

Using Time-of-Flight Range Data for Occlusion Handling in Augmented Reality



Jan Fischer¹, Benjamin Huhle², Andreas Schilling²

¹ Island Graphics Group, University of Victoria, Canada

² WSI / GRIS, University of Tübingen, Germany

I P T - E G V E 2 0 0 7

13th Eurographics Symposium on Virtual Environments
10th Immersive Projection Technology Workshop

Motivation

- Context: **Monoscopic video** see-through augmented reality
- Problem: Lack of **depth information**



Video see-through AR



Hand should occlude virtual model!

Related Work



Phantom models [Breen et al. 1996]



Stereo matching [Wloka, Anderson 1995]



Semi-automatic, advanced computer vision
[Lepetit, Berger 2000]



Visual hull computation [Lok et al. 2003]

Outline



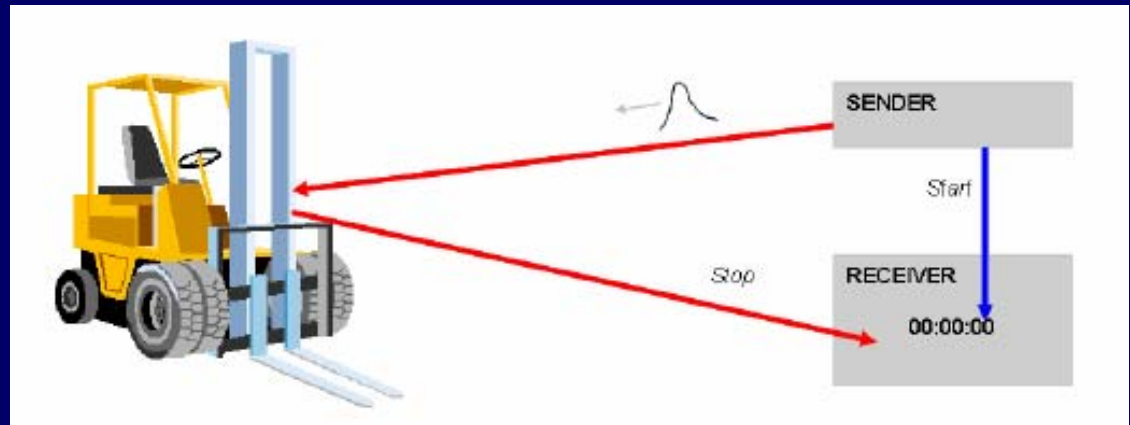
- Introduction
- **Hardware Setup**
- Image Generation Process
- Results
- Conclusions

Time-of-Flight Camera

- Idea: Introduce **dedicated sensor** for object distances
- **Time-of-Flight camera** measures return time of **infrared pulse**



PMD[vision][®] 19k,
PMDTechnologies GmbH



Time-of-Flight principle [Moeller et al., 2005]

- Technology has recently become **commercially available**

Alternative Technologies

- Other similar products exist, e.g., **3DV's Z-Cam**



Z-Cam™ (www.3dvsystems.com)



Depth Keying [Gvili et al. 2003]



- No known application to (real-time, registered) **augmented reality**

Hardware Setup

- Experimental setup
 - PMD **depth sensor** (160x120)
 - High-resolution **color camera** (Matrix Vision BlueFox)
- Similar **fields of view** (ca. 30°)
- Fixed assembly



Color image



Depth image

Outline



- Introduction
- Hardware Setup
- **Image Generation Process**
- Results
- Conclusions

Process Overview

Preprocess:

Camera
calibration / image
registration

Per frame:

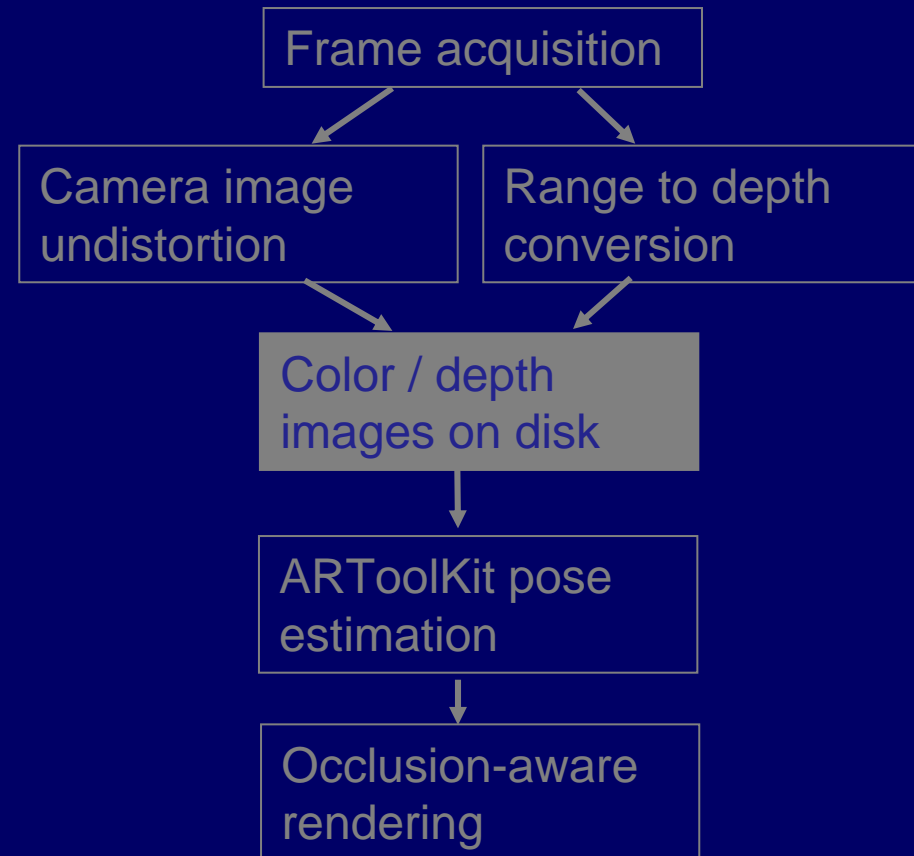


Image Registration

- Internal and external **camera parameters** are calculated
- Low-resolution **intensity image** is delivered by PMD camera



Low-resolution PMD intensity image

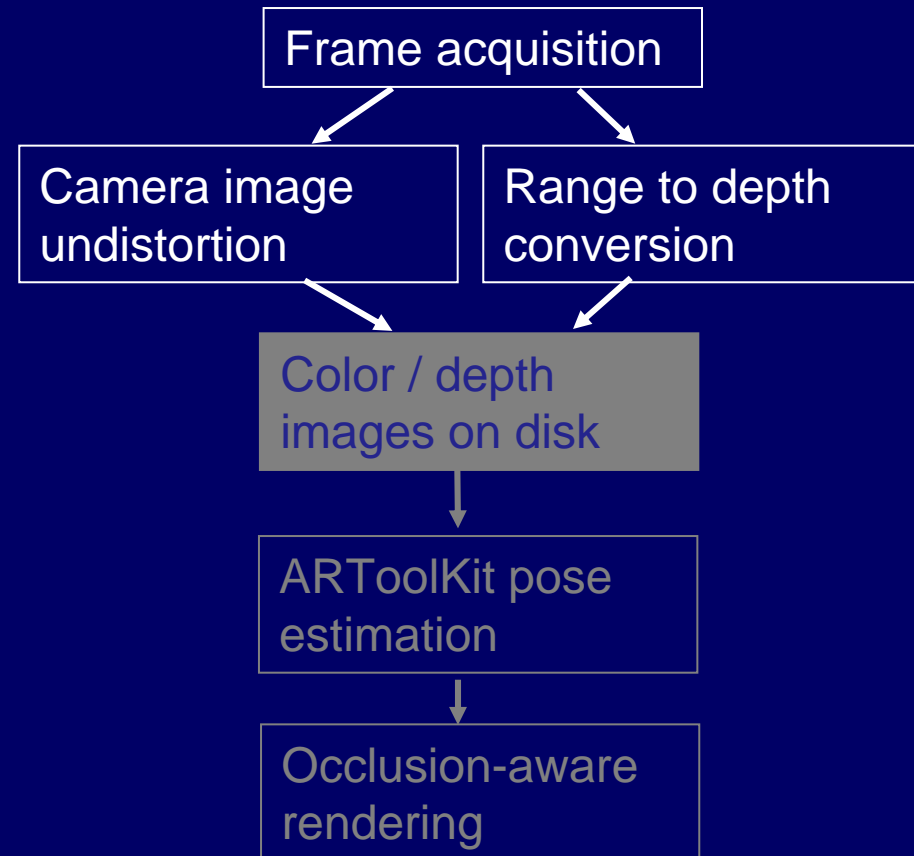
- **Camera Calibration Toolbox** for Matlab is used
[by J.-Y. Bouguet]

Process Overview

Preprocess:

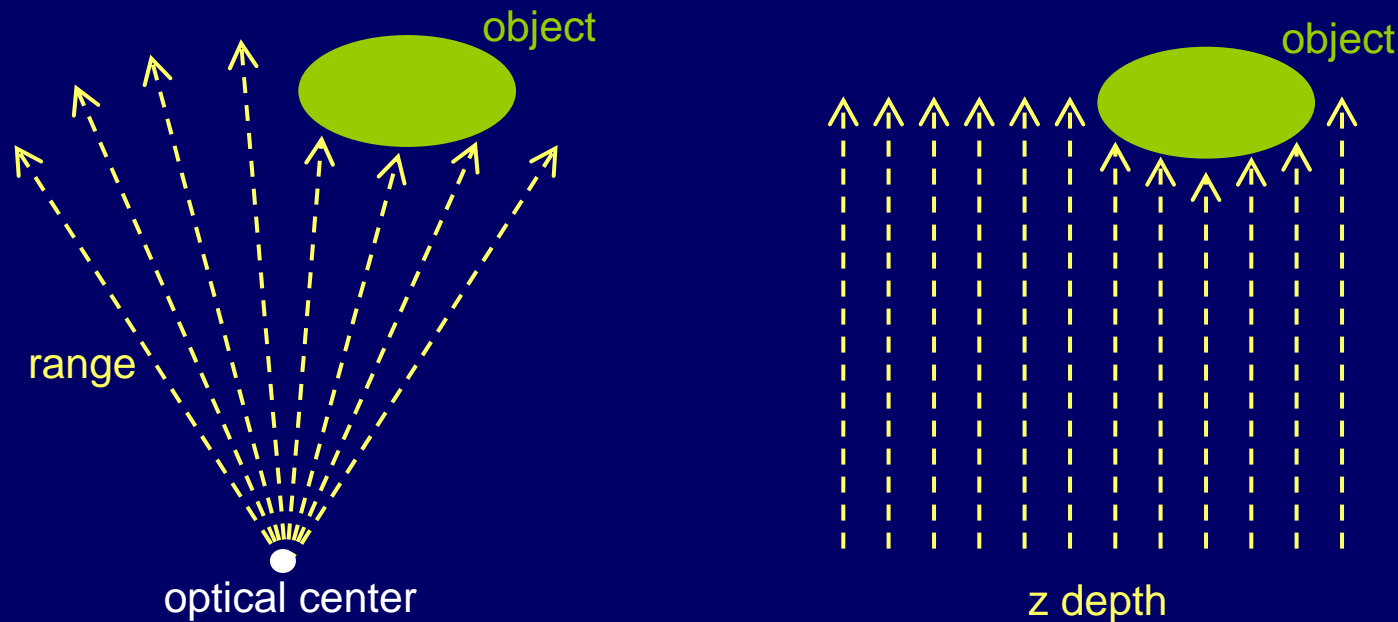
Camera
calibration / image
registration

Per frame:



Range to Depth Conversion

- PMD sensor delivers **range**, i.e., distance to **optical center**



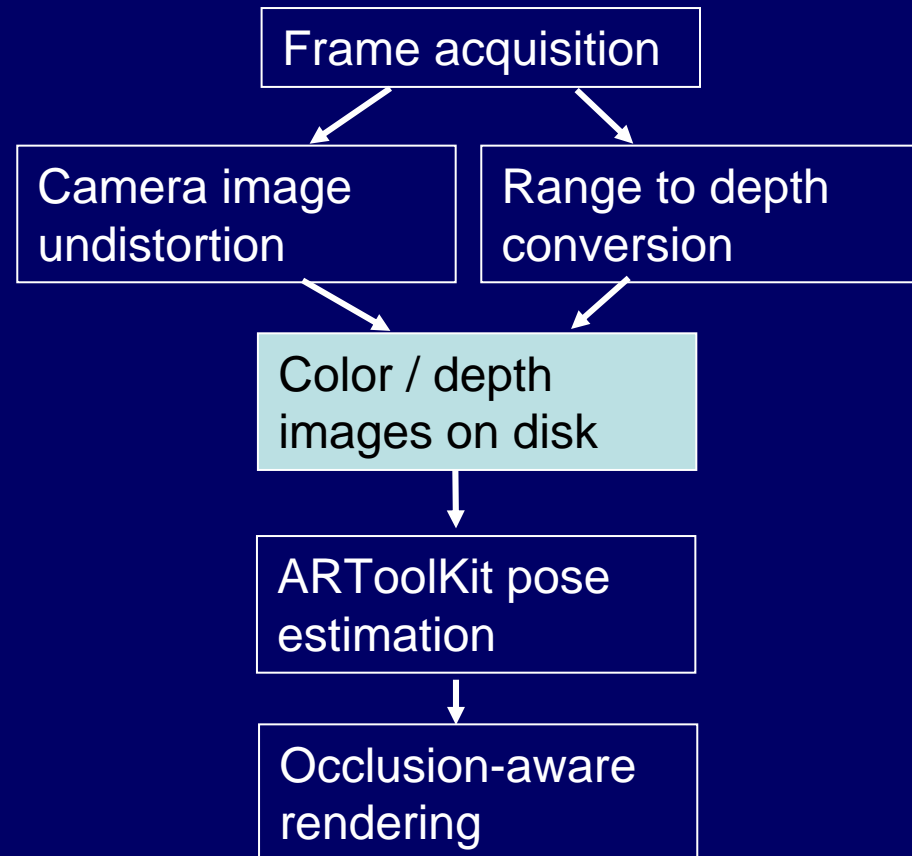
- Conversion to Cartesian **z depth** using standard **projective geometry**

Process Overview

Preprocess:

Camera
calibration / image
registration

Per frame:



Occlusion Handling



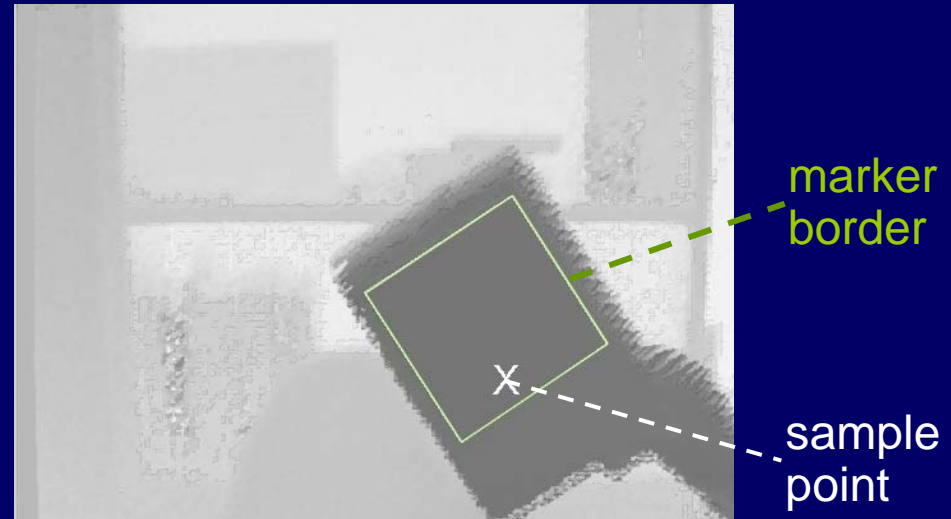
- Prototype system with **offline image generation** (images stored on disk)
- ARToolkit for **marker tracking** in color image
- Depths calibrated to **millimeters in real-world** coordinates, both:
 - Computed time-of-flight depth maps
 - ARToolkit tracking

Depth Comparison Example

Color camera image



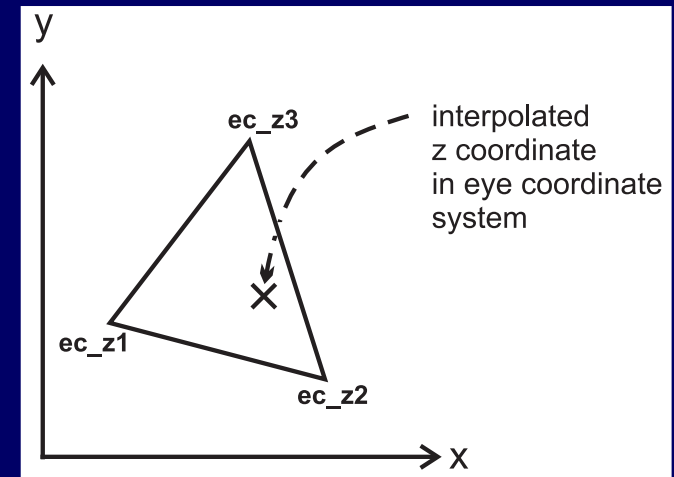
Time-of-flight depth map



- **ARToolKit** depth: 1156 mm
- **Time-of-Flight** depth: 1176 mm
- Deviation 2 cm, ca. **2%**
- **Similar accuracies** can typically be achieved in **well calibrated** setups

Modified Rendering Pipeline

- Specialized **shader program** is used
- Time-of-Flight depth map supplied as **texture**
- Real-world object depth computed in **eye coordinates**
- **Interpolation** over primitive area (**varying variable**)
- **Comparison** between **texel value** and interpolated **eye coordinate depth** in shader

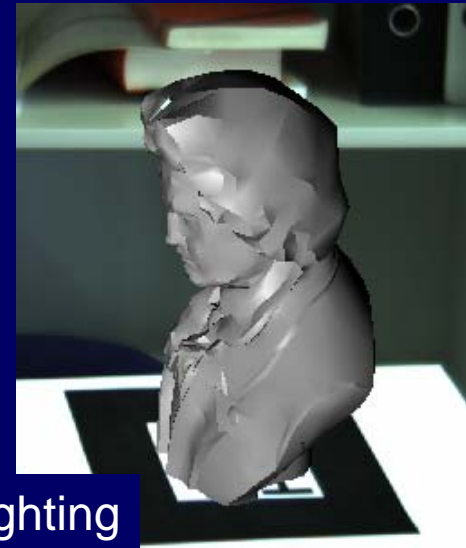


Implementation Details (1)

- Prototype system uses images **stored on disk**
- Images **already undistorted** based on the Camera Calibration Toolbox
- **Disable** ARToolKit undistortion:
`dist_factor = {0, 0, 0, 1}`
- Use computed **calibration matrix** as ARToolKit parameter `mat[]`

Implementation Details (2)

- **Depth test** in in shader
 - Conditional branching with `if(...)`
 - Fail: call to `discard;`
- **Diffuse lighting** computation added to shader



(Monochrome) diffuse lighting

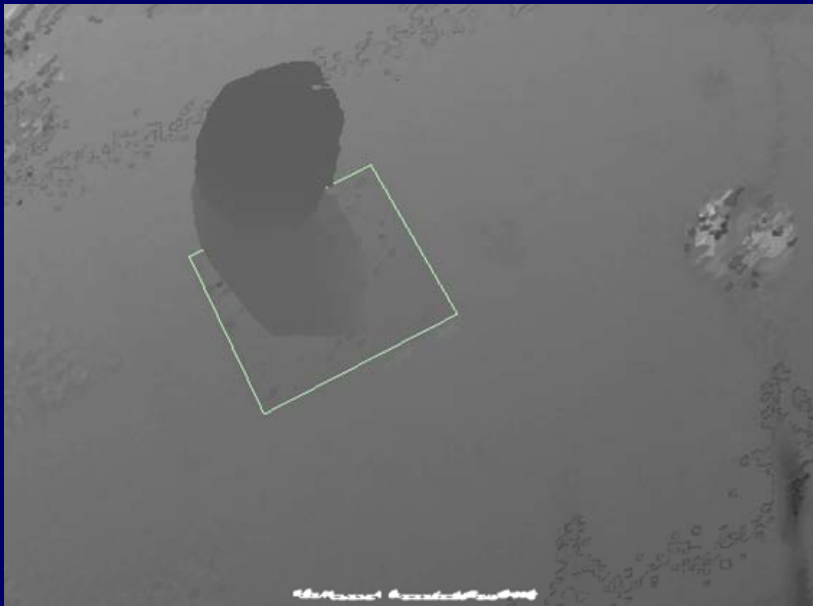
Outline



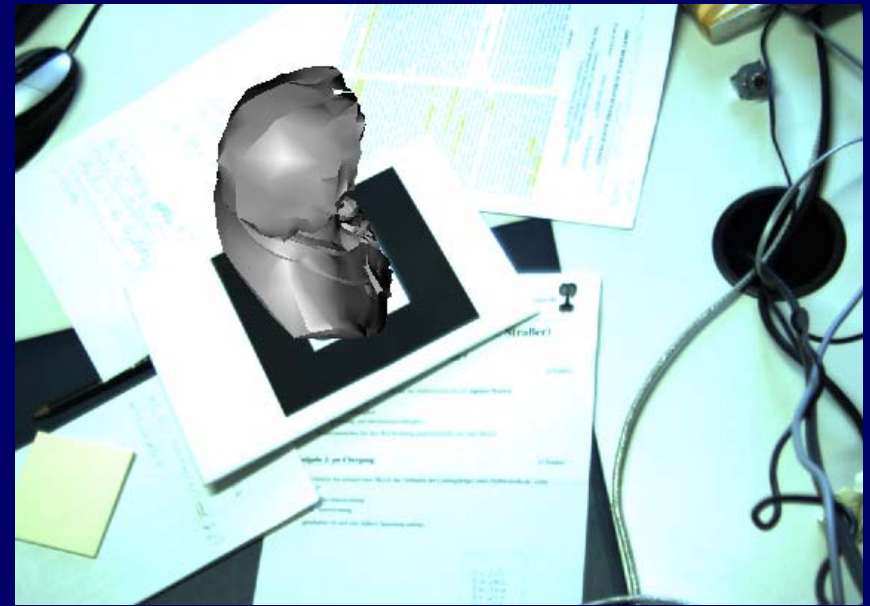
- Introduction
- Hardware Setup
- Image Generation Process
- **Results**
- Conclusions

Results (1)

- Image resolution **640x480** pixels

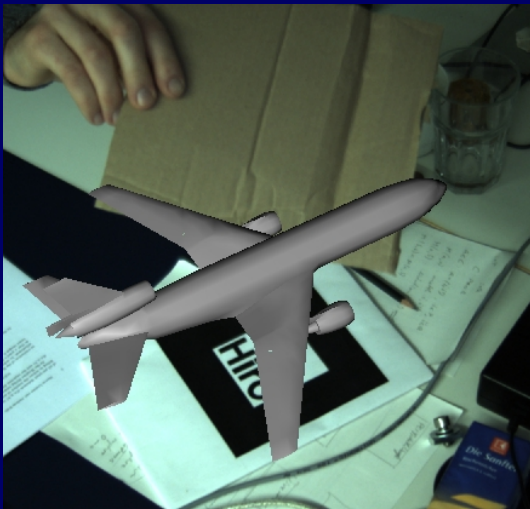


Depth (real and virtual objects)

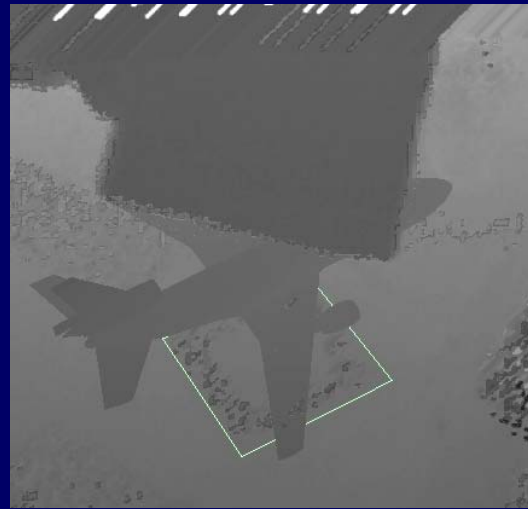


Augmented image with
depth test

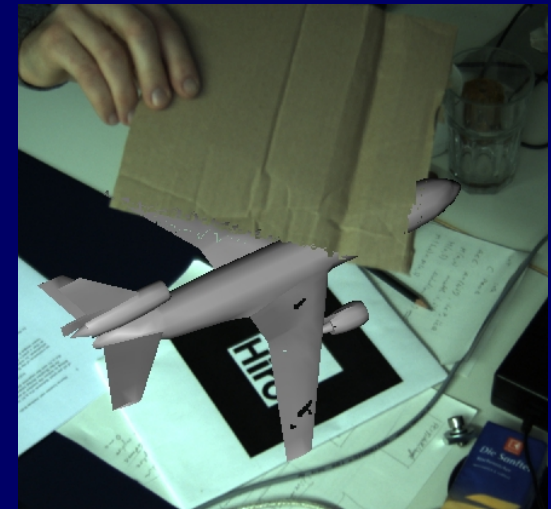
Results (2)



Without occlusion
handling

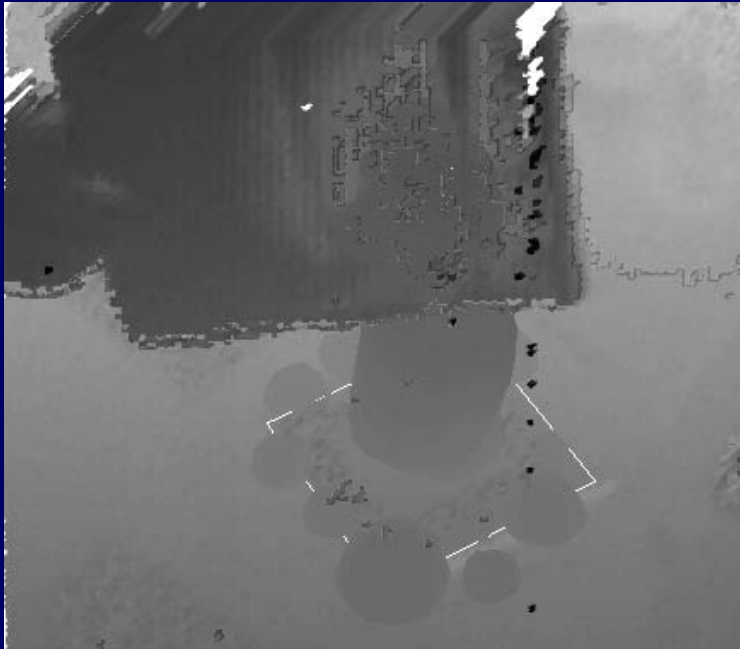


Depth (real and virtual
objects)

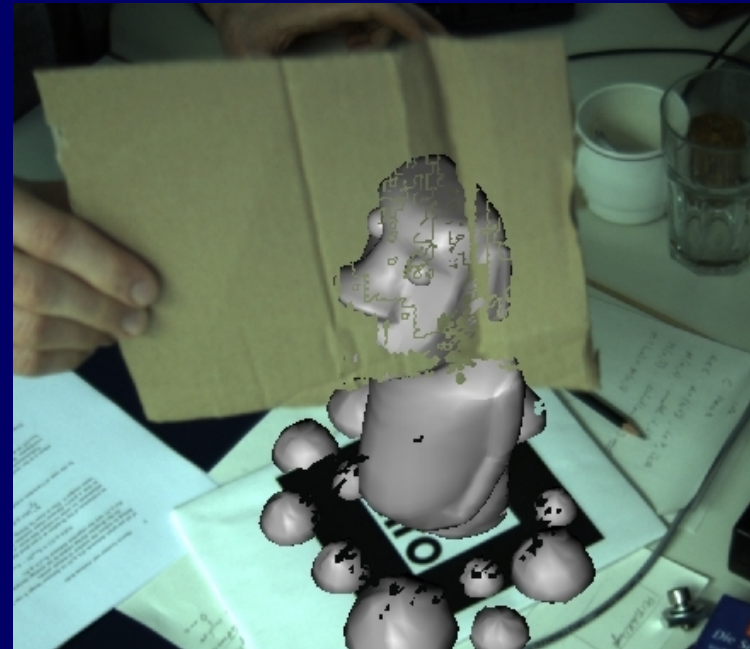


Augmented image with
occlusion handling

Results (3)

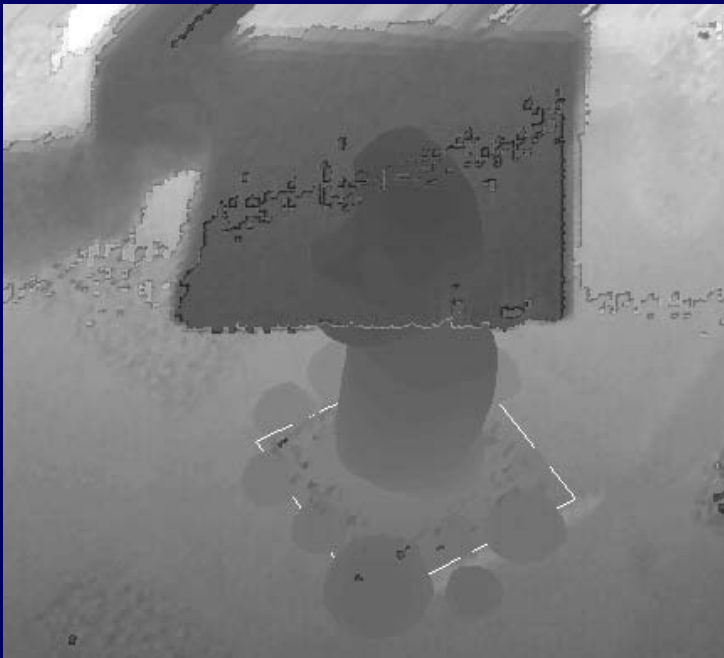


Depth (real and virtual objects)

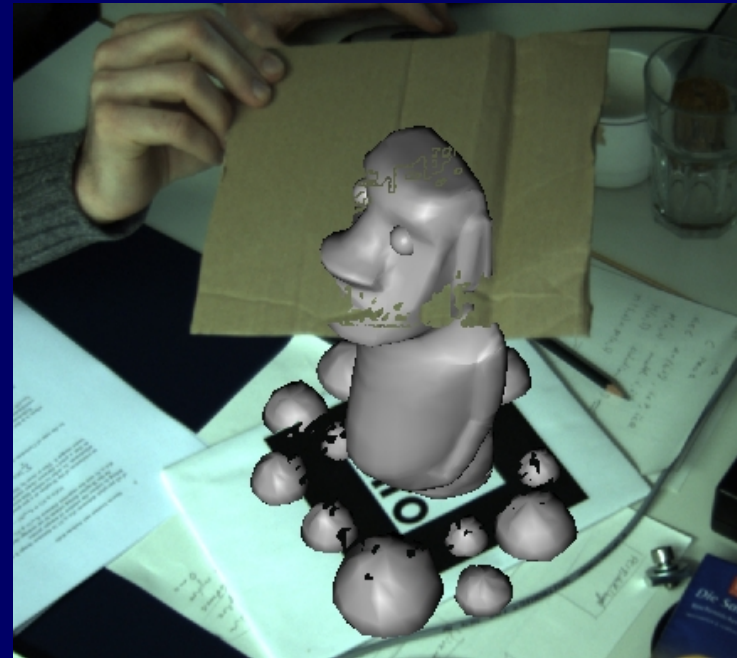


Augmented image with
occlusion handling

Results (4)



Depth (real and virtual objects)



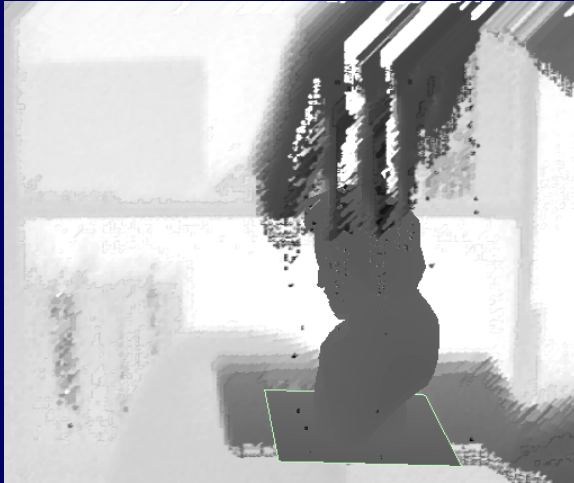
Augmented image with
occlusion handling

Summary

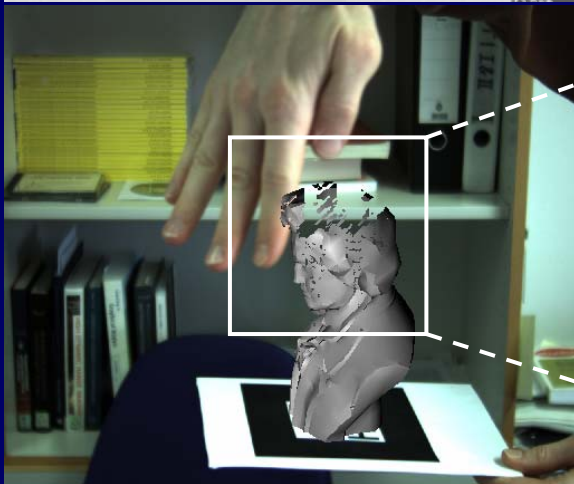


- Usage of **Time-of-Flight sensor** in AR
- **Registration process** for color and range images
- **Occlusion handling** based on Time-of-Flight depth
- Identified future **challenges**

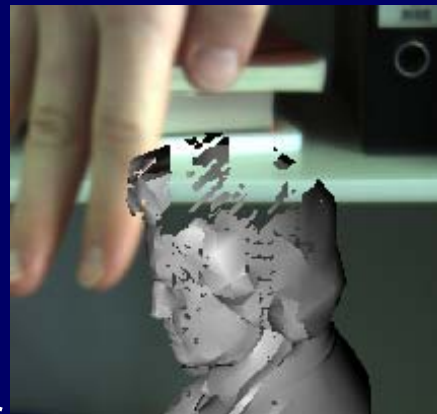
Challenges (1)



- Color and range image acquisition currently **not synchronized**



Frame #61



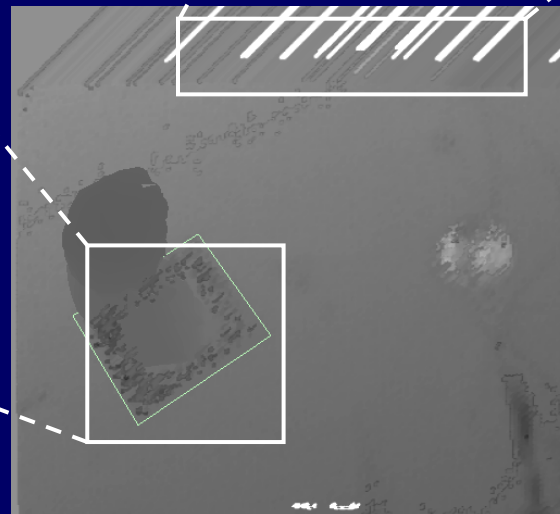
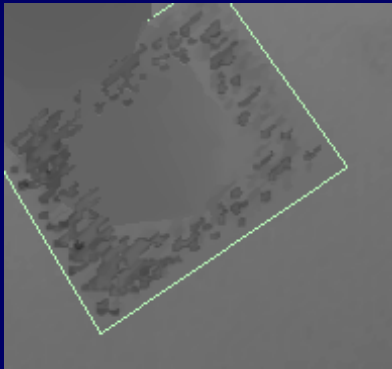
Extreme case of mismatch

Challenges (2)

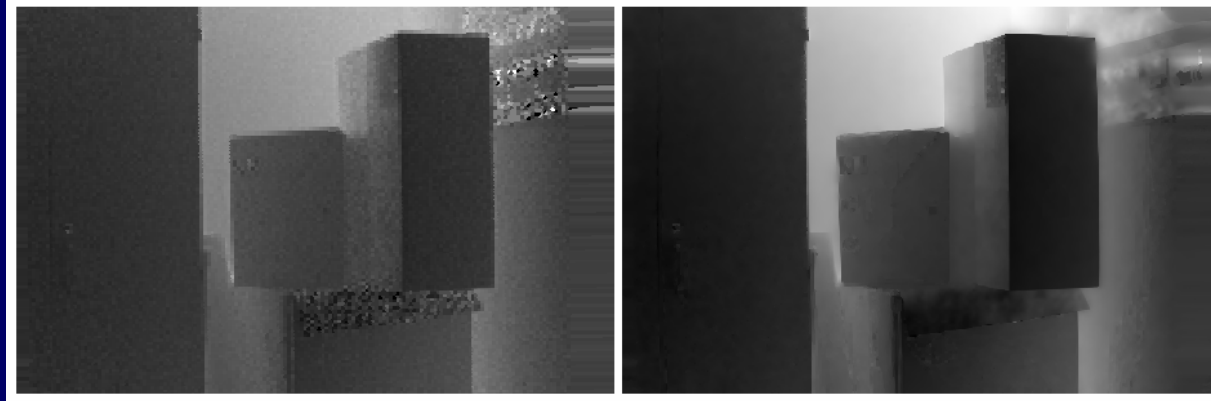
Interpolation **artifacts** /
missing depth data



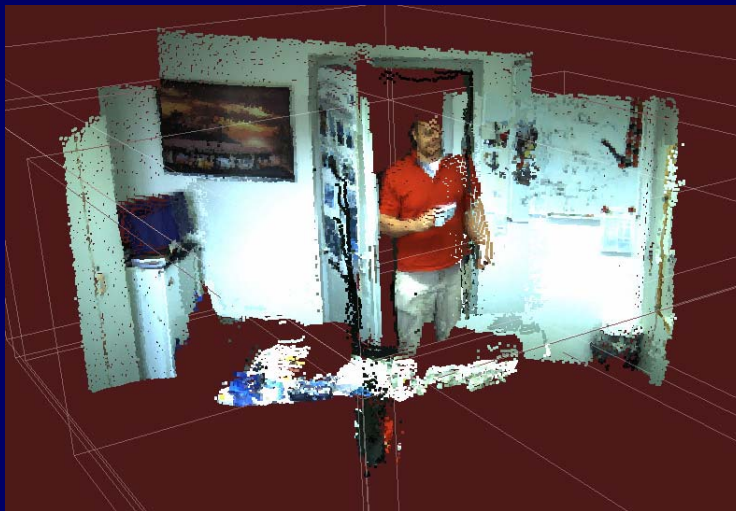
Noise / errors in data (e.g., total
reflection or oversaturation)



Ongoing Work



Improved **smoothing** / interpolation of depth map [Huhle et al. 2007]



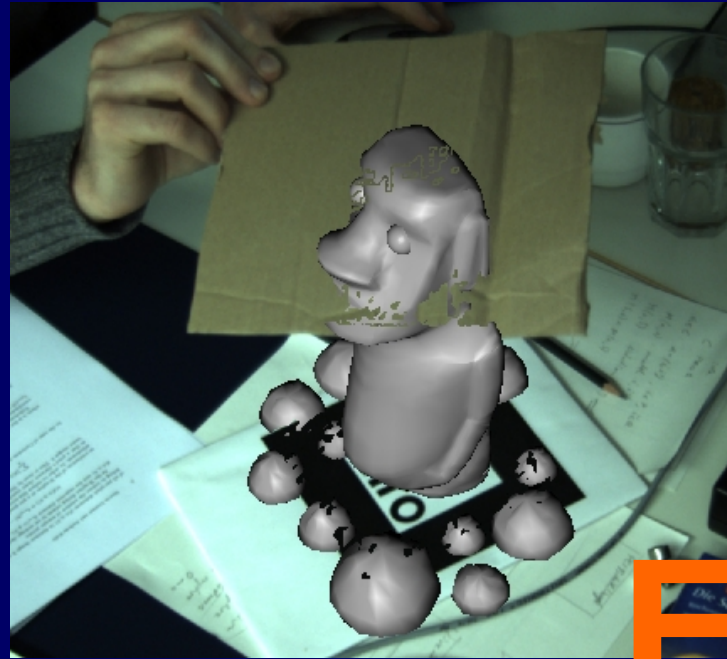
Three-dimensional registration /
scene **acquisition**

Future Work



- Integration into **real-time system**
- Integration of **advanced computer vision** algorithms
- Application to other AR rendering problems (e.g., **shadow casting**)
- (Hopefully) **lighter / smaller** depth sensor for mobile applications

END



END

Acknowledgments: Deutsche
Forschungsgemeinschaft
DFG