

Detecting Dynamic Occlusion in front of Static Backgrounds for AR Scenes

WSI
GRIS



Jan Fischer¹, Holger Regenbrecht², Gregory Baratoff²

¹ WSI / GRIS, Universität Tübingen

² Virtual Reality Competence Center, DaimlerChrysler Ulm

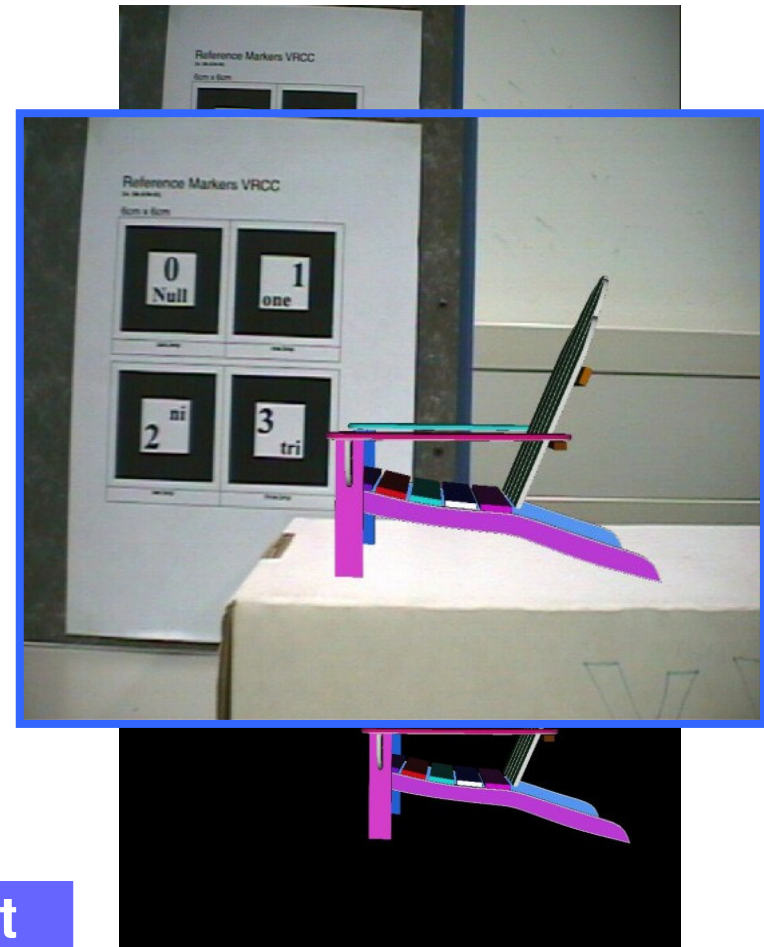


- Introduction and problem specification
- Related work
- The suggested approach
- Implementation of the algorithm
- (Preliminary) results
- Conclusion



23.05.2003

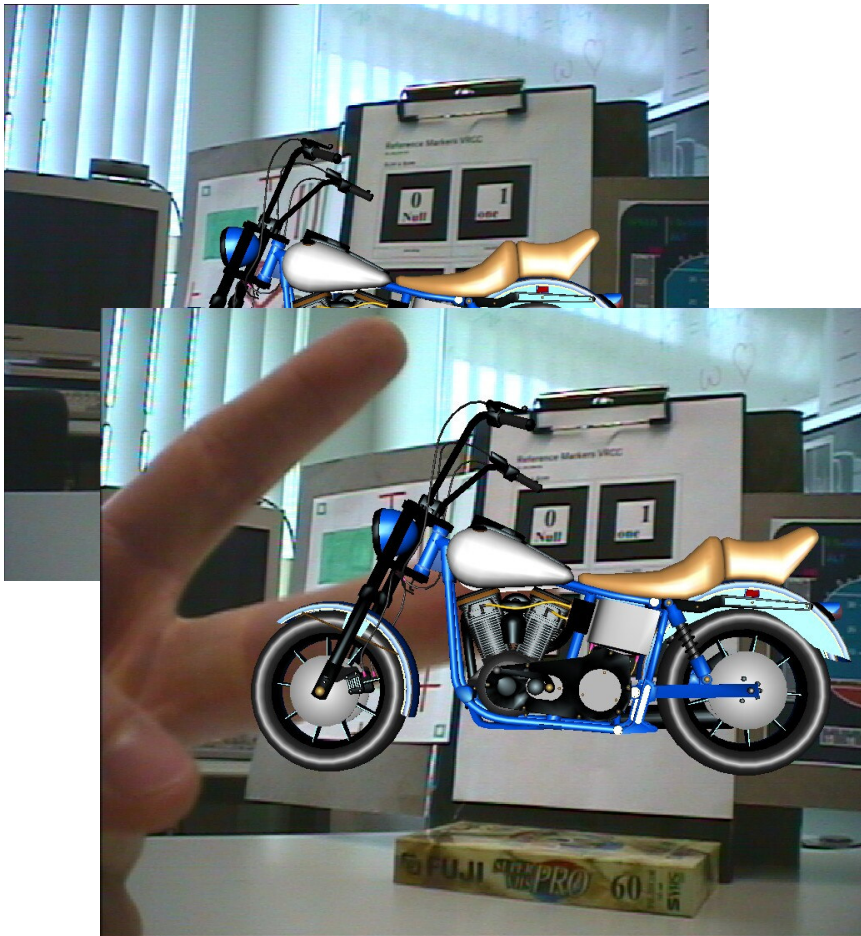
- Real camera images are combined with virtual graphical objects
- A tracking system determines the current pose (position + orientation) of the viewer
- Correctly transformed graphical objects are drawn over current camera image
- Resulting combination is displayed in HMD worn by user



Advantage: Easy to implement and fast



23.05.2003

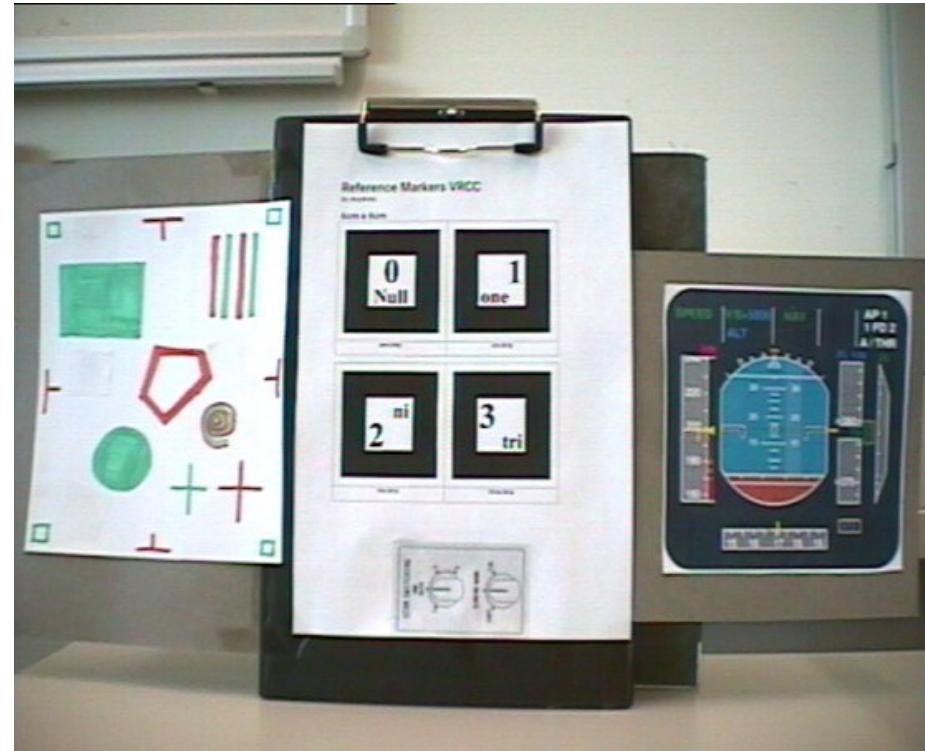


- This simple approach cannot handle occlusion between different types of objects in the scene
- Our area of interest: graphical objects occluded by real objects

Real objects always appear occluded by virtual ones, regardless of their actual spatial relationship



- Detect occlusion of graphical objects by real ones
- The occlusion may be dynamic (nothing is known about real occluder)
- **Detection only in front of previously defined textured planes in the real scene**
- „Dynamic occlusion backgrounds“ (DOBs)



⇒ Typical scenario: Manual interaction with virtual cockpit instruments

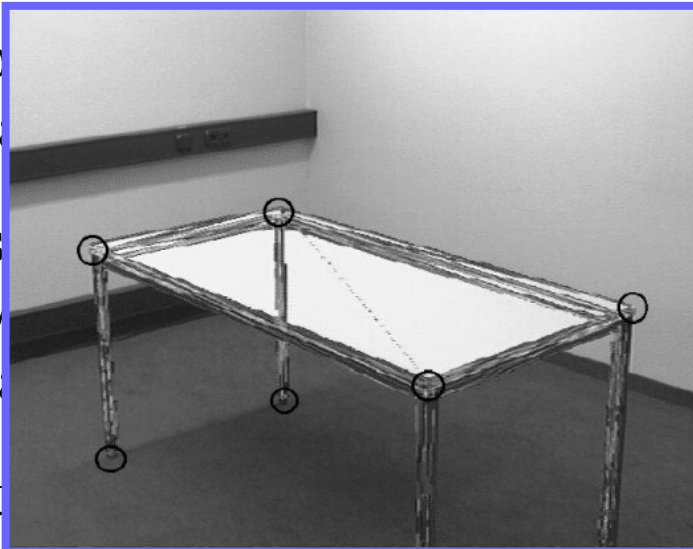


- Static occlusion with phantom models [Breen et. al 1996]

- Co
et

- Us
[W
et

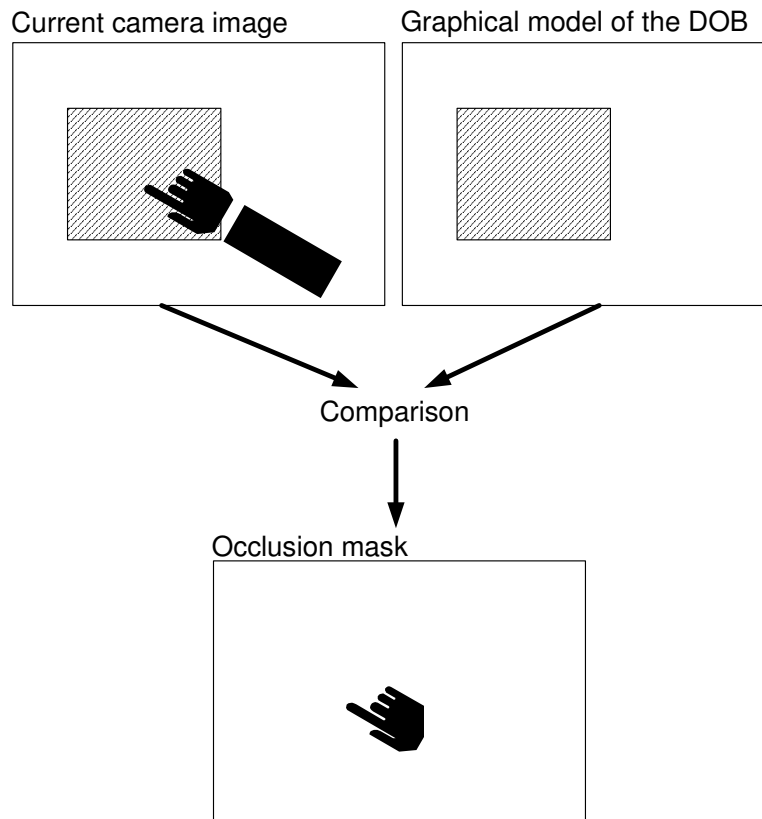
- Int
et al. 2002]



- Image-based tracking system [Kato et al. 2002]

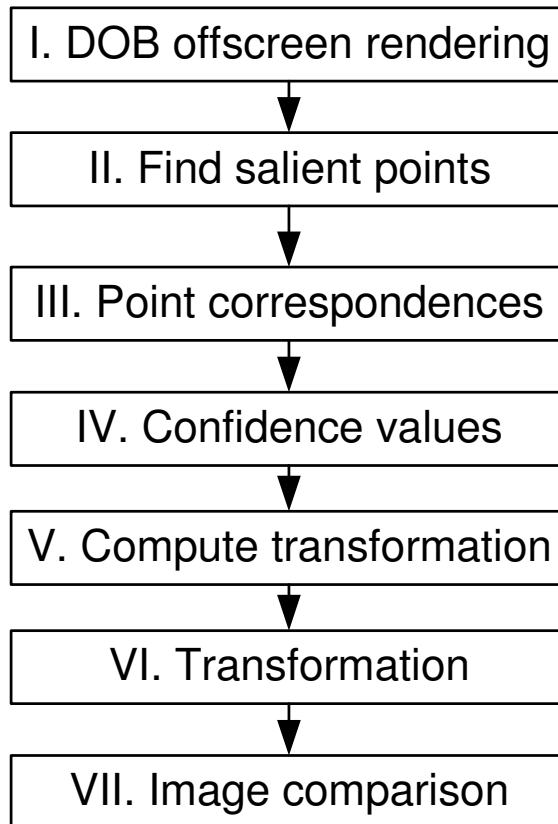


- Introduction and problem specification
- Related work
- The suggested approach
- Implementation of the algorithm
- (Preliminary) results
- Conclusion

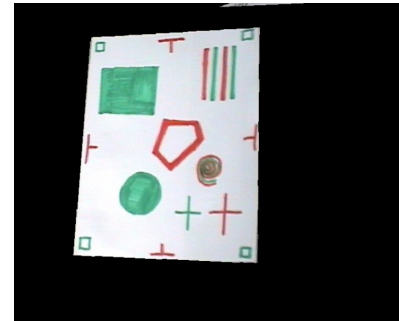


1. Draw DOB into offscreen buffer
2. Correct DOB image according to marker tracking inaccuracies
3. Compare corrected DOB and camera image

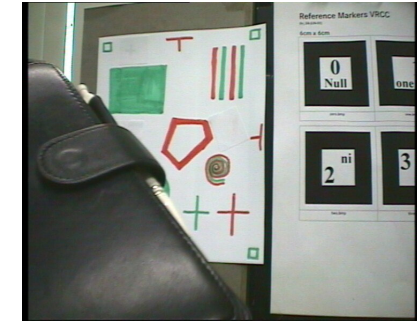
- The result is a binary occlusion mask
- Step 2 consists of several sections and is the most expensive part of the algorithm



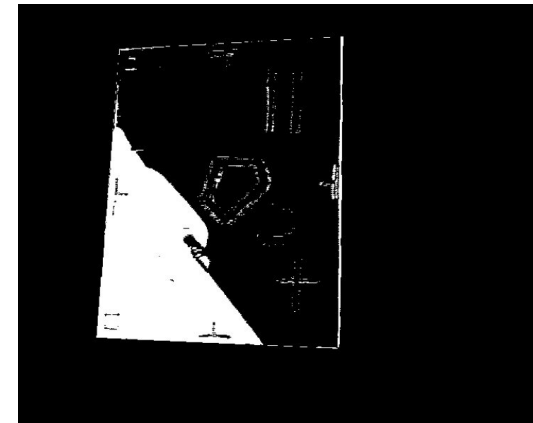
(Correctly transformed) DOB



Current camera image



Resulting occlusion mask

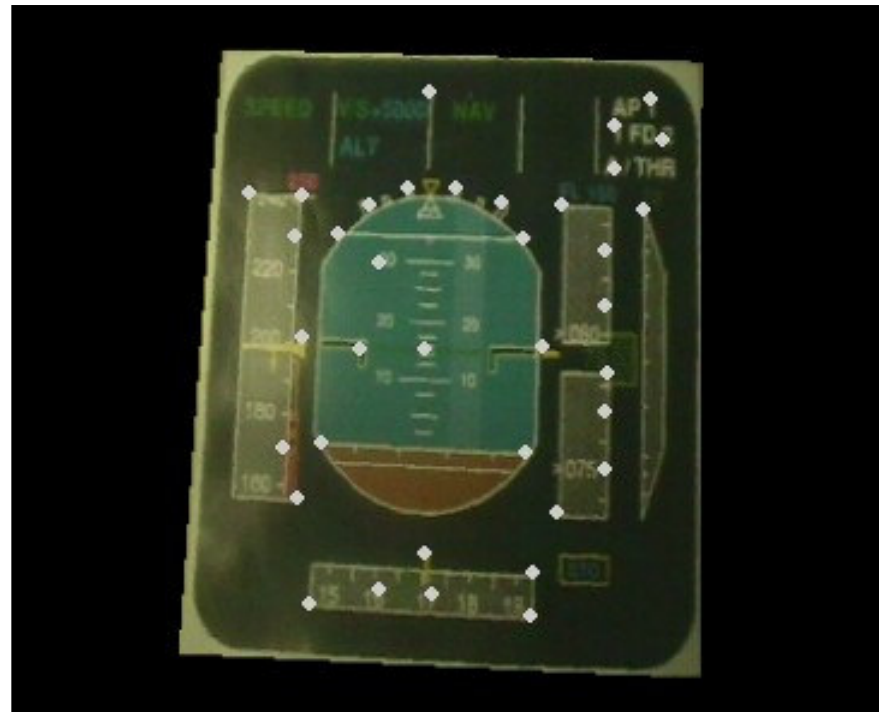




- Introduction and problem specification
- Related work
- The suggested approach
- Implementation of the algorithm
- (Preliminary) results
- Conclusion



- First the DOB is rendered into an offscreen buffer using the marker tracking process
- The structure features in the image are rendered into a 2D image



vo-dimensional



- Candidates for salient points are filtered according to a specific set of rules



- Places corresponding points in the camera images
- Normalized cross-correlation differences for



searched in

relation of

- The corresponding coefficient

maximum

- Detected maximum coefficient defined as confidence

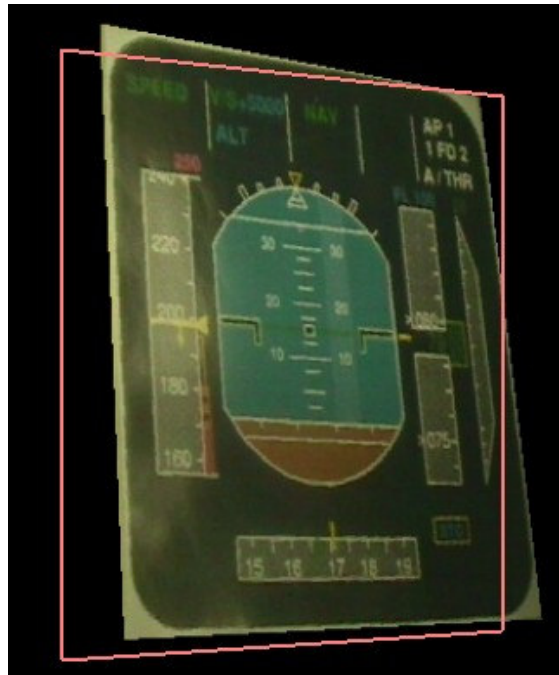


- Only correspondences with a certain minimum confidence are used

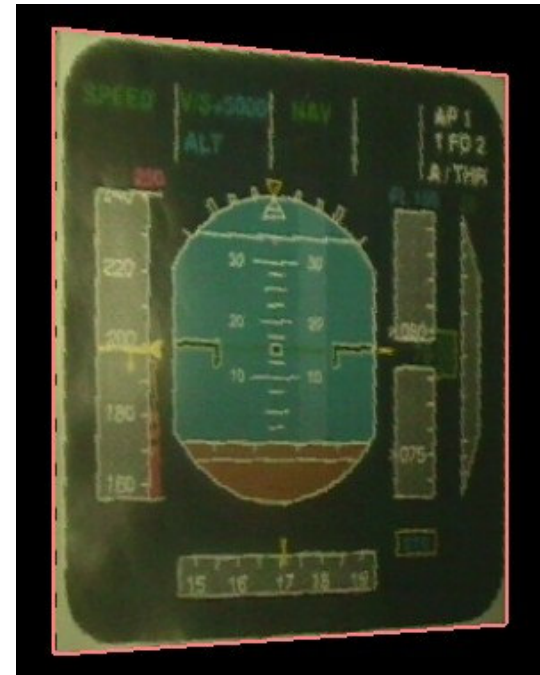
$$\tilde{q}_i = A \cdot \tilde{p}_i \quad \tilde{q}_i, \tilde{p}_i \in \mathbb{R}^3 \quad A \in \mathbb{R}^{3 \times 3} \quad i \in \{1, \dots, l\}$$

$$\tilde{q}_i = \begin{pmatrix} x_{q_i} \\ y_{q_i} \\ 1 \end{pmatrix} \quad \tilde{p}_i = \begin{pmatrix} x_{p_i} \\ y_{p_i} \\ 1 \end{pmatrix} \quad A = \begin{bmatrix} \frac{a_1^T}{a_2^T} \\ \frac{a_2^T}{a_3^T} \end{bmatrix}$$

- Construct equation system from point correspondences
- At least 5 valid point correspondences are required. Equation system is over-determined
- Find best solution using singular value decomposition



Original internal DOB based
on marker tracking pose



Internal DOB image corrected
according to computed trans-
formation



- Pixelwise comparison of corrected DOB and camera images
- Several different criteria for the image comparison were tested
- Best results were achieved with the "adaptive HSV" criterion

$$\alpha(x, y) = \beta \cdot \min(V_i(x, y), V_w(x, y))$$

$$o(x, y) = \alpha(x, y) \cdot \frac{\Delta H(x, y) + \Delta S(x, y)}{2} + (1 - \alpha(x, y)) \cdot \Delta V(x, y)$$

$$occ(x, y) = \begin{cases} 1, & o(x, y) > threshold \\ 0 & \end{cases}$$



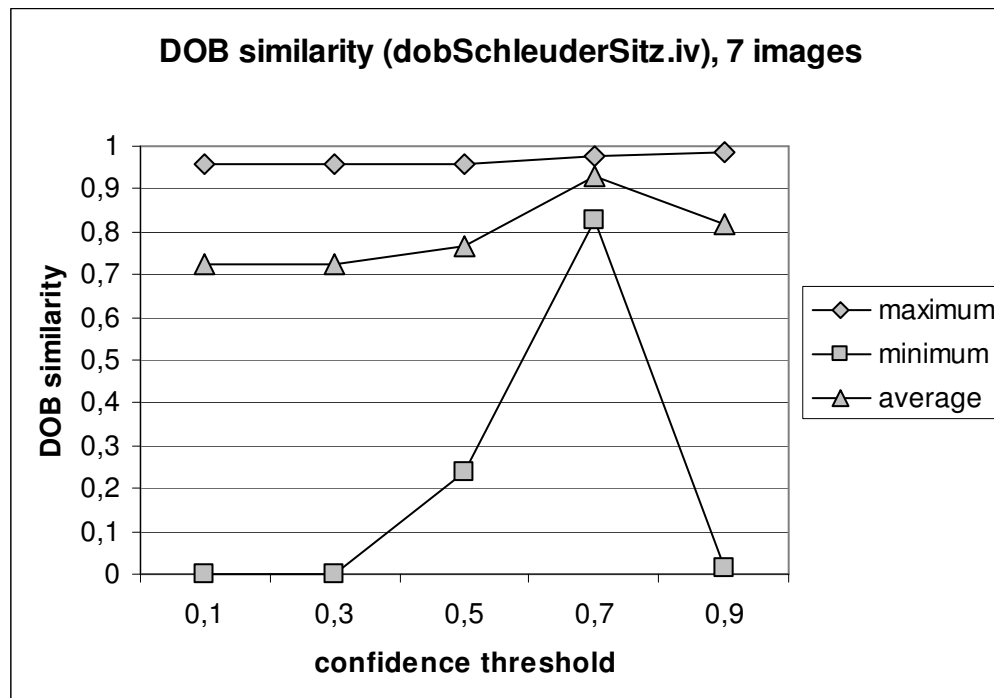
- Introduction and problem specification
- Related work
- The suggested approach
- Implementation of the algorithm
- (Preliminary) results
- Conclusion



- Performance not yet good enough for realtime framerates
- Searching correspondence points is most expensive part of algorithm
- Implementation was tested on a 550 MHz CPU
- Certain optimization techniques should improve the performance considerably

Algorithm data dialog	
Total time	1100 msecs
Salient points detection	70 msecs
Correspondence establishment	560 msecs
Correspondence filtering	0 msecs
Computation of 2D transformation	0 msecs
2D Warping	40 msecs
Finding occlusion	310 msecs
Filtering occlusion mask	110 msecs
Rendering of last DOB	20 msecs
Algorithm benchmark data	
Close	

Analysis of typical algorithm run

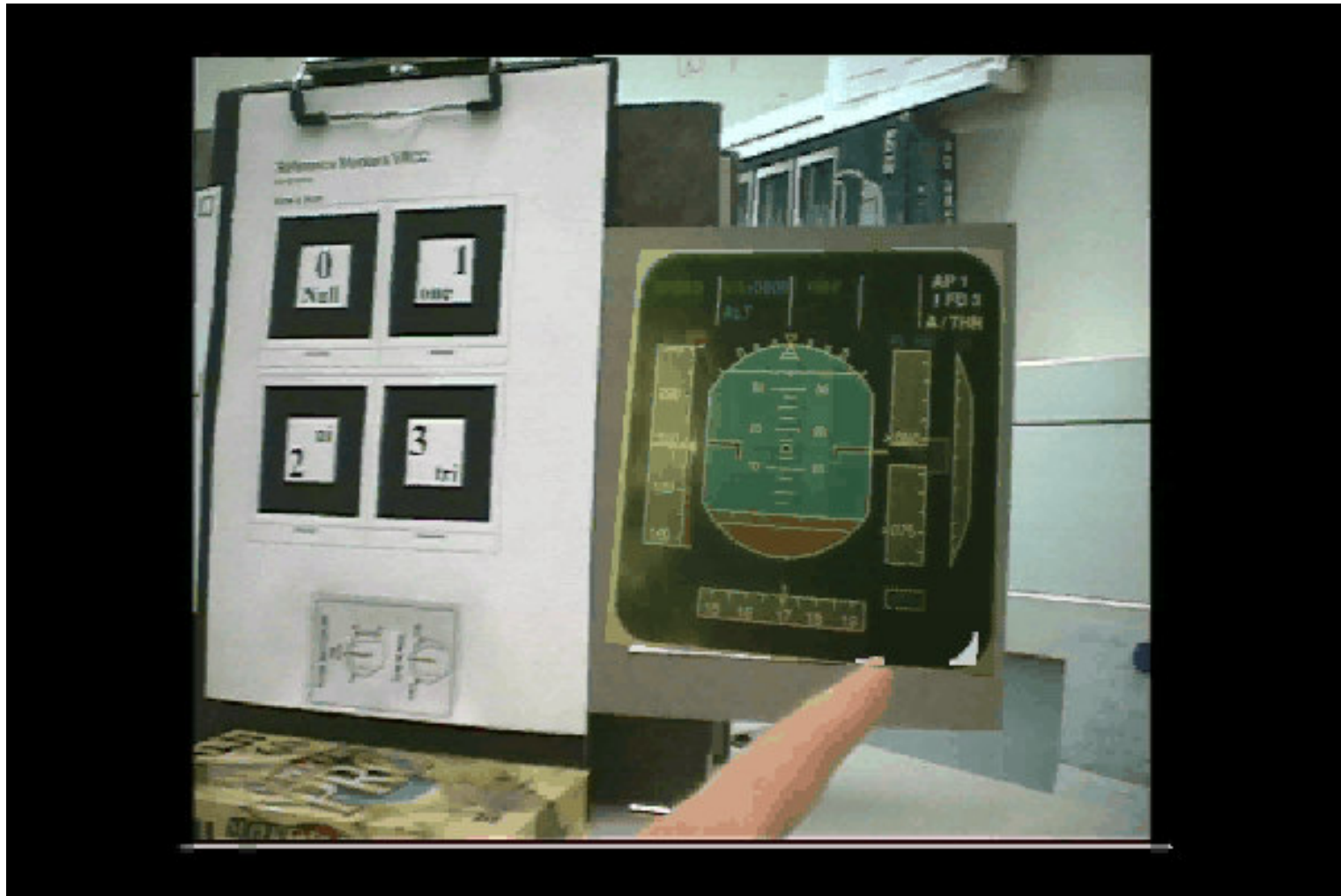


- Software tool for comparing computed and actual geometry of DOB
- Influence of confidence threshold on DOB quality was examined
- Confidence threshold 0.7 appears to deliver very good results

- No method for evaluating correctness of identified occluders yet



23.05.2003





- New, model-based approach to detecting dynamic occlusion in Augmented Reality
- Design aims at easy integration into existing AR systems
- Proof-of-concept implementation might be a promising starting point, although it lacks speed
- A number of effective optimization techniques are conceivable (see paper and thesis)

⇒ Method might be the basis for a combined markerless tracking / occlusion detection system



Questions?

Full text of thesis can be found at <http://gris.uni-tuebingen.de/~fischer/>