Image-Based Rendering using State of the Art Graphics Hardware

und

Image-Based Editing

Moulaye Ouldelmehdi, Wolfgang Wirth

tum.3D
computer graphics & visualization

Überblick

- Einführung
- Light Field Rendering
- The Lumigraph
- Implementierung Hardwarebeschleunigtes IBR
  - Triangulierung
  - Rekonstruktion
  - Hardwarebeschleunigung
- Implementierung Image-Based Editing

Einführung

- Alternative zur geometrie-basierten Methoden
- Nutzt 2D Daten und Zusatzinformationen
- Ermöglicht das Rendern natürlicher Szenen ohne Modellinformationen
- Ermöglicht die Erzeugung von Ansichten aus neuen Blickwinkeln

Light Field Rendering

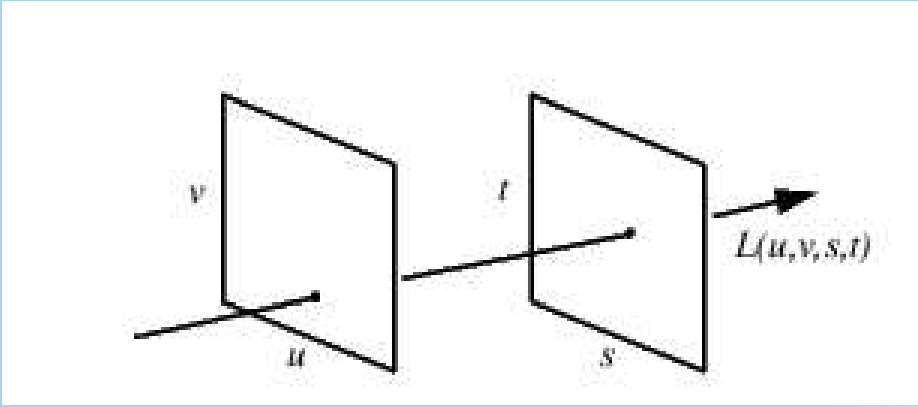
- Entwickelt von Marc Levoy und Pat Hanrahan
- Geeignet für natürliche und gerenderte Szenen
- Benötigt Daten über die Kamerapositionen

Light Field Rendering

- Farbwerte als Funktion von Position und Richtung
- Abgeleitet von der plenoptischen Funktion
- Reduktion von 5D auf 4D, da Farbwerte der Strahlen in transparentem Medium unverändert bleiben

Light Field Rendering

- Wenn ein Lichtstrahl zwei parallele Ebenen schneidet kann er durch die Schnittpunkte eindeutig definiert werden



Light Field Rendering

- Erzeugung geschieht durch Einfügen von 2D Schichten in 4D Datensatz
- Erzeugung einer neuen Ansicht durch Entnahme einer 2D Schicht
- Benötigt eine hohe Abtastrate der Szene um Löcher oder Artefakte zu vermeiden

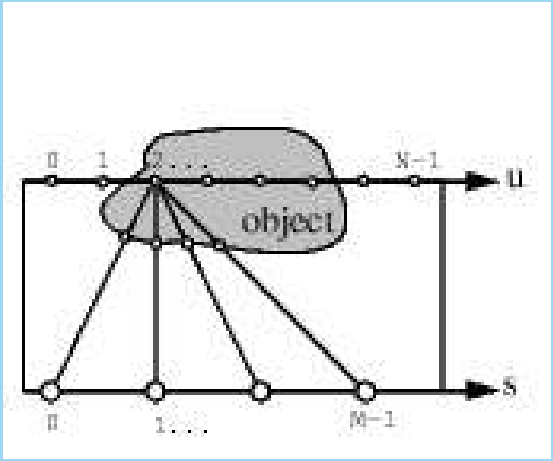
  => Überabtastung

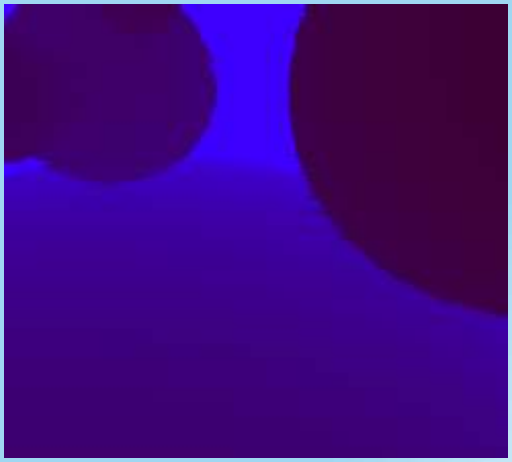
The Lumigraph

- Entwickelt von u.a. Gortler et al., SIGGRAPH 1996
- Entstand zur gleichen Zeit
- Reduktion von 5D auf 4D der plenoptischen Funktion aufgrund der gleichen Annahme wie beim Light Field Rendering
- Verfahren ähnelt dem Lightfield

The Lumigraph

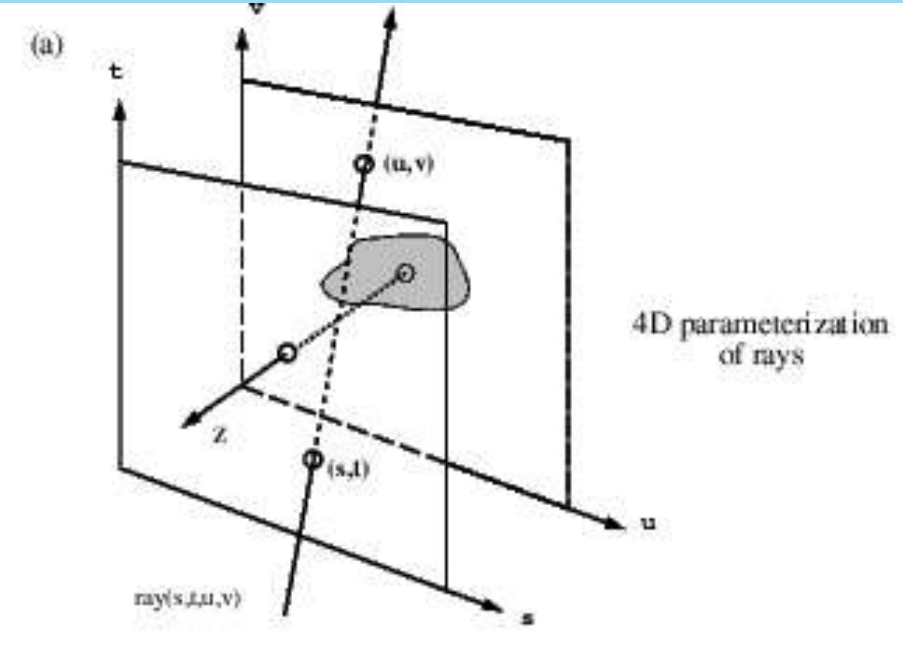
- Verwendet die Tiefe der Szene um bei geringerer Datenmenge eine bessere Darstellungsqualität zu ermöglichen





The Lumigraph

- Two-plane Parametrisierung

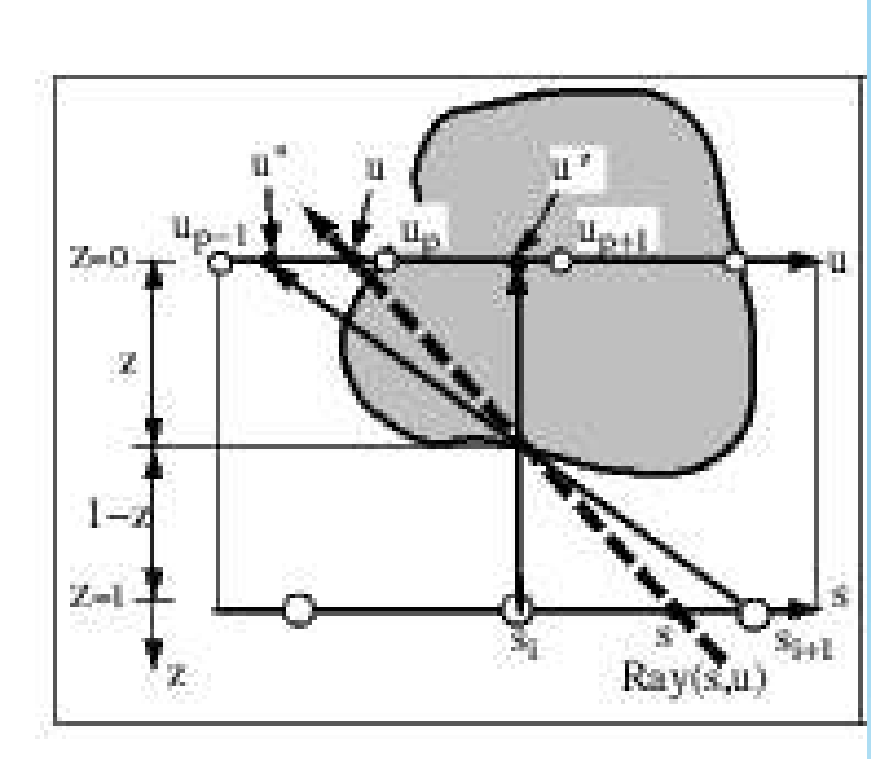


The Lumigraph

- Erzeugung einer neuen Ansicht
  - Ermitteln der Schnittpunkte des Lichtstrahles mit der s,t- und u,v-Ebene
  - Auslesen oder Berechnen der Tiefe in der ein Objekt vom Strahl getroffen wird
  - Tiefenkorrektur zur Auswahl des Farbwertes
  - Anzeige

The Lumigraph

- Tiefenkorrektur:



Implementierung - Triangulierung

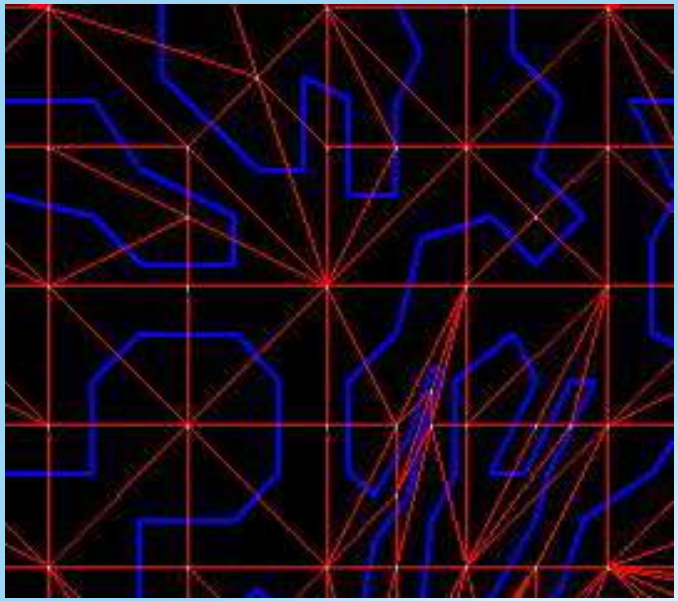
- Triangulierung ist sinnvoll, um Löcher bei der Interpolation der Farbwerte zu vermeiden
- Anforderungen:
  - Adaptiv, abhängig von Gradienten
  - Möglichst wenig Vertices
  - Geringer Rechenaufwand

Implementierung - Triangulierung

- Zerlegung in einen Triangle Strip
- Triangulierung muss einen Hamiltonpfad enthalten
- Hierarchische Triangulierung mit adaptiver Verfeinerung
- Unterteilung eines Dreiecks entspricht dem Einfügen einer Kette in die bereits existierende Kette

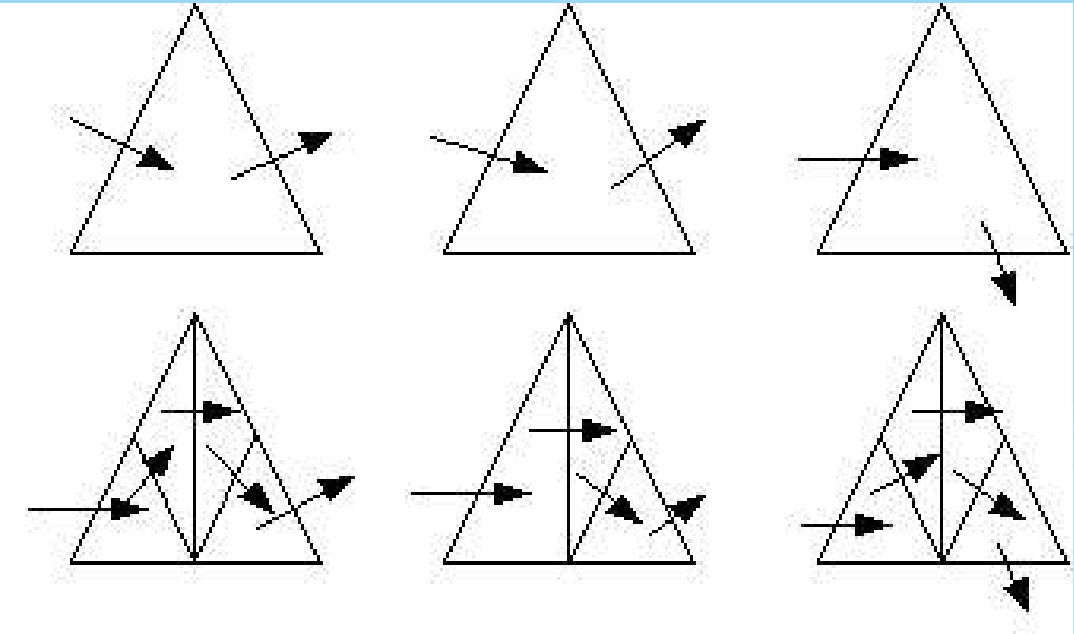
Implementierung - Triangulierung

- Unterteilung erfolgt nach 3 Kriterien
  - Länge der Kante
  - Tiefenunterschied
  - Von Außen erzwungen



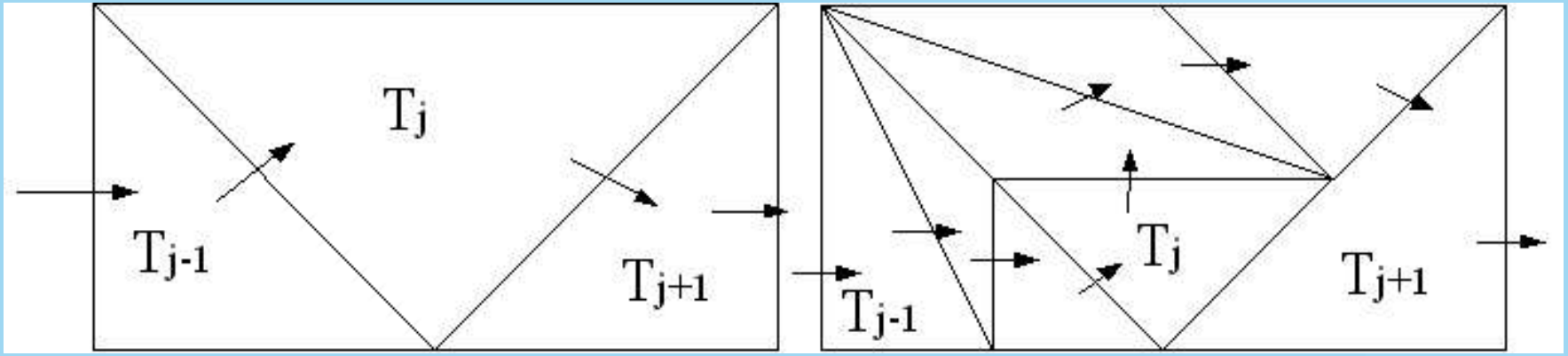
Implementierung - Triangulierung

- Je nach Situation werden Musterunterteilungen eingefügt, so daß der Hamiltonpfad erhalten bleibt

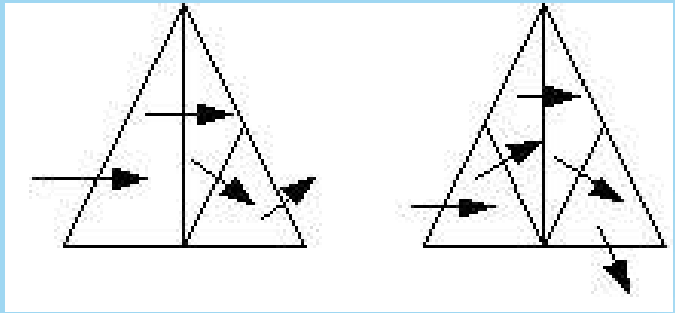


Implementierung - Triangulierung

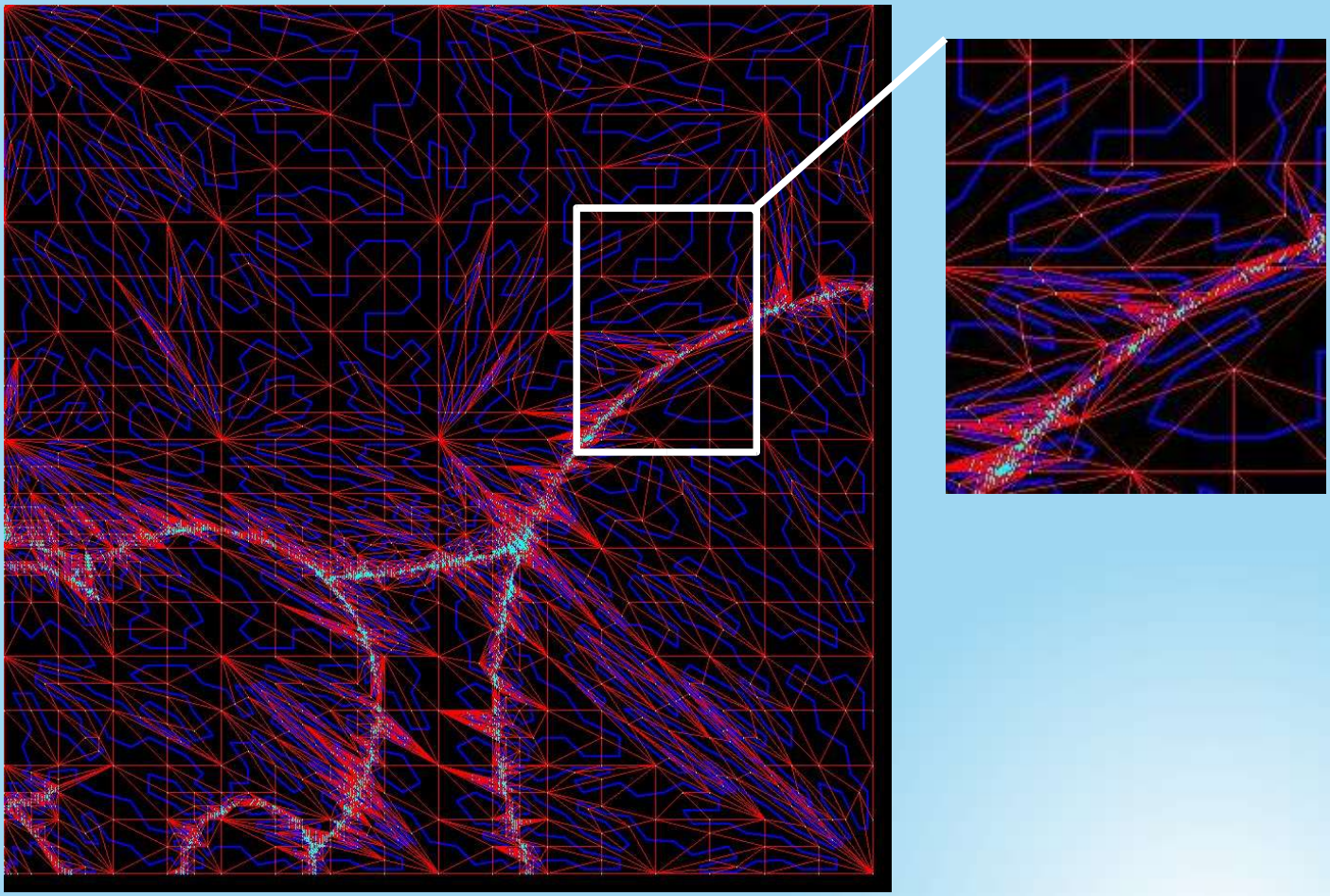
Beispiel:



Verwendete Muster:

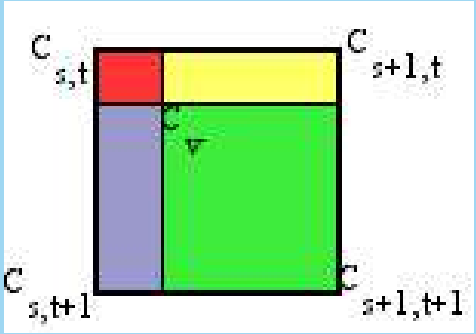


Implementierung - Triangulierung



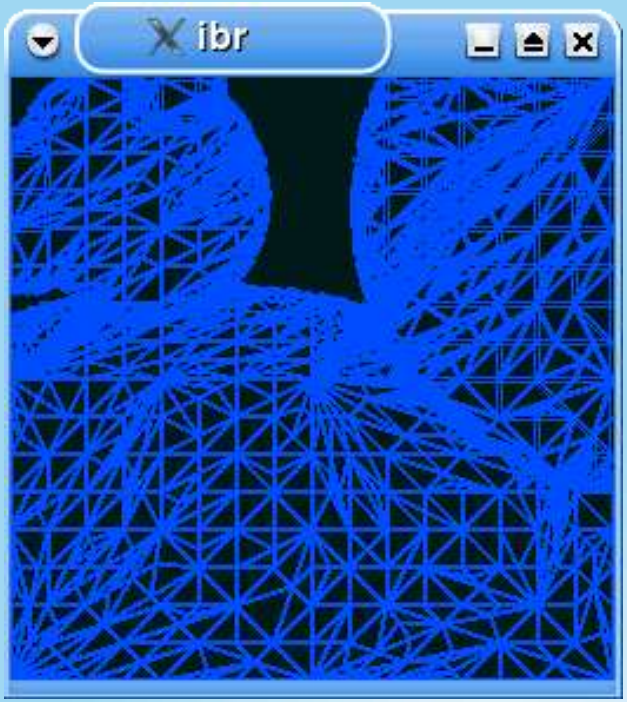
Implementierung - Rekonstruktion

- Bestimmung der virtuellen Kamera C_v
- Auswahl der besten Kameras für aktuellen Ausschnitt anhand des Abstands zu C_v
- Kopieren der Farbinformationen in den Texturspeicher



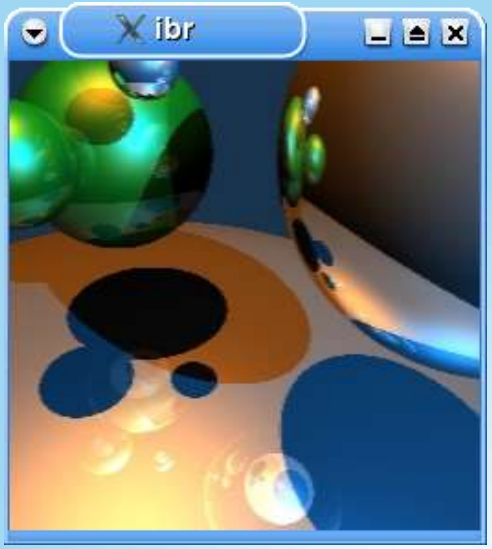
Implementierung - Rekonstruktion

- Zeichnen der 4 Triangulierungen in den Depthbuffer
- Durch den Depth Test von OpenGL ergeben sich die Tiefeninformationen der Szene
- Der Depthbuffer wird in eine Textur gerendert und steht damit für den nächsten Schritt zur Verfügung



Implementierung - Rekonstruktion

- Für alle Fragmente erfolgt eine Tiefenkorrektur bei der die Texturkoordinaten berechnet werden
- Die Farbwerte aus den Texturen werden mit dem Abstand der Kameras zur
- virtuellen Kamera gewichtet und angezeigt



Implementierung - GPU

- Rechenleistung der CPU und Busbandbreite beschränken die Bildwiederholrate
- GPU ist ideal für Verarbeitung paralleler Daten

=> Die Ermittlung der Tiefenwerte, die Tiefenkorrektur und die Kombination der Farbwerte werden an die GPU ausgelagert

=> Die Triangulierung wird auf der CPU durchgeführt

Implementierung - GPU

- Tiefenwerte müssen in ihrer Position korrigiert werden um zur virtuellen Kamera zu passen
- Erfolgt im Vertexshader
- Mehrere geometrische Operationen
- Ergebnis wird in Textur gerendert

Implementierung - GPU

- Für alle Fragmente wird zuerst der zugehörige Tiefenwert ausgelesen
- Durch die Tiefenkorrektur ergeben sich die Texturkoordinaten für die 4 Farbtexturen
- Diese werden anhand der jeweiligen Gewichtung kombiniert und angezeigt

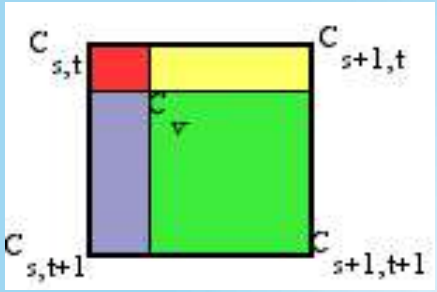


Image-Based Editing

Übersicht:

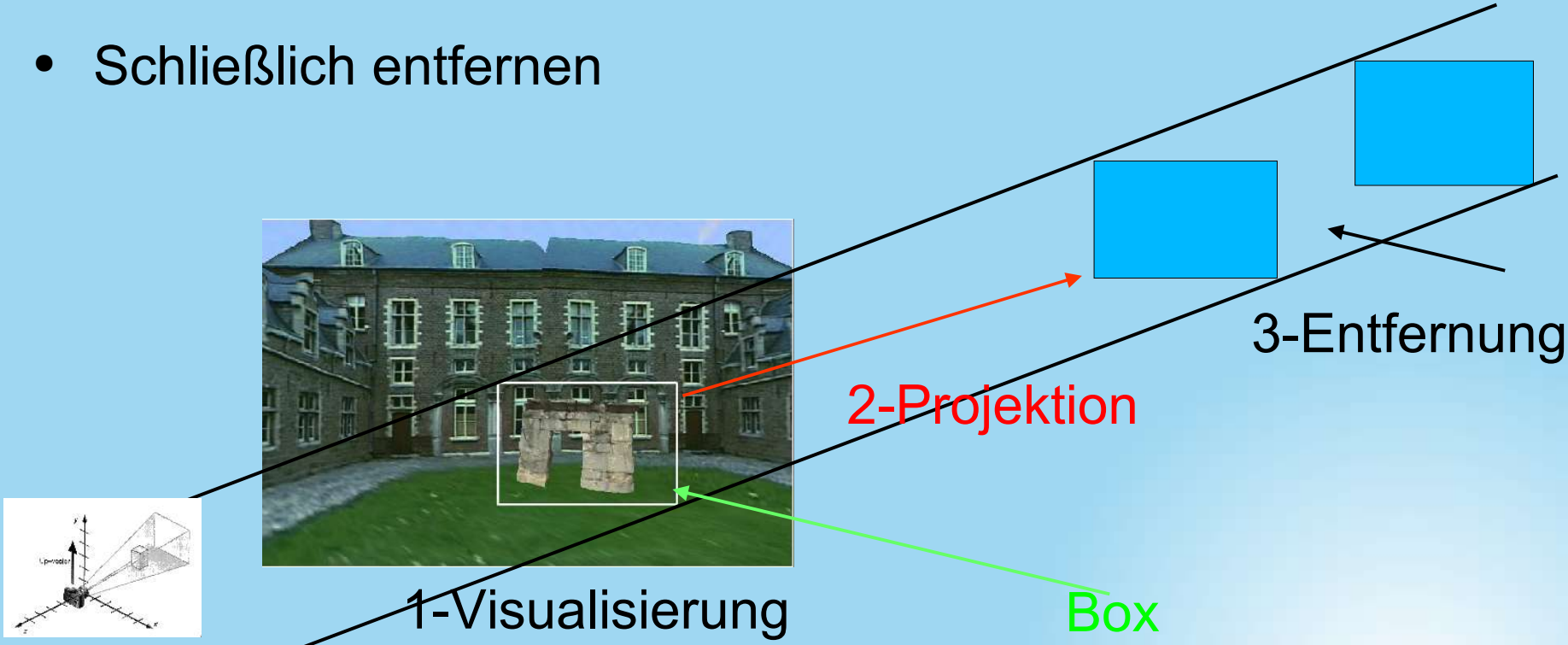
- die Bedeutung der Image-Based-Editing
- Ziel der Image-Based-Editing
- Wie wird das Image-Based-Edititing realisiert
- Prinzipien des Algorithmus
- Das eigentliche Verfahren
- Effiziente Image-Based Editing Verfahren

Bedeutung & Ziele des IB Editing

- Das Bild in eine Menge von homogen Regionen zu zerlegen.
- Objekte von einem Szene zu entfernen.
- Achtung: Eine universelle Strategie mit einem Algorithmus einwandfrei zu identifizieren, gibt es nicht.

Wie wird das IB-Edititing realisiert

- Das Objekt selektieren
- Projektion machen
- Schließlich entfernen



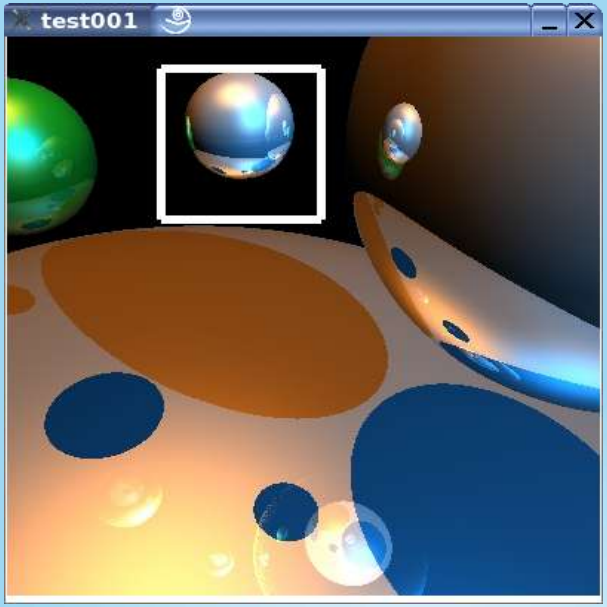
Prinzipien des Algorithmus

- Objekte können ohne Wissen über ihre geometrische Form entfernt werden
- Interaktives Verfahren in dem der Benutzer eine Rolle spielt
- Der Benutzer kann das Objekt markieren
- Anhand die Tiefenwerte kann man das Objekt lokalisieren und danach entfernen

Implementiertes Verfahren

Markierung des Objektes:

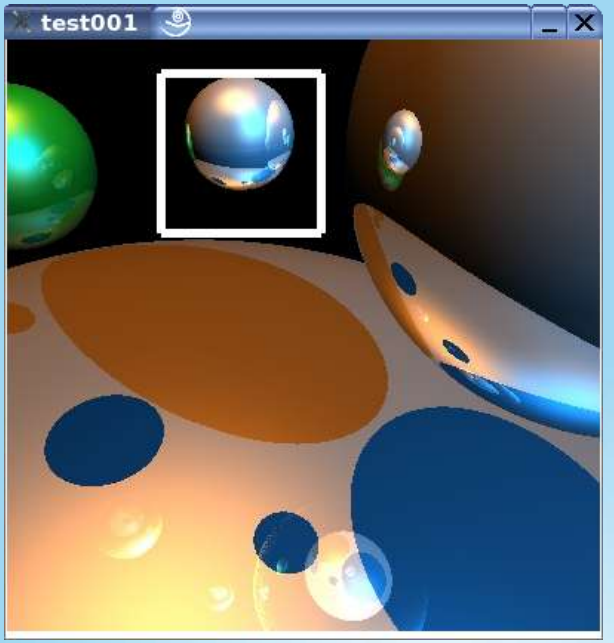
- Der Benutzer zieht eine Box mit der Maus, in dem das gewünschte zu entfernende Objekt liegt
- Danach klickt er mit der Maus zweimal auf das Objekt
- Die Koordinaten dieses (x,y,z) Punktes werden gespeichert



Implementiertes Verfahren

Markierung des Objektes:

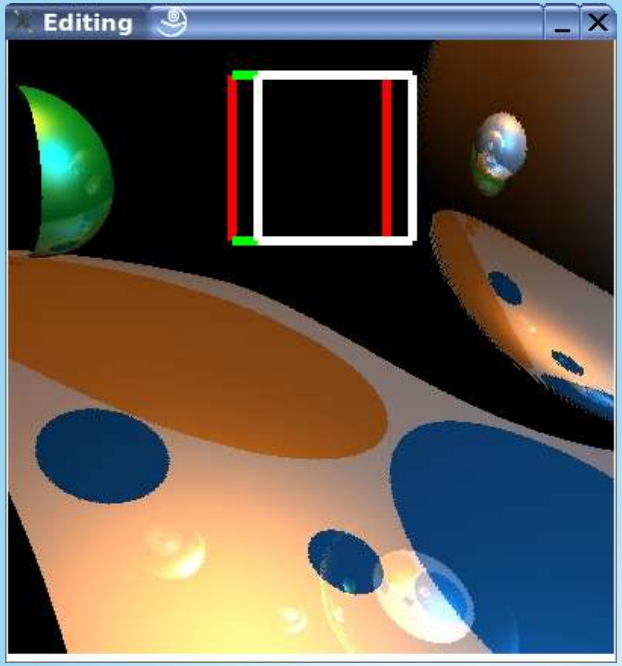
- Die Boxkoordinaten (Boxmin, Boxmax) werden in 3D-Koordinaten transformiert und die Box aufs neue gezeichnet



Implementiertes Verfahren

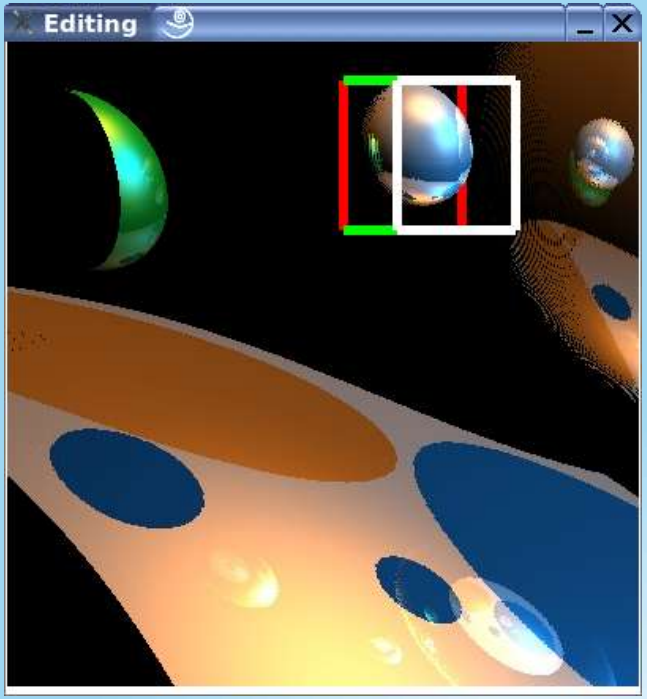
Entfernung des Objektes:

- Anhand der Tiefe dieses Punktes wird die Box im 3D-Koordinatensystem gezeichnet und gerendert
- Alle Pixel die nicht im Bereich der Box liegen werden normal gerendet



Implementiertes Verfahren

- Rendern von Objekten, die hinter dem entfernten Objekt liegen
- Mittels anderen Kameras versucht man hier die hinteren Objekte zu rendern



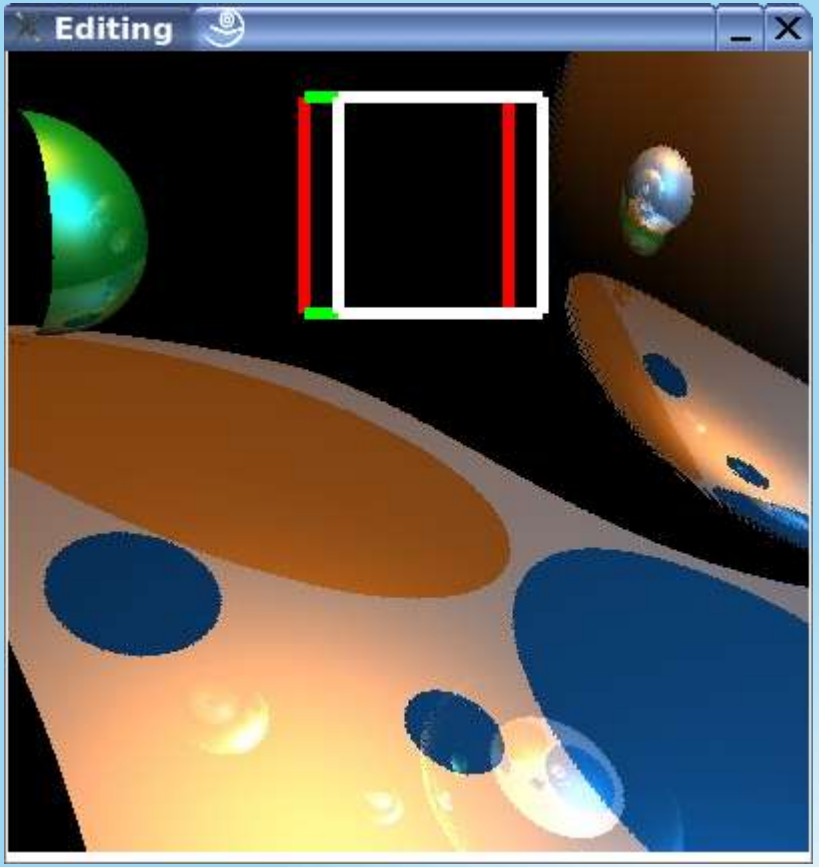
Vorteile & Nachteile des Verfahrens

Vorteile:

- einfach

Nachteile:

- Spiegelungen sind problematisch



Effiziente Image-Based Editing Verfahren

- Je unabhängiger dabei ein Algorithmus von wissensbasierter Vorinformation ist, desto breiter ist sein Einsatzbereich und somit auch sein allgemeiner Nutzen

=> Automatische Segmentierungsverfahren.

Zusammenfassung

- Nutzung von OpenGL oder DirectX ideal für Problemstellung
- Aktuelle Grafikhardware erlaubt dank Shaderprogrammierung die Auslagerung der meisten Berechnungen auf die GPU
- Dadurch werden hohe Bildwiederholraten und damit interaktives Betrachten einer Szene möglich

Vielen Dank für die Aufmerksamkeit