



Nexus (SFB 627)

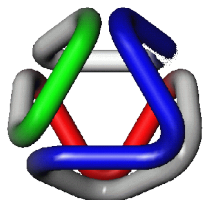
Umgebungsmodelle für mobile kontextbezogene Systeme

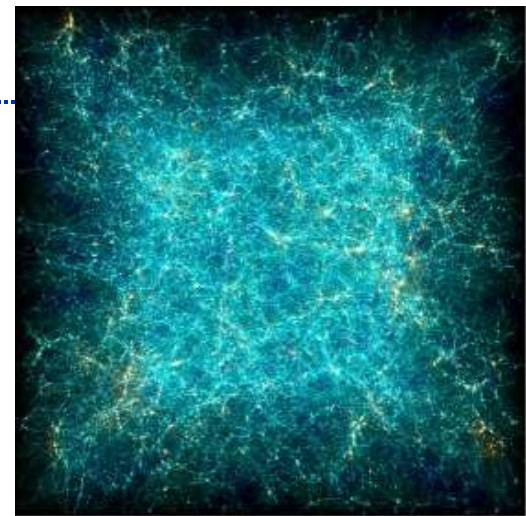
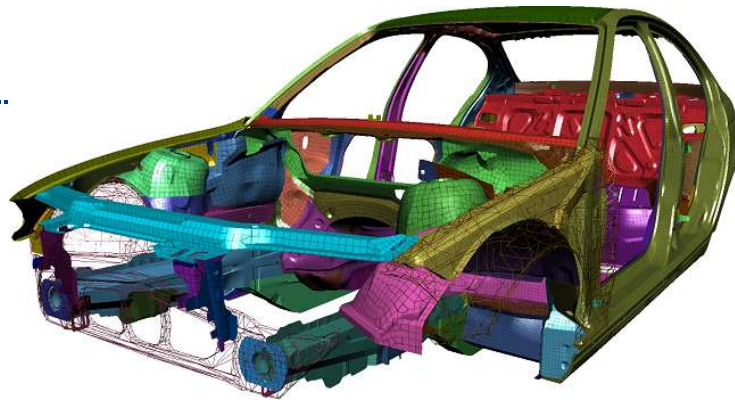
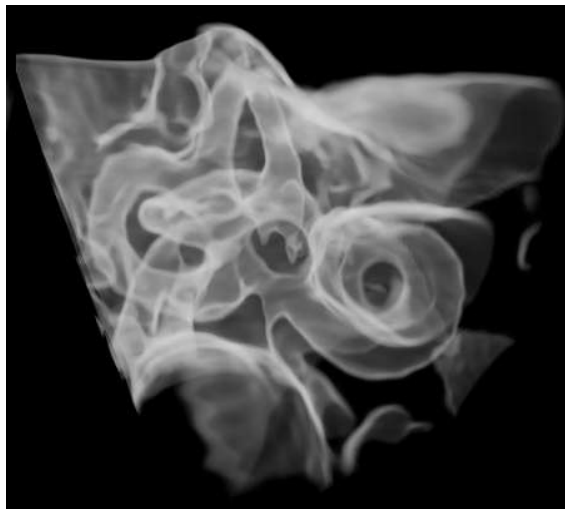
Teilprojekt C5

Abstraktion des Umgebungsmodells für AR-Anwendungen

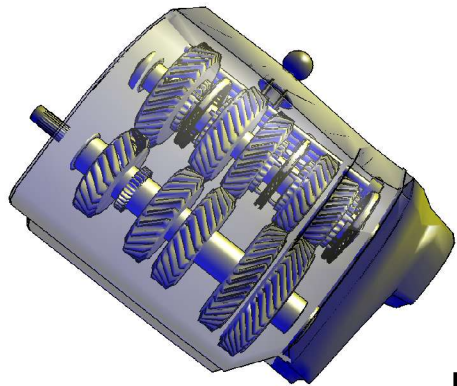
Mike Eissele, Daniel Weiskopf, Thomas Ertl

Institut für Visualisierung und Interaktive Systeme (VIS)



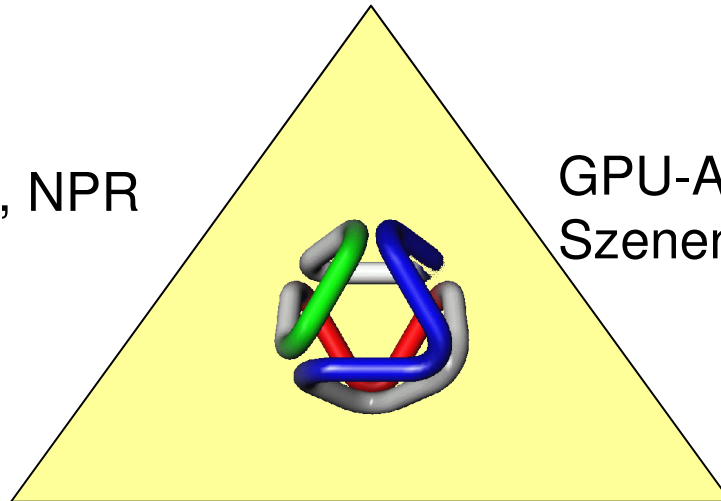


Visualisierung

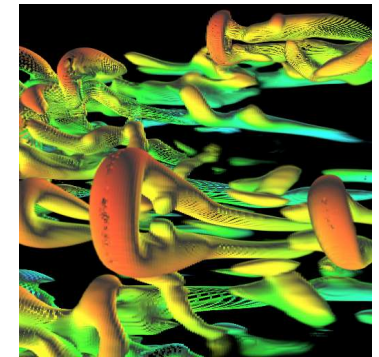


Perzeption, NPR

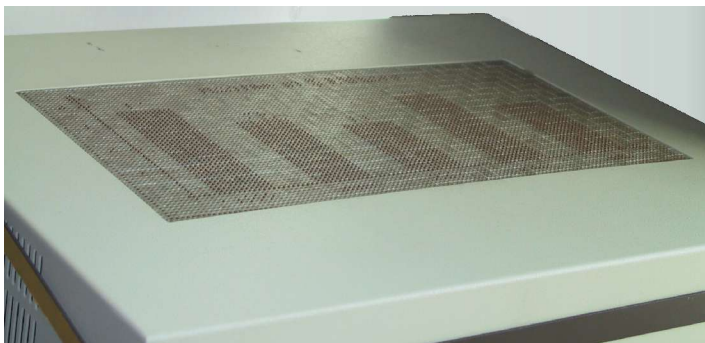
**Mensch-
Maschine-
Schnittstelle**



GPU-Algorithmen
Szenengraphen



**Interaktive
3D-Graphik**



Web-Graphik
Mobile Graphik



3D-Graphik auf PDAs noch schwierig:

- Geringe Hardware Unterstützung

- Wenige Software APIs

- Beschränkte Bandbreite

Zwei Ansätze:

- Beschränkung auf unterstützte Funktionalität

- z.B. Liniengraphik

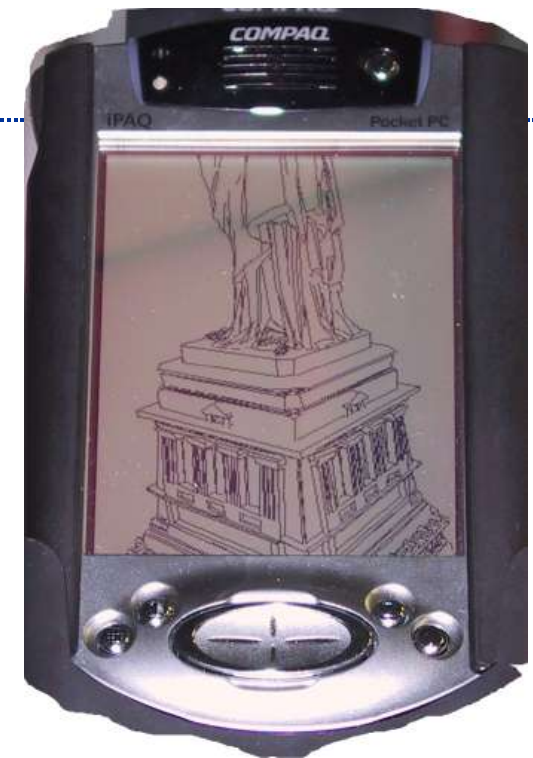
- Image Streaming:

- 3D Rendering am Graphik-Server

Probleme

- Modell-Abstraktion

- Kompression



Daten im Umgebungsmodell mit Raumbezug
3D-Graphik auf mobilen Geräten

Abstrakte Daten mit indirektem Raumbezug
Visualisierung auf kleinen Geräten

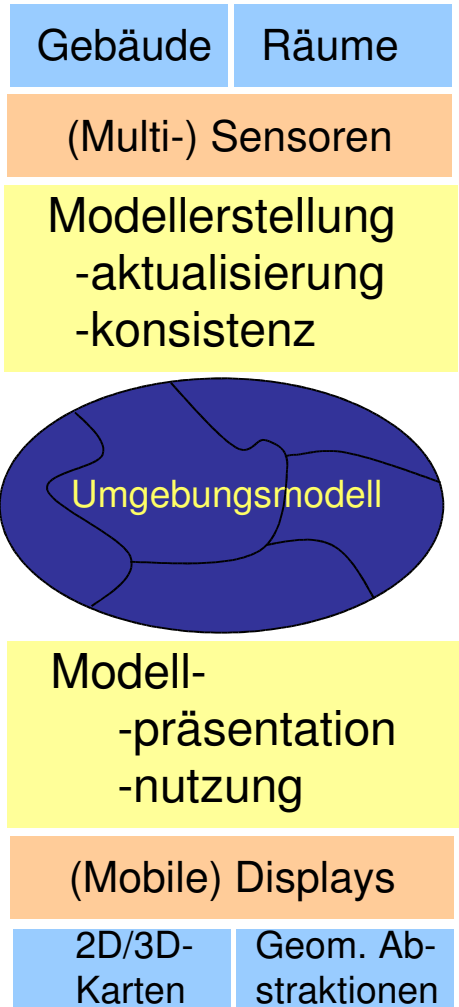
Herstellung des realen Raumbezugs
Augmented Reality

Direkte (3D-)Interaktion (ortsbasiert)
Sensoren (Position, Orientierung)

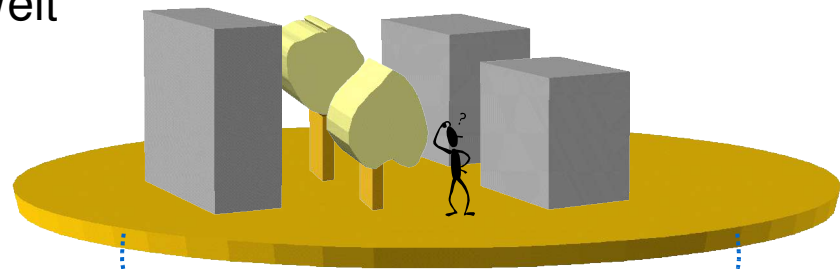
2 Projekte

C5: Abstraktion für Augmented Reality

D2: Navigationsunterstützung für Blinde

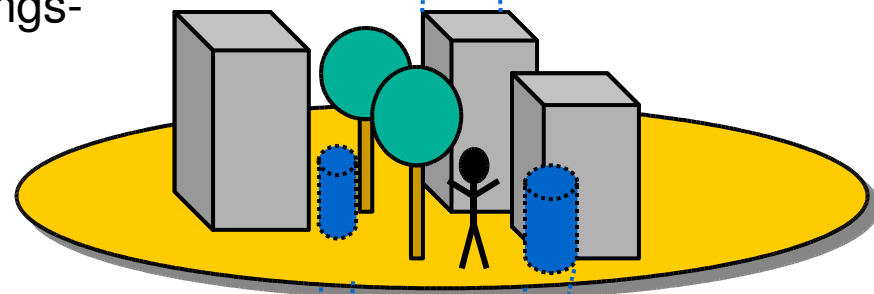


Reale Welt

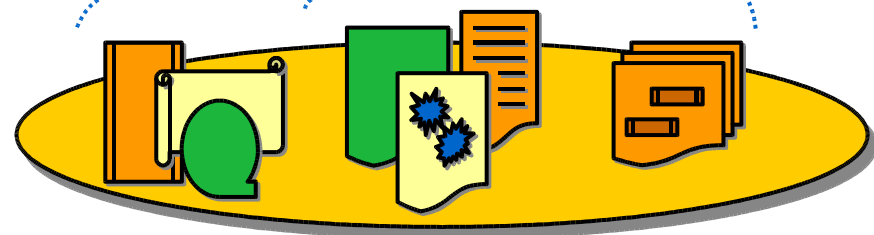


Sensordaten

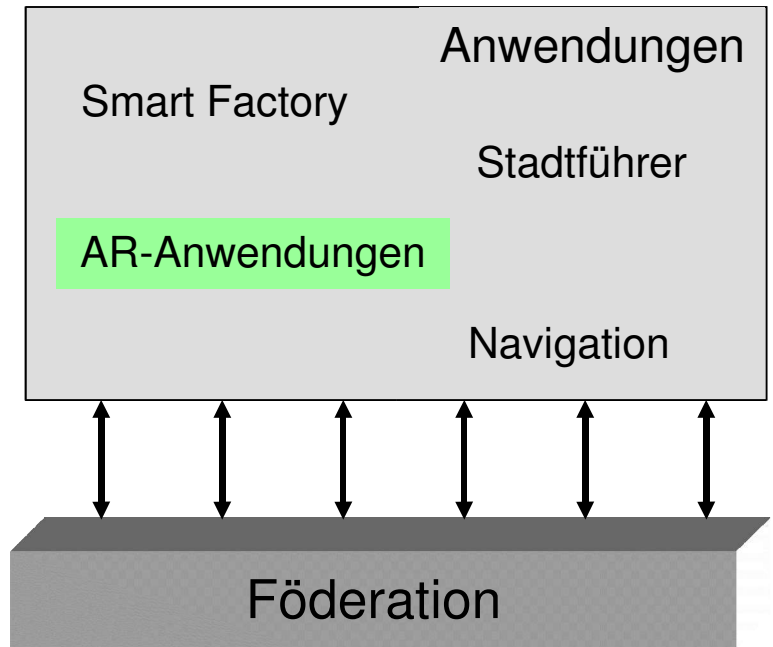
Umgebungsmodell



Informationsräume



Multimedia WWW Digitale Bibliotheken



Entwicklung eines Basissystems zur Visualisierung der im Umgebungsmodell vorliegenden Daten

Darstellung durch AR-Techniken

Abbilden von abstrakten, zeitabhängigen Objekteigenschaften auf Geometrien

Generisches System zur automatischen Auswahl geeigneter Mapping-Verfahren

Abstraktion von komplexen Geometrien zur Echtzeitdarstellung

Navigationsunterstützung

Szenario: Smart Factory

Intelligente Fabrikhalle (Teilprojekt D1)

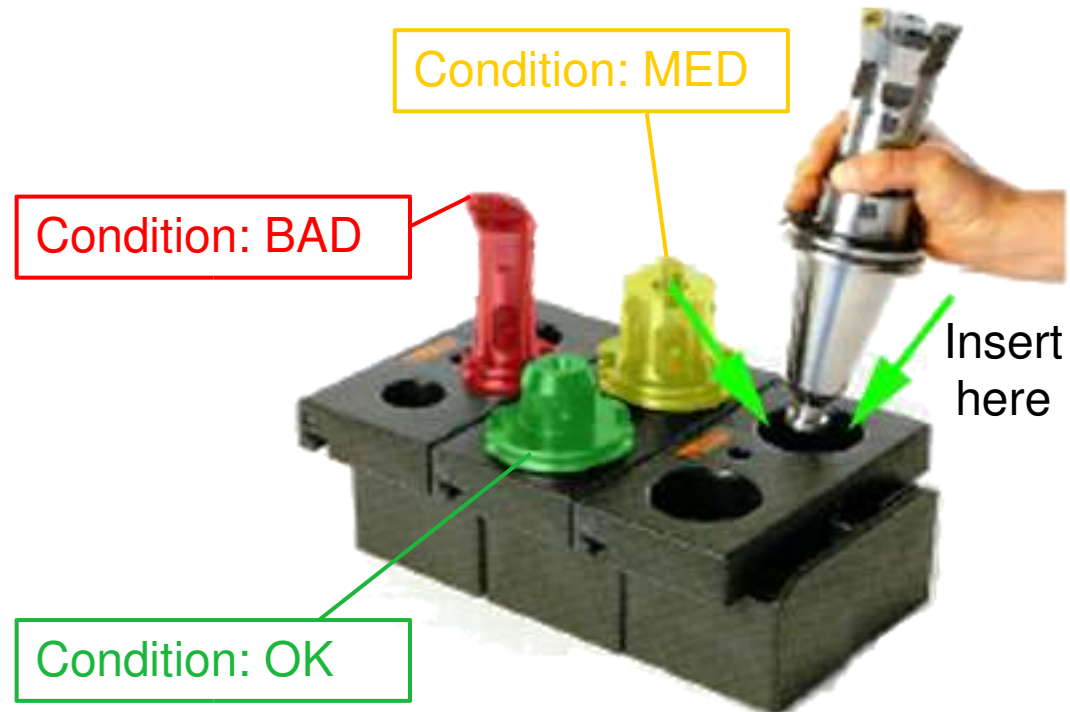
Erfasste Werkzeug/Maschinenzustände darstellen

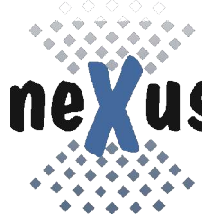
Navigation

Montageanleitungen

Fabrikplanung

Gefahrensituationen



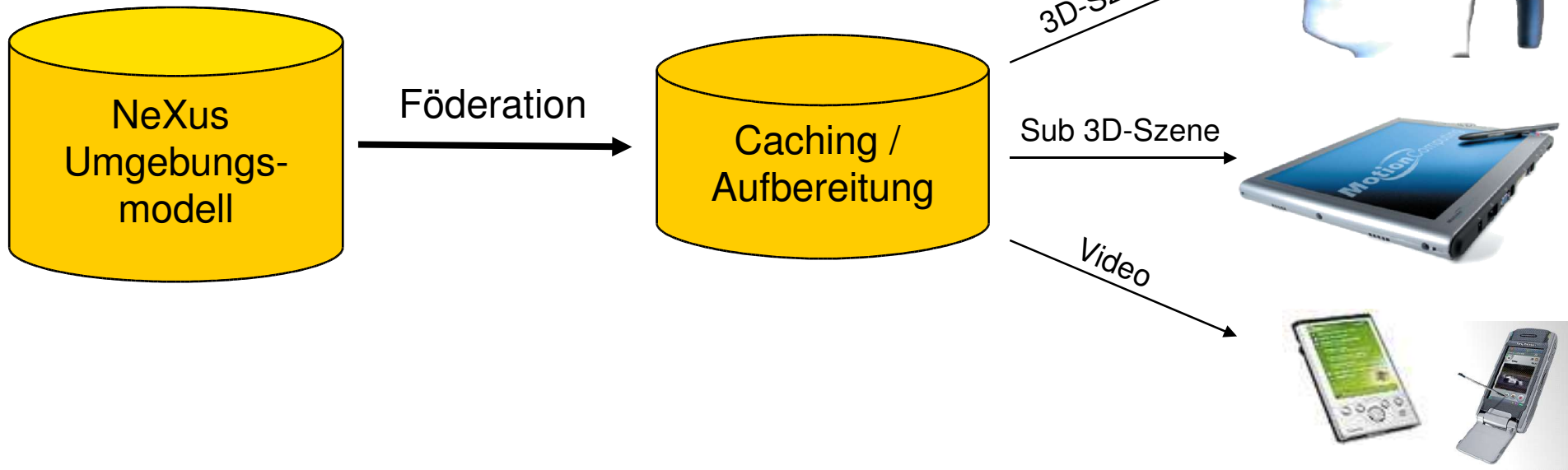


Client / Server Architektur

Aufbereitung der Daten anhängig von:

Client Hardware

Aktuelle Szenenkomplexität



Hält die Vereinigungsmenge aller von den Clients benötigten Teile (Daten) des Umgebungsmodells

Hält pro Client eine Liste der bereits übertragenen Umgebungsmodellteile (nicht für Video Übertragung)

Caching-Strategien um nicht mehr/selten benötigte Teile des Modells aus dem Cache zu entfernen

Extensive Nutzung der Nexus Plattform:

- Spatial Model Server, Location Server, ...

- Registriert Events beim Nexus Event Service (Cache Update, Client Events, ...)

Visualisierungstechnik der Daten ist abhängig von

Daten Typ

Skalar, Vektor, Text, Geometrie, ...

Client/Benutzer Einstellungen

Balkendiagramm bevorzugt, mittlere Größe, ...

Leistungsfähigkeit der Client Hardware

Modelle simplifizieren, Ersatzrepräsentationen zeichnen,...

Caching-Server kennt alle Entscheidungskriterien
sendet entsprechende Visualisierung an den Client

Laptop mit Head-Mounted Display:

Leistungsstark, kann großen Teil des Modells verwalten

Hochwertige Darstellungstechniken können im Client erzeugt werden

Bei gleichzeitiger Darstellung mehrerer Datenquellen teilweise Vorberechnung der Darstellung im Caching-Server

SmartPhone

Geringe Leistung, nur minimale Verarbeitung im Endgerät (Video Streaming)

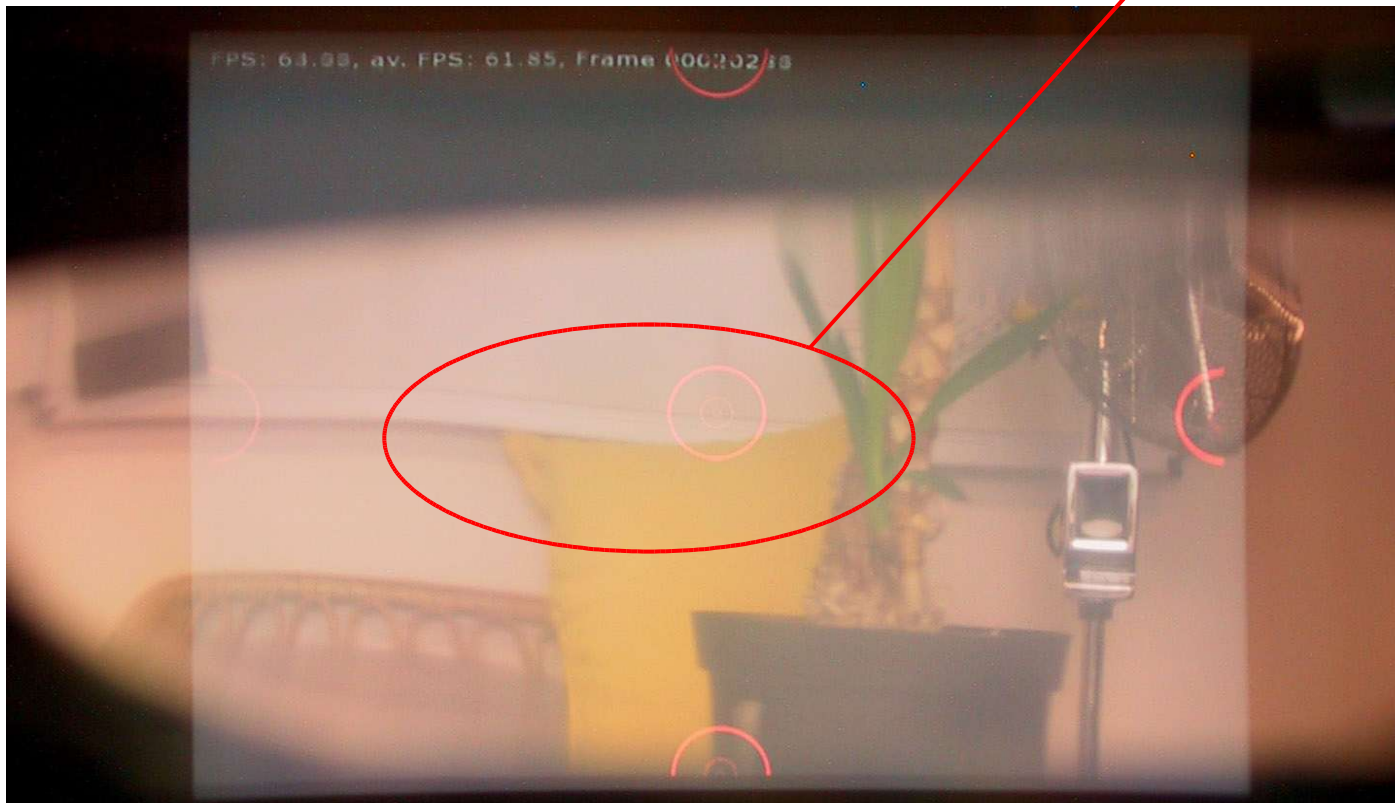
Kleines Display unterschiedliche Darstellungstechniken

Cybermind HI-RES 800 HMD (entspr. Sony Glastron)

Optical See-Trough Option

LCD 800x600 Punkte, 16.8 M Farben

Qualitätsprobleme
(Verzerrungen,
Öffnungswinkel)



Integration des Orientierungssensors zur Bestimmung der Blickrichtung

Intersense Inertia Cube²

3 Achsen 360°, 150 Hz update Rate

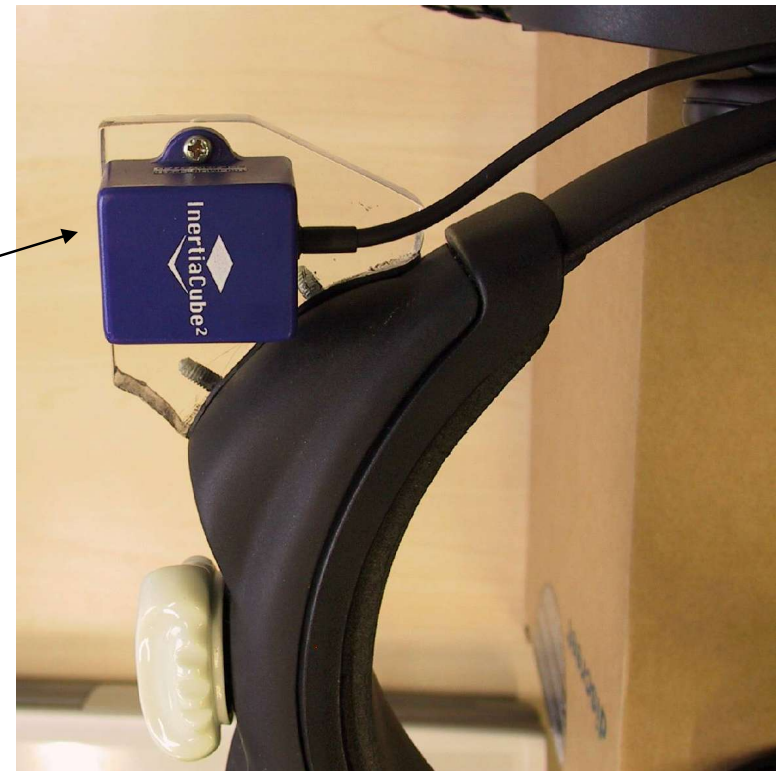
3 Sensoren pro Achse:

Linear- und Rotationsbeschl.

Magnetfeld



Inertia Cube²



Microvision Nomad ND1000

Optical See-Trough

Retina Display 800x600,
monochrom rot (32 Abstufungen)

Ansteuerung über VGA-Eingang

Laptop Pentium M1700

54 Mbit WLAN

GeForce FX 5650GO

Entsprechender Laptop Rucksack

Zur Ansteuerung der Head-Mounted Displays



Motion Computing M1300 Tablet PC

11 MBit WLAN, Intel Extreme Graphics (1024x768)

Inertia Cube² Orientierungssensor

Erweiterung durch
USB-Kamera



Div. PDAs (Toshiba e800, Compaq iPAQ, ...)

Anschaffung einer CF-Card Kamera für PDAs (für AR)

Anschaffung eines SmartPhones

ARM Prozessor mit Symbian OS

Integrierte Kamera und Bluetooth, ev. WLAN



Positionierungssystem (voraussichtlich Ubisense)

Begrenzt auf einen Raum

RF basiert, arbeitet im Ultra-Wide Band

Bei herkömmlicher Abdeckung ca. 15 cm genau

Auch bei hoher Abdeckung max. 8-5 cm Genauigkeit



Optisches Trackingsystem

4 ART track1 IR-Kameras

Bis zu 1mm Genauigkeit,
Position & Orientierung

Auf kleinen Raum beschränkt (ca.5m²)

Durch optisches System Verdeckungsprobleme
nicht für alle Applikationen einsetzbar

Benötigt „große“ Marker

Nicht geeignet für
Smart Factory Szenario

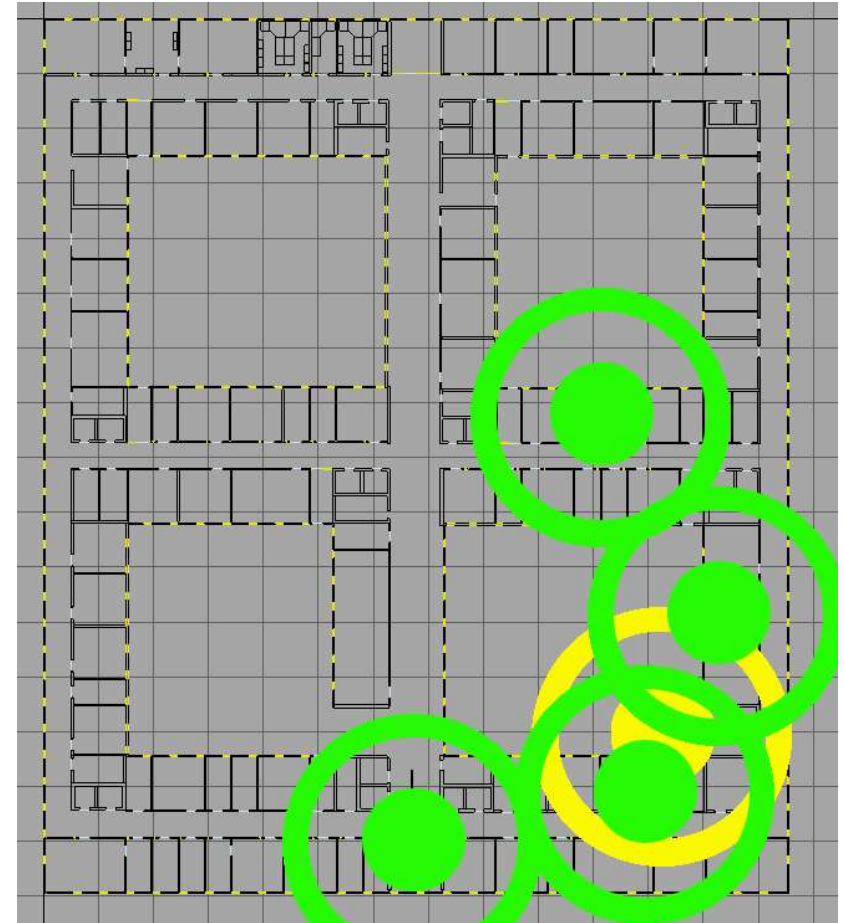


Endgerät misst die empfangene Signalstärke

Positionserver ermittelt Position durch Vergleich der aktuellen Signalstärke mit Referenzwerten in der Datenbank

Benötigt Referenzwerte der Signalstärken des Zielgebiets

Abdeckung durch mindestens 3 Access Points erforderlich



Messwerte unterscheiden sich je Gerätetyp (iPAQ, e800, Laptop, ...)

Gebäudemodell um „ungültige“ Positionen und Fortbewegungen auszuschließen (Sprung „durch“ die Wand)

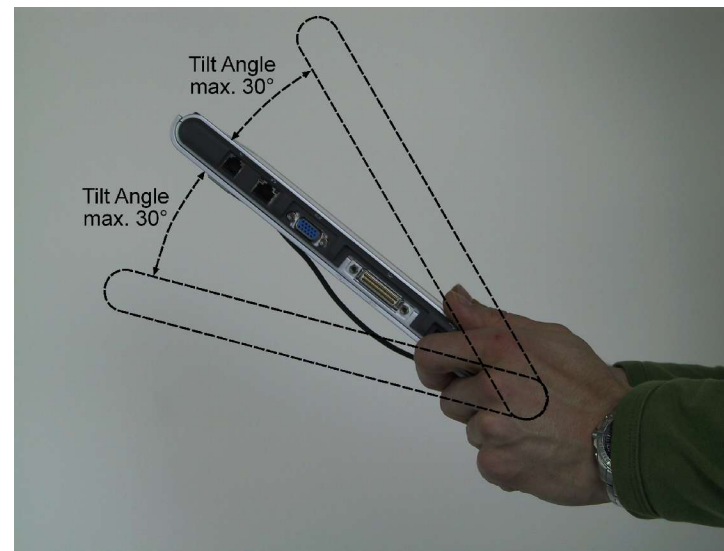
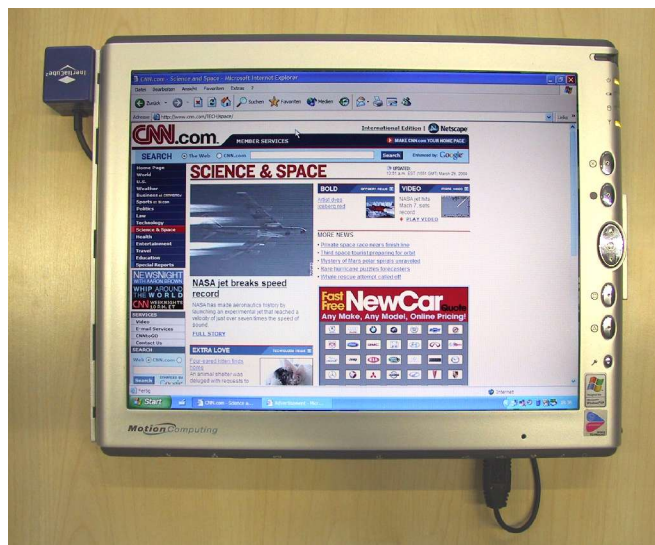
Genauigkeit bei 4 APs ca. 5m bei 3 APs ca. 6-7m

Gebäude Navigation

Modell mit topologischen Informationen erweitert

Einfache Routenplanung

TabletPC mit Orientierungssensor ausgerüstet
 Orientierung dient als weiteres Eingabeinterface
 Generische Integration für Windows und Linux
 Einsetzbar in beliebigen Applikationen (Acrobat, Browser, E-Book Reader, Spiele, AR, ...)
 Orientierung wird auf bestimmte Interaktionen abgebildet



Direkte Abbildung der TabletPC Orientierung für Spiele
(Marble Madness, Kugel-Labyrinth, ...)

Objekte können durch Änderung der Orientierung aus
unterschiedlichen Richtungen betrachtet werden
(Objekt Explorer)



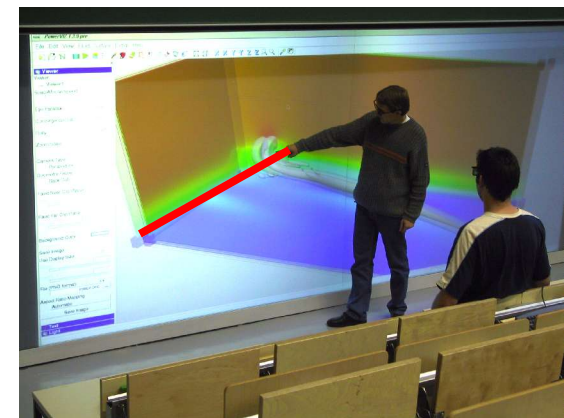
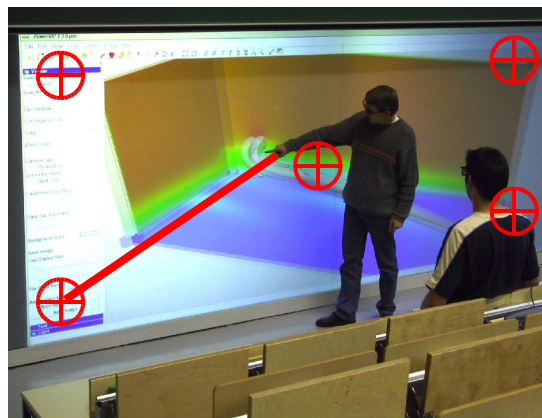
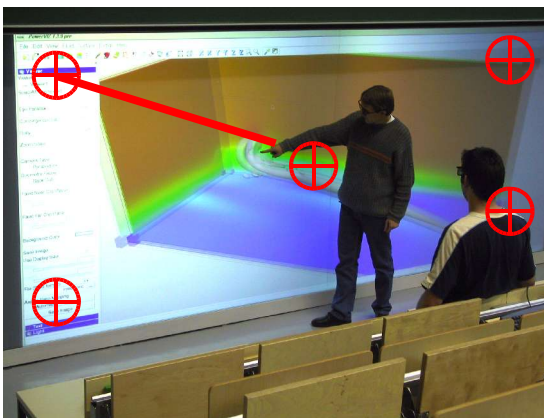
3D VR/AR-Zeigegerät ohne Positionierungssystem

Position kann sehr schnell und einfach kalibriert werden

Orientierungssensor-Daten werden aufgezeichnet bei Ausrichtung auf vordefinierte Ziele (ca. 5-8)

Berechnung der Position durch mehrfache Triangulation und anschließender Fehlerminimierung

Annahme: Benutzer bewegt sich nicht



Auffinden von zur Eingabe ähnlichen 3D Modellen

Suchergebnisse werden über einen 3D Viewer in einer Web-Oberfläche präsentiert

Modell der Suchanfrage wird simplifiziert und durch Subdivision geglättet

Resultierendes Modell dient als Grundlage zur Merkmalsberechnung

Das Suchergebnis wird durch vergleichen der Merkmale berechnet

Aufbau einer Datenbank mit 3D Modellen

Simplifizierung der Modelle, Berechnung und
Speicherung der Vergleichsmerkmale pro Modell

Simplifizierung zunächst durch bekannte Verfahren

M. Garland and P. Heckbert: “*Surface Simplification Using Quadric E...*” [SIGG97]

Vergleichen der Objekte durch einfach Verfahren (z.B. Vertex-Distanz-H.)

M. Novotni, R. Klein: “*A Geometric Approach to 3D Object Comparison*” [ECDL01]

R. Osada et al.: “*Matching 3D Models with Shape Distributions*”. [SMI02]

Einsatz von OpenMesh

Library zum Verwalten von Polygonnetzen

Positionierungssysteme für viele AR-Anwendungen zur ungenau

Position und Orientierung optisch nachschärfen

Grobe Orientierung durch Sensor

Features im Kamerabild mit im Umgebungsmodell gespeicherten Features abgleichen

Orientierung und Position entsprechend korrigieren

Basierend auf Arbeiten zu Marker-Less Tracking

D. Beirer et al.: „*Marker-less Vision Based Tracking for Mobile AR*“ [ISMAR03]

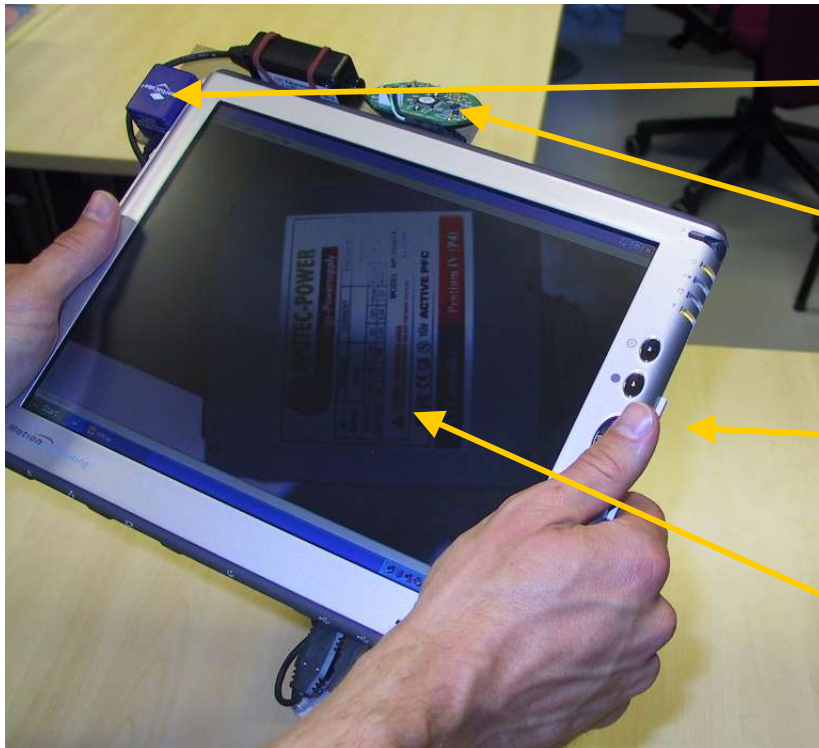
T. Okuma et al.: „*Fiducial-less 3-D Object Tracking in AR Systems*“ [ISMAR03]

Y. Yokokohji et al.: „*Accurate Image Overlay on Video See-Through...*“ [VR00]

Prototyp:

AR-Explorer: TabletPC mit Orientierungssensor und Kamera um Objekt zu erkunden

Position relativ zum Testobjekt



Inertia Cube²

Orientierungssensor

USB Kamera

Zu untersuchendes Objekt
(hinter TabletPC)

Kamerabild mit AR-
Überlagerungen

Visualisierung „aller“ Daten des Umgebungsmodells durch den Einsatz von AR-Techniken

Generisches, automatisches und adaptives System zur Auswahl und Einstellung der Visualisierungsverfahren

Entwurf und Aufbau eines Basissystems für Augmented Reality Anwendungen in NeXus

Weitere Infos:

www.nexus.uni-stuttgart.de (Teilprojekt C5)

Mailto: mike.eissele@vis.uni-stuttgart.de