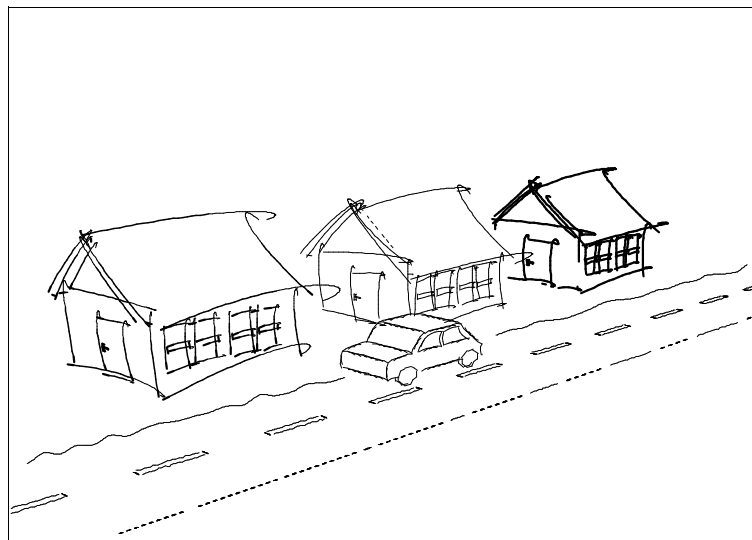


Abbildung 2: Ein Beispiel für Ergebnisse des Sketch-Renderers unter Benutzung von verschiedenen Modellinien.

Das Ergebnis der Arbeit von JUTTA SCHUMANN wurde in das Programm *Sketch-Renderer*

integriert, das am ISG entwickelt wurde. Ein Beispiel einer Graphik, die vom *Sketch-Renderer* unter Benutzung der vorgestellten Linienstile erstellt wurde, zeigt Abbildung 2.

Mit dem vorgestellten Linienmodell ist es möglich, Liniengraphiken zu verfremden. Dabei wurde sich an Attributen von realen Zeichenwerkzeugen orientiert.

Es gibt jedoch einige Nachteile am beschriebenen Linienmodell. Zum einen sind als Pfade nur Geradensegmente möglich. Damit ist es auch nicht möglich, aus mehreren Segmenten bestehende Linienzüge als eine Eingabelinie zu betrachten. Ein weiterer Kritikpunkt ist, daß Informationen aus einem 3D-Renderingprozeß wie z. B. Tiefeninformationen nicht in den Prozeß der Liniengenerierung übernommen werden.

2.2 Linienmodell nach Lars Schumann

LARS SCHUMANN¹ beschäftigte sich in seiner Diplomarbeit [Sch97] mit der Erweiterung des eben besprochenen Modells, so daß die erzeugten Liniengraphiken z. B. in Illustrationen genutzt werden können.

Die Begriffe Pfad und Stil werden weiter gefaßt, damit auch andere Eigenschaften von Zeichenwerkzeugen simuliert werden können. Der Stil gibt jetzt an, wie ein Geradensegment (die Stillinie) gezeichnet würde. Er wird auf den Pfad angewendet, der durch eine Polynomkurve gegeben ist und aus mehreren Segmenten bestehen kann.

Sowohl Pfad als auch Stil werden nun durch stückweise polynomiale Kurven beschrieben (jedoch unabhängig voneinander). Auch der Verlauf der Werte der Linienattribute wie Liniendicke oder Farbsättigung wird durch zugeordnete Kurven beschrieben. Außerdem kann jetzt auch der Pfad Attribute besitzen. Dem Stil wird weiterhin eine Dominanz zugeordnet die angibt, wie stark der Stil (und analog der Pfad) Einfluß auf die resultierende Kurve ausübt.

Die stückweise polynomiale Kurven können z. B. durch uniforme rationale B-Splines, (chordal modifizierte) rationale Catmull-Rom-Splines oder durch eine lineare Interpolation beschrieben werden. Es sind aber auch ohne Probleme weitere Kurven integrierbar. Die Kurven werden als Funktionen in parametrischer Form aufgefaßt, d.h. in Abhängigkeit von einem Parameter t aus dem Intervall $[0, 1]$.

Für die geometrische Überlagerung von Pfad und Stil werden mehrere Varianten vorgeschlagen:

¹LARS SCHUMANN ist nicht mit JUTTA SCHUMANN verwandt

Mittelwertmethode

Entsprechend der Dominanz d des Stils (bzw. des Pfades) werden sowohl der Ort als auch die Attribute von Pfad und Stil gemittelt (arithmetisches Mittel) – nachdem sie vorher auf gleiche Orientierung und gleiches Größenverhältnis gebracht wurden – und das Ergebnis ausgegeben (siehe Abbildung 3).

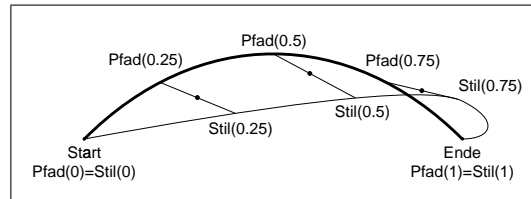


Abbildung 3: Überlagerung von Pfad und Stil durch Mittelwerte. Nach [Sch97]

Nachteile sind hier der schlecht abschätzbare Einfluß der Dominanz und die Nichteignung für zyklische Pfade.

Differenzvektormethode

Bei dieser Methode werden für jede Stillinie relativ zu einer Grundlinie (Geradensegment) bei bestimmten Positionen (bestimmte t) Differenzvektoren zwischen Stillinie und Stil bestimmt (Abbildung 4).

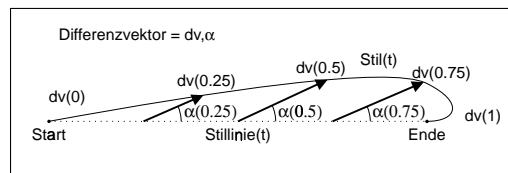


Abbildung 4: Bestimmung der Differenzvektoren. Nach [Sch97]

Die Differenzvektoren werden dann skaliert und an Pfad bei dessen korrespondierenden t abgetragen und so die Position der Ergebniskurve bestimmt. Die Attribute werden wieder entsprechend der Dominanz gemittelt (siehe Abbildung 5).

angepaßte Differenzvektormethode

Wird beim Antragen der Differenzvektoren auch noch die Richtung des Pfades beachtet, d. h. werden die Differenzvektoren relativ zur Tangente des Pfades angetragen, lassen sich bessere Ergebnisse erzielen (siehe Abbildung 6).

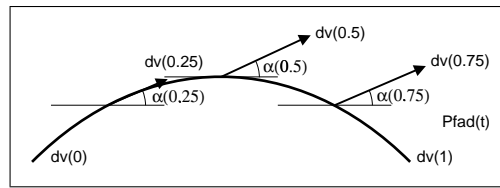


Abbildung 5: Überlagerung von Pfad und Stil durch Differenzvektoren. Nach [Sch97]

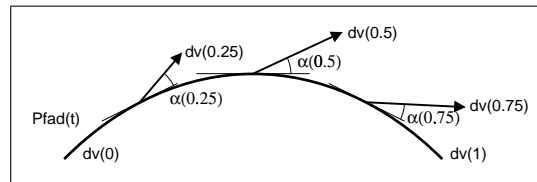


Abbildung 6: Überlagerung von Pfad und Stil durch angepaßte Differenzvektoren. Nach [Sch97]

Durch die Definition zusätzlicher Attribute wird es möglich, weitere Eigenschaften von realen Zeichenwerkzeugen nachzubilden. Zu diesen zusätzlichen Attributen zählen beispielsweise der Zeichenmodus einer Schreibfeder – deren Linienbreite ist immer entlang einer Richtung (Anstellwinkel) konstant – mit deren Anstellwinkel und verschiedene Linienenden.

Durch die Definitionen eines Stils durch mehrere Kurven ist es jetzt auch möglich, einem Pfad mehrere Linien zu überlagern und somit neue Effekte zu erzielen. So lassen sich beispielsweise mehrere nebenläufige Kurven erzeugen, wie sie u. a. in Illustrationen als doppelte Kontur eingesetzt werden.

Weiterhin ist es möglich, Linien zu stricheln. Hierbei wird spezifiziert, ob sich die Linien überlappen sollen oder ob es Zwischenräume geben soll. Es stellt sich hierbei noch die Frage, ob dann die restlichen Attribute global oder lokal wirken sollen (siehe Abbildung 7). Dies läßt sich aber durch ein weiteres Attribut spezifizieren.

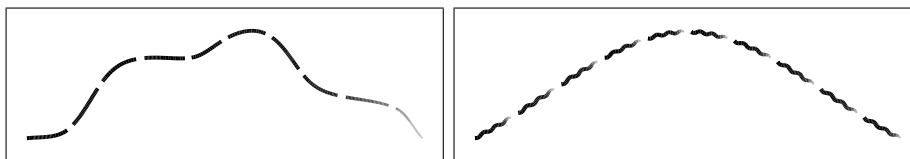


Abbildung 7: Globale und lokale Wirkung von Attributen bei gestrichelten Linien. Nach [Sch97]

Mehrere Attribute lassen sich nun zu sogenannten Linienstilen zusammenfassen und abspeichern und so einfach wiederbenutzen.

Ein Beispiel, das mit den Linienstilen von LARS SCHUMANN erzeugt wurde, ist in Ab-

bildung 8 zu sehen.

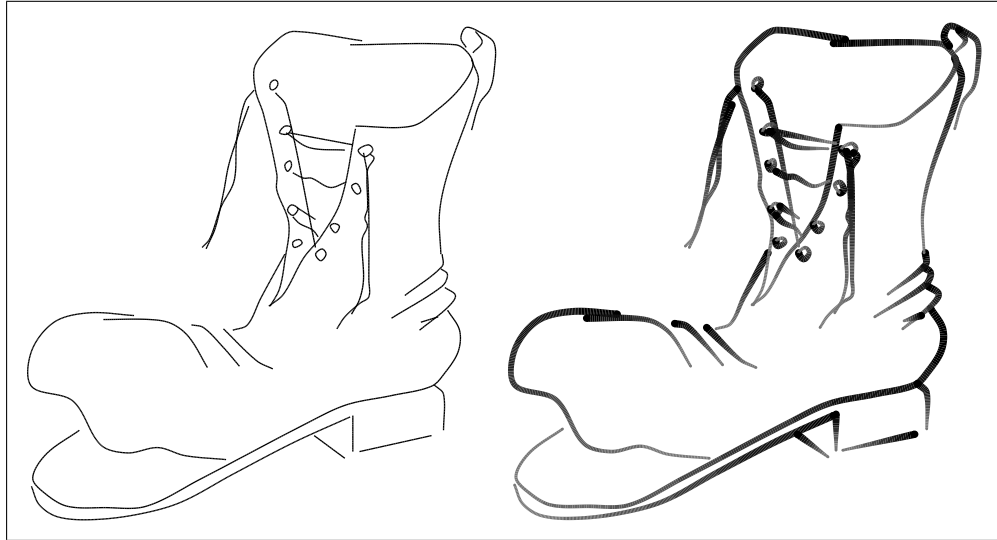


Abbildung 8: Beispiel für die Benutzung von Liniestilen (links: Original, rechts: unter Verwendung der Liniestile). Nach [Sch97]

Beide besprochenen Modelle lehnen sich in ihren Attributen an Eigenschaften herkömmlicher Zeichenwerkzeuge an und versuchen, diese nachzuahmen. Das Modell von LARS SCHUMANN ist gegenüber dem von JUTTA SCHUMANN erweitert, integriert zusätzliche Attribute und bietet mehr Freiheiten. So können auch ein nichtlineares Verhalten von Farbsättigung und Linienbreite modelliert werden. Es können verschiedene Stile definiert werden, die die entsprechenden Werte der Attribute enthalten – sozusagen virtuelle Zeichenwerkzeuge. Diese können ähnlich herkömmlichen wirken, aber auch völlig neuartige Effekte erzeugen.

3 Texturgraphiken und Stroke Textures

Die Linienmodelle von JUTTA SCHUMANN und LARS SCHUMANN werden vorrangig zum Gestalten und Manipulieren von Konturen genutzt. Ein weiteres wichtiges Gestaltungsmittel in Liniengraphiken stellen Schraffuren dar, wie bereits in Abschnitt 1 angesprochen. Daher soll im folgenden ein Ansatz zum Generieren von Texturen in Liniengraphiken vorgestellt werden.

3.1 Realisierung von Texturgraphiken mittels Stroke Textures

In photorealistischen Computergraphiken ist die Textur für die Struktur der Oberfläche, nicht jedoch für die Helligkeit verantwortlich. Im Gegensatz dazu ist in Liniengraphiken eine Textur für Struktur- und Helligkeitseindruck gleichzeitig verantwortlich.

WINKENBACH und SALESIN stellen in [WS94] als einen Ansatz dazu sogenannte *Stroke Textures* vor.

Ein *Stroke* ist hier ein Geradensegment, das durch eine *Waviness Function* (Welligkeit) und eine *Pressure Function* (Linienbreite) verändert wird. Mehrere dieser *Strokes* werden nun zu einer *Stroke Texture* zusammengefaßt.

Um nun einen bestimmten Helligkeitswert zu erreichen, werden solange *Strokes* aus der *Stroke Texture* ausgewählt und gezeichnet, bis der gewünschte Helligkeitswert erreicht ist. So wird erreicht, daß Helligkeits- und Strukturinformation der Textur gemeinsam behandelt werden.

Werden den verschiedenen *Strokes* Prioritäten zugeordnet, so spricht man von einer *Prioritized Stroke Texture*. Die Notwendigkeit für eine solche Priorisierung sieht man am Beispiel einer *Stroke Texture* für Ziegelsteine, wie sie unter anderem in Abbildung 9 in verschiedenen Helligkeitsabstufungen zu sehen ist. Die höher priorisierten *Strokes* werden zuerst gezeichnet – sie bilden die Grobform der Textur. Je dunkler die Textur sein soll, desto mehr Details werden gezeichnet.

Einige Beispiele für verschiedene *Stroke Textures* sind Abbildung 9 zu entnehmen.

Im Gegensatz zu den gespeicherten *Strokes* lassen sich natürlich auch prozedurale *Stroke Textures* erzeugen. Es werden also nicht gespeicherte *Strokes* auf ihre Eignung für die Textur überprüft, sondern jeweils neu errechnete. Beispiele für prozedurale Texturen sind „bricks“ und „shingles“ in Abbildung 9.

Computergenerierte Liniengraphiken haben gegenüber photorealistischen Graphiken einen entscheidenden Vorteil: sie sind auflösungsunabhängig². Skaliert man eine fertige Lini-

²Speicherung in einen Vektorgraphikformat vorausgesetzt

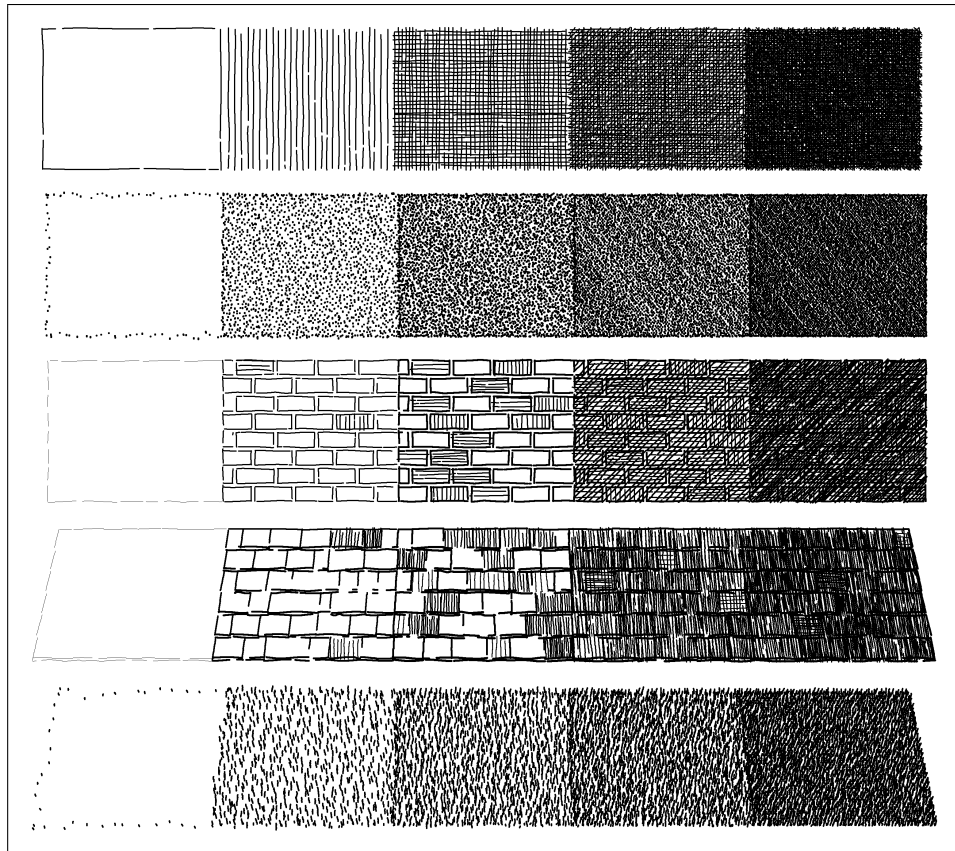


Abbildung 9: Beispiel für *Stroke Textures* (von oben: „cross-hatching“, „stippling“, „bricks“, „shingles“ und „grass“). Nach [WS94]

engraphik mit *Stroke Textures* jedoch, so wird die subjektive Helligkeit der Textur verändert. Verkleinert man das Bild, wird die Textur dunkler – vergrößert man es, wird sie heller. Ein fertiges Bild ist also doch auflösungsabhängig.

Dahingegen kann das Verfahren der *Stroke Textures* für jede Auflösung eine passende Textur liefern und ist damit auflösungsabhängig. Beispiele für eine Textur bei unterschiedlichen Auflösungen findet man in Abbildung 10.

3.2 Anwendungen

WINKENBACH und SALESIN setzen die *Stroke Textures* in einem abgeänderten 3D-Rending-System ein, in dem herkömmliche Bitmap-Texturen ersetzt wurden. Sie ergänzten ihr System zusätzlich noch um eine Möglichkeit, interaktiv Prioritäten bzw. Detailisie-

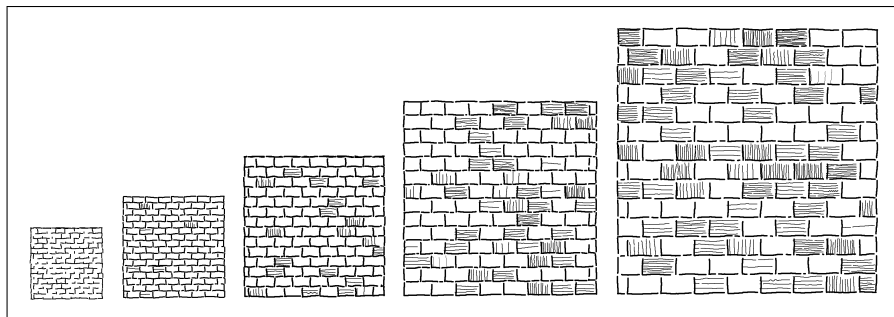


Abbildung 10: Auflösungsunabhängigkeit der *Stroke Textures*. Nach [WS94]

rungssegmente zu definieren (Abbildung 11). Die darzustellende Helligkeit einer Textur wird dann aus dem Abstand des betrachteten Ortes zum nächsten der Detailisierungssegmente bestimmt (Abbildung 12).

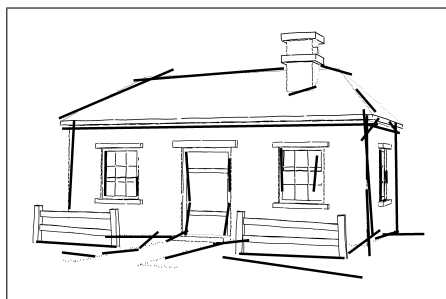


Abbildung 11: Detailisierungssegmente. Nach [WS94]

Eine weitere Einsatzmöglichkeit für *Stroke Textures* wird in [SABS94] angesprochen. Hier wird ein interaktives System beschrieben, das zur Bestimmung der Helligkeit ein Referenzbild (Grauwertbild) benutzt. Dieses wird jedoch auch noch zu anderen Zwecken genutzt, z. B. zur Kanten-Extraktion (für Konturlinien, als Clippingkanten und zur Orientierung für die *Stroke Textures*).

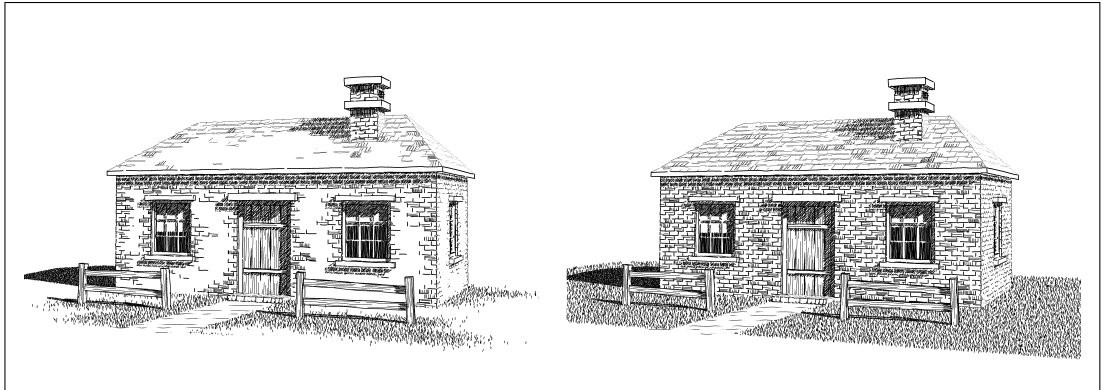


Abbildung 12: Benutzung der Detailisierungssegmente aus Abbildung 11 (links, Original rechts). Nach [WS94]

4 Liniengraphiken in der MCI

Zum Abschluß sollen jetzt noch versucht werden, Liniengraphiken in den Kontext der Mensch-Computer-Interaktion einzuordnen.

Obwohl es heutzutage durchaus möglich ist, qualitativ gute photorealistische Graphiken zu erzeugen, werden trotzdem stattdessen oder statt Photos als Illustrationen häufig Liniengraphiken eingesetzt. Einige Gründe dafür wurden bereits im Abschnitt 1 angesprochen.

In [SSRL96] wurden Untersuchungen durchgeführt, um am Beispiel von Architekturzeichnungen die Eignung von ungenauen Liniengraphiken im Gegensatz zu photorealistischen Graphiken und genauen Liniengraphiken zu bestimmen.

Es wurde untersucht, wie gut die verschiedenen Darstellungsformen in den unterschiedlichen Phasen der Gestaltung genutzt werden können, welchen Eindruck sie auf Personen machen und wie stark sie zu Änderungen stimulieren.

Dazu wurden den Versuchspersonen (Architekten und Architekturstudenten) das selbe Modell in drei unterschiedlichen Formen vorgelegt: als „normale“ CAD-Zeichnung, als mit Flat-Shading gerendertes Bild und als skizzenartige Liniengraphik:

Ein Ergebnis war, daß über die Hälfte (52,8%) der befragten Personen die skizzenartige Darstellung zur Präsentation eines ersten Entwurfes benutzen würden.

Bei der Untersuchung der Stimulation zu Veränderungen zeigte sich bei verbaler Beschreibung, Gestik und beim Zeichnen auf ein extra Blatt kein signifikanter Unterschied zwischen dem skizzenartigen Bild und dem flat-geshadeten Bild. Allerdings bei der Sti-

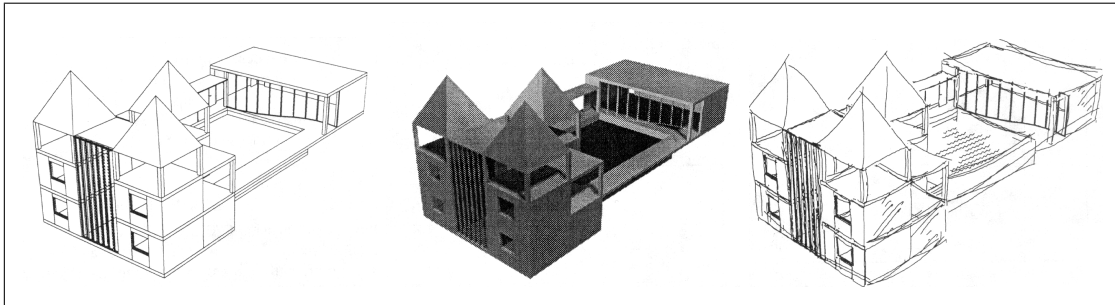


Abbildung 13: Drei Versionen des gleichen Modells. Nach [SSRL96]

mulation zum „Herummalen“ im Bild zeigte sich für die skizzenartige Darstellung ein eindeutiger Vorteil: 60% im Gegensatz zu 33% sahen sich dazu veranlaßt.

Dies deckt sich auch mit dem vorher erwähnten Ergebnis, daß eine skizzenartige Darstellung sich eher zum Darstellen eines ersten Entwurfs eignet. Es zeigt sich also, daß hier ein guter Einsatzbereich für skizzenartig verfremdete Liniengraphiken besteht.

Abbildungsverzeichnis

| | | |
|----|--|----|
| 1 | Transformation der Modelllinie in die Ziellinie. Nach [Sch92] | 4 |
| 2 | Ein Beispiel für Ergebnisse des Sketch-Renderers unter Benutzung von verschiedenen Modelllinien. | 4 |
| 3 | Überlagerung von Pfad und Stil durch Mittelwerte. Nach [Sch97] | 6 |
| 4 | Bestimmung der Differenzvektoren. Nach [Sch97] | 6 |
| 5 | Überlagerung von Pfad und Stil durch Differenzvektoren. Nach [Sch97] . . | 7 |
| 6 | Überlagerung von Pfad und Stil durch angepaßte Differenzvektoren. Nach [Sch97] | 7 |
| 7 | Globale und lokale Wirkung von Attributen bei gestrichelten Linien. Nach [Sch97] | 7 |
| 8 | Beispiel für die Benutzung von Linienstilen (links: Original, rechts: unter Verwendung der Linienstile). Nach [Sch97] | 8 |
| 9 | Beispiel für <i>Stroke Textures</i> (von oben: „cross-hatching“, „stippling“, „bricks“, „shingles“ und „grass“). Nach [WS94] | 10 |
| 10 | Auflösungsunabhängigkeit der <i>Stroke Textures</i> . Nach [WS94] | 11 |
| 11 | Detailisierungssegmente. Nach [WS94] | 11 |
| 12 | Benutzung der Detailisierungssegmente aus Abbildung 11 (links, Original rechts). Nach [WS94] | 12 |
| 13 | Drei Versionen des gleichen Modells. Nach [SSRL96] | 13 |

Tabellenverzeichnis

| | | |
|---|--|---|
| 1 | Entwurf für Linienattribute. Nach [Sch92]. | 3 |
|---|--|---|

Literatur

- [FS94] Adam Finkelstein und David H. Salesin. Multiresolution Curves. In *Proceedings of SIGGRAPH '94 (Orlando, Florida Juli 1994)*, Computer Graphics Proceedings, Annual Conference Series, Seiten 261–267. ACM SIGGRAPH, ACM Press, Juli 1994.

- [SABS94] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel und David H. Salesin. Interactive Pen-and-Ink Illustration. In *Proceedings of SIGGRAPH '94 (Orlando, Florida Juli 1994)*, Computer Graphics Proceedings, Annual Conference Series, Seiten 101–108. ACM SIGGRAPH, ACM Press, Juli 1994.
- [Sch92] Jutta Schumann. Linienqualität in ungenauen Grafiken. Diplomarbeit, TU Otto von Guericke, Institut für Simulation und Graphik, Magdeburg, Dezember 1992.
- [Sch97] Lars Schumann. Ein parametrisierbares Modell zur Darstellung von Linien. Diplomarbeit, Otto-von-Guericke Universität, Institut für Simulation und Graphik, Magdeburg, März 1997.
- [SPRF94] Thomas Strothotte, Bernhard Preim, Andreas Raab und David R. Forsey. How to Render Frames and Influence People. In *Computer Graphics Forum*, Nummer 13 in 3, Oslo, September 1994.
- [SS96] Christine Strothotte und Thomas Strothotte. *Seeing Between the Pixels: Pictures in Human-Computer Interaction*. Springer-Verlag, Berlin – Heidelberg – New York, 1. Auflage, 1996. Manuskript.
- [SSRL96] Jutta Schumann, Thomas Strothotte, Andreas Raab und Stefan Laser. Assessing the Effect of Non-Photorealistic Rendered Images in CAD. In *Proceedings of the CHI'96, April 13–18, Vancouver, BC Canada*, Seiten 35–41. ACM, New York, April 1996.
- [WS94] Georges Winkenbach und David H. Salesin. Computer-Generated Pen-And-Ink Illustration. In Andrew Glassner, Hrsg., *Proceedings of SIGGRAPH '94 (Orlando, Florida, Juli 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, Seiten 91–100. ACM SIGGRAPH, ACM Press, Juli 1994.