Erschien zuerst in: Tagungsband Graphiktag 2004, GI Chemnitz, 2004

Nichtphotorealistische Echtzeitdarstellung von skizzierten Gesichtern

Thomas Luft
Fachbereich Informatik und Informationswissenschaft
Universität Konstanz
78457 Konstanz
Tel.: +49 7531 88 5108
E-Mail: luft@inf.uni-konstanz.de

Abbildung 1: Verschiedene Ansichten des *female*-Modells.

Zusammenfassung: Diese Arbeit stellt ein Rendering-System vor, das die Echtzeitdarstellung von gezeichneten Gesichtern erlaubt. Auf Basis einer dreidimensionalen Beschreibung eines Kopfes werden signifikante Linien extrahiert und auf zeichnerische Weise dargestellt. Details, die nicht oder unzureichend geometrisch modelliert sind, können manuell mit Hilfe von Surface- und Anchor-Strokes ergänzt werden. Auf diese Weise können charakteristische, detaillierte Gesichter in Echtzeit dargestellt und animiert werden, wodurch sich interessante Anwendungsgebiete im Bereich virtueller Avatare ermöglichen.

Stichworte: Nichtphotorealistisches Rendering, Echtzeitgrafik, Linienzeichnung

1 Einführung

Die nichtphotorealistische Computergrafik hat sich in den letzten Jahren als eigenständiges Forschungsthema etabliert. Es beschäftigt sich mit alternativen Darstellungsmöglichkeiten, die we-

Konstanzer Online-Publikations-System (KOPS)
URL: http://www.ub.uni-konstanz.de/kops/volltexte/2007/2428/
URN: http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-24286

sentliche Details gezielt betonen, bestimmte Eindrücke beim Betrachter erwecken oder auch künstlerische Aspekte in eine Darstellung einfließen lassen.
Diese Arbeit beschäftigt sich mit der Modellierung und Echtzeitdarstellung von zeichnerisch wirkenden, detaillierten Gesichtern. Dadurch lässt sich beispielsweise der große Aufwand des manuellen Erstellens von handgezeichneten Animationssequenzen stark vereinfachen. Des weiteren stellen detaillierte Gesichtszeichnungen eine interessante Alternative für Mensch-Maschine-Schnittstellen dar. Durch die Echtzeitfähigkeit bieten sich Anwendungsgebiete im Bereich der Avatare, in Chat- und Messagesystemen oder für die automatische Generierung von Gebärdensprache an. Besonders im Bereich der Lernsoftware für Kinder ist eine höhere Akzeptanz durch die zeichnerische Visualisierung zu erwarten, da in diesem Alter die Akzeptanz stark vom visuellen Eindruck abhängt.

2 Bisherige Arbeiten

Das Erstellen von Linienzeichnungen anhand dreidimensionaler Geometriebeschreibungen wurde in den letzten Jahren von vielen Autoren behandelt. Die Vielzahl der existierenden Algorithmen wird nach bildraumbasierten und objektraumbasierten Verfahren unterschieden.
Bildraumbasierte Algorithmen können auf moderner Grafikhardware effizient mittels Pixelshadern und Multipassverfahren implementiert werden [MBC02]. In [Ras01] werden Verfahren vorgestellt, die auf einer prozeduralen Geometrieerstellung basieren. Diese Methoden erzeugen einfache Linienzeichnungen, die jedoch kaum eine Stilisierung der Linien über die Variation von Breite, Farbe oder Deckkraft hinaus erlauben.
Analytische Verfahren im Objektraum liefern hochqualitative Ergebnisse, die jedoch für Echtzeitanwendungen nur begrenzt einsetzbar sind, z.B. [HZ00, RK00, WS96]. In [MKT+97] wurde ein optimiertes Verfahren vorgestellt, das auf der quantitativen Unsichtbarkeit [App67] basiert und Szenen moderater Größe in Echtzeit darstellt. Hybride Verfahren eignen sich ebenfalls für die Echtzeitdarstellung, wobei die Sichtbarkeitsberechnung mit bildraumgenauen Verfahren kombiniert wird [IHS02, NM00].
Neben dem Finden von Silhouetten, die das charakteristische Merkmal eines Objektes darstellen, beschäftigen sich viele Arbeiten mit der automatischen Extraktion von so genannten *feature lines*. Diese Linienarten werden anhand von geometrischen Merkmalen aus den 3D-Daten extrahiert. Beispielsweise stellen *boundaries* und *creases* topologische und geometrische Diskontinuitäten dar. Das visuelle Ergebnis von *creases* ist für weiche Oberflächen jedoch unbefriedigend, da auch viele ungewollte Linien extrahiert werden. Um signifikante Merkmale weicher Oberflächen darzustellen, eignen sich *suggestive contours*, die in [DFRS03] vorgestellt wurden. Diese Linienart ist in der Nullmenge der radialen Krümmung lokalisierbar und wird aus silhouettennahen Blickwinkeln sichtbar.

3 Linienzeichnungen

3.1 Silhouetten

Unser Rendering-System [LD04] verwendet die in [GSG+99] vorgestellte srf-Methode für die Berechnung der Silhouette. Das Verfahren erzeugt eine stückweise lineare Approximation der Silhouette und vermeidet Artefakte, die bei Verfahren auftreten, die auf Dreieckskanten basieren [IHS02, NM00].
Für die Berechnung dienen die Normalen der Dreieckseckpunkte, deren Skalarprodukt mit dem Sichtvektor für die Interpolation der Silhouettenkanten verwendet wird. Die Verkettung der Silhouettenkanten erzeugt entweder geschlossene Silhouettenringe oder Silhouettenabschnitte, beginnend und endend an *boundaries*. *Boundaries* werden ebenso als Teil der Silhouette betrachtet und können immer als geschlossene Ringe gefunden werden.

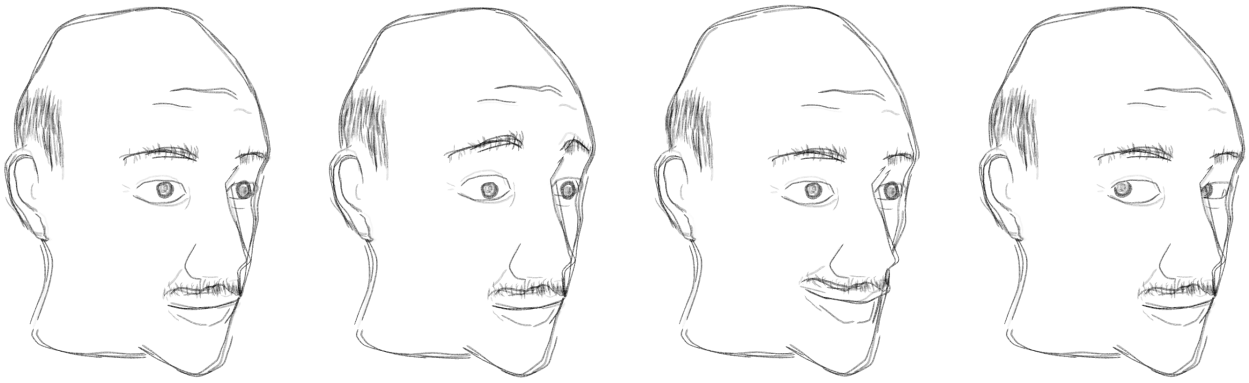

Abbildung 2: Verformung der *surface strokes* durch das gegebene 3D-Modell (*male B*).

3.2 Surface Strokes und Anchor Strokes

Für die Darstellung von Details, wie z.B. Augen, Mund, Haaren oder Accessoires, werden in dieser Arbeit manuell ergänzte Linien verwendet: *surface strokes* und *anchor strokes* (Abb. 1).
Surface strokes sind Linien, die vollständig auf der Objektoberfläche mittels baryzentrisch definierter Kontrollpunkte beschrieben werden. Dadurch lassen sich Verformungen der Geometriedaten automatisch auf die Detaillinien übertragen (Abb. 2). Das Vorbild dieser Linienart sind *decal strokes*, die im Rendering-System von [KMM+02] das direkte Zeichnen auf der Objektoberfläche ermöglichen.
Dagegen besitzen *anchor strokes* nur einen Ankerpunkt auf der Oberfläche sowie weitere im Raum befindliche, relativ beschriebene Kontrollpunkte. Diese Linienart wurde durch die *graphtal strokes*, vorgestellt in [KMN+99], inspiriert. Jedoch basieren *graphtal strokes* nicht auf individuellen Linien, sondern auf polygonalen Primitiven.
Durch die Anwendung dieser beiden Linientypen können charakteristische Details eines Gesichtes dargestellt werden, ohne dass diese Details geometrisch modelliert werden müssen. Andererseits

muss ein entsprechender Aufwand in die Erstellung der Linien sowie in die Modellierung von so genannten Trägerobjekten (Abschnitt 5.2) eingeplant werden.

4 Rendering Pipeline

4.1 Konzept

Für die Berechnung und Animation haben wir eine nichtphotorealistische Rendering-Pipeline entworfen. Die Pipeline ermöglicht die Berechnung der Linien, die Bestimmung von sichtbaren und verdeckten Abschnitten sowie die zeichnerische Visualisierung.

4.2 Vorverarbeitung

Für die Echtzeitfähigkeit ist eine geeignete Datenhaltung notwendig, die eine effiziente Extraktion und Berechnung der gesuchten Linien ermöglicht. So müssen beispielsweise Informationen über benachbarte Dreiecke zur Verfügung stehen, die eine schnelle Suche und Verkettung von Silhouettenkanten ermöglicht. Außerdem werden die Oberflächennormalen zur Laufzeit berechnet, um während der Animation und Verformung die Konsistenz sicherzustellen.
Aufgrund der Verformbarkeit der Geometrie muss auf Algorithmen verzichtet werden, die eine effizientere Silhouettensuche ermöglichen, z.B. *gauss map* [BE99][GSG+99] oder die *normal-cone hierarchy* [SGG+00].

4.3 Linienberechnung

Die Linienextraktion umfasst die Suche und Verkettung von Silhouettenkanten sowie die Umrechnung der baryzentrischen Koordinaten der Detaillinien. Anschließend erfolgt die Projektion der gefundenen Kontrollpunkte in den Bildraum. Für die endgültige Liniendarstellung werden diese Kontrollpunktlisten ausgedünnt, indem Punkte ausgeschlossen werden, deren euklidischer Abstand unter einen vorgegebenen Schwellwert fällt.

4.4 Hidden Line Removal

Als Vorbild für das Bestimmen sichtbarer Segmente dient der Tiefenbildansatz von [IHS02], ein hybrides Verfahren. Das Tiefenbild wird dabei entlang der darzustellenden Linien abgetastet und die Tiefenwerte verglichen. Bei dieser Methode können durch numerische Instabilitäten und Quantisierungsfehler sichtbare Artefakte auftreten, die mittels verschiedener Techniken unterdrückt werden können.
Zum einen wird die 8-Nachbarschaft betrachtet, um einen Abtastpunkt genauer zu untersuchen, zum anderen können mittels zweier Schwellwerte kurze sichtbare oder kurze verdeckte Abschnitte zusammengefasst werden. Ein anderes Problem entsteht bei einer Approximation des Objektes

Abbildung 3: Verschiedene Zeichenstile des Liniengenerators (*male A*).

durch zu große Dreiecke. Durch die stückweise Interpolation der Silhouette auf der Objektoberfläche gibt es Silhouettenkanten, die auf einem rückseitig orientierten Dreieck liegen. Dadurch kommt es bei der Abtastung des Tiefenbildes zu falschen Messwerten. Dem kann nur durch eine ausreichend feine Triangulierung oder mittels aufwendiger, analytischer Methoden entgegengewirkt werden.
Das Rücklesen des gerenderten Tiefenbildes (512×512 Pixel) aus dem Grafikkarten-Speicher begrenzt die Bildrate auf einer GeforceFX 5800 auf ca. 50 Bildern pro Sekunde.

4.5 Liniengenerator

Abschließend visualisiert ein Liniengenerator die erzeugten Linieninformationen auf zeichnerische Weise. Der durch die Kontrollpunkte beschriebene Linienpfad wird durch kubische B-Splines approximiert und mit Hilfe von Normalenvektoren zu einem *quadstrip* aufgespannt. Dadurch erhalten die Linien eine Linienstärke und Linienbreite, die entlang des Pfades stochastisch variiert werden können. Um ein bestimmtes Zeichenmedium nachzuahmen, besteht die Möglichkeit eine ortsstabile Textur auf den *quadstrip* abzubilden.
Die Nachahmung der Zeichenstile wird durch eine Kombination verschiedener, mathematisch beschriebener Effekte erreicht (Abb. 3). Auf diese Weise wird die Polygonstruktur stochastisch modifiziert. Dies erlaubt einen höheren Freiheitsgrad im Vergleich zu Verfahren, die auf Beispiellinien basieren [FTP99, HOCS02].
Für alle stochastischen Einflüsse werden Perlinsche Rauschfunktionen verwendet [Per85], wodurch eine Voraussetzung für kohärente Darstellungen geschaffen ist. Die frame-to-frame-kohärente Darstellung von Silhouetten ist jedoch durch die topologische Veränderung während einer Bewegung sehr schwierig. Speziell für kohärente, stilisierte Silhouetten wurden Verfahren in [Bou98, KDMF03] vorgestellt, welche dies mit gewissen Einschränkungen ermöglichen. Die temporäre Kohärenz wird in unserem Rendering-System zur Zeit nur durch den Determinismus der Algorithmen erzeugt, dadurch kann es unter Umständen zu sichtbaren Artefakten kommen.

5 Modellierung

5.1 Interaktiver Editor

Wir stellen einen interaktiven Editor zur Verfügung, mit dem Detaillinien auf der Oberfläche gezeichnet werden können. Auf diese Weise können mit Hilfe von *surface strokes* Details wie Mund, Augen, Augenbrauen oder Falten nachgebildet werden. Eine weitere Möglichkeit bestände darin, die Linien mit Hilfe von Filtern aus den originalen Texturen zu extrahieren. Das Erstellen der *anchor strokes*, um z.B. Haare nachzubilden, ist in professionellen Modellierungstools am einfachsten und in unserem Editor nur zu Testzwecken implementiert.
Für die Modellierung von homogenen Strukturen sind sowohl zweidimensionale als auch dreidimensionale Linientexturen denkbar. Diese Texturen würden aus einer Anzahl von vordefinierten Linien bestehen, welche im Vorverarbeitungsschritt mittels herkömmlicher Texturkoordinaten auf die Objektoberfläche abgebildet werden. Somit ließe sich der Modellierungsaufwand für Haare oder Bart verringern. Ebenso könnte auf diese Weise eine Wiederverwendbarkeit von gezeichneten Details erfolgen.

(a) Trägerobjekt mit aufgezeichneten Details

(b) verborgenes, verdeckungsneutrales Trägerobjekt

Abbildung 4: Trägerobjekte zum Modellieren von Details.

5.2 Unsichtbare Zeichenebenen

Für eine charakteristische Nachbildung der Gesichter, speziell von Details wie Haaren oder Accessoires, reicht die gegebene Geometrie nicht aus. Es ist sinnvoll und notwendig Objekte zu modellieren, die nur als Träger der Detaillinien dienen. Voraussetzung für den Einsatz solcher Trägerobjekte ist die volle Kontrolle über deren Einfluss auf das *hidden line removal*. Für jedes Objekt lassen sich dessen Sichtbarkeit sowie dessen Verdeckungseigenschaften einstellen (Abb. 4). Beispielsweise lassen sich somit Trägerobjekte erstellen, die Verdeckung erzeugen, jedoch ansonsten unsichtbar sind, d.h. es wird keine Silhouette berechnet.

6 Performance und Ergebnis

Die Performance wurde anhand verschieden komplexer Gesichtsmodelle getestet. Als Plattform diente ein 3GHz Pentium4 mit einer GeForceFX5800 Grafikkarte. In Tabelle 1 können die Berechnungszeiten verglichen werden (*#triang* – Anzahl Dreiecke, *#surf* – Anzahl surface strokes, *#anch* – Anzahl anchor strokes, *prep* – Vorverarbeitung, *lineComp* – Linienberechnung, *depth* – Tiefenbild erzeugen, *hlr* – Sichtbarkeit bestimmen, *render* – Visualisierung der Linien, *#quads* – Anzahl von OpenGL-Quads, *fps* – Bildrate).
Die Ergebnisse zeigen, dass sich das System für die zeichnerische Visualisierung detaillierter Gesichter eignet. Es erlaubt eine Echtzeitdarstellung und Animation von Szenen moderater Komplexität. Mit Hilfe der eingeführten Detaillinien und Trägerobjekte können eine Vielzahl von Details, die nur aufwendig in der Geometrie zu modellieren wären, nachgebildet werden.

7 Ausblick

Nichtphotorealistische Visualisierungen sind durch die Konzentration auf wesentliche Details hoch informativ. Aus diesem Grund besteht eine wichtige, zukünftige Aufgabe darin, die Akzeptanz bei unterschiedlichen Benutzergruppen und die kognitiven Eigenschaften von nichtphotorealistischen Avataren zu untersuchen. Dabei spielen Aspekte der Wahrnehmung, Psychologie und Kommunikation eine große Rolle. Aus den Ergebnissen lassen sich speziellere Anwendungsgebiete ableiten und Vorteile bzw. Nachteile der Darstellung spezifizieren.
Die temporäre frame-to-frame-Kohärenz stellt ein weiteres Problem dar, welches wir in Zukunft lösen wollen. Ausschlaggebend sind dabei Inkohärenzen, die bei der Silhouettenberechnung entstehen und Artefakte, bei der Berechnung der sichtbaren Liniensegmente. Neben dem Ziel der temporären Kohärenz, kann in weiteren Arbeiten mittels beliebig definierter Kohärenzverhalten die visuelle Erscheinung gezielt beeinflusst werden und neuartige Effekte und Eindrücke im Bereich der zeichnerischen Visualisierung untersucht werden.
Die in der Arbeit vorgestellte Modellierung von Trägerobjekten und Detaillinien und dessen Einfluss auf die zeichnerische Qualität muss untersucht werden. Ziel ist es, die modellierte Szene als gültige Zeichnung identifizieren zu können und nicht nur als Abbild von dreidimensionalen Objekten anzusehen. Hierbei spielt die Art der Linienvisualisierung und das Kohärenzverhalten eine entscheidende Rolle.

8 Danksagung

Das *female*-Modell wurde von Stefan Hertz (Filmakademie Ludwigsburg) zur Verfügung gestellt. Das *maleB*-Modell ist Eigentum von Marc Alexa (TU-Darmstadt) und Wolfgang Mueller (PH Ludwigsburg).

model	#triang	#surf	#anch	prep	lineComp	depth	hlr	render	#quads	fps
male A	32085	712	1000	5.6ms	4.0ms	14.6ms	4.9ms	12.3ms	12487	24.2
female	34964	874	4032	6.1ms	5.1ms	18.3ms	7.4ms	37.1ms	21981	13.5
male B	6818	801	0	1.0ms	0.9ms	13.5ms	1.2ms	11.2ms	6498	36.1

Tabelle 1: Berechnungszeiten für die verschiedenen Renderingschritte.

Literatur

[App67] Arthur Appel. The notion of quantitative invisibility and the machine rendering of solids. *Proc. ACM Natl. Mtg.*, page 387, 1967.

[BE99] Fabien Benichou and Gershon Elber. Output sensitive extraction of silhouettes from polygonal geometry. In *Pacific Graphics '99*, October 1999.

[Bou98] Lubomir Bourdev. Rendering nonphotorealistic strokes with temporal and arc-length coherence. Master's thesis, Brown University, 1998.

[DFRS03] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics*, 22(3):848–855, July 2003.

[FTP99] W. T. Freeman, J. B. Tenenbaum, and E. Pasztor. An example-based approach to style translation for line drawings. Technical Report TR-99-11, MERL - A Mitsubishi Electric Research Laboratory, 1999.

[GSG+99] Bruce Gooch, Peter-Pike J. Sloan, Amy Gooch, Peter S. Shirley, and Rich Riesenfeld. Interactive technical illustration. In *1999 ACM Symposium on Interactive 3D Graphics*, pages 31–38, April 1999.

[HOCS02] Aaron Hertzmann, Nuria Oliver, Brian Curless, and Steven M. Seitz. Curve analogies. In *Rendering Techniques 2002: 13th Eurographics Workshop on Rendering*, pages 233–246, June 2002.

[HZ00] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 517–526, July 2000.

[IHS02] Tobias Isenberg, Nick Halper, and Thomas Strothotte. Stylizing Silhouettes at Interactive Rates: From Silhouette Edges to Silhouette Strokes. *Computer Graphics Forum (Proceedings of Eurographics)*, 21(3):249–258, September 2002.

[KDMF03] Robert D. Kalnins, Philip L. Davidson, Lee Markosian, and Adam Finkelstein. Coherent stylized silhouettes. *ACM Transactions on Graphics*, 22(3):856–861, July 2003.

[KMM+02] Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. Wysiwyg npr: drawing strokes directly on 3d models. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 755–762. ACM Press, 2002.

[KMN+99] Michael A. Kowalski, Lee Markosian, J. D. Northrup, Lubomir Bourdev, Ronen Barzel, Loring S. Holden, and John Hughes. Art-based rendering of fur, grass, and trees. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 433–438, Los Angeles, 1999. Addison Wesley Longman.

[LD04] Thomas Luft and Oliver Deussen. Non-photorealistic real-time rendering of characteristic faces. In *Pacific Graphics 2004*, 2004. To appear.

[MBC02] Jason L. Mitchell, Chris Brennan, and Drew Card. Real-time image-space outlining for non-photorealistic rendering. In *Proceedings of SIGGRAPH 2002*, Sketches and Applications, page 239, 2002.

[MKT+97] Lee Markosian, Michael A. Kowalski, Samuel J. Trychin, Lubomir D. Bourdev, Daniel Goldstein, and John F. Hughes. Real-time nonphotorealistic rendering. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pages 415–420, August 1997.

[NM00] J. D. Northrup and Lee Markosian. Artistic silhouettes: A hybrid approach. In *NPAR 2000 : First International Symposium on Non Photorealistic Animation and Rendering*, pages 31–38, June 2000.

[Per85] Ken Perlin. An image synthesizer. In *Computer Graphics (Proceedings of SIGGRAPH 85)*, volume 19, pages 287–296, July 1985.

[Ras01] Ramesh Raskar. Hardware support for non-photorealistic rendering. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 41–47. ACM Press, 2001.

[RK00] Christian Rössl and Leif Kobbelt. Line-art rendering of 3d-models. In *8th Pacific Conference on Computer Graphics and Applications*, pages 87–96, October 2000.

[SGG+00] Pedro V. Sander, Xianfeng Gu, Steven J. Gortler, Hugues Hoppe, and John Snyder. Silhouette clipping. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 327–334, July 2000.

[WS96] Georges Winkenbach and David H. Salesin. Rendering parametric surfaces in pen and ink. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pages 469–476, August 1996.