Multi-Camera Systems for Virtual Reality: The Grimage Experience

Bruno Raffin

INRIA

SVR 2009, Porto Alegre

Presence, Interaction and immersion in VR

Presence: user presence in VR environment

Depend on the capabilities of the system to extract data about the user and its environment

Interaction: modifications of the VR environment triggered from presence data

Immersion: user feeling that one (or more) of his senses is fully focused on the corresponding rendering of the VR world.

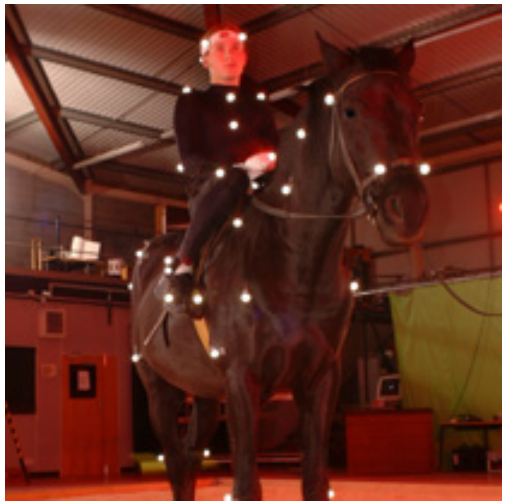
Cameras for VR Presence

Camera: 2D real-time video stream

Camera pair: 2D + depth

Multiple cameras: 3D

Markers



Goal: 3D tracking of markers

Data retrieved:

identification, position, orientation, velocity, acceleration

Data density: limited by the number of markers used

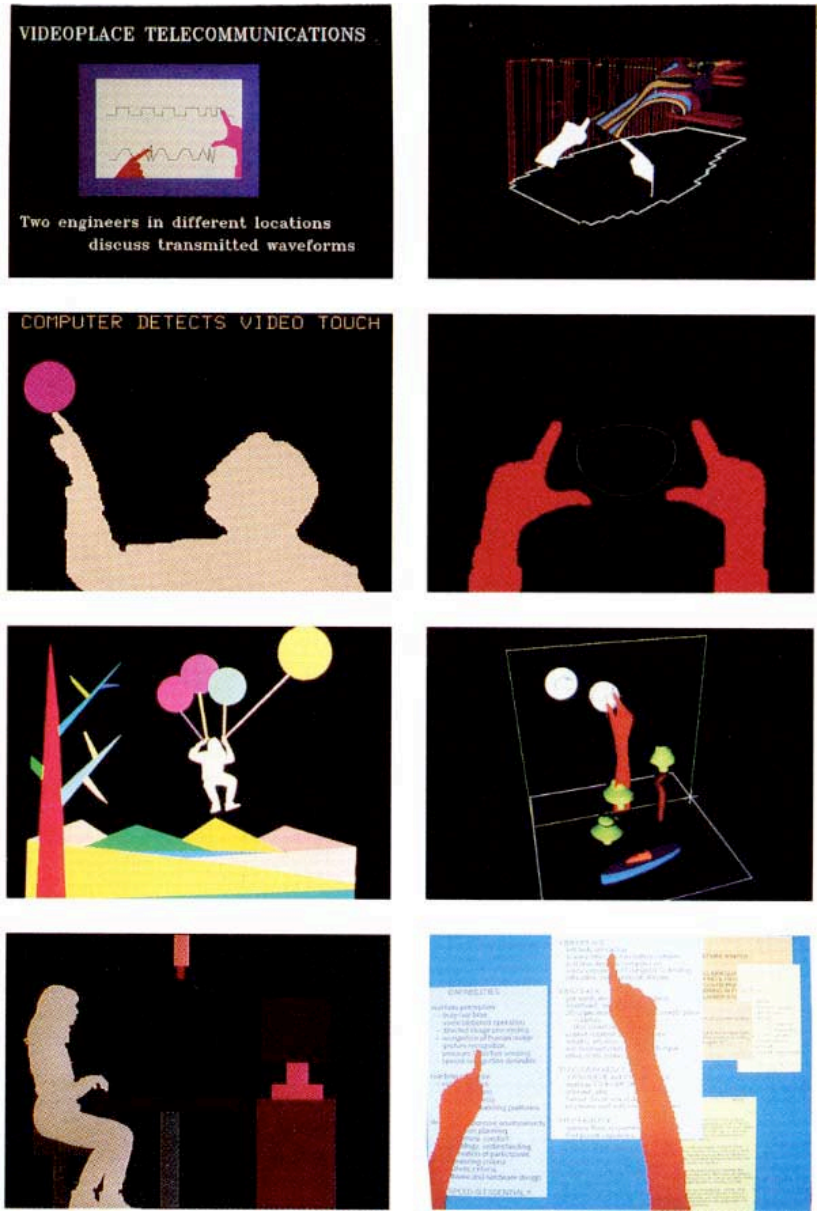
Classically used for motion capture, 3D trackers, tangible interfaces

Markerless

Pioneer work:

Myron Krueger (74)

Single camera



bruno.raffin@inria.fr

Markerless

- 2001: 30 cameras for the 2001 Super Bowl (Takeo Kanade)

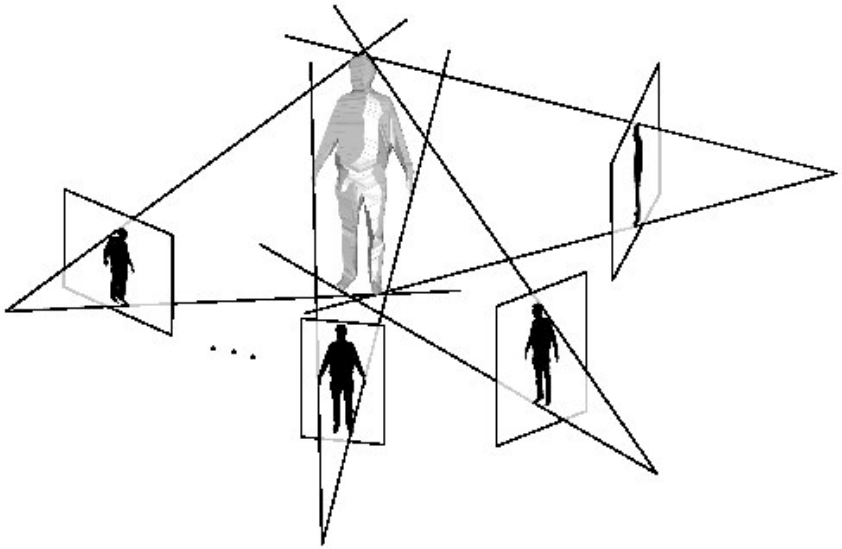
  Video 1

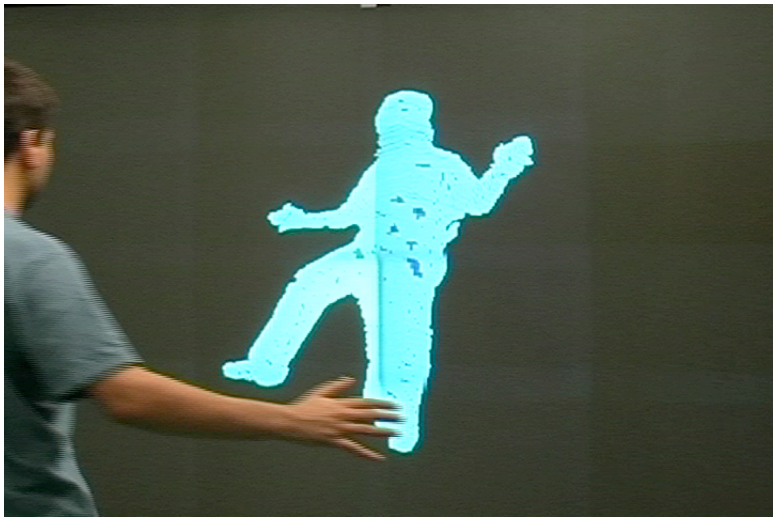
  Video 2

- CNN Hologram
  - 2008
  - 35 cameras

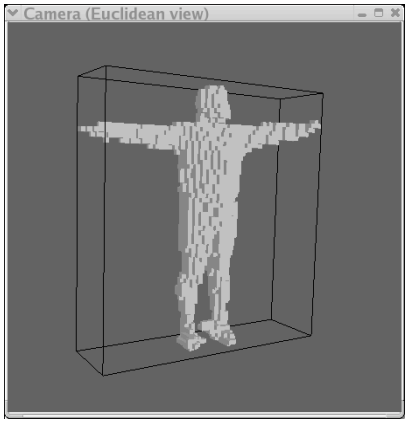
Markerless

3D modeling : extract from multiple cameras a 3D volume or surface of the scene being observed



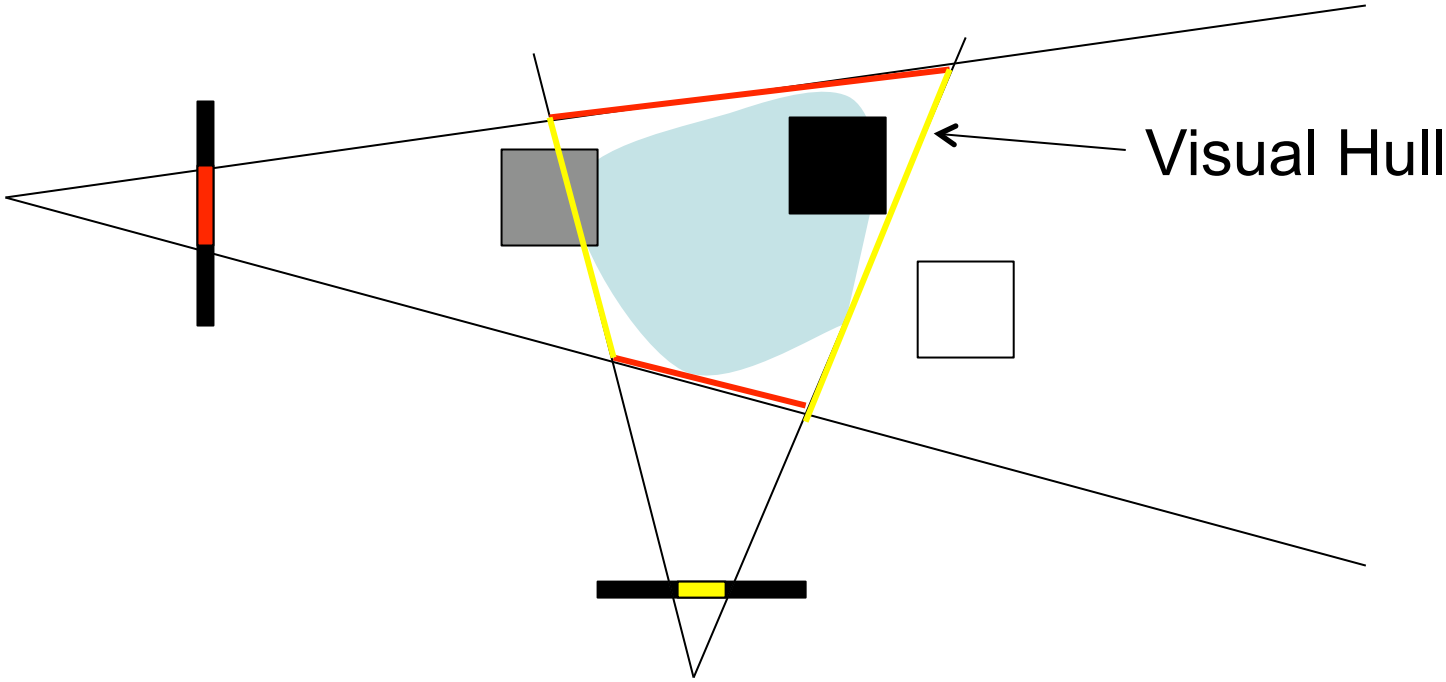


Visual Hull



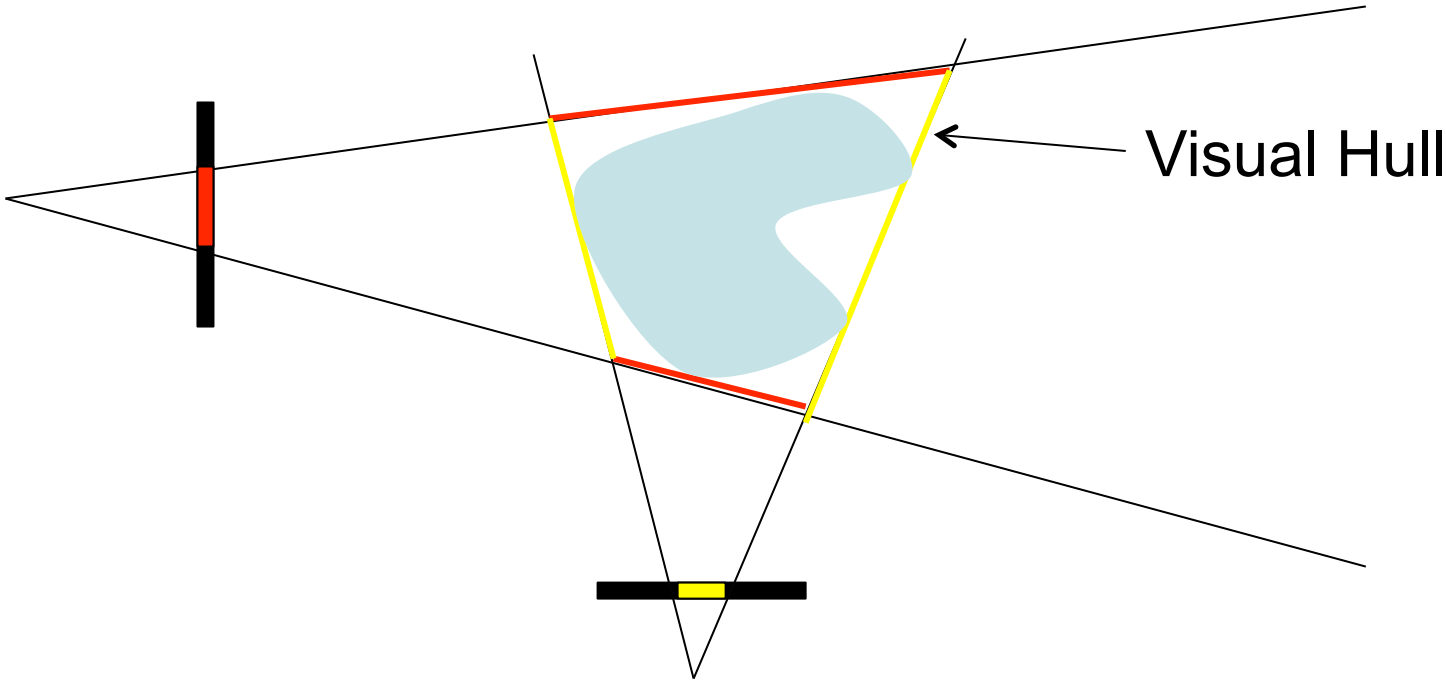
Shape from silhouettes (Laurentini 94)

Octree carving (Szeliski 93)



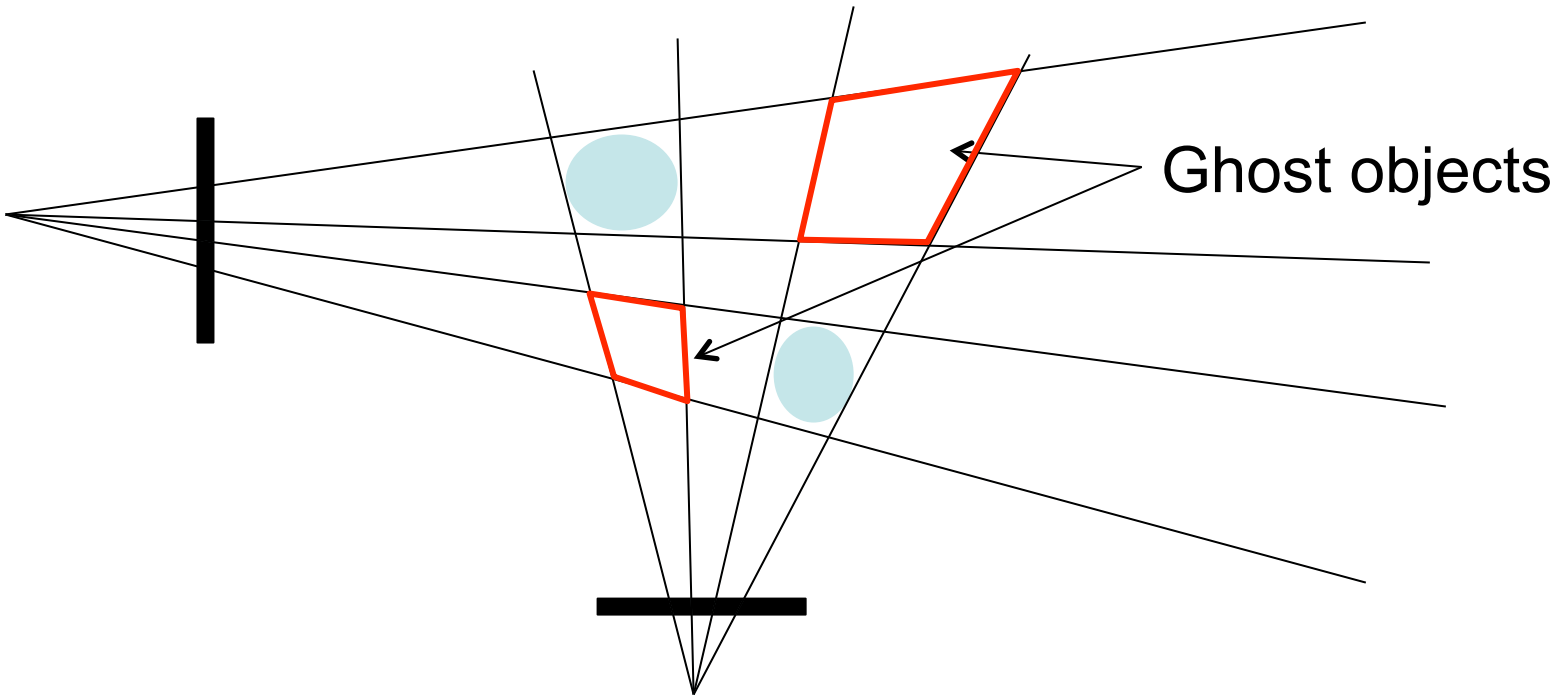
Visual Hull

Cannot extract concavities

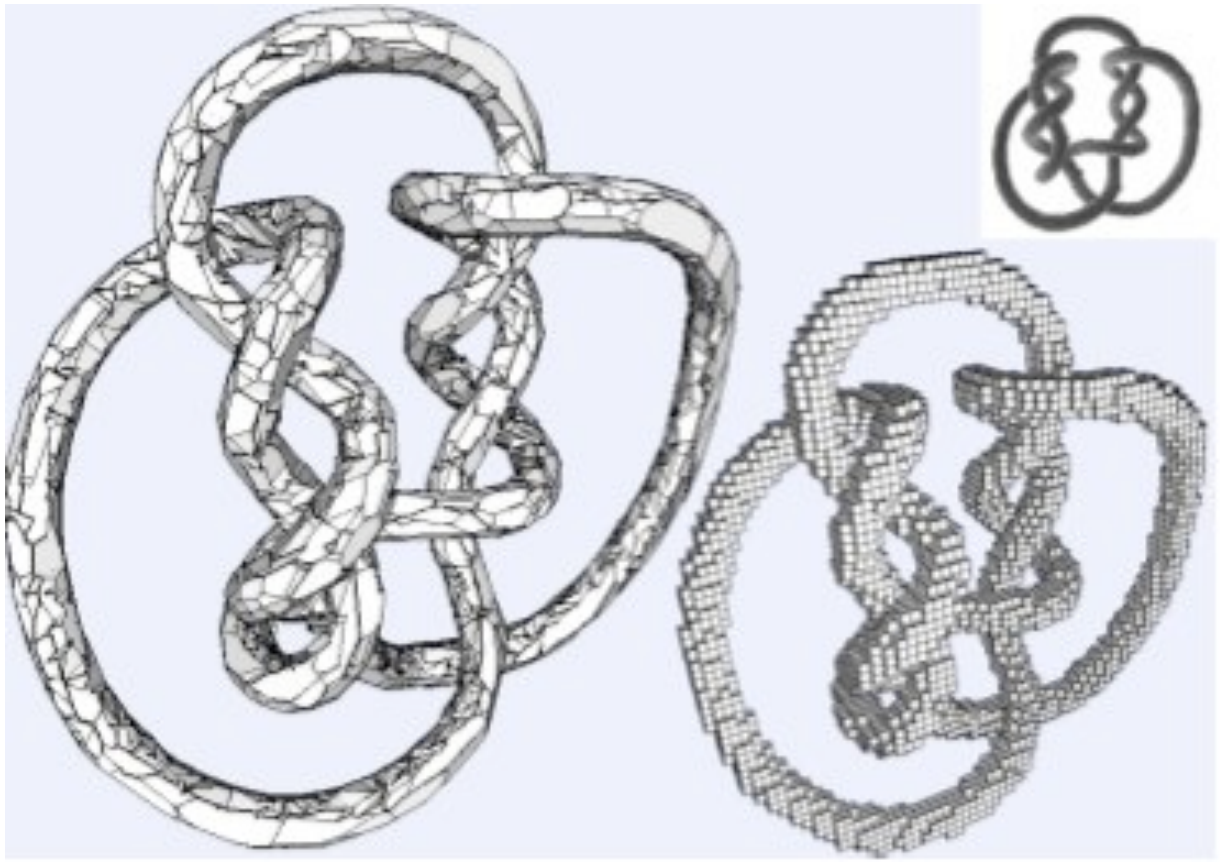


Visual Hull

Ghost objects



Visual Hull



Visual hull computed from 32 images

Texturing the Visual Hull

Use photometric data from cameras to improve the visual appearance

EPVH algorithm - 2004

Texturing the Visual Hull

- Octree Carving + Marching Cube: no
- Exact Marching Cube (Liang & Wong 2008): yes
- Exact Polyhedral Visual Hull (Franco & Boyer 2003): yes

Texturing the Visual Hull

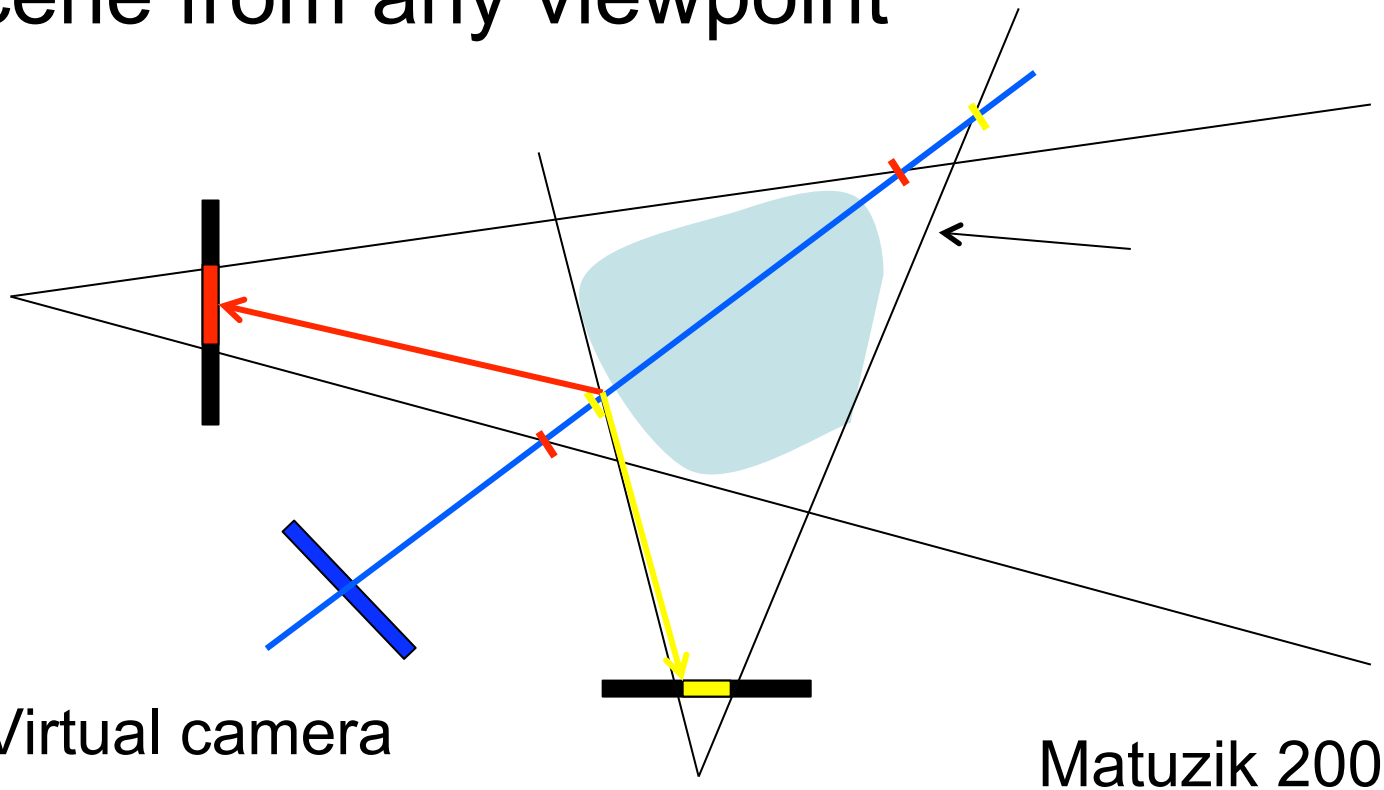
Texturing algorithms:

- Visibility issues
- Mix between photometric data from various cameras

EPVH algorithm - Off-line - 2004

View Dependent “3D Modeling”

- Compute a 2D+depth information
- Free viewpoint: observation of the scene from any viewpoint



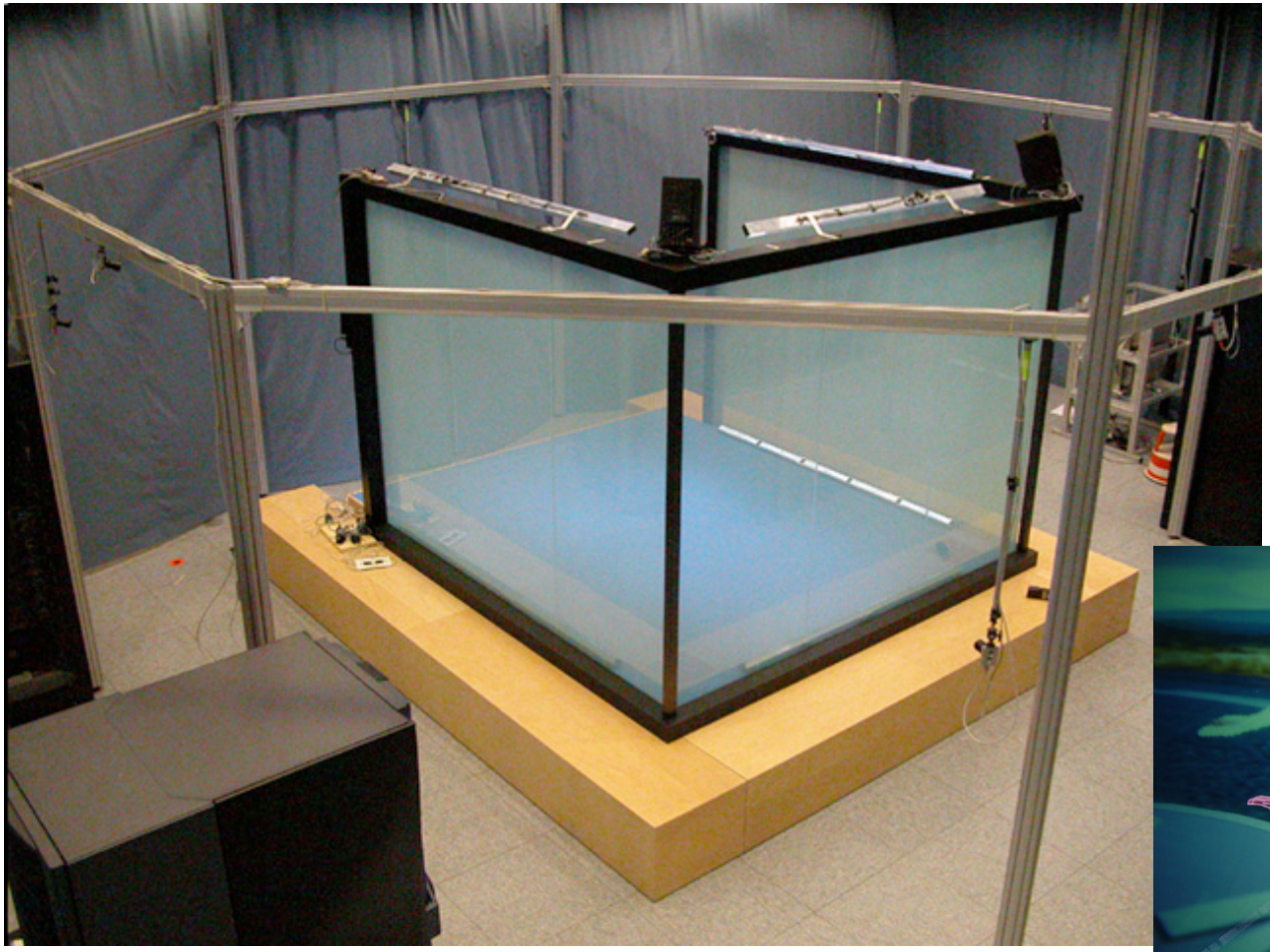
Virtual camera

Matuzik 2000

View Dependent “3D Modeling”

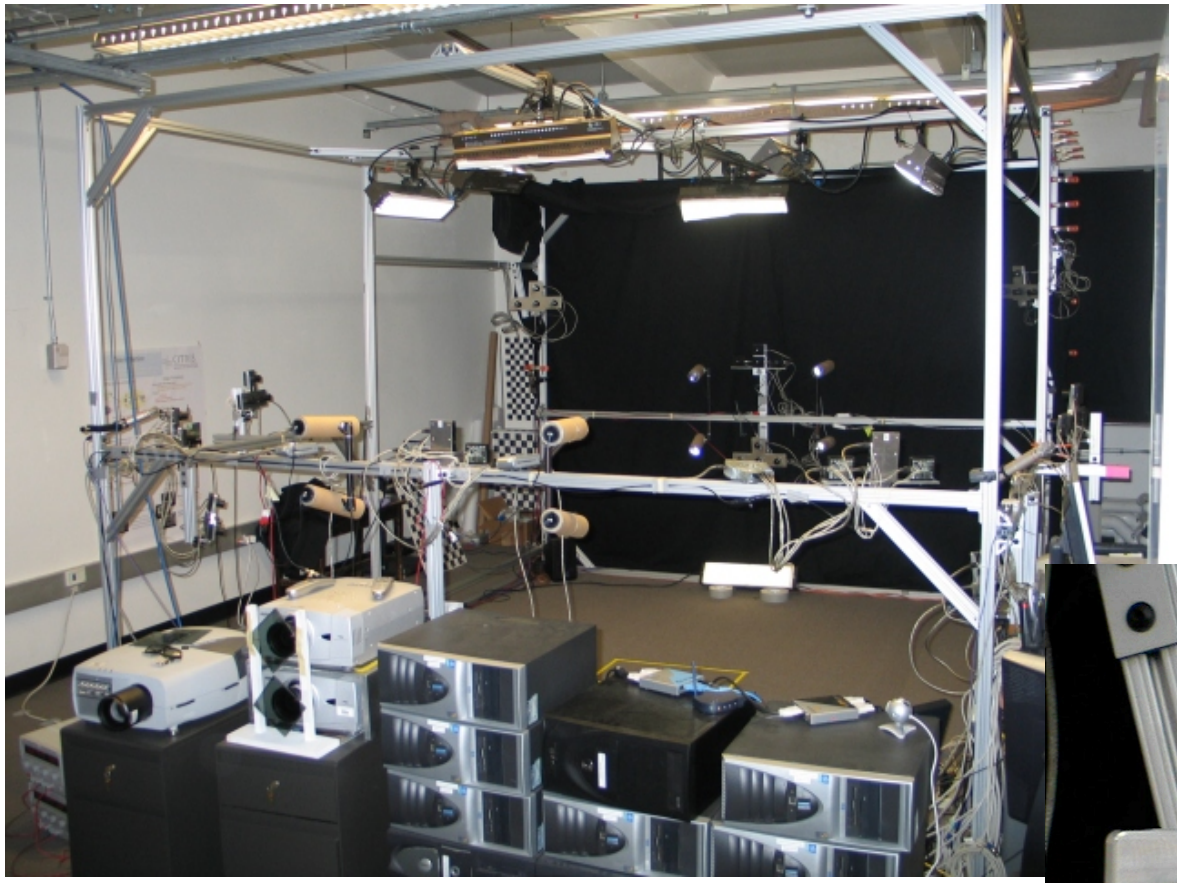
- Often used in telepresence environments for bandwidth limitations concerns
- Visual presence
- No mechanical presence:
  - no data on the invisible parts
  - cannot compute collision with virtual objects

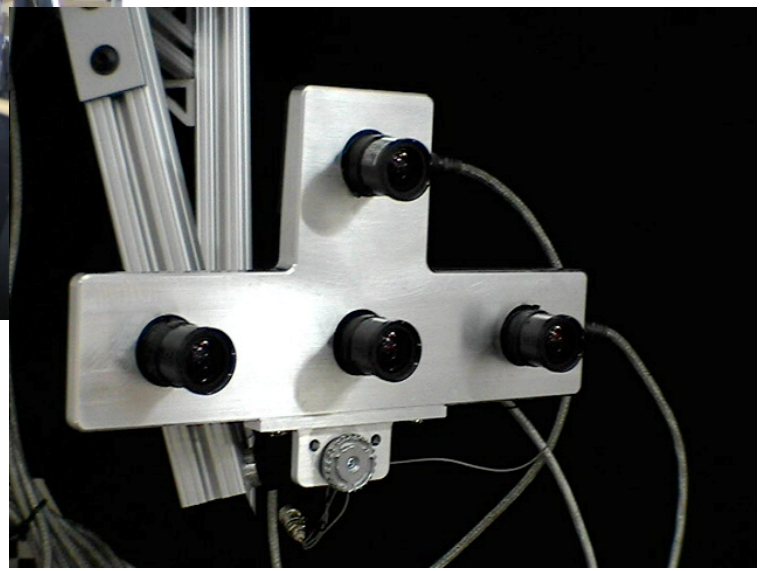
Blue-C (2003)



Blue-C (2003)

Tele-Immersion@UCB





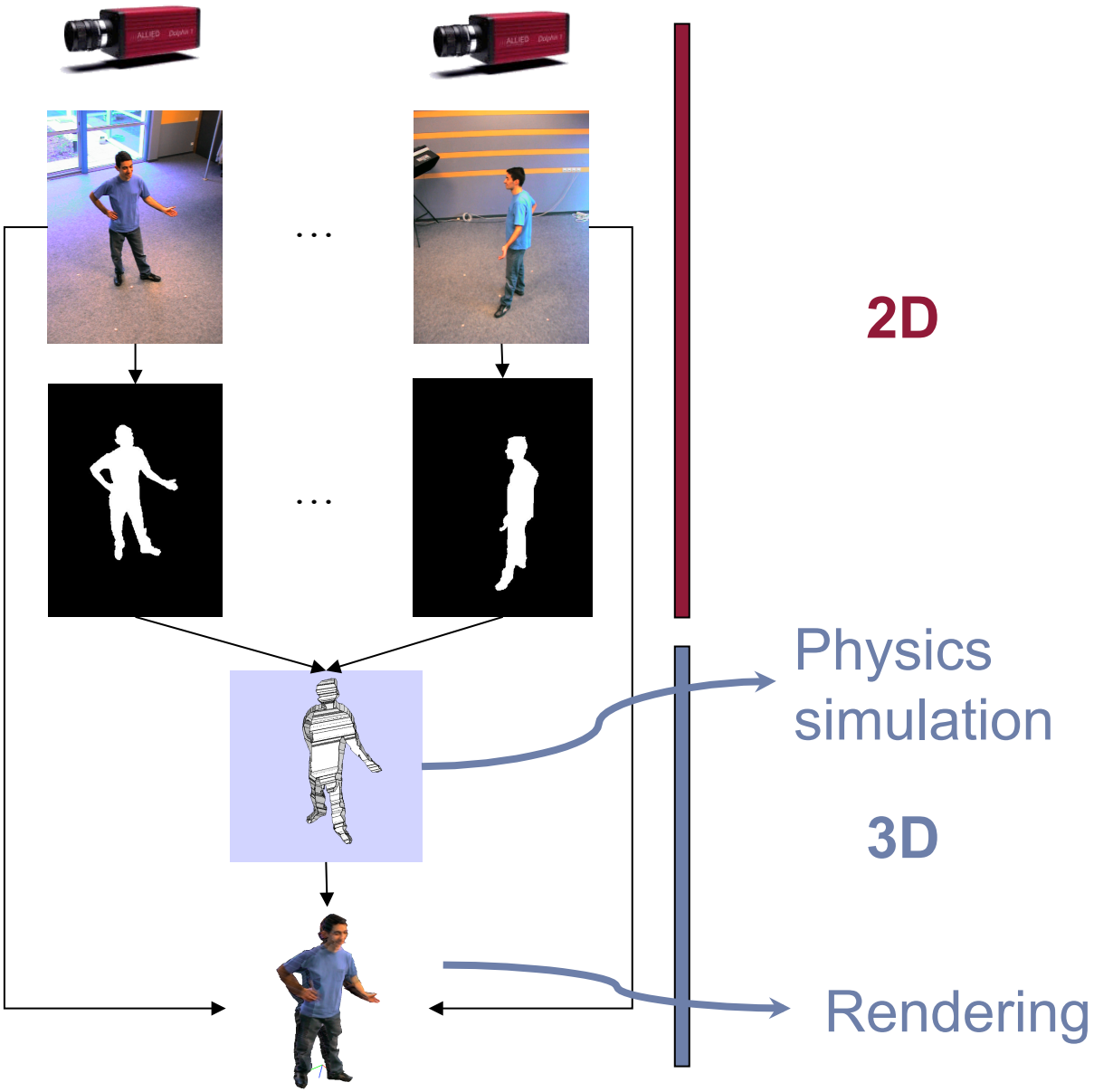
48 cameras arranged in 12 stereo clusters

Tele-Immersion@UCB

- video

2009

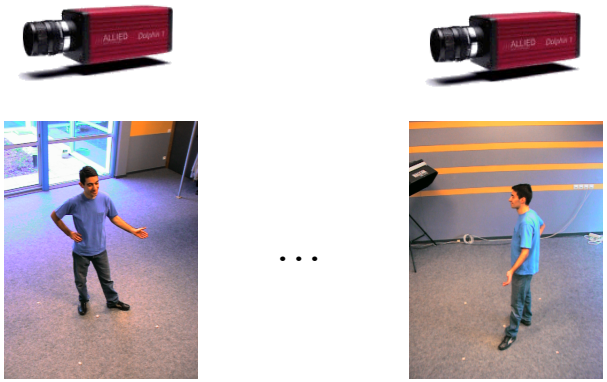
3D Modeling Pipe-Line



Calibration & Acquisition

Calibration matrix:

- Chessboard
- Light stick

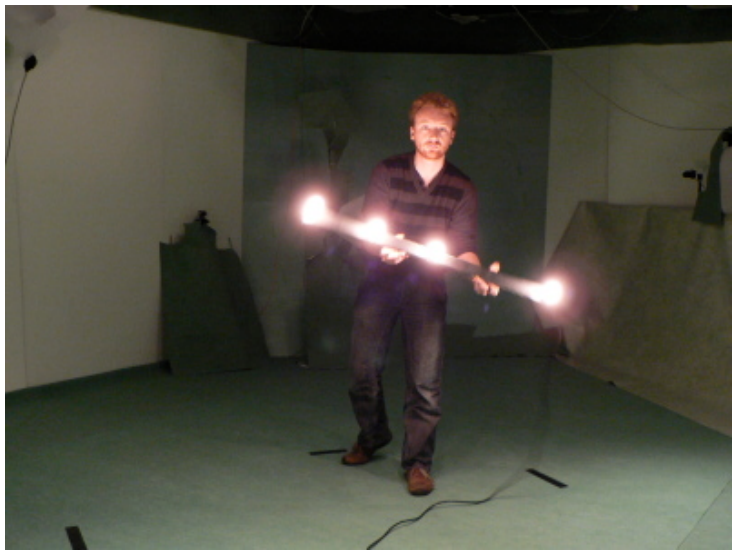


Synchronisation:

- Synchronize (genlock) camera triggers
- Timestamp images with global wall clock (rtp) to identify metaframes

I/O intensive:

- 1 to 4 cameras per PC
  -> PC cluster required



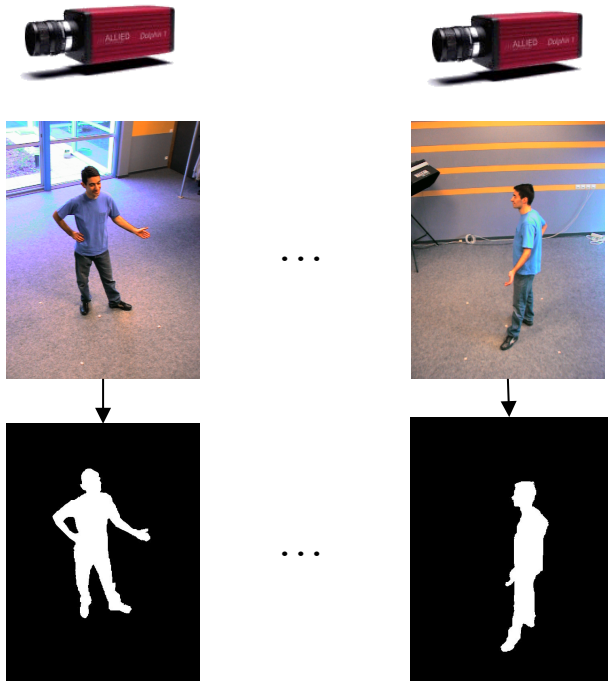
Background Subtraction

Remove background pixels

A key step that directly affects the quality of the 3D model

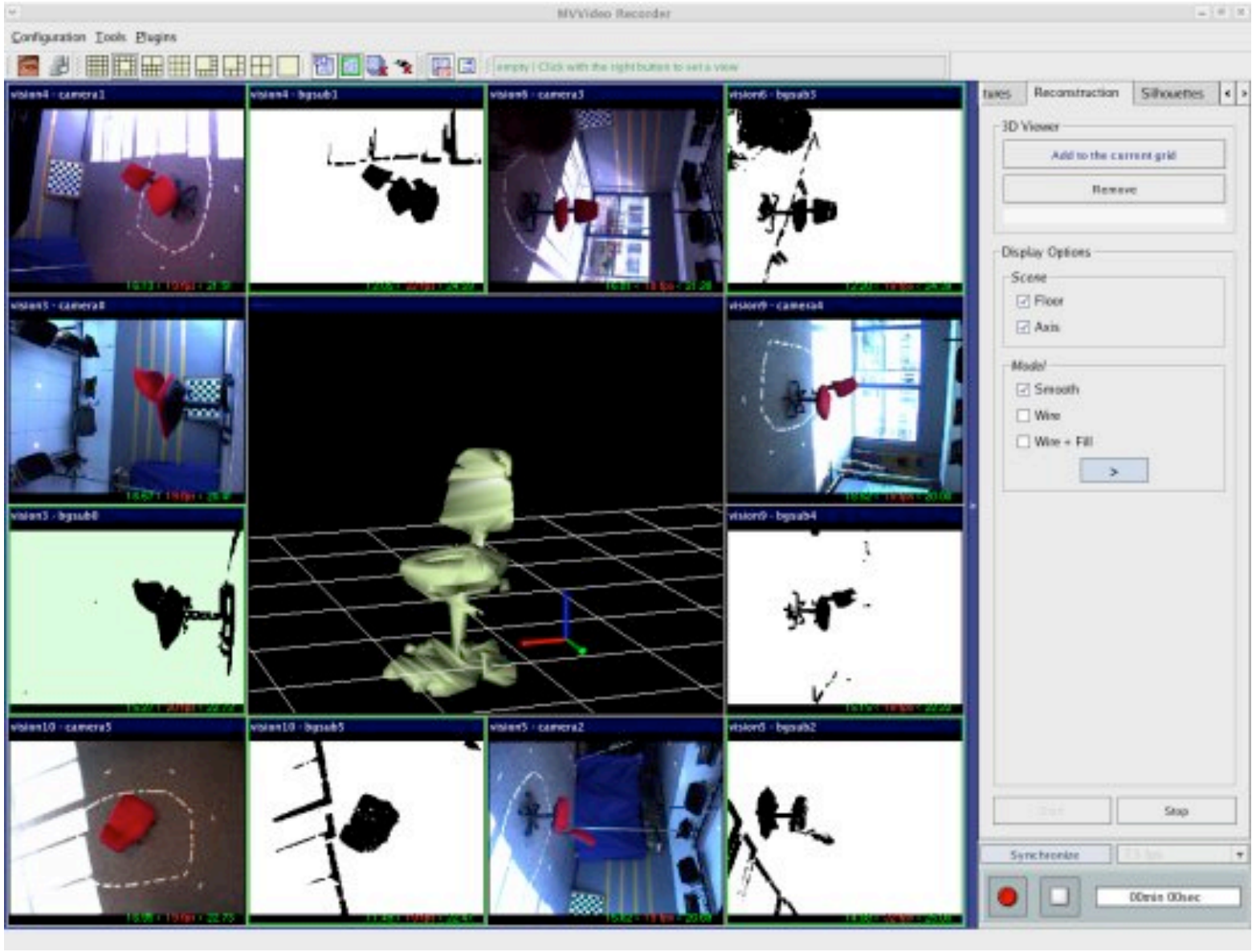
- Fixed background (no mobile object or changing light conditions)
- Lighting system to remove shadows
- Chromakey fabric

Real-time constraints: difficult to use sate-of-the-art algorithms



Plenty of room for improvement

Control Interface

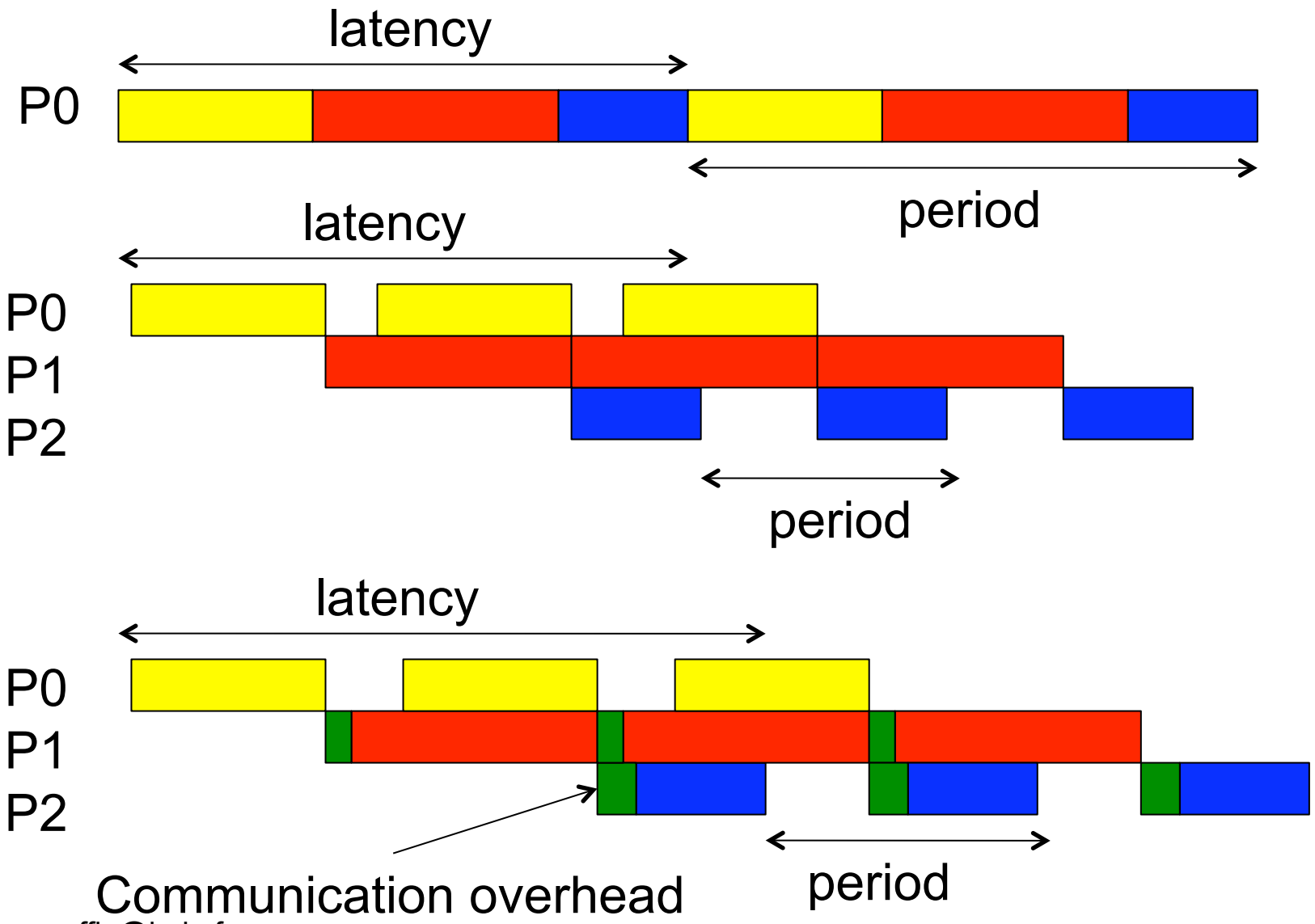


Parallelization & Distribution

- I/O and compute intensive application
- Interactive time constraints
  - 30 metaframes per second
  - Bellow 100 ms of latency
- Processors are now multi-cores

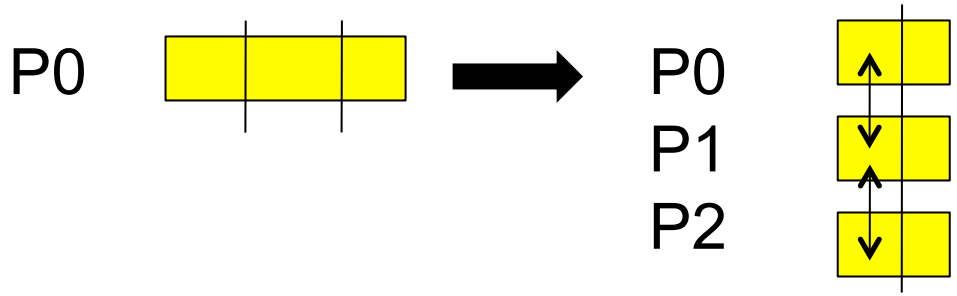
PC cluster + Code optimization +
Parallelization

Pipe-Lining ?


latency
P0
period
latency
P0
P1
P2
period
latency
P0
P1
P2
Communication overhead
period

Parallelization

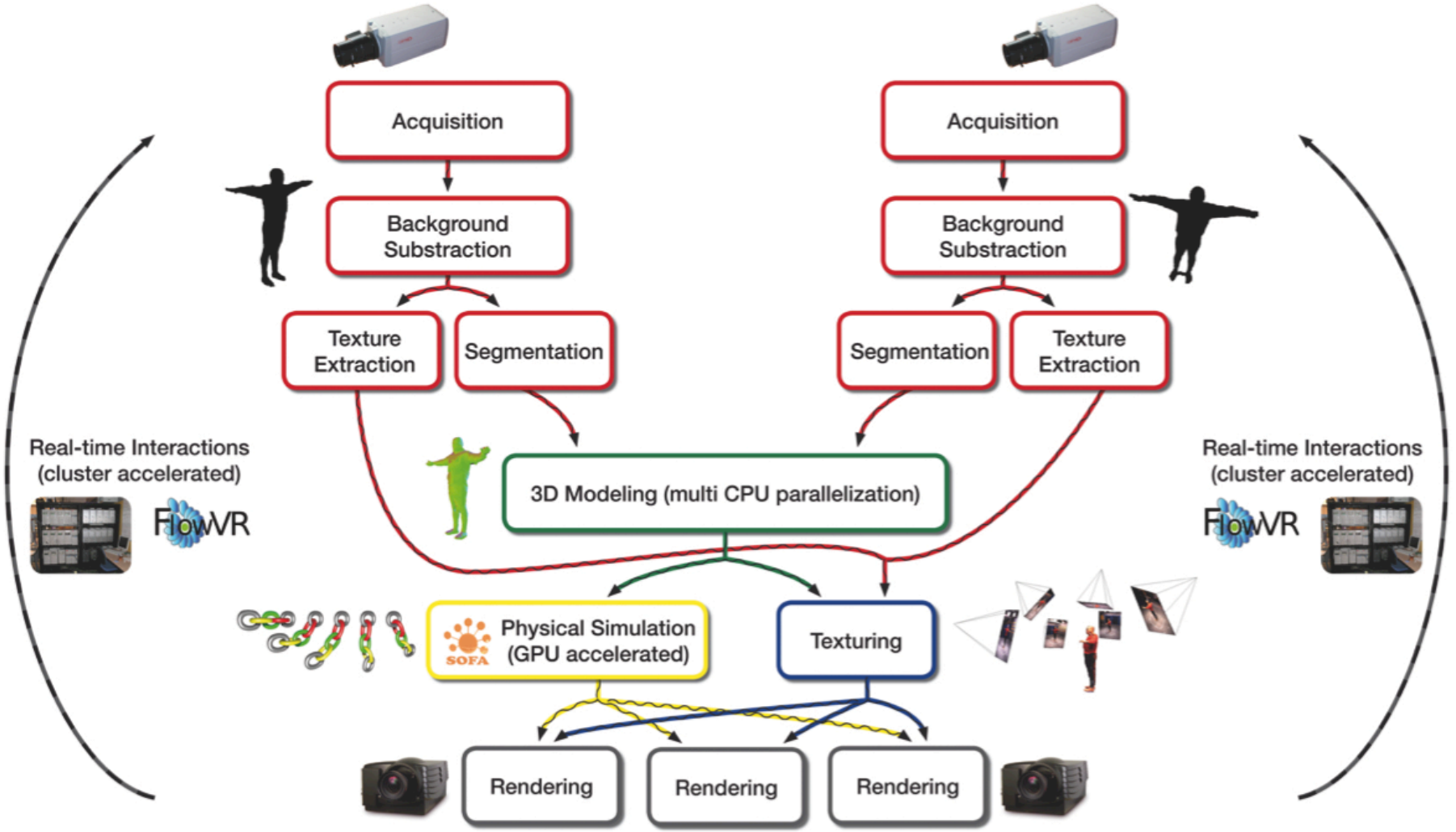
- Avoid pipe-line abuse
- Combine with efficient parallelizations



- Beware of the GPU mirage:
  - Cost of CPU/GPU memory transfer
  - Efficient for regular data parallelism

Intel Larrabee ?

Software Architecture



Large Interactive Application

Need for

flexibility, scalability, performance

in a distributed context

- Sockets: NO
- Parallel programming (MPI): not flexible enough
- Components (Corba): better but does not support parallel code coupling

Middelware dedicated to interactive applications

FlowVR enforces a **modular programming** that leverages software engineering issues while enabling high performance executions on distribued and parallel architectures

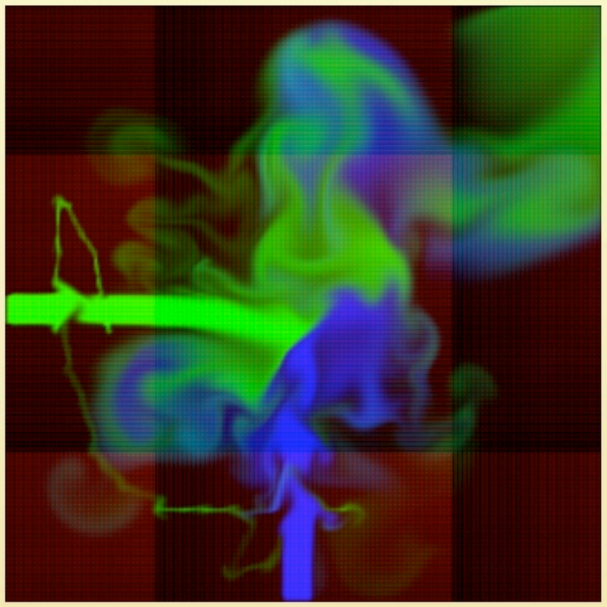
Component oriented

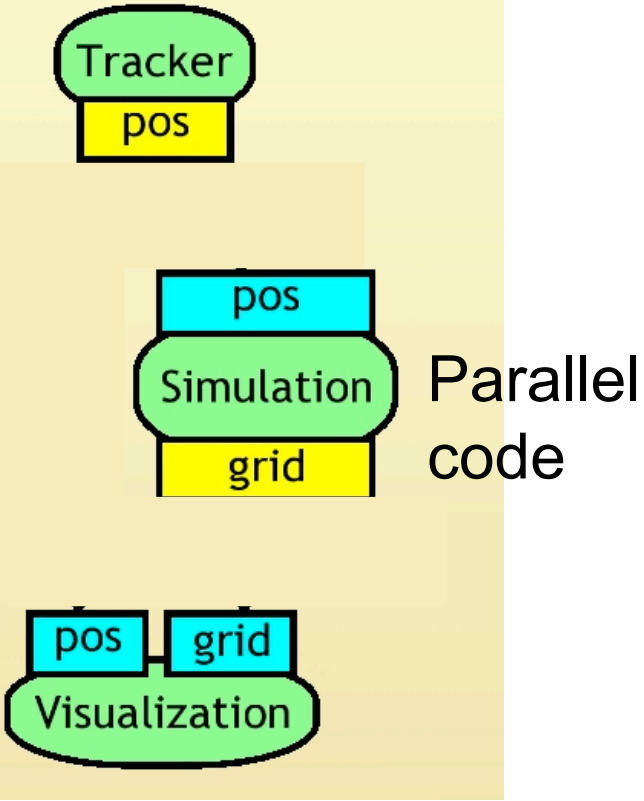
- Component hierarchy (design patterns)
- Parallel
- Communication schemes for interactive applications (push, pull, sampling, collectives, etc.)

FlowVR
High Performance Virtual Reality

3 programming steps:

1. Individual component programming

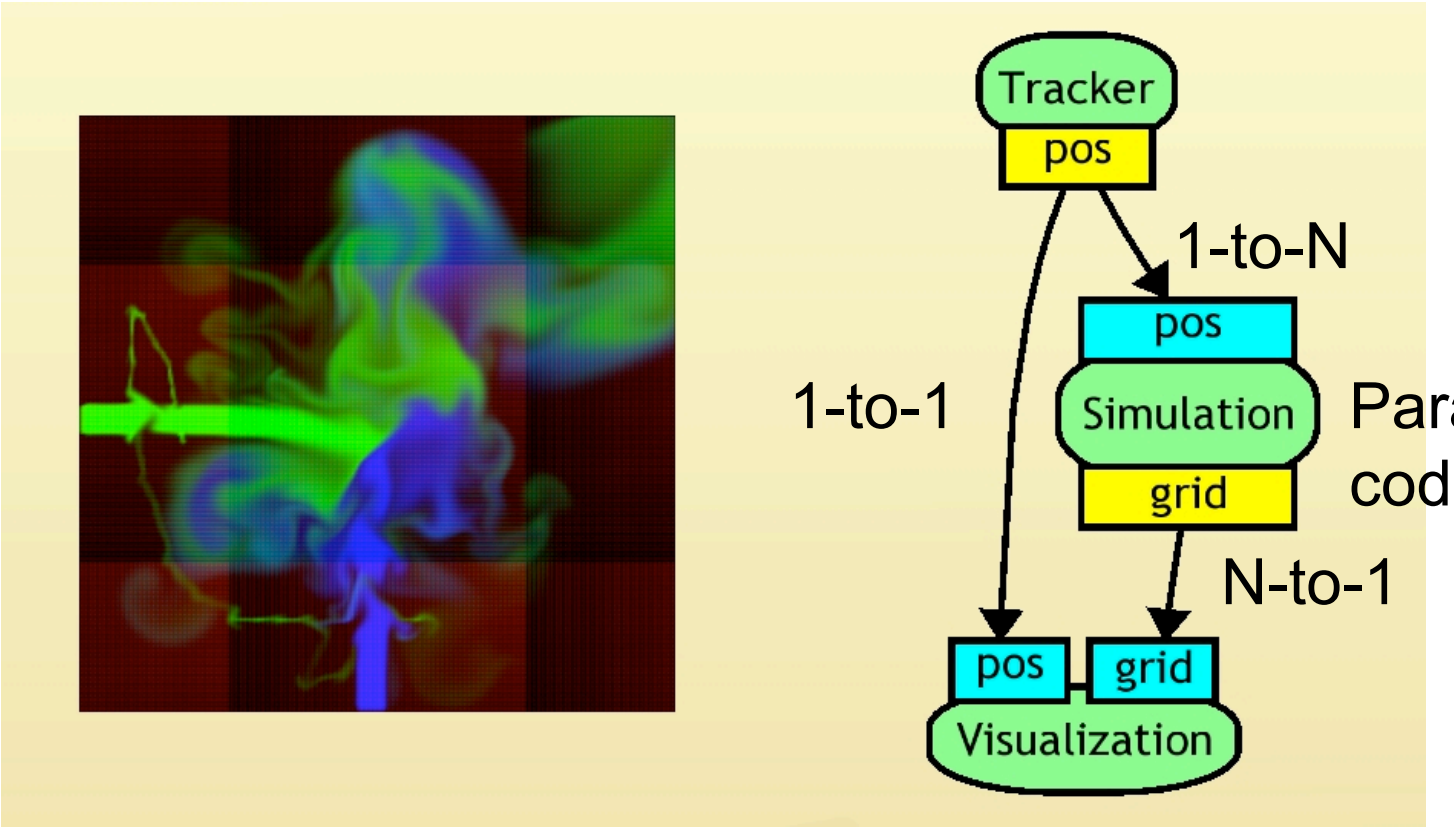




FlowVR
High Performance Virtual Reality

3 programming steps:

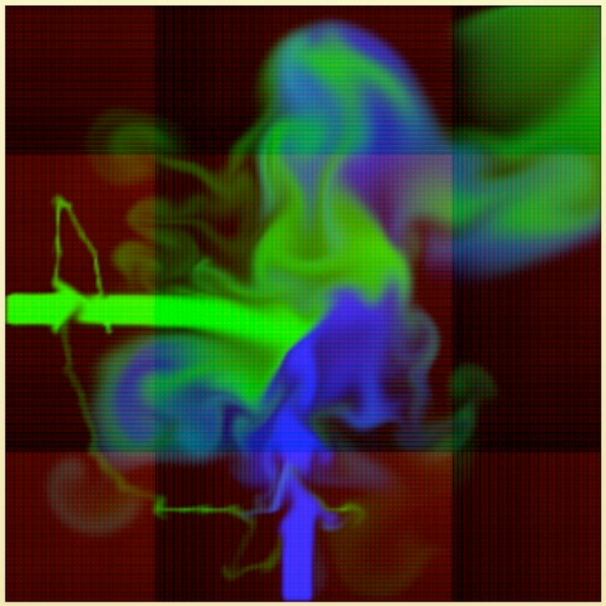
2. Application design (assembly of components)

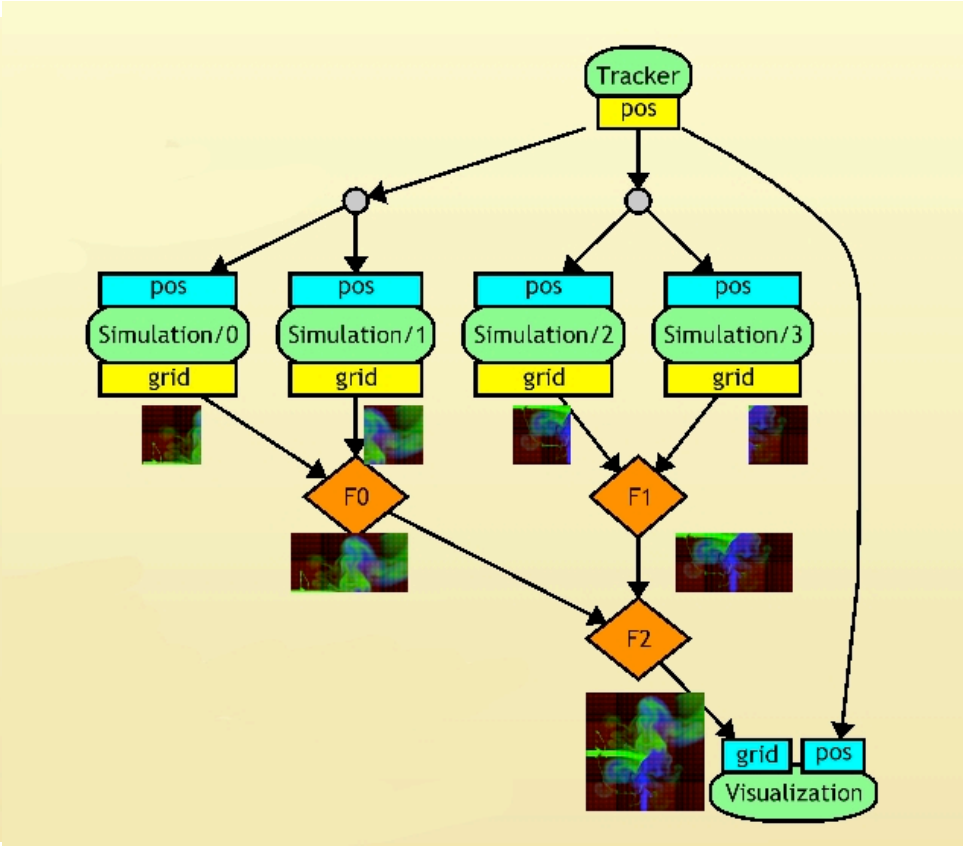


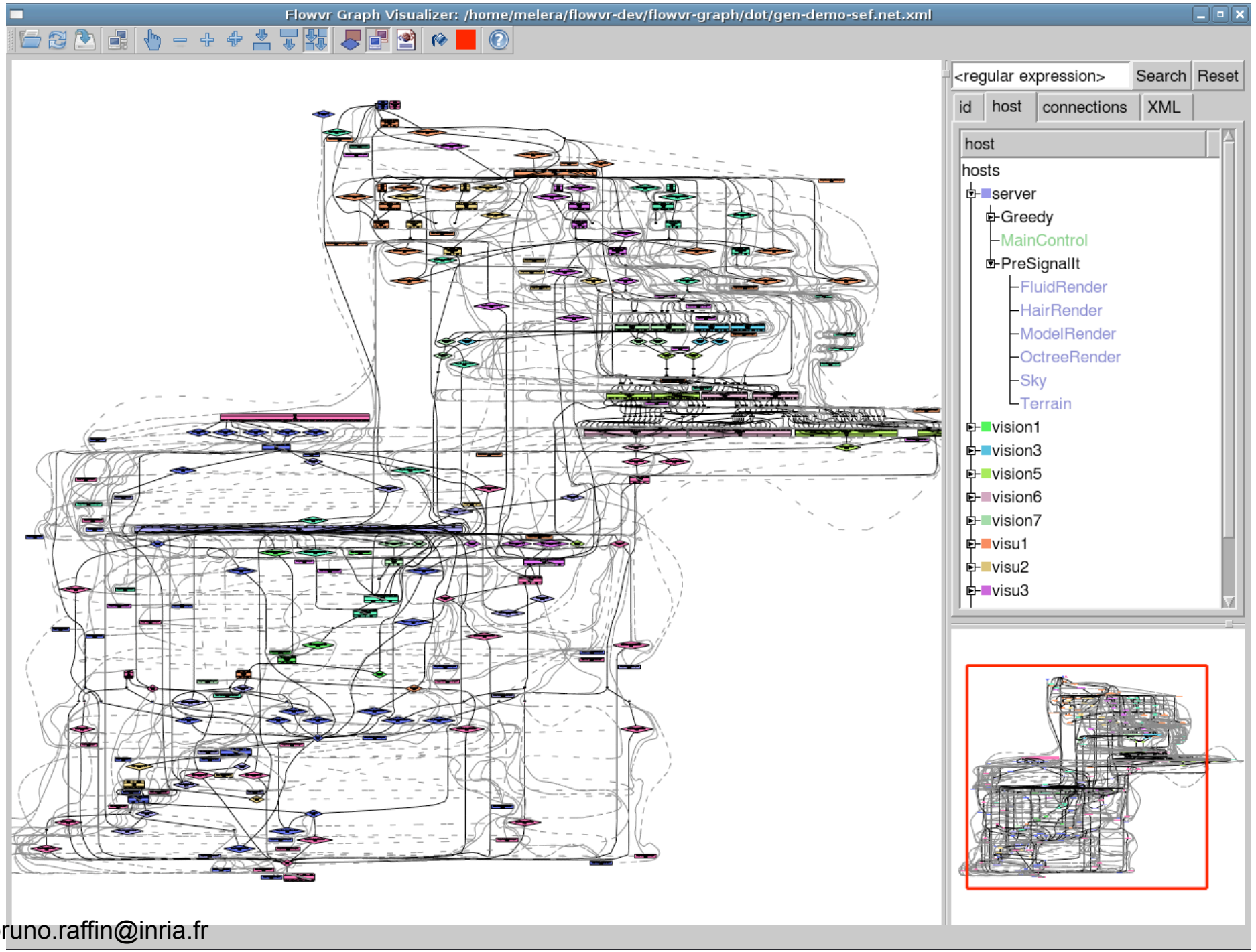
FlowVR
High Performance Virtual Reality

3 programming steps:

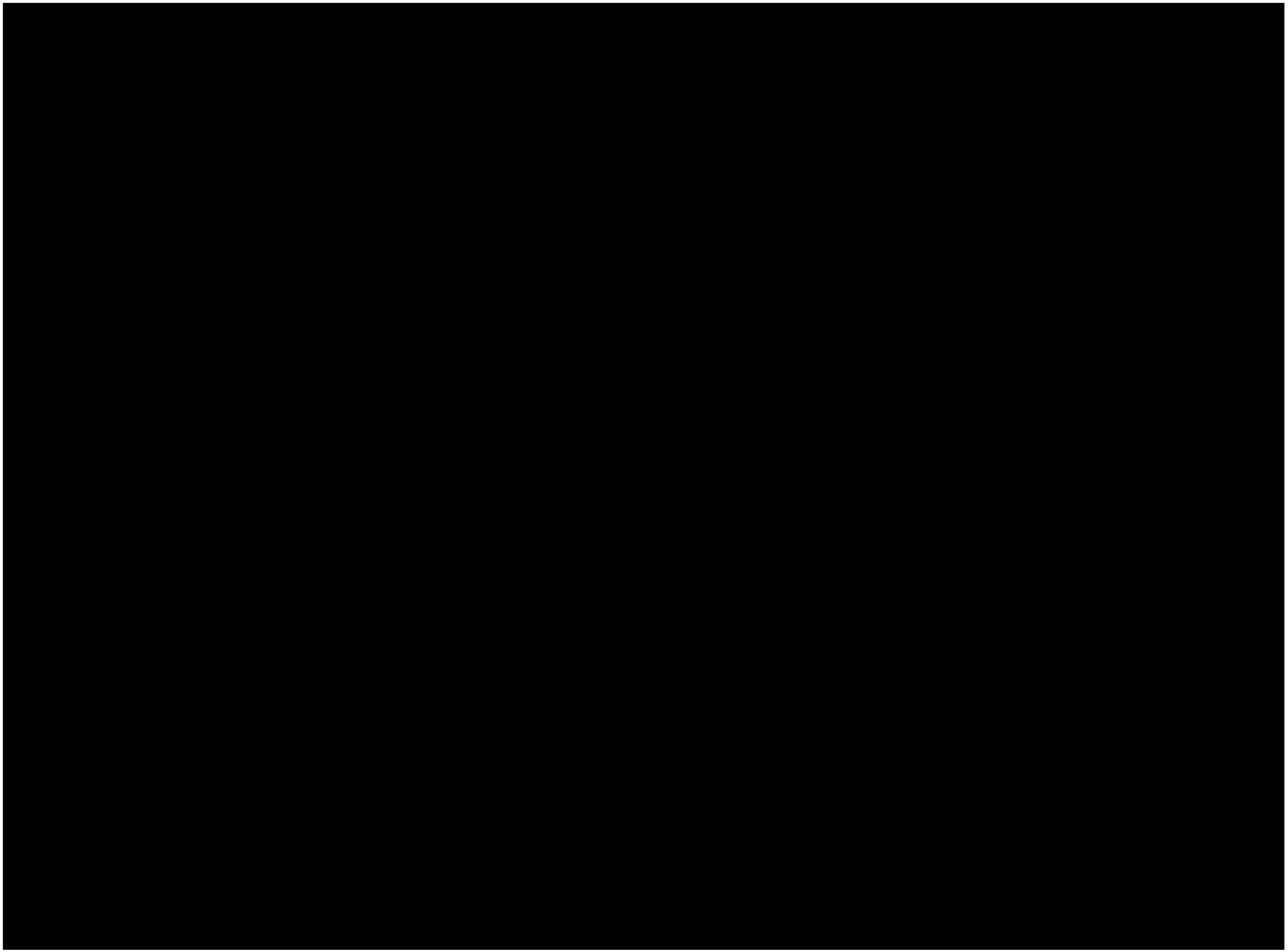
3. Customize according to target machine







bruno.raffin@inria.fr



Presence: what we get

3D volume/surface:

- Instantaneous space occupancy:
  - Correct mixing between real and virtual
  - Collisions (mechanical presence)

Textures: visual presence

Presence: what we don’t get (yet)

3D volume/surface:

- Visual Hull intrinsic limitations:
  - Approximation of real volume: no concavity
  - Visibility related limitations
- Velocity
- Acceleration
- Identification
- Tracking

Textures:

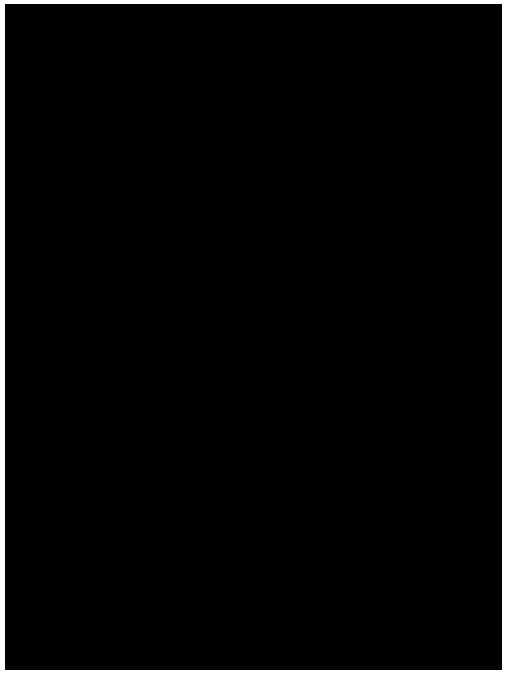
-Missing parts due to visibility occlusions

Presence

Several approaches to explore:

– New pure camera based algorithms
– Combine with data acquired with other devices (Z camera for concavities for instance)

Presence directly affects the interaction capabilities



Tsuyoshi Horo

- Video

Octree carving

Tsuyoshi Horo

Video

DreamWorld

- video

ExMC algorithm, 2009

Immersion: CAVE

Distant displays:

- Pro: stability, user sees its own body
- Cons: user's body block projector beams

3D Modeling: telepresence, mechanical presence

Technical issues: difficult to integrate cameras within a cave

Blue-C

- Video

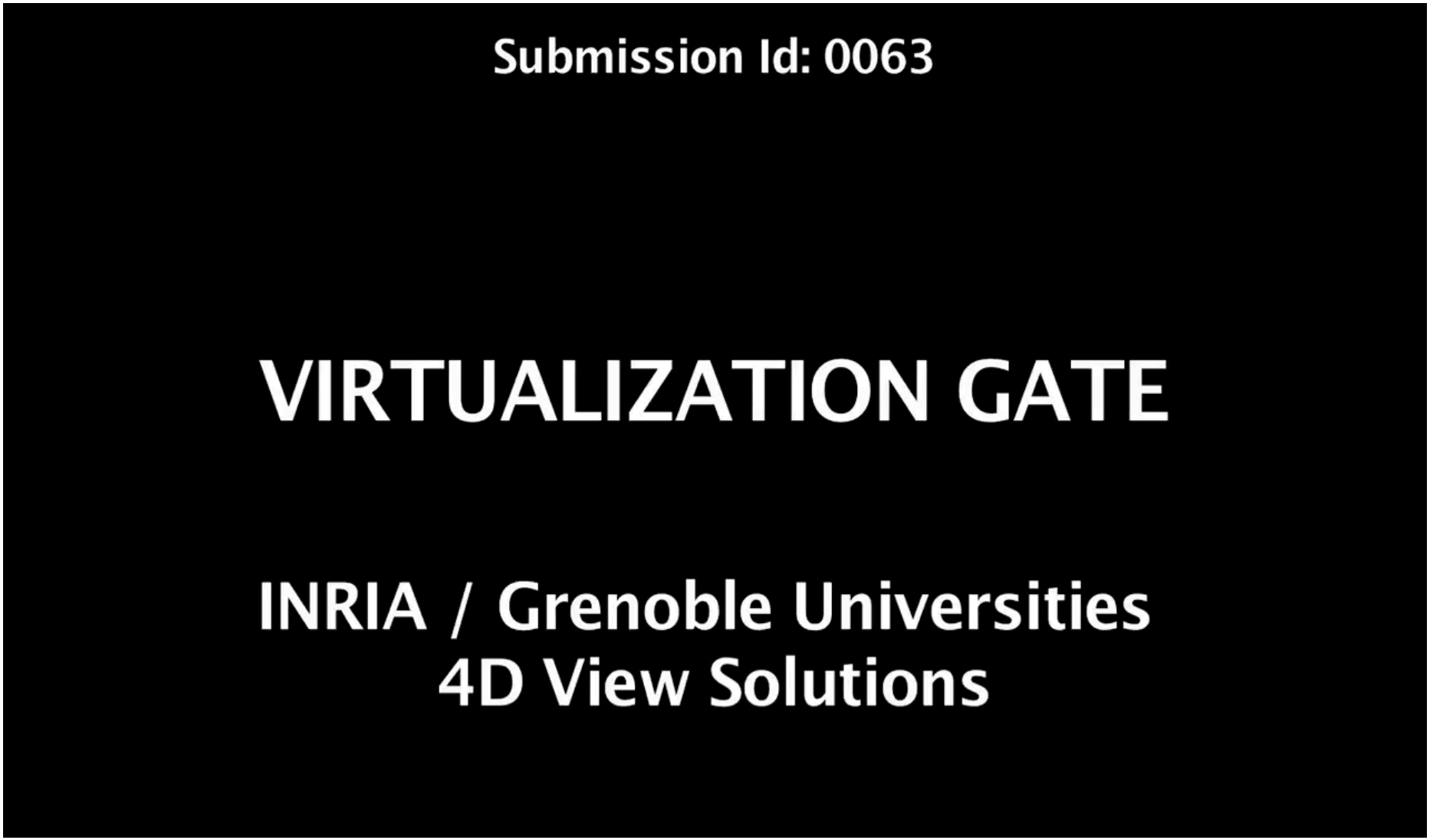
Immersion: HMD

Head mounted display

- Pro: no beam occlusion issues, multi-user
- Cons: immersion quality, stability, user does not see his body

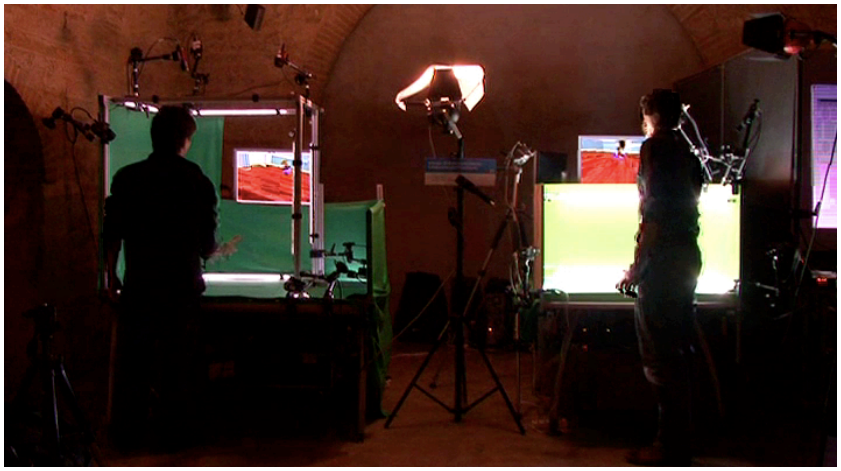
3D Modeling: user sees his modeled body's

No technical issue for integration with cameras



Epvh algorithm - 2009

Conclusion



Multi-cam for VR:

- Marker based tracking
- 3D modeling

3D makes it easier than 2D+depth

Watch computer vision algorithms:

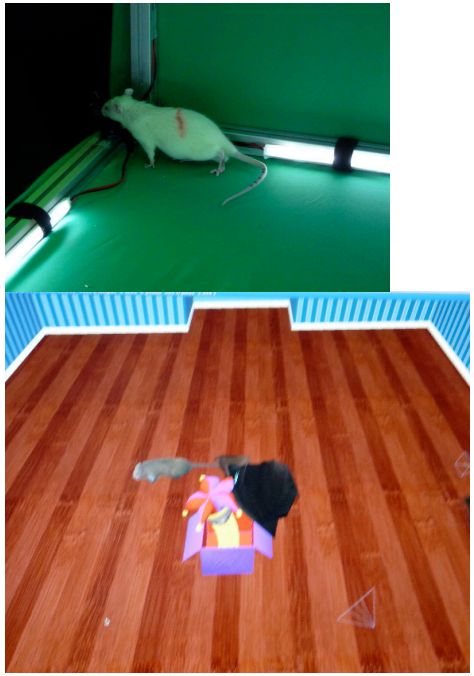
CVPR 2008: 1500 papers submitted (500 in 2000)

Conclusion

Presence: today 3D modeling and marker based tracking, tomorrow …

Interactions:

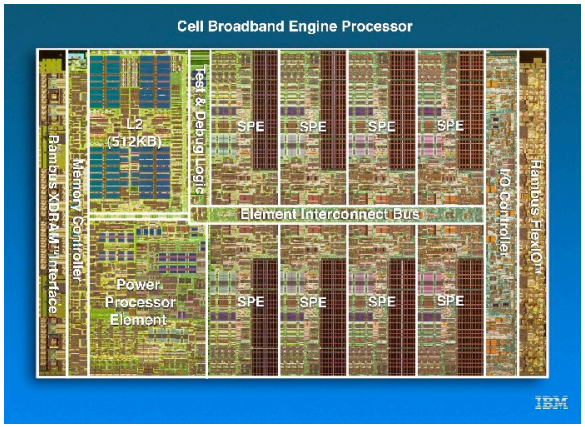
- Physics based interactions
- Symbolic interactions



Immersion: HMD + 3D modeling

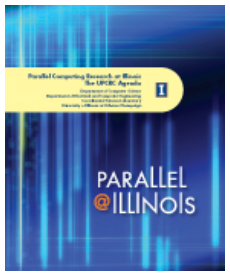
Applications: …

Conclusion



Multi-core killer apps (UPCRC):

- Tele-immersive environments
- Dynamics virtual environments



Be ambitious. Don’t limit yourself because it does not fit in one PC.

Muito Obrigado!

http://grimage.inrialpes.fr

http://flowvr.sf.net

http://www.sofa-framework.org

http://www.4Dviews.com