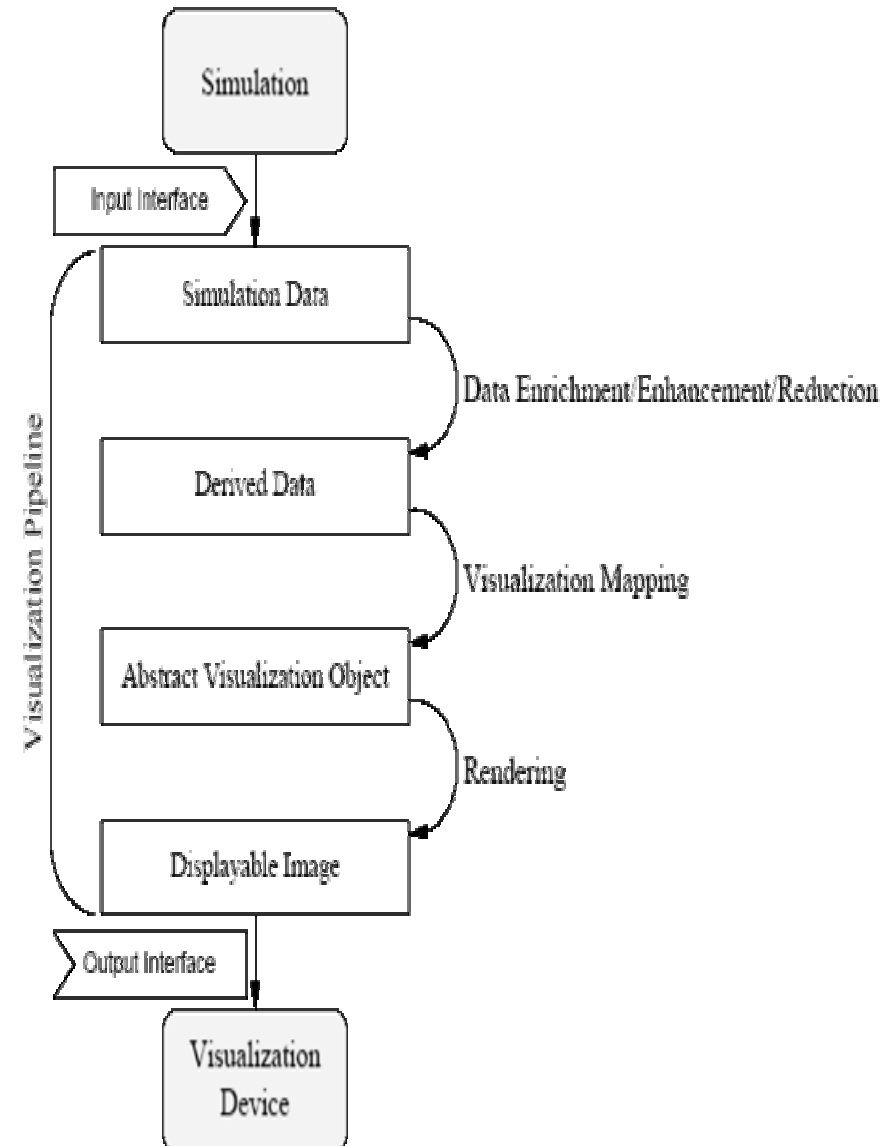


VTK-Filter für effizientes
netzwerkbasiertes Rendern inkl.
Interaktionsmöglichkeit

Warum?

- Netzwerkbasiertes Rendering muß die Renderpipeline am effizientesten Punkt teilen
- An den unterschiedlichen Übergabepunkten können die zu transportierenden Datenmengen sehr groß sein
- Wo ist der richtige Punkt?
 - für optimale Leistung
 - bestmögliche Interaktion
 - vorhandene Bandbreite





VTK – Visualization ToolKit

- Weit verbreitetes Toolkit
- Gut strukturiert
- Frei verfügbar
- Objektorientierte Struktur
- Visualisierungs-Pipeline wird vom Programmierer definiert
- Rendering und Interaktion wird transparent
- Bietet seit Version 4.0 Kommunikationsklassen
 - Transparente Teilung der Pipeline an beliebigen Punkten inkl. Interaktion
 - >erweiterung des Konzepts/Toolkits für GVK



Bloodflow Visualization zeigt

- Daten kommen in Serie von Vektorwürfeln
 - Visualisierungsabbildung auf Glyphen bringt eine Vervielfachung der Daten
 - Die pro Frame zu übertragende Datenmenge explodiert
 - GSS-Verschlüsselung erledigt den Rest
 - Die Applikation wird grid-basiert langsamer
 - > es muss ein besserer Teilungspunkt gefunden werden



Neue Lösung

- Trennung der Pipeline nach dem Rendern
- Direkt vor der Darstellung->Übertragung von Bildern
- Zu transportierende Datenmenge
 - proportional zur Displaygröße
 - unabhängig von den Visualisierungsdaten
- Aber:
 - Unkomprimert sehr große Datenmengen
- Lösung: VideoStreaming!!



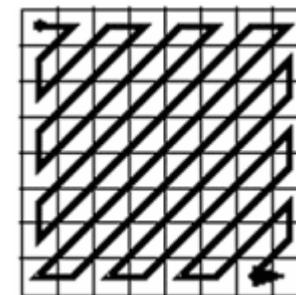
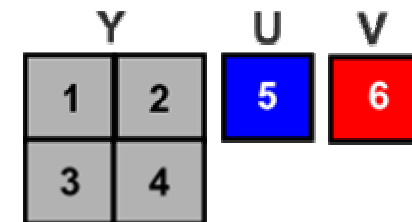
XVid-Codec

- OpenSource ISO-MPEG-4 kompatible Video Codec Implementierung
- Hohe Bildqualität bei hoher Auflösung
- Und geringen Bitrate (ca. 0.5-1Mbit)
- Spielfilm in DVD-Qualität auf CD-Medium
- Verwendet I,P und B-Frames
- Effizienter Datentransport für GVK



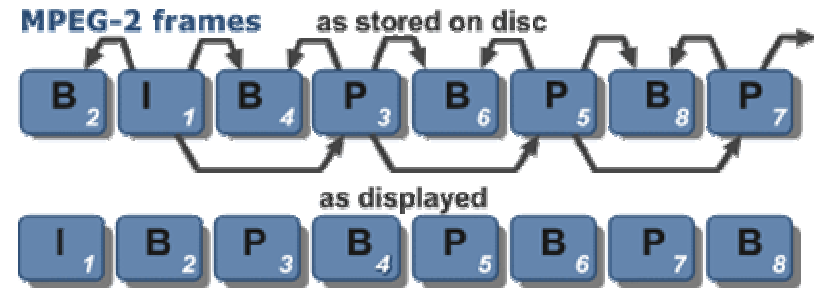
MPEG-Kompression

- Group of Pictures (GOP)
- Macroblocks
- 4:2:2 YUV downsampling
- DCT + quantization + coding
- MotionCompensation
- Different Frametypes





- **I-Frames** (intra coded frames)
 - Nur DCT kodierte Vollbilder
- **P-Frames** (predicted frames)
 - Kodiert Unterschied zum letzten I oder P Frame
 - Motion Prediction und DCT
 - Der nächste Frame wird „predicted“ und die Differenz zum wirklichen Frame gespeichert
 - Effizient hängt von der „Motion“ ab
- **B-Frames** (bidirectional frames)
 - „Predicten“ in beide Richtungen
 - Liefern bessere Kompressionswerte
 - Erfordern kompliziertere Speicherung/Verarbeitung

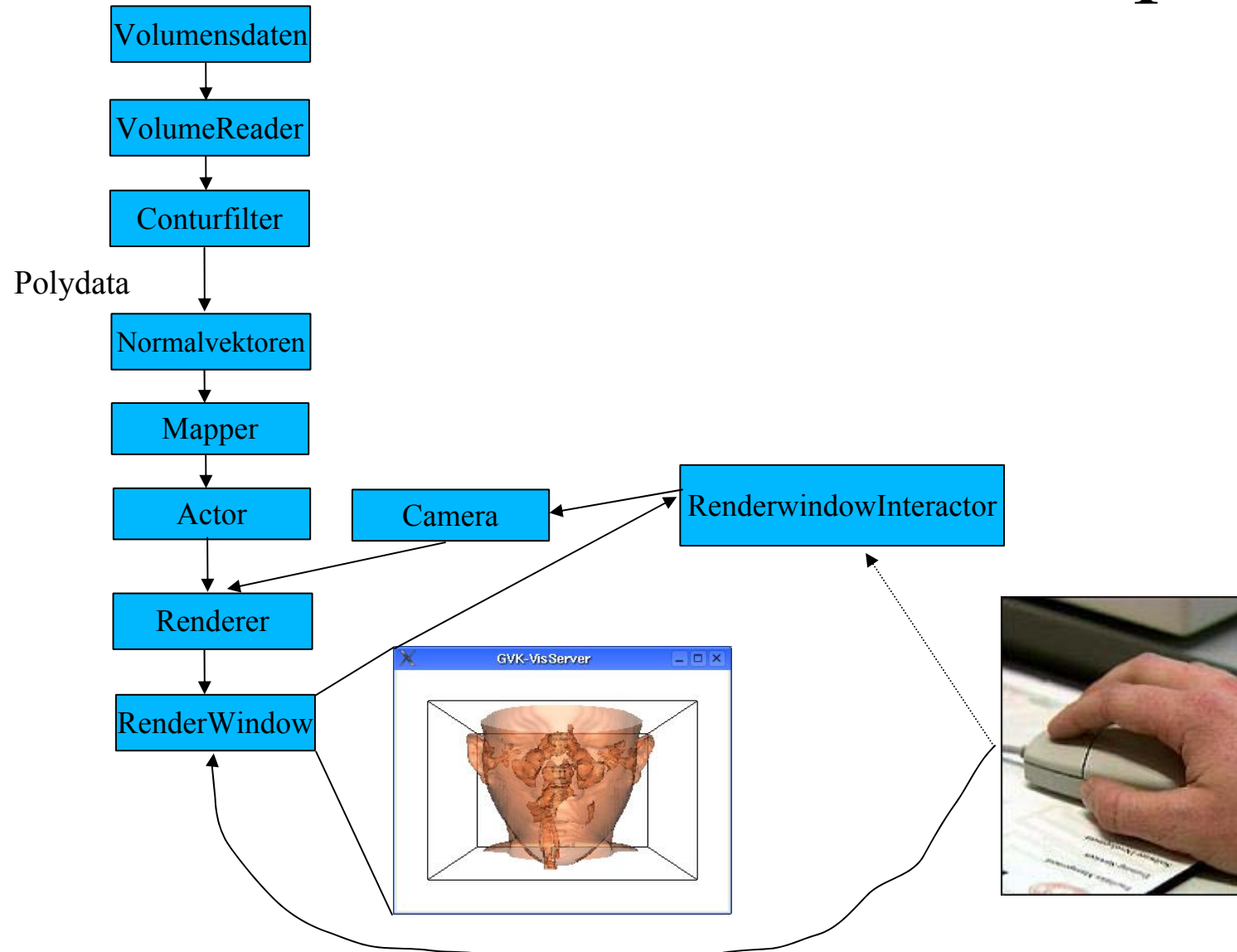


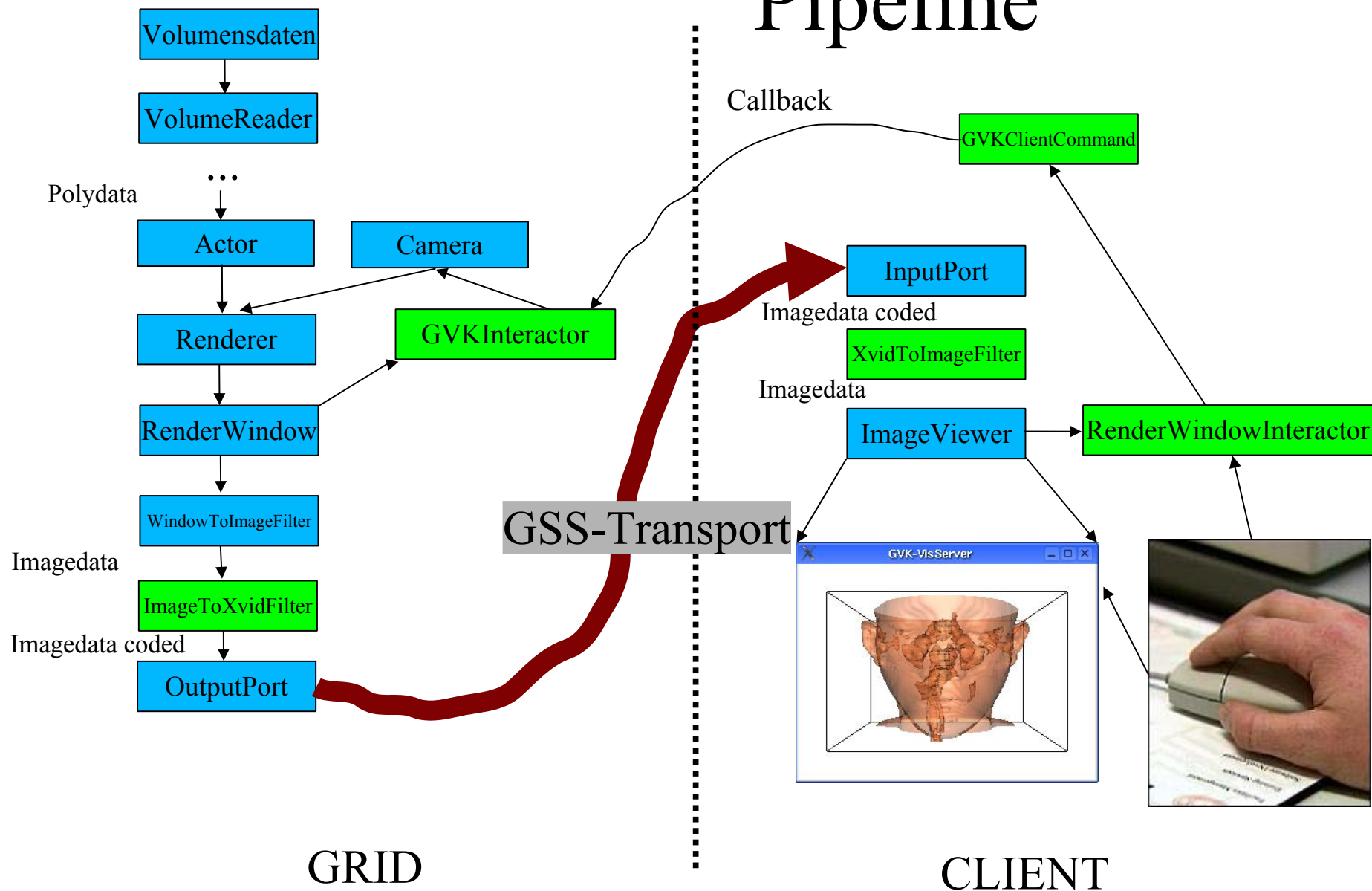


Anpassung für VTK

- Entwicklung spezieller Wrapper-Klassen für VTK
 - Anpassungen an den Pipeline-Betrieb von VTK
 - Anpassungen für dynamische Bildgrößen
 - „Transport“ der Interaktion

Die klassische Pipeline







TODO

- Dynamische Anpassung
 - der Bitrate
 - der Quantisierung
 - Stichwort „Quality of Service“
 - Stabilisierung
- -> Diplomarbeit Köckerbauer