

## Modern Volumenvisualisierung

Hanspeter Pfister

TR2004-066 January 2004

### Abstract

Over the last decade, volume rendering has become an invaluable visualization technique for a wide variety of applications. In this paper, we discuss modern, hardware-accelerated methods for the visualization of volume datasets; ray casting, texture slicing, shear-warp and shea-image rendering and splatting.

*Information Technology 2004*

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



# Moderne Volumenvisualisierung

## Modern Volume Visualization

Hanspeter Pfister, Mitsubishi, Cambridge (U.S.A.)

**Zusammenfassung** Die Volumenvisualisierung hat sich in den letzten zehn Jahren zu einer unentbehrlichen Visualisierungstechnik für viele Anwendungsbereiche entwickelt. In diesem Beitrag präsentieren wir einen Überblick über moderne, hardwarebeschleunigte Methoden zur Visualisierung von Volumendaten: Raycasting, Texture-Slicing, Shear-Warp-Rende-

ring und Splatting. ▶▶▶ **Summary** Over the last decade, volume rendering has become an invaluable visualization technique for a wide variety of applications. In this paper, we discuss modern, hardware-accelerated methods for the visualization of volume datasets: ray casting, texture slicing, shear-warp and shear-image rendering, and splatting.

**KEYWORDS** 1.3 [Computer Graphics] Visualisierung, Volumenvisualisierung, Raycasting, Texture-Slicing, Shear-Warp-Rendering, Splatting

### 1 Einleitung

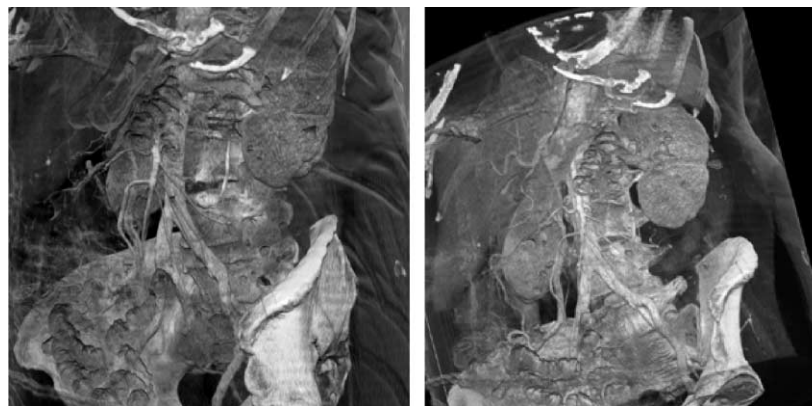
In den letzten zehn Jahren ist die Volumenvisualisierung von wissenschaftlichen Daten unentbehrlich für viele Anwendungen in der Medizin, der Biotechnologie, den Ingenieurwissenschaften, der Astrophysik und anderen Bereichen geworden. Die untersuchten Objekte oder Phänomene sind volumetrischer Natur, typischerweise repräsentiert als ein dreidimensionales Gitter von Volumenelementen, oder *Voxel*. Beispiele volumetrischer Daten sind dreidimensionale Datensätze aus der bildgebenden Medizin (CT oder MRT), numerische Simulationen der Strömungsdynamik (*computational fluid dynamics*, CFD) oder Berechnungen von Finite-Elemente-Modellen.

Einer der größten Vorteile volumetrischer Daten ist, dass sie die innere Struktur der Objekte repräsentieren können. Die direkte Volumenvisualisierung generiert Bilder direkt aus den Volumendaten ohne den Umweg über eine Oberflächenrepräsentation zu nehmen.

Sie ermöglicht es, interne Strukturen qualitativ darzustellen, inklusive amorpher und transparenter Merkmalen. Die Volumenvisualisierung ist besonders geeignet, strukturell schwache oder „fuzzy“ Informationen aus den Volumendaten hervorzuheben (Bild 1).

Obwohl die Volumenvisualisierung seit langem ein populäres Forschungsgebiet ist, dauerte es wegen mangelnder Interaktivität sehr lange

bis sie weite Verbreitung in der Anwendung fand. Glücklicherweise ermöglicht der technische Fortschritt moderner PC-Grafikkarten mittlerweile interaktive und sogar Echtzeit-Bildraten. Schnelle Bilderzeugung ist essentiell für die Analyse und Interpretation von Volumendaten. Die Bewegungsparallaxe führt zu einem höheren Realismus, wenn der Benutzer die 3D-Objekte flüssig rotieren kann. Eine unmittelbare visuelle



**Bild 1** Volumetrische CT-Daten eines menschlichen Oberkörpers, dargestellt mit Volumenvisualisierung. Die Bilder wurden von Yin Wu und Jan Hardenbergh, TeraRecon Inc., zur Verfügung gestellt.

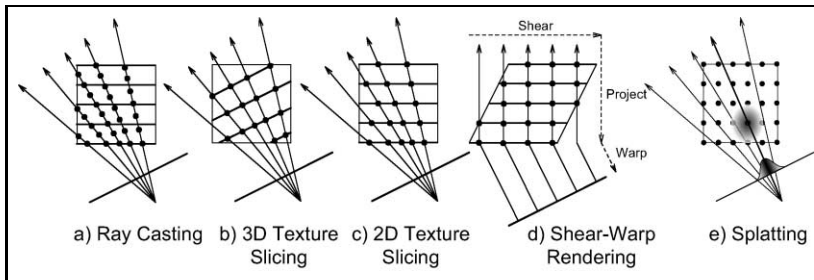


Bild 2 Überblick über die Algorithmen.

Rückkopplung ermöglicht das interaktive Experimentieren mit verschiedenen Visualisierungsparametern. Die Bildqualität von hardwarebeschleunigten Methoden reicht heute an die der besten Software Algorithmen heran.

In diesem Beitrag beschreiben wir die gängigsten Methoden der modernen, hardwarebeschleunigten Volumenvisualisierung. Wir besprechen *Raycasting*, 3D- und 2D-*Texture-Slicing*, *Shear-Warp-Rendering* und *Splatting* (Bild 2). Wir konzentrieren uns auf die allgemeinen Algorithmen und Prinzipien und hoffen beim Leser das Interesse zu wecken, die weiterführende Literatur zu studieren.

## 2 Raycasting

Eine der beliebtesten Volumenvisualisierungstechniken ist Raycasting, das mit einem einfachen und optisch korrektem Algorithmus eine sehr hohe Bildqualität erreicht [2]. Dabei werden simulierte Sehstrahlen vom Projektionszentrum durch Bildschirmpixel ins Volumen geschossen (Bild 2a). Entlang der Strahlen werden in regelmäßigem Abstand so genannte Resampling-Punkte berechnet. An jedem Resampling-Punkt werden die Volumendaten interpoliert, typischerweise unter Verwendung trilinearere Interpolation in einer Zelle mit  $2 \times 2 \times 2$  Voxel.

Für die Volumenklassifizierung und die Beleuchtung wird an jedem Resampling-Punkt ein Gradientenvektor berechnet. Die meisten Gradientenfilter sind eine einfache Erweiterung von 2D-Kantenerkennungsfiltren in die dritte Dimen-

sion. Gradientenvektoren werden während der Bildgenerierung berechnet oder mit den Volumendaten gespeichert.

Die Volumenklassifizierung bildet die physikalischen Eigenschaften der Volumendaten, wie zum Beispiel Materialdichte, auf optische Eigenschaften ab. Mögliche optische Parameter sind zum Beispiel Farbe (RGB), Absorptionseigenschaften und Opazität ( $\alpha$ ). Man unterscheidet zwischen *Pre-* und *Post-Klassifizierung*, je nachdem ob die Volumendaten vor oder nach der Interpolation klassifiziert werden.

Die Gradientenvektoren werden als Approximation einer Oberflächennormalen benutzt, um ein lokales Beleuchtungsmodell zu berechnen. Die Beleuchtung der Volumenpunkte führt zu realistischeren Bildern mit abgestuften Oberflächenbeleuchtungen. Alle klassifi-

zierten und beleuchteten Punkte entlang eines Sehstrahls werden in einem Pixel akkumuliert. Die Summe aller Sehstrahlen liefert schließlich das berechnete Bild. Um die Bildqualität zu erhöhen, können auch mehrere Strahlen pro Pixel berechnet werden, die dann mit einem 2D-Filter kombiniert werden.

Raycasting liefert eine hohe Bildqualität und ist konzeptionell einfach zu implementieren. Leider stellt es auch sehr hohe Ansprüche an die Rechenleistung und liefert typischerweise keine interaktiven Bildraten. Roettger et al. [6] haben vor kurzem eine Implementation von Raycasting auf modernen, handelsüblichen Grafikkarten präsentiert (Bild 3). Das Verfahren ist interaktiv und liefert eine höhere Bildqualität als schnellere Texture-Slicing-Methoden.

## 3 Texture-Slicing

Texture-Slicing ist die vorherrschende Volumenvisualisierungsmethode auf modernen Grafikkarten (*programmable graphics processing units*, GPUs). Texturbasierte Techniken können mit 2D- oder 3D-Texturen implementiert werden.

### 3.1 3D-Texture-Slicing

Wird das Volumen als 3D-Textur gespeichert, können so genannte

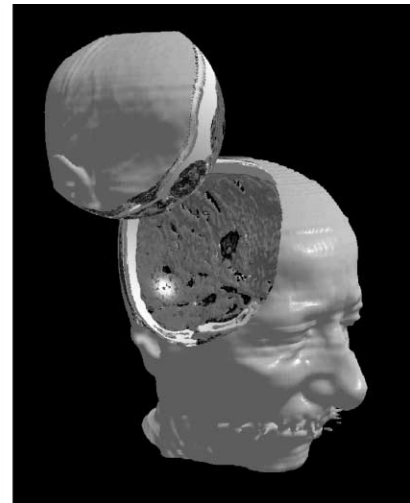
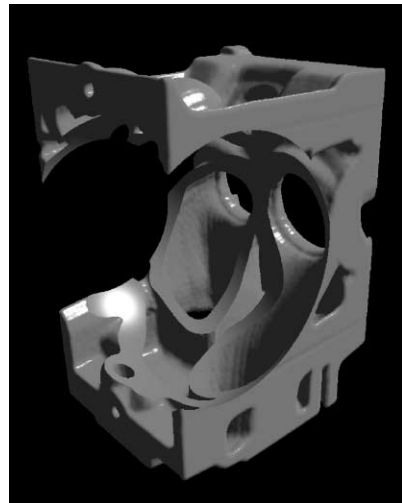


Bild 3 Hardwarebeschleunigtes Raycasting des CT-Scans eines Bonsai Baums mit  $256^3$  Voxel. Das Bild wurde von Stefan Roettger, Universität Stuttgart, Deutschland, zur Verfügung gestellt.

*Texture-Slices* parallel zur Bildebene berechnet werden (Bild 2b). Das Volumen wird dabei mit Polygonen geschnitten, die parallel zur Bildebene liegen. Texturkoordinaten an den Eckpunkten der Polygone werden entsprechend der Volumen-Bounding-box berechnet. Die Polygone werden dann von der Grafikhardware auf den Bildschirm projiziert, wobei die Volumendaten automatisch von der Hardware trilinear-interpoliert werden. Die Polygone mit RGB $\alpha$ -Texturen werden dann ins Bild akkumuliert. Programmierbare Hardware erlaubt es, diffuse und spekulare Beleuchtung sowie Schatten für jedes Pixel zu berechnen (Bild 4).

Heutige GPUs unterstützen noch nicht die Berechnung von Gradientenvektoren mit der Hardware. Gradientenvektoren werden in Software vorberechnet und mit den 3D-Texturen gespeichert. Die Volumendaten werden post-klassifiziert, indem die Klassifikationstabelle als 1D- oder 2D-Textur gespeichert wird. Die interpolierten Volumendaten werden dann als Lookup-Koordinaten in die Klassifikationstabelle verwendet.

Die nichtlineare Klassifizierung von Volumendaten kann zu Artefakten führen, wenn die Abtastrate nicht groß genug ist, d.h., wenn



**Bild 5** Volume-Clipping des CT-Scans eines Motorblocks (links,  $256^2 \times 110$ ) und eines MRT-Scans eines Kopfes (rechts,  $256^3$ ). Die Bilder wurden von Daniel Weiskopf, Universität Stuttgart, Deutschland, zur Verfügung gestellt.

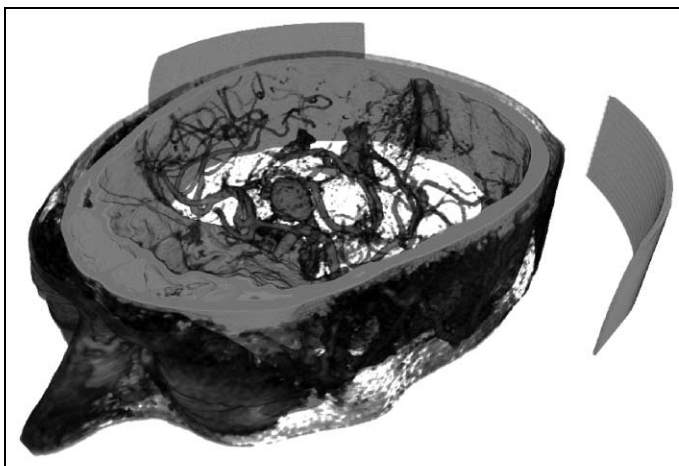
nicht genügend Polygonschnitte durch das Volumen berechnet werden. Eine elegante Lösung dieses Problems ist die so genannte *Pre-Integration* [1]. Anstatt einzelne Polygone zu klassifizieren, wird die Klassifizierung auf die Volumendaten zwischen zwei nacheinander folgenden Polygonen angewendet. Die Integration von RGB- und  $\alpha$ -Werten für diese so genannten *Slabs* wird vorberechnet und in einer Tabelle gespeichert. Die Bildqualität der Pre-Integration ist wesentlich besser als traditionelle Techniken, ohne höhere Rechenzeiten zu benötigen.

Das Innere von Volumendaten kann mit so genannten *Volume-Clipping*-Techniken dargestellt werden, die auch mit 2D- und 3D-Texturhardware implementiert werden können [7]. Beliebige Clipping-Geometrie kann mit Polygonen oder durch zusätzliche Volumina definiert werden (Bild 5).

### 3.2 2D-Texture-Slicing

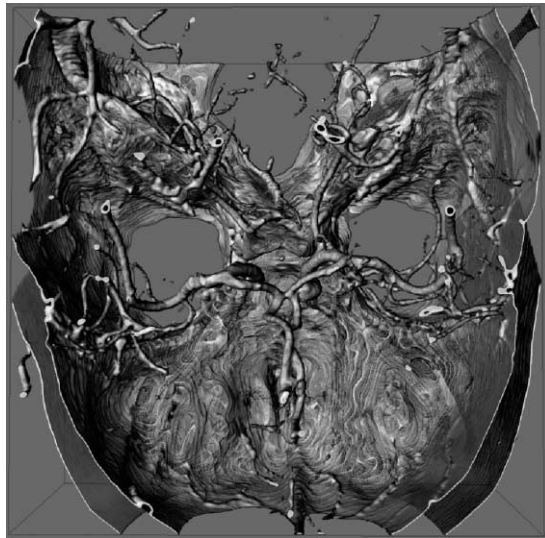
Obwohl 3D-Texturen auf modernen Grafikkarten standardmäßig vorhanden sind, werden 2D-Texturen immer noch häufig in der Volumenvisualisierung eingesetzt. Einer der Hauptgründe ist die höhere Geschwindigkeit von 2D-Texture-Mapping.

2D-Texturmethode verwenden achsparallele *Texture-Slices*, die an einer Seite der Volumendaten ausgerichtet sind (Bild 2c). Wie bei 3D-Methoden werden die *Texture-Slices* von Polygonen aufgespannt. Die Polygone verlaufen stets parallel zur Seite des Volumens, die der Bildebene zugewandt ist. Diese Seite wird auch *Basisebene* genannt. Die texturierten Volumenschnitte werden bei der Projektion der Polygone in die Bildebene bi-linear interpoliert. Um die Geschwindigkeit zu erhöhen, können die Volumendaten dreifach in den Hauptrichtungen abgespeichert werden. Die Methoden für Klassifizierung, Beleuchtung



**Bild 4** CT-Angiographie eines menschlichen Gehirns ( $512^2 \times 128$ ). Berechnet auf einer ATI Radeon 9700 Grafikkarte mit 3D-Textur und Post-Klassifizierung. Das Bild wurde von Christof Rezk-Salama, Universität Erlangen, Deutschland, zur Verfügung gestellt.

**Bild 6** CT-Angiographie eines menschlichen Gehirns ( $512^2 \times 128$ ). Berechnet auf einer NVIDIA GeForce-4 Grafikkarte mit 2D-Texturen und Multi-texturing. Das Bild wurde von Christof Rezk-Salama, Universität Erlangen, Deutschland, zur Verfügung gestellt.



und Akkumulierung von Texturen entsprechen denen des 3D-Texture-Slicing.

Werden keine zusätzlichen Texture-Slices zwischen den Originaldaten interpoliert, kann dies zu Alias-Artefakten führen, wenn die Volumendaten oder die Volumenklassifizierung hohe Frequenzen enthalten. Um das zu verhindern, können während der Visualisierung zusätzliche Texture-Slices mit Hilfe von Multi-Texturing berechnet werden [5] (Bild 6).

#### 4 Shear-Warp-Rendering

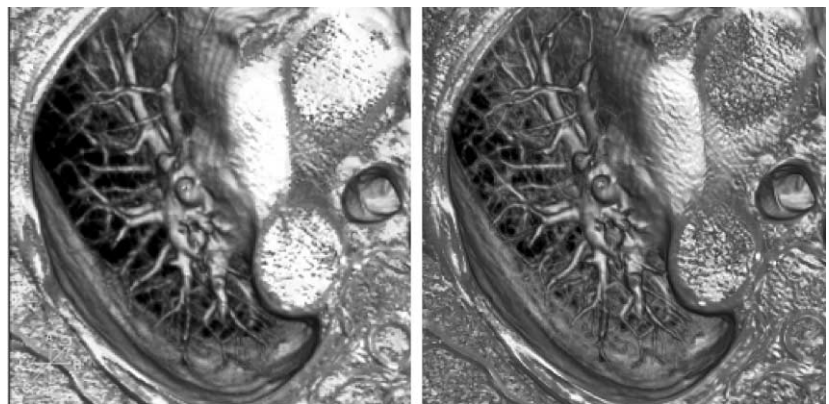
Shear-Warp-Rendering projiziert die Volumendaten so in die Bildebene, dass entlang eines Sehstrahls interpolierte Voxel aufeinander zu liegen kommen (Bild 2d). Die Projektion der Volumenscheiben kann in eine Scherung und einen 2D-Warp separiert werden. Für Parallelprojektionen liefert Shear-Warp-Rendering die höchste Leistung aller Volumenvisualisierungsalgorithmen und lässt sich gut in Hardware implementieren.

##### 4.1 VolumePro

Die VolumePro 500 Architektur [3] verwendet den Shear-Warp-Algorithmus für Parallelprojektionen der Volumendaten. Das neuere VolumePro 1000 System verwendet einen neuen Ansatz, das *Shear-Image-Order-Raycasting*. Analog zum Raycasting werden dabei Sehstrahlen

durch Pixel geschickt, aber die Volumendaten werden wie bei 2D-Texture-Slicing parallel zur Basisebene interpoliert [8]. Im Gegensatz zum traditionellen Shear-Warp-Rendering muss das endgültige Bild nicht gewarpt werden, weil die interpolierten Voxel mit den Pixeln ausgerichtet sind. Dies führt zu einer höheren Bildqualität (Bild 7).

Ein besonderes Merkmal der VolumePro-Architekturen ist eine spezielle Speicheradressierung der Volumendaten mit so genanntem 3D-Skewing, das optimalen Zugriff auf die Volumendaten erlaubt [3]. Blöcke und 2D-Schichten parallel zu den Basisebenen können sehr effizient vom Speicher gelesen werden, ohne dass die Volumendaten mehrmals gespeichert werden müssen.



**Bild 7** Vergleich von Shear-Warp-Rendering (links, berechnet mit VolumePro 500) und Shear-Image-Order-Raycasting (rechts, berechnet mit VolumePro 1000). Die Bilder wurden von Yin Wu und Jan Hardenbergh, TeraRecon Inc., zur Verfügung gestellt.

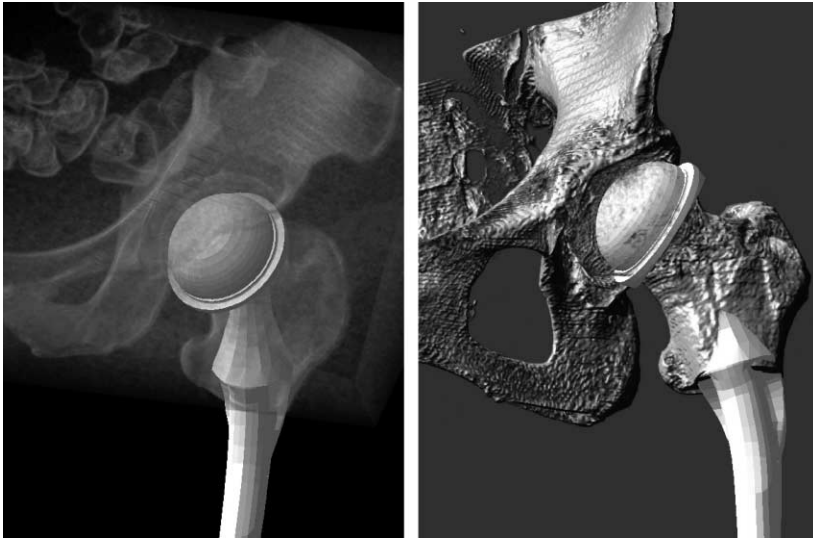
VolumePro implementiert die tri-lineare Interpolation und die Berechnung von Gradientenvektoren in Hardware. Ähnlich zu 2D-Texture-Slicing-Methoden [5] können zusätzlich interpolierte Schichten generiert werden, um eine höhere Bildqualität zu erreichen. Die interpolierten Daten werden mit hochwertiger Phong-Beleuchtung in Hardware beleuchtet.

Für die Volumenklassifizierung verwendet VolumePro 1000 mehrere kaskadierte Lookup-Tabellen, die mit einer Hierarchie von arithmetischen und logischen Operationen verknüpft werden können. Voxel können bis zu vier Materialien speichern, und jedes Material wird mit einer separaten Lookup-Tabelle klassifiziert. Die Klassifizierung und Interpolation sind querverbunden, um der Applikation sowohl Pre- als auch Post-Klassifizierung zu ermöglichen.

VolumePro 1000 implementiert zusätzliche Features wie Volumen-Clipping, perspektivistische Projektion und einige Optimierungen, um unnötige Berechnungen zu vermeiden. Volumendaten können auch mit Oberflächendaten (Polygonen) gemischt werden, was einige interessante Anwendungen in der Chirurgesimulation ermöglicht (Bild 8).

#### 5 Splatting

Splatting faltet jedes Voxel im Volumen mit einem 3D-Rekonstruk-



**Bild 8** Eine Polygon-Prothese ist zusammen mit dem CT-Scan einer menschlichen Hüfte visualisiert. Das Bild wurde von Yin Wu und Jan Hardenbergh, TeraRecon Inc., zur Verfügung gestellt.

tionsfilter und akkumuliert den Beitrag der Voxel auf der Bildebene (Bild 2e). Typischerweise wird auf die Voxel wie bei 2D-Texture-Slicing zugegriffen. Fortgeschrittene Splatting-Methoden mit höherer Bildqualität traversieren das Volumen ähnlich zu 3D-Texture-Slicing.

Splatting ist attraktiv, weil es sehr effizient sein kann, die integrierten Rekonstruktionsfilter vorzuberechnen. Im einfachsten Fall wird ein integrierter Filter vorberechnet und als 2D-Textur gespeichert. Diese 2D-Textur wird dann mit jedem Voxel multipliziert, ins Bild projiziert und akkumuliert. Splatting erlaubt es auch, höherwertige Interpolationsfilter zu verwenden, die tri-linearer Filterung überlegen sind. Die Volumendaten werden typischerweise preklassifiziert und jedes Voxel speichert die resultierenden RGBα-Werte.

Die Splatting-Methode mit der höchsten Bildqualität ist EWA-Splatting [9]. Ein 3D-Gauss-Rekonstruktionsfilter kann analytisch mit einem 2D-Gauss-Tiefpassfilter in der Bildebene integriert werden. Das Ergebnis ist ein Filter im Bildraum, der analytisch definiert ist und sehr effizient in Grafikkhardware evaluiert werden kann [4] (Bild 9).

## 6 Ausblick

Aktuelle Volumenvisualisierung ist von dem konsequenten Einsatz programmierbarer Grafikkhardware geprägt. Nach wie vor ist Texture-Slicing die meistverwendete Methode auf GPUs. Vor kurzem konnte auch Raycasting auf der GPU implementiert werden. Die algorithmischen Vorteile von Raycasting könnten es zu einer sehr populären Methode mit hoher Geschwindigkeit und ausgezeichneter Bildqualität machen lassen.

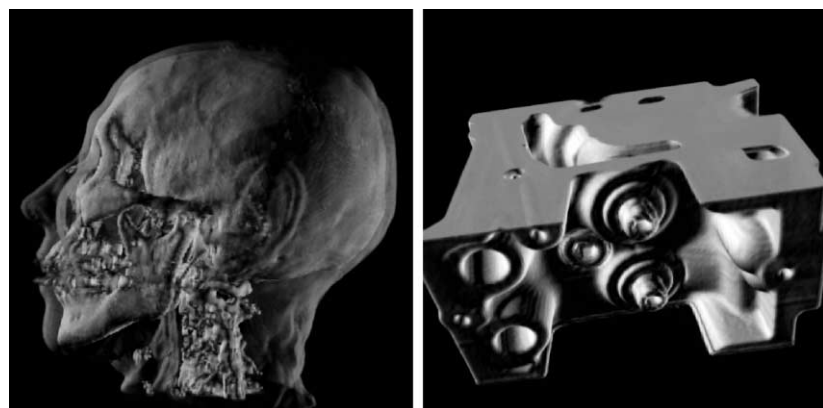
Ein wichtiges Thema ist die schnell wachsende Größe von Volumendaten im Zusammenhang mit der limitierten Speicherbandbreite der GPUs. Zudem werden Gradi-

enten-Vektoren zurzeit noch nicht in der GPU berechnet, was die Speicheranforderung der Volumendaten beträchtlich erhöhen könnte. Vor kurzem wurden einige interessante und vielversprechende Ansätze für dieses Problem vorgestellt, die unter anderem Multi-Resolution-Methoden, Volumenkompensation, und Image-based Volume-Rendering (IBVR) umfassen.

Da die hardwarebeschleunigten Techniken für reguläre Volumendaten recht ausgereift sind, fokussiert sich die Forschung auf die interaktive oder Echtzeit-Visualisierung von unstrukturierten und zeitabhängigen Volumendaten. Neue Visualisierungsverfahren verwenden auch nicht-photorealistische Methoden, wie z.B. die illustrative Volumenvisualisierung. Die nächsten zehn Jahre werden sicher wesentliche Fortschritte in diesen Gebieten hervorbringen.

## Danksagung

Der vorliegende Beitrag ist eine deutsche Kurzfassung des Kapitels „Hardware-Accelerated Volume Rendering“ in „Visualization Handbook“, herausgegeben von Chuck Hansen and Chris Johnson. Herzlichen Dank an Tim Weyrich, der bei der Übersetzung dieses Beitrags ins Deutsche geholfen hat, und den anonymen Rezensenten für die hilfreichen Kommentare.



**Bild 9** CT-Scan eines menschlichen Kopfes ( $256^2 \times 225$ ) und einer Maschine ( $256^2 \times 110$ ), berechnet mit hardware-beschleunigtem EWA-Splatting auf einer GeForce FX Ultra 5900. Die Bilder wurden von Wei Chen, Zhejiang University, China, und Liu Ren, Carnegie Mellon University, USA, zur Verfügung gestellt.

## Literatur

- [1] K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Proc. of the ACM SIGGRAPH/Eurographics Workshop on Graphics hardware*, pp. 9–16. ACM Press, 2001.
- [2] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics & Applications*, 8(5):29–37, May 1988.
- [3] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The volumepro real-time ray-casting system. In *Proc. of the 26th Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH 99)*, pp.251–260. ACM Press/Addison-Wesley Publishing Co., 1999.
- [4] L. Ren, H. Pfister, and M. Zwicker. Object-space ewa surface splatting: A hardware accelerated approach to high quality point rendering. In *Computer Graphics Forum*, volume 21, pp. 461–470, Sep. 2002. *Proc. of Eurographics 2002*.
- [5] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume rendering on standard pc graphics hardware using multi-textures and multi-stage rasterization. In *Eurographics/SIGGRAPH Workshop on Graphics Hardware*, pp. 109–118, Interlaken, Switzerland, August 2000.
- [6] S. Roettger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser. Smart hardware-accelerated volume rendering. In *Eurographics/IEEE TCVG Symp. on Visualization 2003*. Eurographics, 2003.
- [7] D. Weiskopf, K. Engel, and T. Ertl. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Trans. on Visualization and Computer Graphics*, 9(3):298–313, July/Sep. 2003.
- [8] Y. Wu, V. Bhatia, H. C. Lauer, and L. Seiler. Shear-image order ray casting volume rendering. In *Symp. on Interactive 3D Graphics*, pp. 152–162, Monterey, CA, June 2003.
- [9] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross. Ewa splatting. *IEEE Trans. on Visualization and Computer Graphics*, 8(3):223–238, 2002.



**Dipl.-Ing. Ph.D. Hanspeter Pfister** ist seit 1996 Vize-Direktor und Senior Research Scientist bei Mitsubishi Electric Research Laboratories (MERL) in Cambridge, USA. Dr. Pfister studierte Elektrotechnik an der ETH Zürich und machte 1991 seinen Abschluß als Diplomingenieur. Er hat sein Ph.D. in Informatik an der State University of New York in Stony Brook, USA, abgeschlossen. Seine Forschungsschwerpunkte betreffen die wissenschaftliche Visualisierung, die Computer Grafik, sowie die moderne Bildverarbeitung. Adresse: Mitsubishi Electric Research Laboratories, Broadway 201, Cambridge, MA, U.S.A., E-Mail: pfister@merl.com