

# Generierung synthetischer Hologramme durch Texture Mapping

Alf Ritter, Joachim Böttger, Oliver Deussen\*  
Otto-von-Guericke-Universität Magdeburg

## Zusammenfassung

Die synthetische Erzeugung von Hologrammen stellt hohe Anforderungen an Rechenleistung und Speicherkapazität. Traditionelle Methoden erlauben daher lediglich eine zeitaufwendige Hologrammgenerierung. Eine Beschleunigung wird durch den Einsatz von Methoden der Computergrafik erreicht, die die Ausnutzung von Grafikhardware gestatten. Dieser Beitrag beschreibt die Erzeugung synthetischer Hologramme mit Hilfe von Texturen. Dazu werden unterschiedliche Techniken der Überlagerung von Texturen diskutiert. Der Ansatz umfaßt sowohl punkt- als auch linienbasierte Methoden.

## 1 Einleitung

Die Holografie ist eine Methode, die die Aufnahme und dreidimensionale Visualisierung von Gegenständen gestattet. Bei der Aufnahme eines Hologramms wird der Gegenstand (Objekt) mit kohärentem Licht beleuchtet und das Interferenzmuster der Objekt- und einer Referenzwelle (aus der gleichen kohärenten Lichtquelle eines Lasers) auf einer hochauflösenden fotografischen Platte festgehalten. Für die Rekonstruktion (Wiedergabe) des Objektes wird die entwickelte Platte mit der gleichen Referenzwelle beleuchtet. Ein Teil des Lichts wird dabei so gebeugt, daß eine genaue Kopie des vom Objekt ausgehenden Wellenfeldes entsteht, und das Objekt wird in seiner räumlichen Ausdehnung dargestellt wird [BS93].

Die synthetische Holografie verkörpert die rechnergestützte Simulation der Vorgänge von Hologrammaufnahme und Rekonstruktion. Sie ist eine sehr attraktive Visualisierungsmethode, da sie die Umsetzung aller vom natürlichen Sehen her bekannten Phänomene der Tiefenwahrnehmung erlaubt [MZ92].

Einer der ältesten und am meisten verbreiteten Ansätze der synthetischen Hologrammgenerierung ist das Zusammensetzen des darzustellenden Objektes aus Punktlichtquellen [Wat66, LH68]. Hierbei wird pro Punktlichtquelle eine Zonenplatte berechnet und auf dem Hologramm überlagert, was einen enormen Rechenaufwand bedeutet. Einsparungen lassen sich durch die Vorabberechnung von Punkteinflüssen auf das Hologramm erreichen. Das wird in [Luc93] für die Erzeugung von holografischen Stereogrammen demonstriert.

---

\*Fakultät für Informatik, Institut für Simulation und Graphik, {alf.jboettge,deussen}@isg.cs.uni-magdeburg.de

Eine Fortentwicklung [LG95, Luc97] gestattet bereits die Ausnutzung von Grafikhardware. Durch Kopplung mit einem speziellen Display [WLS<sup>+</sup>95] ist hier eine Erzeugung holografischer Stereogramme nahezu in Echtzeit möglich. Eine weitere Variante der Einbeziehung von Computergrafiktechniken in die Erzeugung holografischer Stereogramme wird in [HH92] vorgestellt.

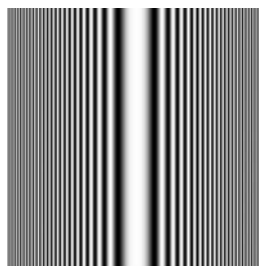
Neben den genannten punktbasierten Ansätzen ergibt sich eine weitere Möglichkeit der Verminderung des Rechenaufwands, wenn das Objekt aus Linien anstatt aus Punkten zusammengesetzt ist [FLB86, Les87]. Eine Linie repräsentiert dabei eine große Menge von Punkten, was sich in der Verminderung der Anzahl der Rechenoperationen widerspiegelt. Die in diesem Beitrag vorgestellte texturbasierte Lösung gestattet die Zusammensetzung des darzustellenden Objektes sowohl aus Linien als auch aus Punkten. Die holografische Repräsentation dieser geometrischen Primitive mit Hilfe von Texturen wird im Abschnitt 2 vorgestellt. Anschließend erfolgt im Abschnitt 3 eine Diskussion verschiedener Techniken zur Überlagerung dieser Texturen. Abschnitt 4 rundet die Darstellung mit einer Zusammenfassung und einem Blick auf zukünftige Arbeiten ab.

## 2 Hologrammgenerierung mit Texture Mapping

### 2.1 Linienbasierte Hologrammerzeugung

Eine unendlich lange, leuchtende Linie emittiert bei konstanter Phase entlang der Linie eine zylindrische Welle [FLB86, Les87]. Durch eine lineare Phasenverteilung entsteht eine konische Welle. Linien endlicher Länge werden nachgebildet, indem ihr Einflußbereich auf das Hologramm durch Clipping [FLB86] begrenzt wird.

Die von Linien ausgehenden und auf ein Hologramm treffenden Wellen werden im holografischen Äquivalent (vgl. [RDWS97]) des Eingabeobjektes nachgebildet. Das Äquivalent besteht aus einem texturierten Rechteck pro Eingabelinie, das entsprechend der Linienlänge angeordnet wird. Die aufgebrauchte Textur wird vorab berechnet. Sie enthält das Muster einer zylindrischen Welle (siehe Abbildung 1).

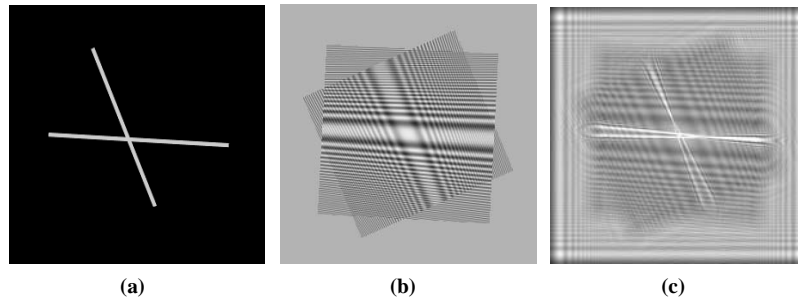


**Abbildung 1:** Holografisches Muster einer zylindrischen Welle

Eine Verzerrung, realisiert durch die Modifikation der Texturkoordinaten entlang des Rechtecks, ermöglicht die Simulation der holografischen Muster konischer Wellen.

Das Hologramm entsteht durch eine Parallelprojektion des Äquivalents auf die Bildebene, die der Hologrammebene entspricht. Dieser Ansatz wurde erfolgreich auf die holografische

Darstellung von Splines in 3D erweitert [RBD<sup>+</sup>97]. Abbildung 2 zeigt ein Beispiel für die linienbasierte Hologrammgenerierung. Im Hologramm in Bild 2(b) sind dabei konische Wellenmuster zu sehen. Alle in den Bildern gezeigten Rekonstruktionen wurden durch

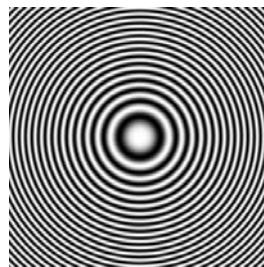


**Abbildung 2:** Holografische Darstellung zweier sich kreuzender Linien: Originalobjekt (a), Hologramm (b) mit konischen Wellenmustern, rekonstruiertes Bild (c) in einem vorgegebenen Abstand vom Hologramm

Simulation mit dem System DIGIOPT [ABST95] erstellt.

## 2.2 Punktbasierte Methode

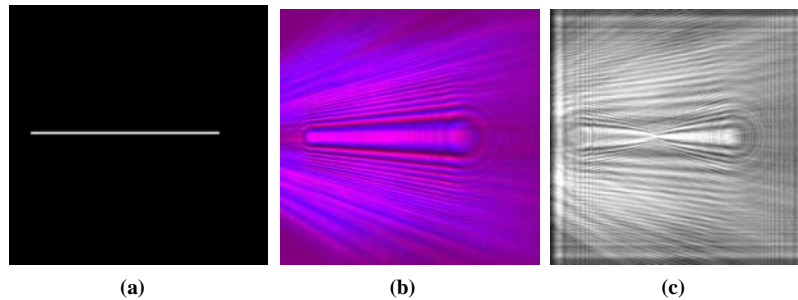
Die holografische Darstellung linienhafter Objekte mit Hilfe des holografischem Äquivalents [RDWS97] bedient sich dem in [FLB86] eingeführten Clipping, um die Linienlänge zu begrenzen. Die Linie kann bei der optischen Rekonstruktion [RB97] lediglich durch das durch Clipping begrenzte Fenster betrachtet werden. Dieser Nachteil wird dann aufgehoben, wenn die Information über das geometrische Eingabeprimativ über das gesamte Hologramm verteilt wird. Das geschieht in einer punktbasierten Variante der Hologrammerzeugung mit Texturen. Die Grundtextur ist in diesem Fall eine Fresnelsche Zonenplatte, wie sie in Abbildung 3 gezeigt wird.



**Abbildung 3:** Fresnelsche Zonenplatte als Textur

Für die Hologrammgenerierung wird das darzustellende Objekt (in Abbildung 4 eine waagerechte, zur Hologrammebene geneigte Linie) in Punkte aufgelöst. Das holografische Äquivalent enthält für jeden dieser Punkte ein Element, das aus einem Quadrat mit Fresnelscher Zonenplatte als Textur besteht. Jedes Element wird anhand der Lage des korrespondierenden Eingabepunktes platziert und entsprechend des Abstandes des Punktes zur

Bildebene vergrößert bzw. verkleinert. Das resultierende Hologramm ist in Bild 4(b) zusammen mit einer Rekonstruktion (Bild 4(c)) zu sehen.



**Abbildung 4:** Holografische Darstellung einer waagerechten, zum Hologramm geneigten Linie mit Hilfe von Zonenplatten-Texturen: Originallinie (a), Hologramm (b), rekonstruiertes Bild (c) mit der Linienmitte im Fokus

Neben dem Clipping wird in der Punktlösung ein zweiter Nachteil der Linienvariante behoben. Die Simulation zylindrischer und konischer Wellen basiert auf bestimmten Annahmen für die Phasenverteilung entlang der Linie [FLB86]. Für die Display-Holografie ist jedoch eine Zufallsphase erforderlich. Deren Realisierung ist in der Linienvariante nicht gelungen. In der Punktlösung wird die Zufallsphase dadurch erreicht, daß den einzelnen Elementen (beispielsweise entlang der Linie) im holografischen Äquivalent Zonenplatten unterschiedlicher Phase als Textur zugewiesen werden. Die Zuweisung erfolgt dabei anhand von Zufallszahlen.

### 3 Überlagerung von Texturen

Die Verwendung von vorab berechneten Texturen hat gegenüber den traditionellen Methoden der Hologrammgenerierung (vgl. [Wat66],[LH68],[FLB86], [RB97]) den Vorteil, daß die aufwendigen Berechnungen holografischer Muster im eigentlichen Prozeß der Hologrammgenerierung eingespart werden. Die texturierten Gebilde sind geeignet zu transformieren und zu überlagern, um Interferenzen korrekt nachzubilden. Nachfolgend werden einige Varianten der Überlagerung vorgestellt.

#### 3.1 Grauwerttexturen

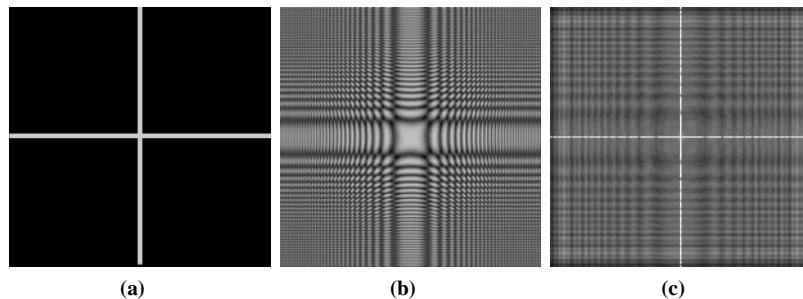
Zuerst wird die additive Überlagerung von Texturen mit Hilfe von Alpha-Blending [NDW93] vorgestellt. Ein Beispiel für diese Form der Hologrammgenerierung ist bereits in Abbildung 2 zu sehen. Hierbei repräsentiert jedes Element des holografischen Äquivalents ein Hologramm eines einzelnen geometrischen Primitivs. Dieses Hologramm enthält schon eine Referenzwelle, womit durch die Texturen lediglich reelle Zahlen nachgebildet werden müssen. Reelle Zahlen können mit Hilfe von Grauwerttexturen kodiert werden.

Diese Methode gestattet die Hologrammgenerierung für eine geringe Anzahl zu überlagernder Elemente (bis ca. 20). Bessere Resultate werden durch die Repräsentation komplexer Zahlen mit einer getrennten Kodierung von Real- und Imaginärteil innerhalb der Textur erreicht. Das wird im folgenden Abschnitt gezeigt.

### 3.2 Verwendung mehrerer Farbkanäle

Die Abbildung von komplexen Zahlen wird durch die Verwendung mehrerer Farbkanäle anstelle von Grauwerten realisiert. Bei der Vorabberechnung der Textur wird für jedes Texel der Realteil der komplexen Zahl als blaue Farbintensität und der Imaginärteil rot gespeichert. Die Flanken rechts und links der dunklen Streifen (vgl. Abbildungen 1 und 3) haben somit eine unterschiedliche Färbung und nicht ein einheitliches Grau wie in der zuerst vorgestellten Methode. Die Referenzwelle wird dabei in einer separaten Textur kodiert und in einem Element des holografischen Äquivalents repräsentiert.

Für die Überlagerung von texturierten Elementen im Bildspeicher müssen sowohl positive als auch negative Zahlen auf positive Farbintensitäten abgebildet werden, negative auf geringere Intensitäten (untere Hälfte der Farbskala) und positive auf größere (obere Hälfte). Dadurch wird nach erfolgter Überlagerung der Texturen bei Abspeicherung des Hologramms aus dem gerenderten Bild ein Nachbearbeitungsschritt nötig. Für jede erfolgte Überlagerung ist pro Pixel und Farbkanal derjenige Farbwert abzuziehen, der bei obengenannter Abbildung der 0 zugeordnet wurde. Das Abziehen wird dadurch realisiert, indem auf den Alpha-Wert zurückgegriffen wird, der für jedes Pixel die Anzahl der Überlagerungen ausweist. Abbildung 5 zeigt ein Beispiel für die Hologrammgenerierung unter Benutzung mehrerer Farbkanäle.



**Abbildung 5:** Hologrammgenerierung durch Repräsentation komplexer Zahlen mit Texturen: Originalobjekt (a), Hologramm (b) und Rekonstruktion (c) des Hologramms aus (b)

Dadurch, daß bei der Nachbearbeitung die Anzahl der Operationen pro Pixel mit einbezogen werden muß, kann dieser Schritt nicht durch grafische Operationen mit Hilfe von OpenGL nachgebildet werden. Er muß vielmehr in einem zusätzlichen Rechenschritt ausgeführt werden, wodurch eine Geschwindigkeitseinbuße eintritt. Nach dem Abziehen des Offsets (pro Überlagerung der der 0 zugeordnete Wert) wird die Amplitude

$$A = \sqrt{\text{real}^2 + \text{imaginär}^2} \quad (1)$$

pro Pixel berechnet. Das resultierende Hologramm wird als Grauwertbild abgelegt. Da das für die Hologrammausgabe verwendete Filmmaterial empfindlich gegenüber der Intensität als dem Quadrat der Amplitude

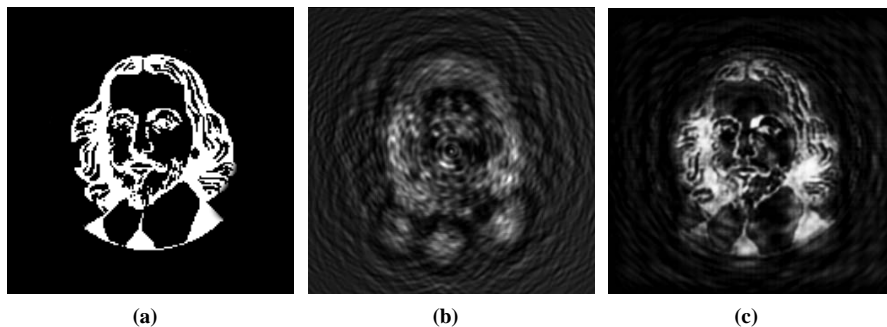
$$I = A^2 = \text{real}^2 + \text{imaginär}^2 \quad (2)$$

ist [Goo68], genügt es, die Intensität auf dem Hologramm abzuspeichern. Dadurch entfällt die Bestimmung der Quadratwurzel.

Die additive Zusammenfassung von „linearen“ Texturen im Bildspeicher hat den Nachteil, daß ähnlich wie in der im Abschnitt 3.1 gezeigten Methode bereits nach relativ wenigen Überlagerungen an den Kreuzungspunkten (Bildmitte in Abbildung 5(b) zeigt einen Kreuzungspunkt) Überläufe auftreten. Diese Partien erscheinen weiß und liefern keine klaren Rekonstruktionen. Eine Lösung dieses Problems wird im folgenden Abschnitt gezeigt.

### 3.3 Benutzung des Akkumulationspuffers

Der Akkumulationspuffer [HA90] ist ein besonderer Grafikspeicher, der gegenüber dem Bildspeicher, der normalerweise für das Rendering benutzt wird, eine größere Tiefe aufweist. Dadurch gestattet er die Überlagerung einer größeren Anzahl grafischer Elemente. Im vorliegenden Fall ermöglicht er eine Verarbeitung holografischer Äquivalente komplizierterer Objekte. Das Äquivalent weist dabei eine entsprechend große Anzahl von Elementen (bis zu einigen tausend) auf. Abbildung 6 zeigt ein Beispiel.



**Abbildung 6:** Hologrammgenerierung mit Hilfe des Akkumulationspuffers: Originalobjekt (a) aus 4000 Punktlichtquellen (in einer Ebene), Hologramm (b) und rekonstruiertes Bild (c)

Pro Renderschritt wird der Inhalt des Bildspeichers in den Akkumulationspuffer übertragen (akkumuliert). Ein wichtiger Vorzug dieses Verfahrens ist, daß dabei pro Pixel eine Addition des Bildspeicherinhalts zum Inhalt des Akkumulationsspeichers möglich ist, aber auch eine Subtraktion durchgeführt werden kann. Im Gegensatz zum Bildspeicher sind dabei in fast allen Implementationen des Akkumulationspuffer auch negative Werte zulässig [OARB93].

Für die Benutzung des Akkumulationspuffers werden die Texturen nach einem anderen Schema als in der Bildspeichervariante aufbereitet. Wie in der herkömmlichen Mehrkanalvariante wird der Realteil der komplexen Zahl auf den Blauanteil pro Texel und der

Imaginärteil auf den Rotanteil abgebildet. Die Akkumulation gestattet jedoch keine gleichzeitige Addition oder Subtraktion, diese müssen in unterschiedlichen Schritten ausgeführt werden. Daher werden positive und negative Werte auch in separaten Texturen kodiert.

Durch diese Kodierung von positiven und negativen Zahlen in separaten Texturen entfällt die in Abschnitt 3.2 erwähnte Nachbearbeitung anhand der Anzahl der Überlagerungen, um pro Pixel jeweils den Wert, auf den die Zahl 0 abgebildet wird, auszufiltern. Dadurch kann die Berechnung der Intensität aus der Amplitude lt. Gleichung 2 effizient mittels Pixel-Map [NDW93] in OpenGL ausgeführt werden. Dazu wird eine Funktion für die Abbildung von Pixelfarbwerten vorgegeben. Im vorliegenden Fall wird beim Auslesen des Hologramms aus dem Grafikspeicher jeweils für den blauen (Realteil der komplexen Zahl) und den roten (Imaginärteil) Farbkanal das Quadrat bestimmt. Die Addition in Gleichung 2 wird durchgeführt, indem von OpenGL anstatt der einzelnen Werte pro Farbkanal die Luminanz als Summe der Farbwerte pro Pixel angefordert wird.

Damit ist ein wichtiges Ziel, die vollständige Abbildung der Operationen zur Generierung des synthetischen Hologramms mittels Grafikoperationen in OpenGL, erreicht. Der Akkumulationspuffer erlaubt durch seine größere Tiefe gegenüber dem Bildspeicher eine Überlagerung einer größeren Anzahl von Elementen des holografischen Äquivalents. Allerdings kann pro Akkumulationsschritt dabei jeweils nur ein Element verarbeitet werden. Das zieht eine Geschwindigkeitseinbuße nach sich. Die holografische Darstellung komplizierter Objekte erfordert die Anwendung des Akkumulationspuffers, einfachere Objekte können jedoch nach wie vor im Bildspeicher dargestellt werden. Nachfolgend wird eine spezielle Variante gezeigt, die die additive Überlagerung aus Abschnitt 3.2 umgeht.

### 3.4 Besondere Farbzuzuordnung

Bei dieser Methode werden Real- und Imaginärteil der komplexen Zahl nicht wie in den vorher gezeigten Varianten linear auf Farbintensitäten innerhalb der Textur abgebildet. Vielmehr erfolgt für eine komplexe Zahl  $x$  eine Zuordnung nach folgendem Schema:

$$blau = \frac{2^{x_{real}}}{2}, \quad rot = \frac{2^{x_{imaginär}}}{2}. \quad (3)$$

Dabei bezeichnet  $x_{real}$  den Realteil und  $x_{imaginär}$  den Imaginärteil von  $x$ .

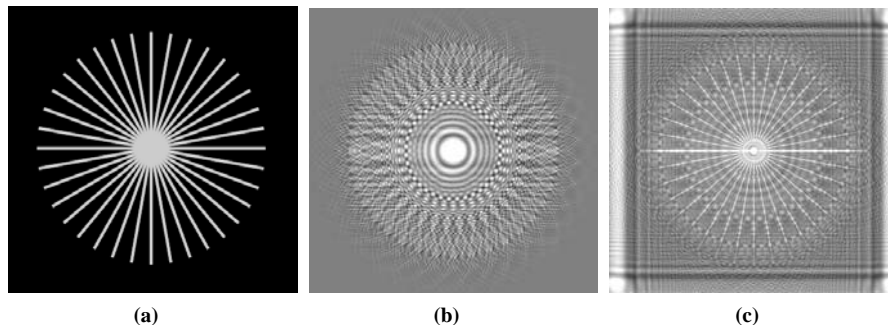
Aufgrund dieser Einteilung erfolgt die Überlagerung der Texturen beim Rendern des Hologramms nun durch Multiplizieren von Farbintensitäten anstelle durch eine Addition. Hier ist es möglich, ebenso wie in der Variante mit Akkumulationspuffer, die Amplitude ohne aufwendige Einbeziehung der Anzahl der Überlagerungen pro Pixel zu bestimmen. Im Gegensatz zur Hologrammgenerierung mit Akkumulationspuffer erfolgt dabei jedoch keine elementweise Verarbeitung. Das Hologramm wird vielmehr in einem Schritt gerendert. Dadurch ist eine schnellere Hologrammgenerierung möglich.

Die Amplitude für jedes Hologrammpixel ergibt sich nach Gleichung 3 aus

$$A = \sqrt{(\log_2(2 \times blau))^2 + (\log_2(2 \times rot))^2}. \quad (4)$$

Auf dem Hologramm wird wiederum die Intensität als Quadrat der Amplitude abgespeichert. Die Bestimmung der Intensität unter Auswertung von Gleichung 4 wird auch hier durch Pixel-Map und Auslesen der Luminanz realisiert.

Ein Beispiel multiplikativer Überlagerung ist in Abbildung 7 zu sehen. Diese Methode ge-



**Abbildung 7:** Multiplikatives Blending: Originalobjekt (a) aus Linien (im Abstand von  $10\theta$ ), Hologramm (b) und Rekonstruktion (c)

stattet eine schnellere Hologrammgenerierung für Objekte bis ca. 100 Punkte bzw. Linien. Die holografische Darstellung komplizierterer Gebilde muß jedoch mit der zuvor gezeigten Variante unter Ausnutzung des Akkumulationspuffers erfolgen.

#### 4 Zusammenfassung und Ausblick

Die Anwendung von Methoden der Computergrafik, insbesondere der Einsatz von Texture Mapping, gestattet eine schnellere Hologrammgenerierung, als das mit traditionellen Methoden der synthetischen Holografie möglich ist. Die Ausnutzung hochentwickelter Grafikhardware bietet hier ein enormes Potential.

Zukünftige Arbeiten untersuchen die Darstellung von Flächen. Dazu existieren mehrere Varianten. Zum einen kann die Fläche aus vielen Punktlichtquellen zusammengesetzt werden, die hinreichend dicht plaziert werden. Jede dieser Punktlichtquellen würde dann, wie im Abschnitt 2.2 gezeigt, durch je ein Element des holografischen Äquivalents mit Fresnelscher Zonenplatte als Textur repräsentiert.

Eine weitere Variante ist die Zusammensetzung von Flächen aus Linien. Wie bei der Scanlinien-Konvertierung werden die Flächen dabei aus Scanlinien zusammengesetzt. Allerdings ist zu erwarten, daß das in der Linienlösung angewandte Clipping zu Aliasing-Effekten entlang der Begrenzungen der Fläche führt.

Drittens soll untersucht werden, inwieweit Dreiecke, aus denen sich beliebige Flächen zusammensetzen lassen, direkt durch ein einzelnes Element im holografischen Äquivalent repräsentiert werden können.



## Literatur

- [ABST95] H. Aagedal, Th. Beth, H. Schwarzer und S. Teiwes. Design of Paraxial Diffractive Elements with the CAD System DigiOpt. In I. Cindrich und S. H. Lee, Hrsg., *Diffractive and Holographic Optics Technology III*, Band 2404 d. *SPIE Proceedings*, Seiten 50–58, 1995.
- [BS93] L. Bergmann und C. Schaefer. *Lehrbuch der Experimentellen Physik*, Band III, Optik. de Gruyter, Berlin, New York, 9. Auflage, 1993.
- [FLB86] C. Frère, D. Leseberg und O. Bryngdahl. Computer-Generated Holograms of Three-dimensional Objects Composed of Line Segments. *Journal of the Optical Society of America (JOSA)*, 3(5):726–730, 1986.
- [Goo68] J. W. Goodman. *Introduction to Fourier Optics*. McGraw-Hill, New York, 2. Auflage, 1968.
- [HA90] P. Haerberli und K. Akeley. The Accumulation Buffer: Hardware Support for High-Quality Rendering. In F. Baskett, Hrsg., *SIGGRAPH 90 Conference Proceedings*, Annual Conference Series, Seiten 309–318, Dallas, August 1990. ACM SIGGRAPH, ACM Press.
- [HH92] K. Haines und D. Haines. Computer Graphics for Holography. *IEEE Computer Graphics and Applications*, 12(1):37–46, Januar 1992.
- [Les87] D. Leseberg. Computer Generated Holograms: Cylindrical, Conical, and Helical Waves. *Applied Optics*, 26(20):4385–4390, 1987.
- [LG95] M. Lucente und T. A. Galyean. Rendering Interactive Holographic Images. In R. Cook, Hrsg., *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, Seiten 387–394, Los Angeles, August 1995. ACM SIGGRAPH, ACM Press.
- [LH68] L. B. Lesem und P. M. Hirsch. Computer Synthesis of Holograms for 3-D Display. *Communications of the ACM*, 11(10):661–674, Oktober 1968.
- [Luc93] M. Lucente. Interactive Computation of Holograms Using a Look-up Table. *Journal of Electronic Imaging*, 2(1):28–34, 1993.
- [Luc97] M. Lucente. Interactive Three-dimensional Holographic Displays: Seeing the Future in Depth. *ACM Computer Graphics*, 31(2):63–66, Mai 1997.
- [MZ92] M. McKenna und D. Zeltzer. Three Dimensional Display Systems for Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 1(4):421–458, 1992.
- [NDW93] J. Neider, T. Davis und M. Woo. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley, Reading, 1993.

- [OARB93] OpenGL Architecture Review Board. *OpenGL Reference Manual: The Official Reference Document for OpenGL Release 1*. Silicon Graphics Inc., 1993.
- [RB97] A. Ritter und Th. Benziger. Synthetische Holografie und Computergrafik. In O. Deussen und P. Lorenz, Hrsg., *Simulation und Animation '97*, Seiten 76–86, Magdeburg, März 1997. SCS – Society for Computer Simulation Int.
- [RBD<sup>+</sup>97] A. Ritter, Th. Benziger, O. Deussen, Th. Strothotte und H. Wagener. Synthetic Holograms of Splines. In H.-P. Seidel, B. Girod und H. Niemann, Hrsg., *3D Image Analysis and Synthesis '97*, Seiten 11–18, Erlangen, 1997. infix.
- [RDWS97] A. Ritter, O. Deussen, H. Wagener und Th. Strothotte. Holographic Imaging of Lines: A Texture-Based Approach. In P. Storms, Hrsg., *International Conference on Information Visualization IV'97*, Seiten 272–278, London, August 1997. IEEE Computer Society.
- [Wat66] J. P. Waters. Holographic Image Synthesis Utilizing Theoretical Methods. *Applied Physics Letters*, 9(11):405–407, 1966.
- [WLS<sup>+</sup>95] J. A. Watlington, M. Lucente, C. J. Sparrell, V. M. Bove und I. Tamitani. A Hardware Architecture for Rapid Generation of Electro-Holographic Fringe Patterns. In S. A. Benton, Hrsg., *Practical Holography IX*, Band 2406 d. *SPIE Proceedings*, Seiten 23–34, 1995.