

Einführung in die Erweiterte Realität

- 8. System Architectures for AR -

Gudrun Klinker

Dec 9, 2003

Augmented reality

Immersivity

- tracker sensor input -
- near real-time performance -
- many communication types -

Context-aware computing

Adaptivity

- sensor input -
- reconfiguration -
- context refinement -
- automatic adaptation -
- user-directed adaptation -

Mobile context-aware augmented reality

Distribution

- combination of local and remote devices -
 - ad hoc system -
 - alternate devices -
 - interdependency-
- integration of unknown components -
- new applications -

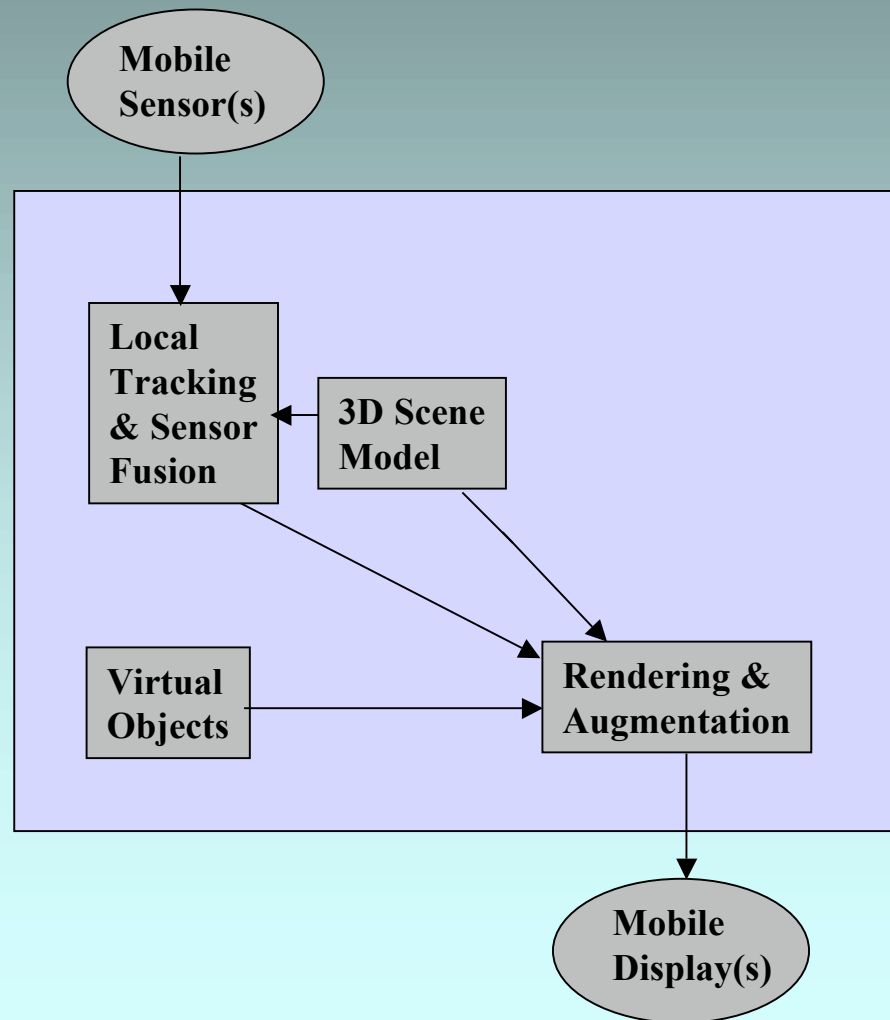
Ubiquitous computing

Overview

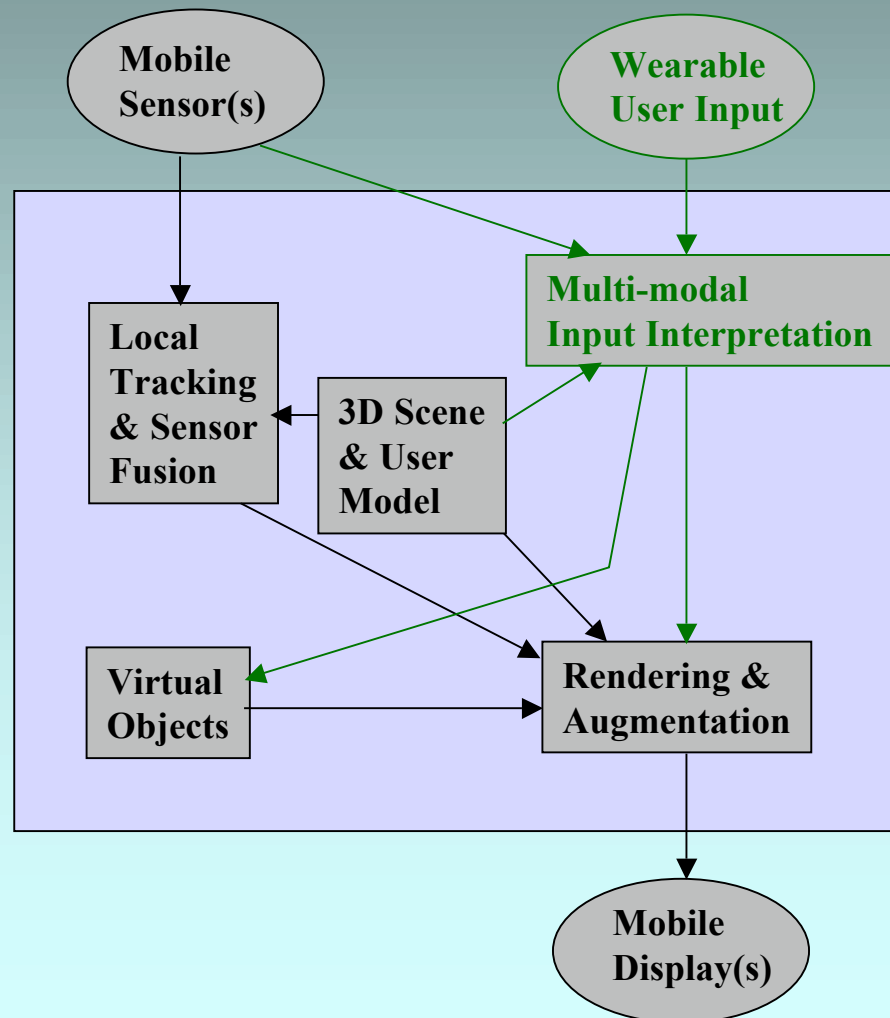
- Architecture for a Mobile AR System
- Architecture for Distributed Tracking
- DWARF

Architecture for a Mobile AR System

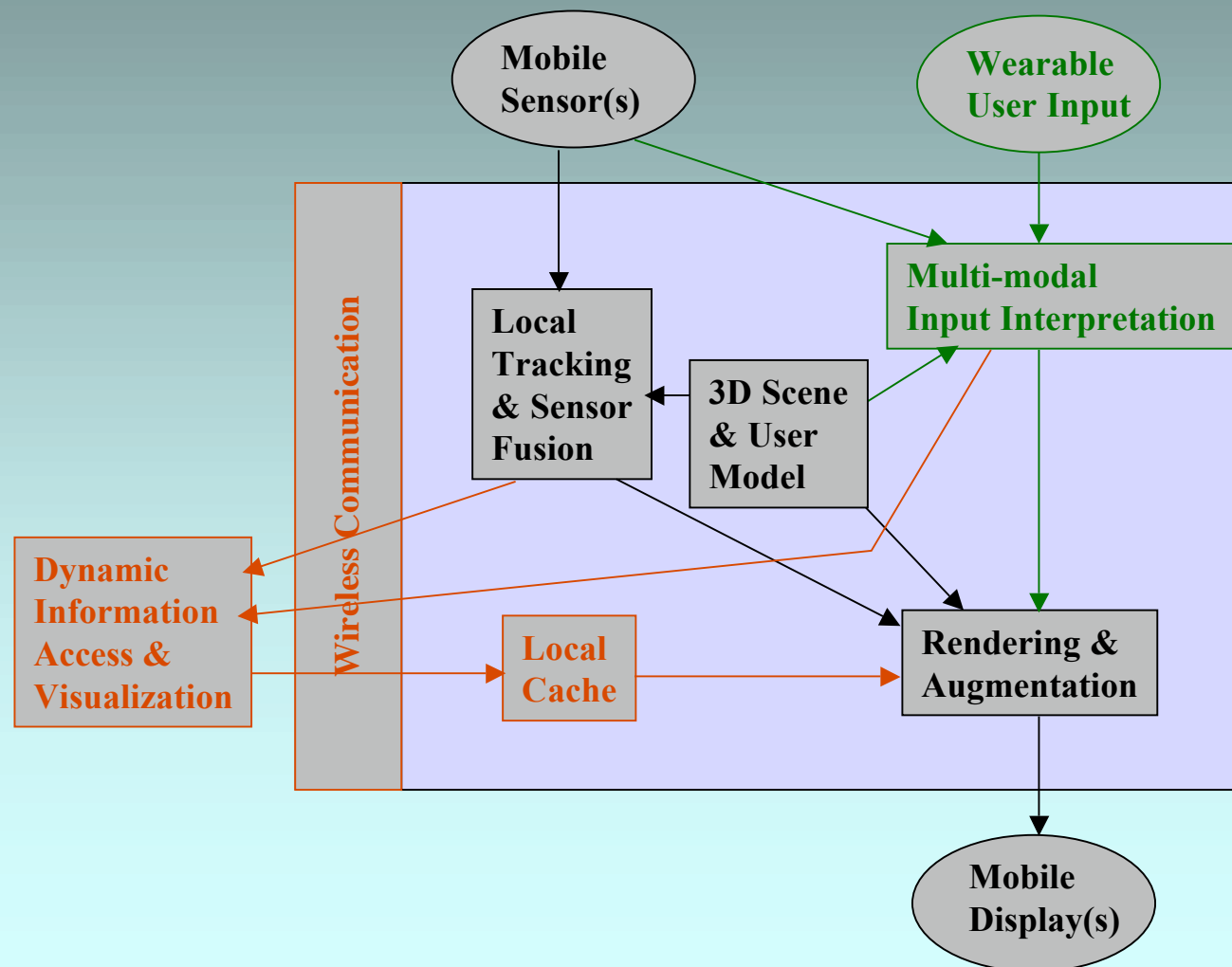
Architecture of a mobile AR System



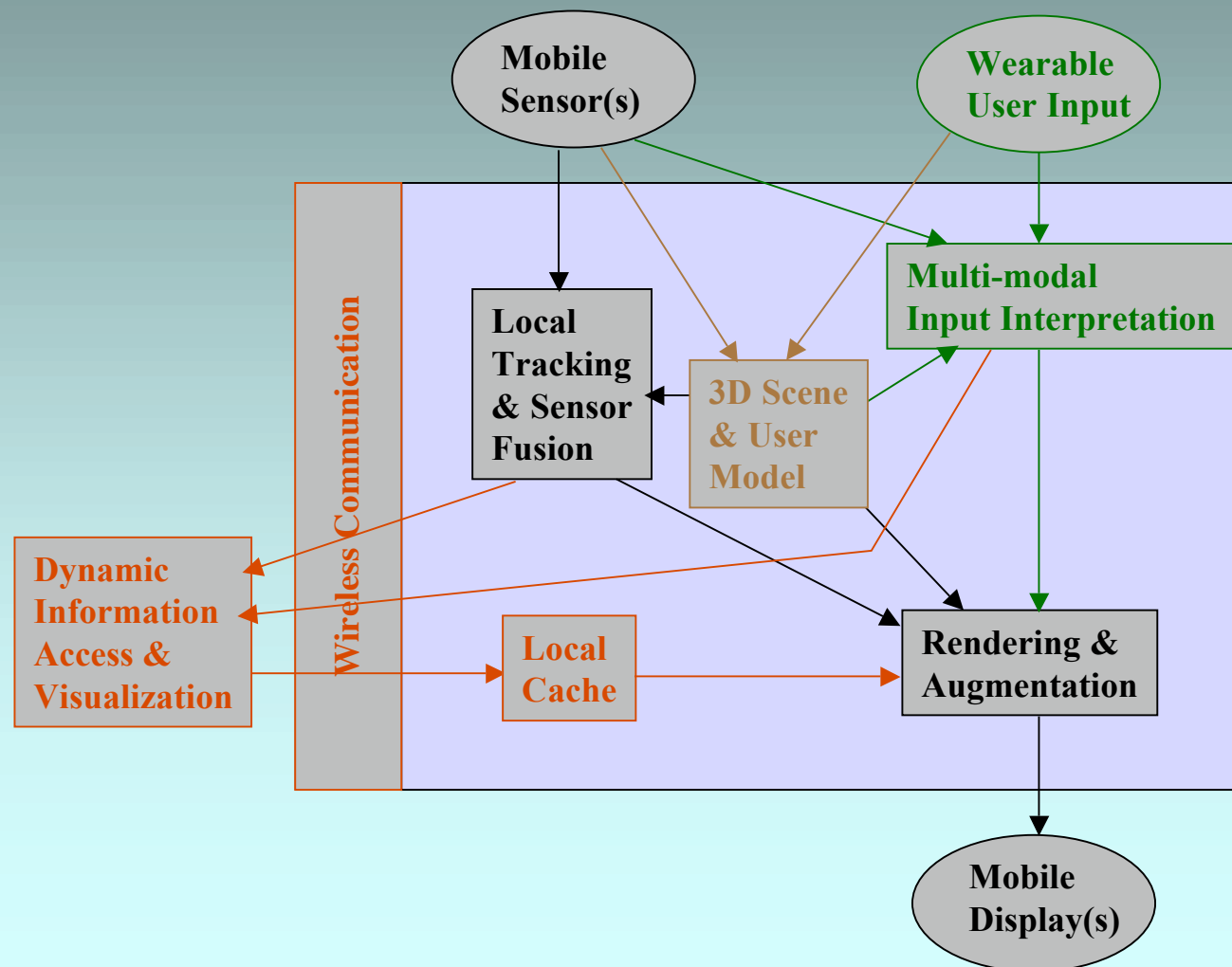
Architecture of a mobile AR System



Architecture of a mobile AR System



Architecture of a mobile AR System

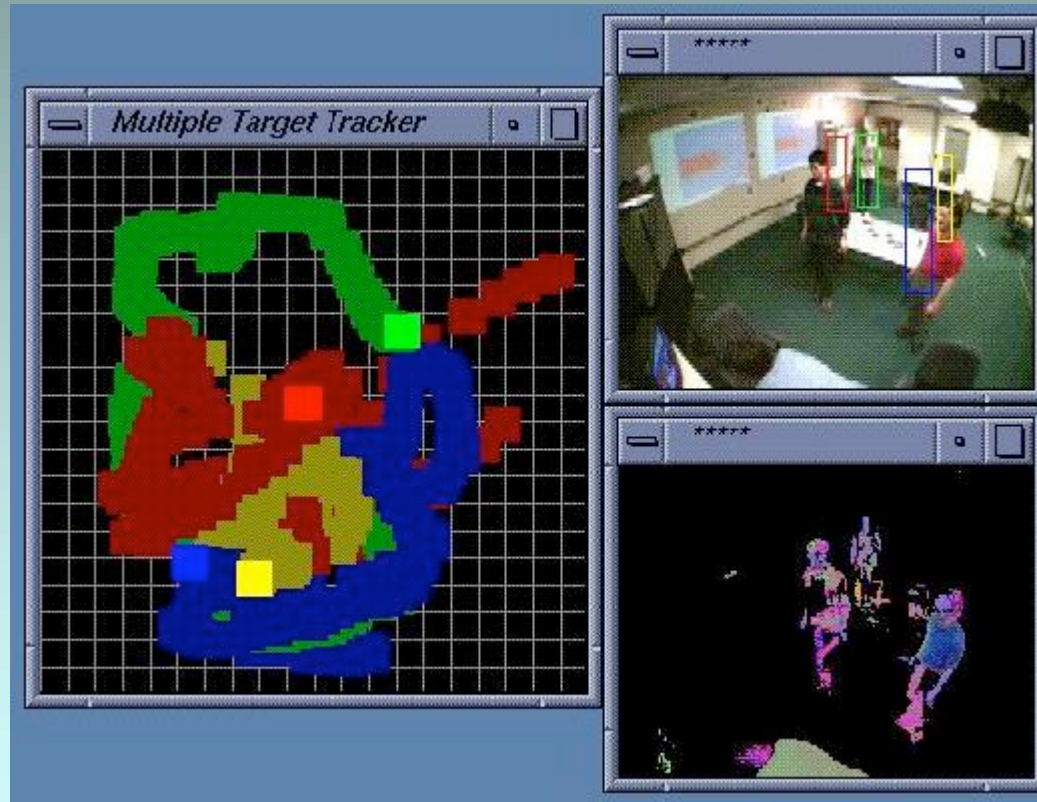


Architecture for Distributed Tracking

Hypothesis

- Dynamic scene analysis with a single mobile camera is very difficult:
 - user motion
 - mobile objects in the scene
 - Simplifications in applications:
 - known stationary 3D features
 - stationary sensors/cameras in the environment
- ⇒ “AR-ready” intelligent environments

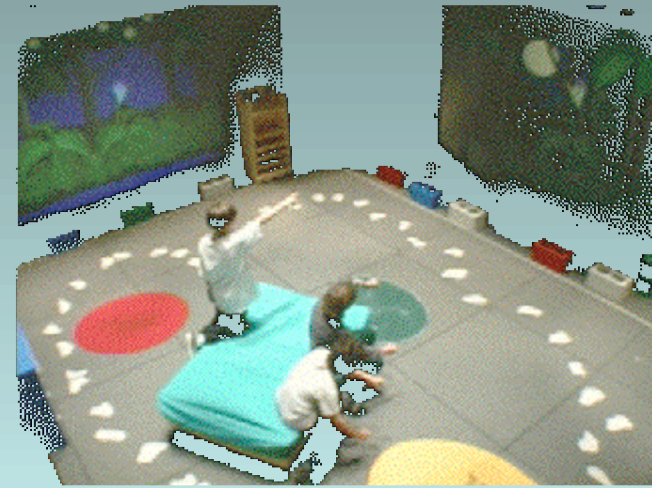
Examples Intelligent Environments (MIT)



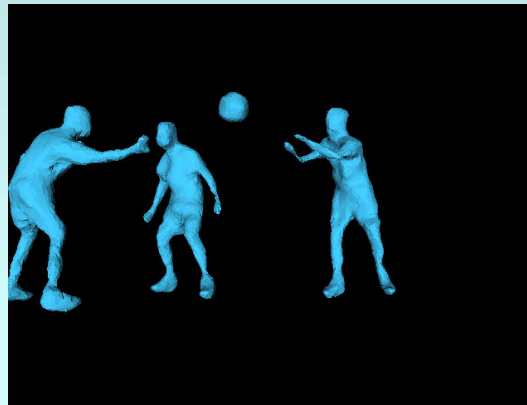
MIT AI Lab
Interactive Rooms



MIT Media Lab
Kidsroom, ALIVE



Example Virtualized™ Reality (CMU)



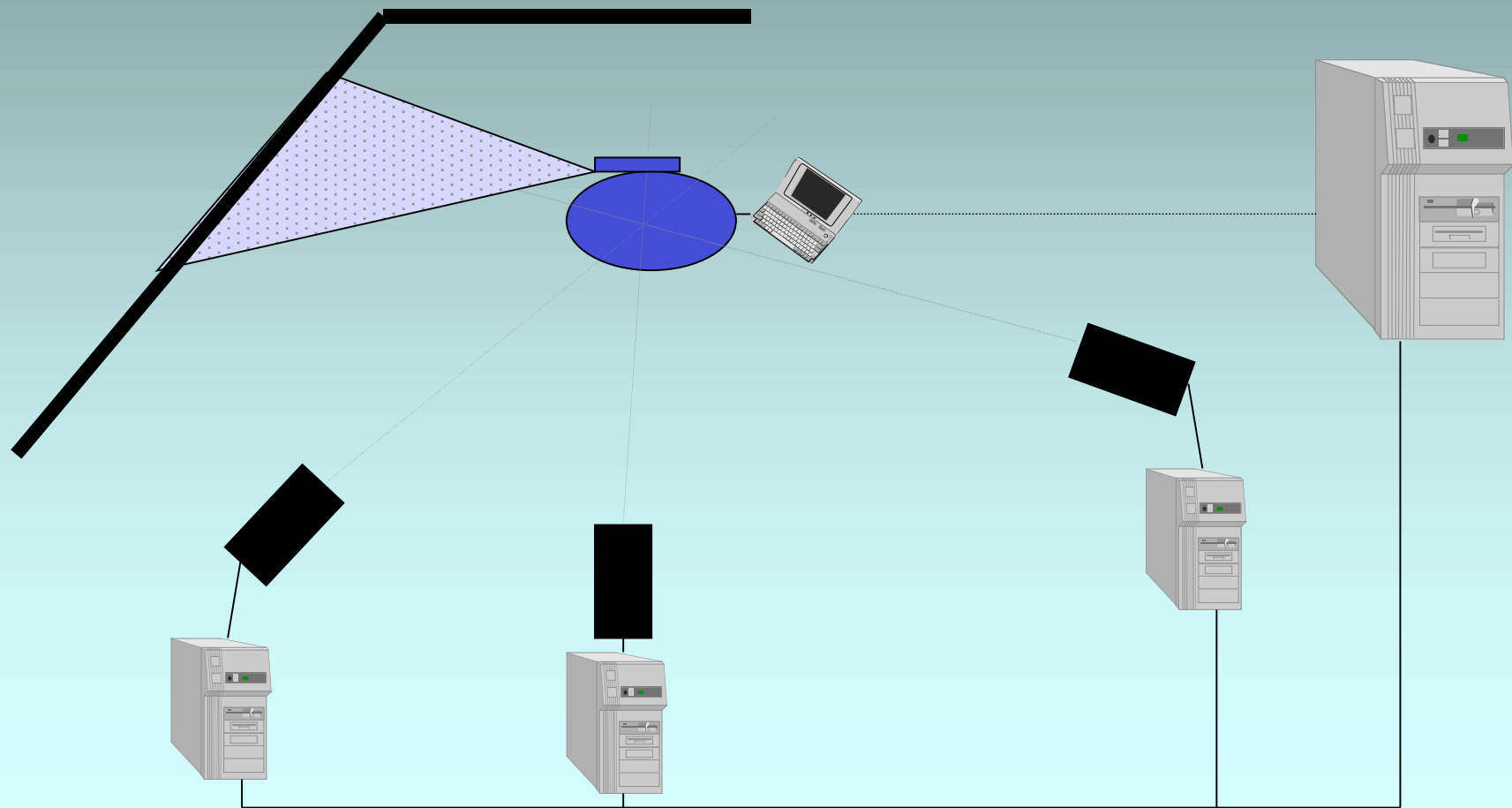
Distributed Tracking for AR

- Concepts -

- Several stationary sensors and displays in suitable locations
- Wearable mobile sensors and displays
- Non-proprietary communication protocols between mobile and stationary components

Distributed Tracking for AR

- Setup -

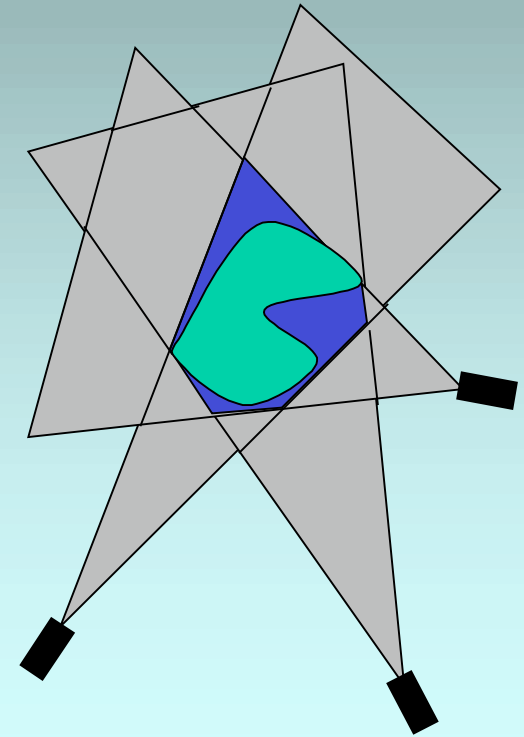


Distributed Tracking

- **Stationary (pre-calibrated) units:** Track the mobile unit(s)
- **Central unit:** Triangulate data from several stationary units
- **Transmission** to mobile unit(s): user motion data, 3D features in field of view
- **Mobile unit(s):** precise tracking
- **Feedback** to stationary units

Distributed Occlusion Detection

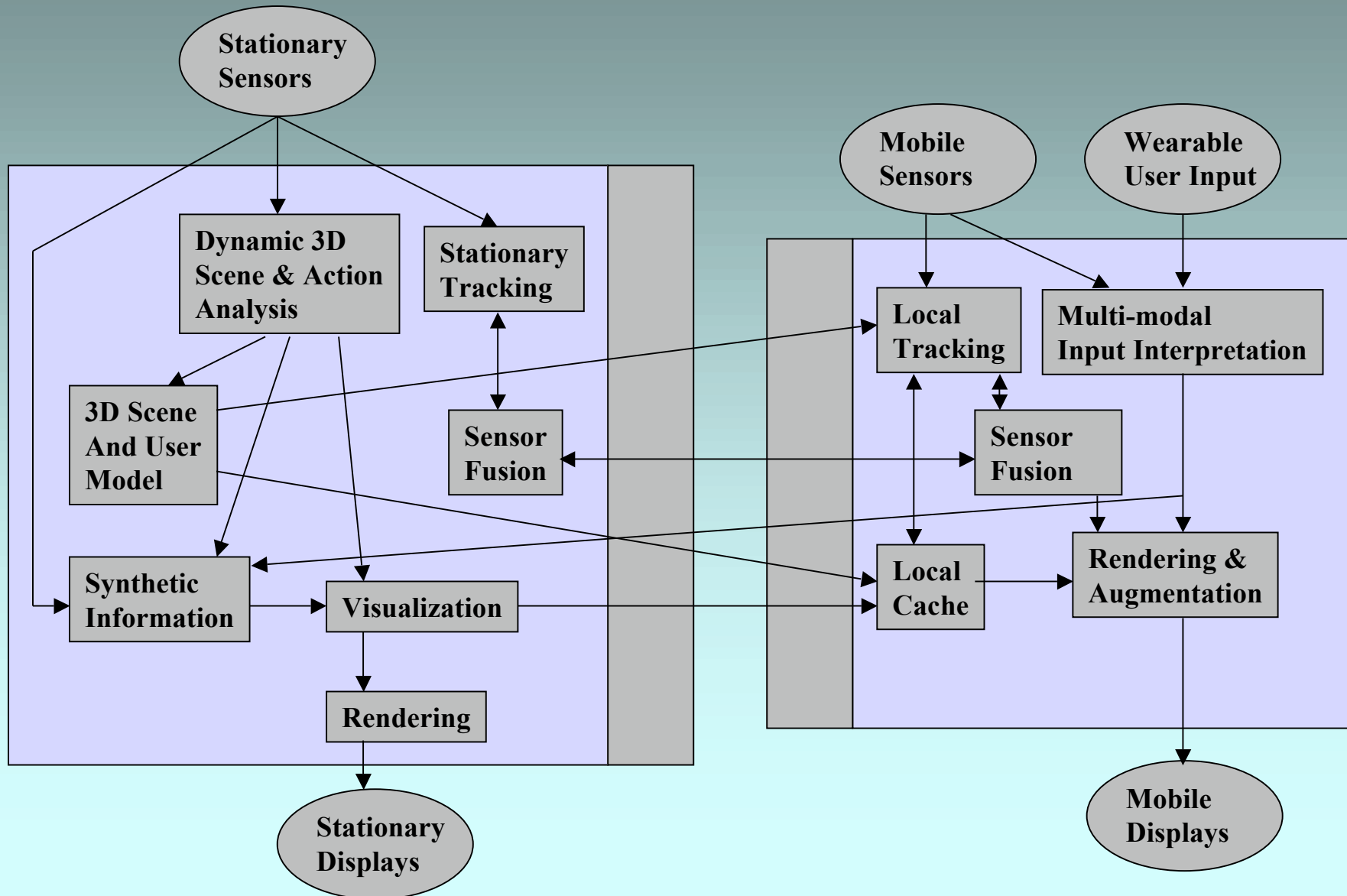
- **Stationary units:** Determine silhouettes of mobile objects (e.g. from difference images)
- **Central unit:** Compute convex hulls of mobile objects from several silhouettes
- **Transmission:** Convex hulls
- **Mobile unit(s):** Occlusion analysis



Distributed Depth Recovery

- **Stationary units:** Compute depth maps (e.g. real-time stereo vision)
- **Central unit:** Combine depth maps into a 3D volumetric model
- **Transmission:** Volumetric Data
- **Mobile unit(s):** Occlusion Analysis, diminished reality, image-based rendering, telepresence, ...

Distributed AR-ready Environment



Important Issues

- Shared networks: Quality of service
- Bottlenecks
- Mobile agents
- Load balancing
- Adaptation to: task, user, position, time, ...
- Security (access rights)
- Safety, user acceptance (gadget or information overload): NO OCCLUDED REALITY

Appropriate User Interfaces

- Multi-media Output: which display for what kind of data?
- Multi-media Input: which interaction method in which situation? Speed of data analysis, complexity of the „communication language“
- Many possibilities to confuse the user!
- Sensor (concentration) overload? Motion sickness

An AR Software Architecture: DWARF

Distributed Wearable Augmented Reality Framework

with

Martin Bauer

Bernd Brügge

Asa MacWilliams

Thomas Reicher

Christian Sandor

Martin Wagner

Basic Design - Distributed Services

- Software services that can run on different hardware modules.
- Modules are connected by wired or wireless networks.
- Modules can be distributed on the body...
- ...or as external devices in intelligent environments.
- Services discover each other and dynamically cooperate to form a complete system.

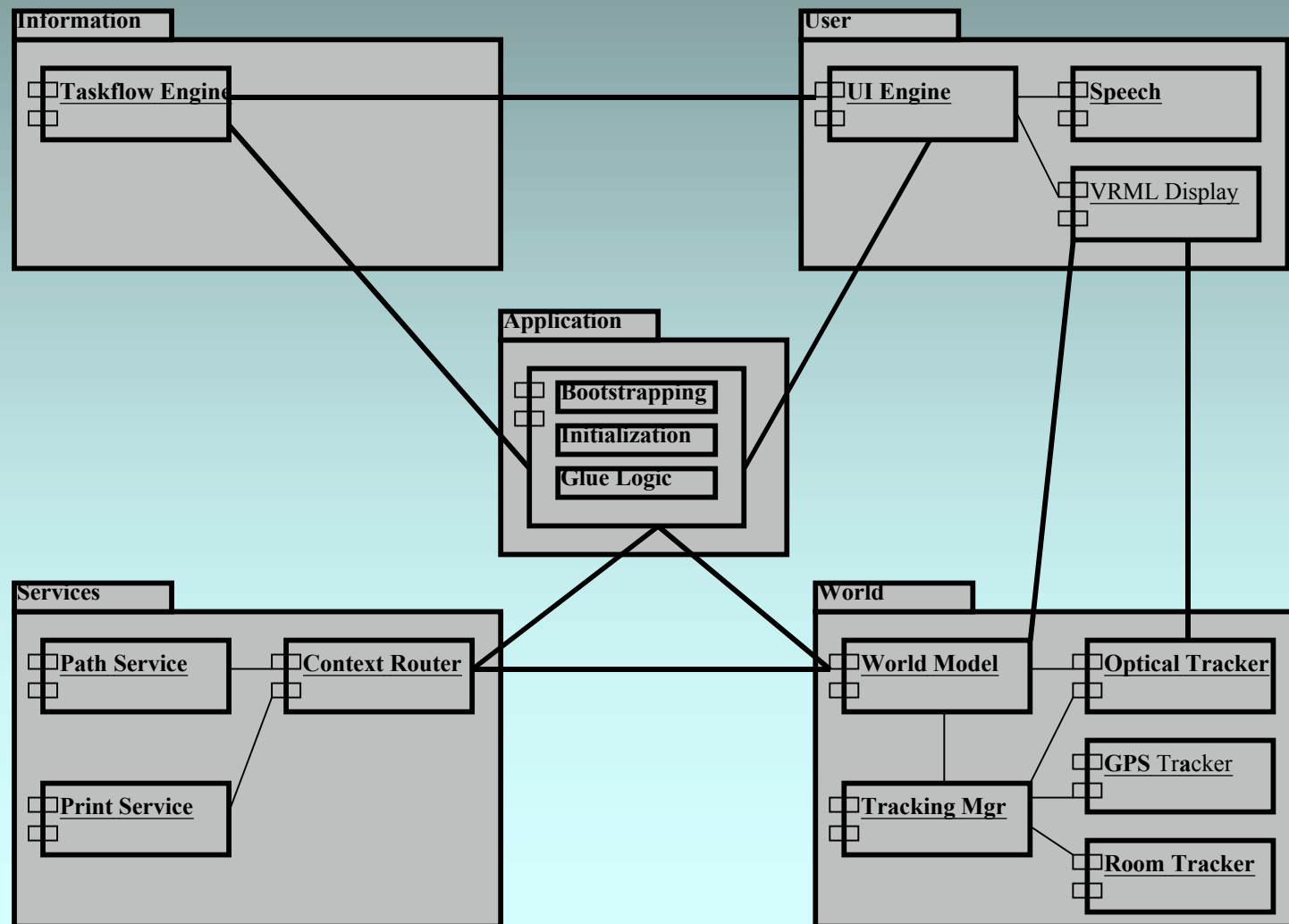
Main Elements

- Services providing general functionality
 - General enough to reuse, yet also efficient
- Architecture to fit the services together
 - Generalization of many different AR systems
- Middleware to let services communicate
 - Must allow fast yet flexible communication

Main Services

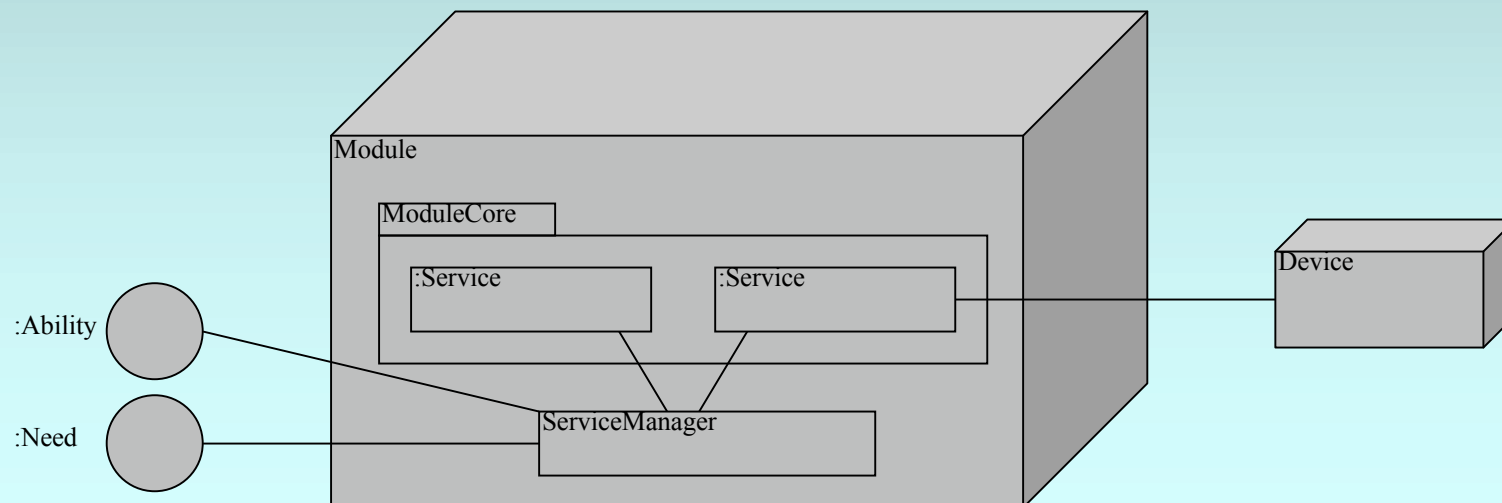
- First DWARF services cover the basic functionalities for AR applications
 - World model, optical tracker, GPS/compass tracker, tracking manager, user interface engine, VRML viewer, HTML viewer, voice recognition, taskflow engine, context router
- Implementation of services uses well-established and third-party technology

Generic Architecture



Middleware

- To find each other spontaneously and communicate with one another, the DWARF services use CORBA-based middleware.
- This is distributed as local Service Managers on each hardware node to provide fault tolerance.



Middleware Challenges

- Middleware needs to be fast, yet flexible
 - Decomposition into communication and location subsystems
- Middleware should not have to be “in the middle”
 - Distributed service managers
- Services that do not know each other have to cooperate
 - Service descriptions with needs and abilities

Results

- Nearly entire functionality is handled by the DWARF services.
 - Navigation sequence, campus geography, multi-modal I/O, printer location, outdoor tracking, indoor tracking, 3D rendering
- First application design is greatly simplified.
 - Model campus and route in markup languages
 - Bootstrapping, glue logic
- Application implemented in three weeks' time.
- Network latency remained below 8 ms.
- Has been used in many projects at TUM since 2000.