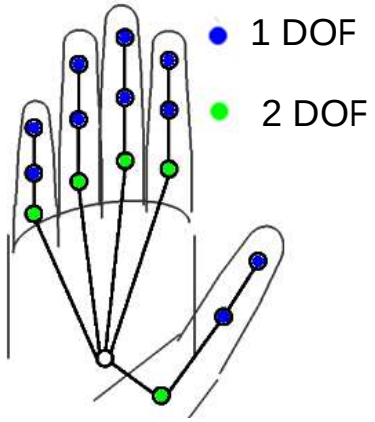
Segmentation-Free, Area-Based Articulated Object Tracking

Daniel Mohr, Gabriel Zachmann
Clausthal University, Germany
{dmoh,zach}@tu-clausthal.de

ISVC 2011, Las Vegas, Nevada, USA

Motivation: Camera Based Hand Tracking

- Estimate hand parameter
  - Global position (3 DOF)
  - Global orientation (3 DOF)
  - Joint angles (20 DOF)



global state local state

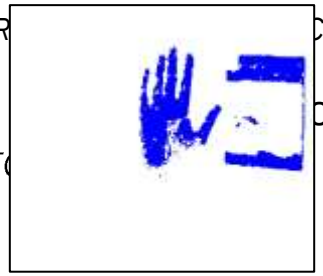
- Tracking approach
  - Sample hand parameter space $\theta_1, \cdots, \theta_N, \quad \theta_i \in \mathbb{R}^{26}$
  - Render hand model for θ_i
  - Compute descriptor for matching

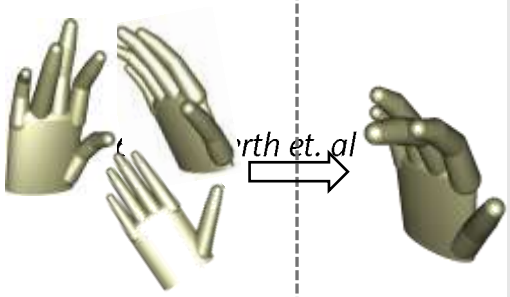
Related Work: Tracking Pipeline

Capture image | Feature extraction | Binarization | Matching | Result

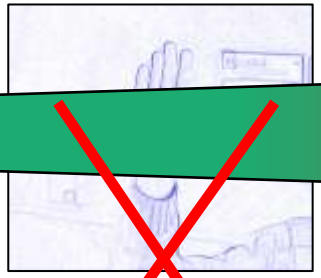
- Binary Segmentation
  - Compare difference vectors between gravity center and points at binary silhouet… *t. al AFGR … CCV '01]*
  - Intersect… …mplate … …on *[Lin et. al A… …FGR '06][…]*
  - …nent… …d
  - …t probability using skin likelihood map *[Stenger et. al … …rth et. al … '04]*

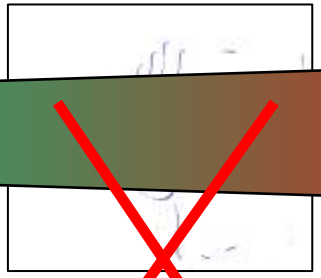
Skin segmentation





Edge detection

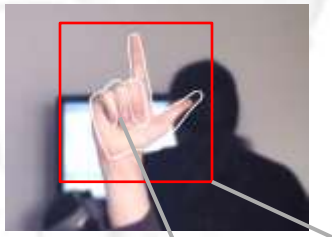




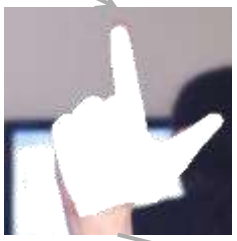
Sources of error

Our Color Divergence-Based Similarity

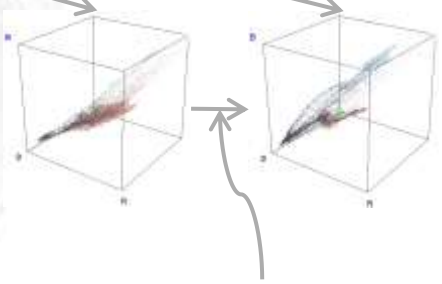
Input Image



Hypothesis about shape



Fore- and Background from Hypothesis



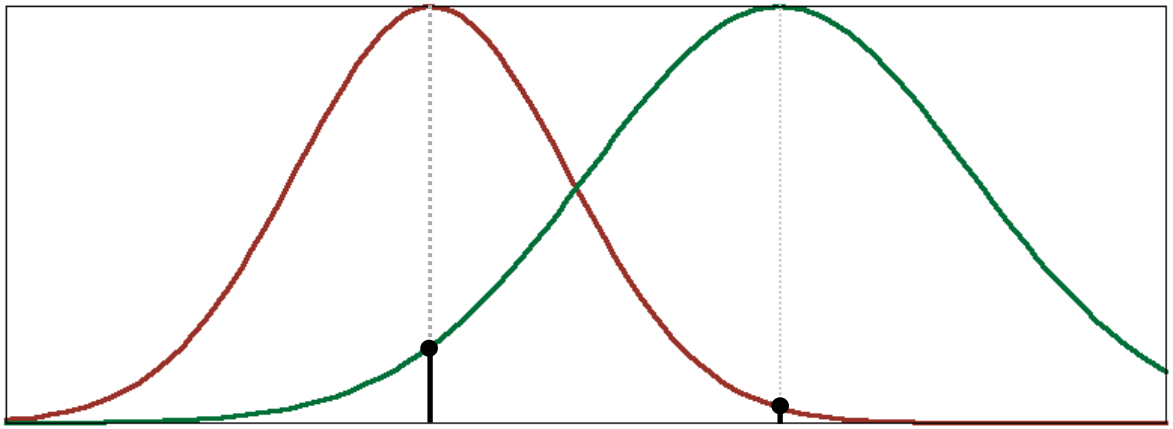
Dissimilarity

Check color distribution

Likelihood

Dissimilarity Measure

- Goal: compute extremely fast
- Gaussian distribution for foreground color:-μ_{fg}, Σ_{fg} and background color: μ_{bg}, Σ_{bg}
- Similarity = $G(\mu_{fg}|\mu_{bg}, \Sigma_{bg}) + G(\mu_{bg}|\mu_{fg}, \Sigma_{fg})$



- Kullback-Leibler divergence for Gaussians performed worse

Our Fast Color Distribution Estimation

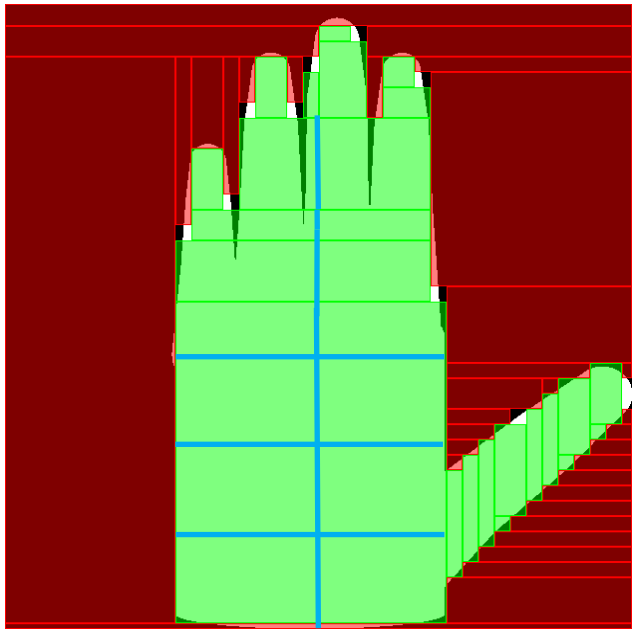
- Color mean μ is normalizing sum of pixel colors
- Color covariance matrix

$$\Sigma \propto \sum_{\mathbf{x} \in \mathscr{R}} \mathbf{x}\mathbf{x}^{\mathsf{T}} - \mu\mu^{\mathsf{T}}$$

$$= \sum_{R_i \in \mathscr{R}} \sum_{\mathbf{x} \in R_i} \mathbf{x}\mathbf{x}^{\mathsf{T}} - \mu\mu^{\mathsf{T}}$$

$$\approx \sum_{R_i \in \mathscr{R}} |R_i| \cdot \mu_i \mu_i^{\mathsf{T}} - \mu\mu^{\mathsf{T}}$$

- For each channel in input image:
  - Compute integral image
- For all templates:
  - Axis-aligned rectangle representation
  - Subdivide large rectangles

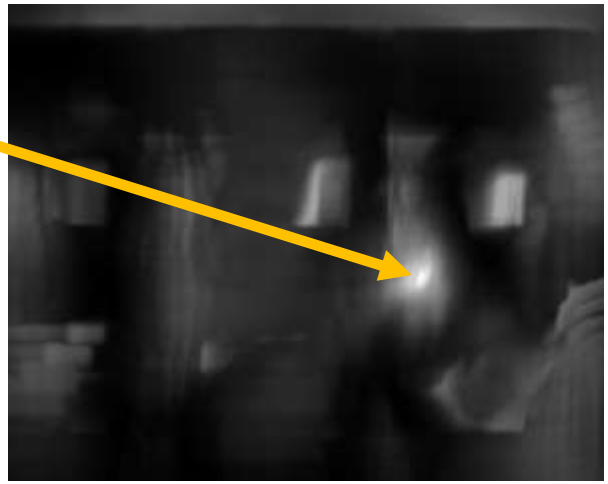


Tracking by Detection

- Tracking by detection
  - No manual initialization
  - No predictive filtering
    - Predictive filtering tends to drift away
    - Real hand movements are unpredictable
- For each frame:
  - find *hand pose & position*

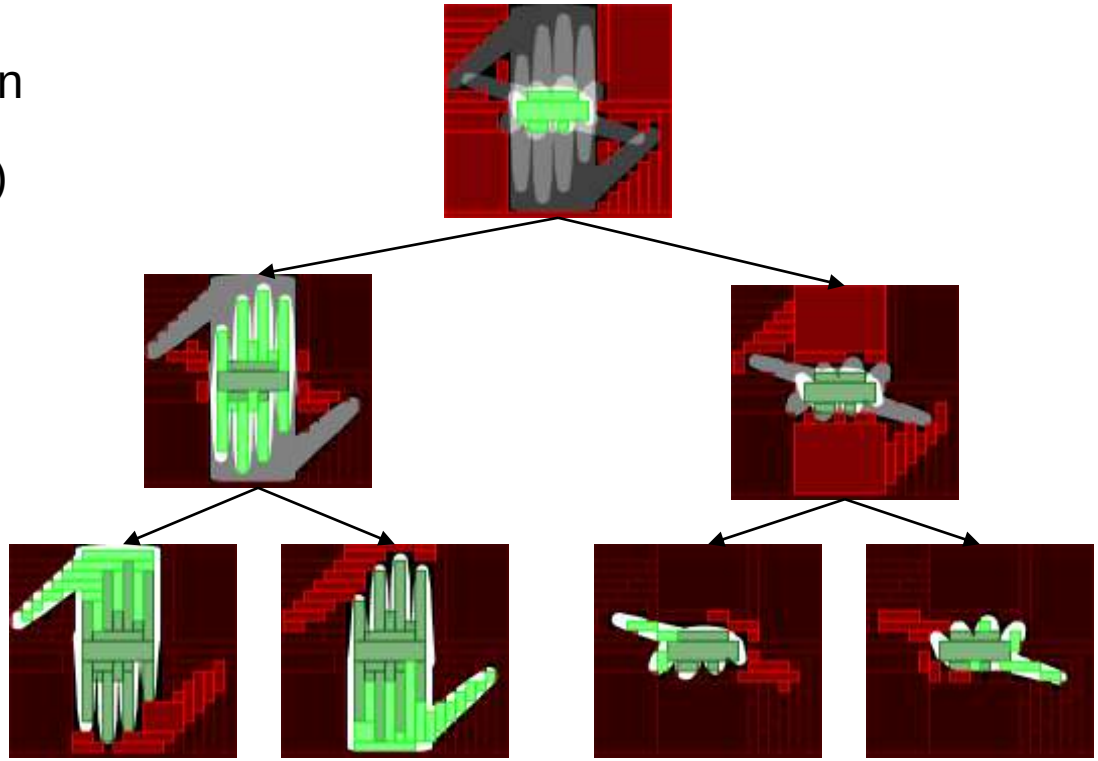
  = find *global max* in likelihood map

  → Approach:
    - Scan input image with step size Δ and
    - Perform local optimization



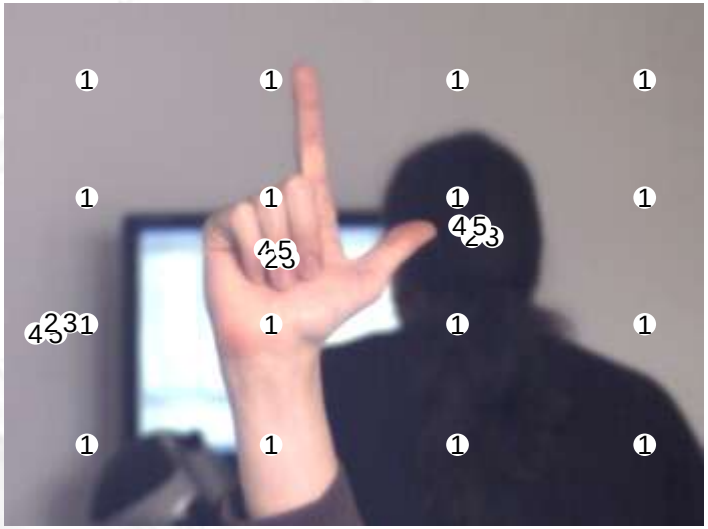
Determine Best Match from Object Database

- Task: Find best of *n* hand poses
- Hierarchy
  - Based on intersection
  - Complexity O(log *n*)

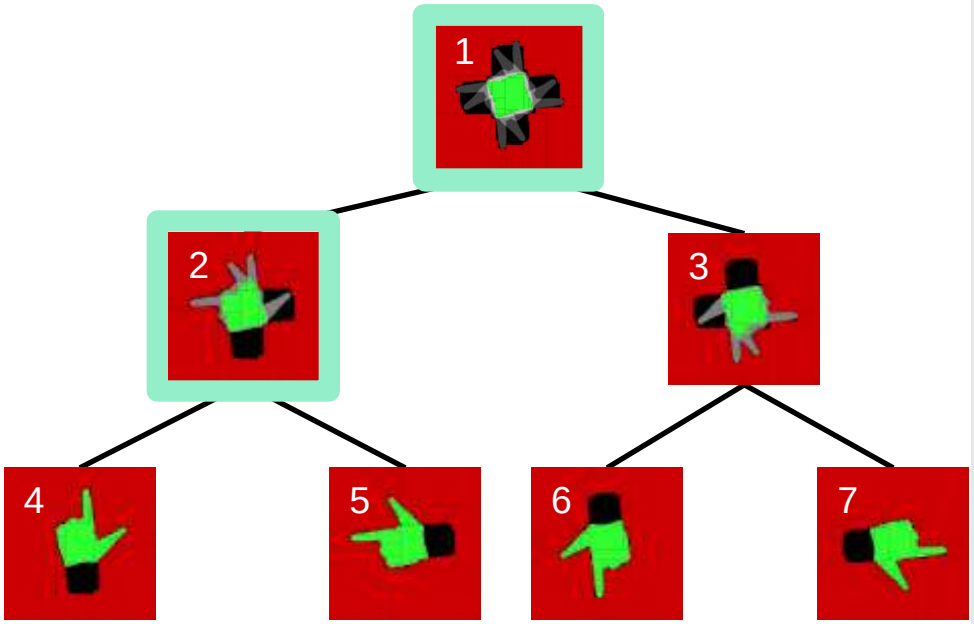


Our Hierarchical Coarse-to-Fine Detection

Input image



Template hierarchy



Algorithm

1. add match candidates
2. local optimization
3. keep *k* best candidates
4. for each candidate
   replace by child node
5. if inner nodes
   goto 2
6. else
   local optimization
   select best candidate

Computation of Step Size Δ

- Need to be computed offline at template generation

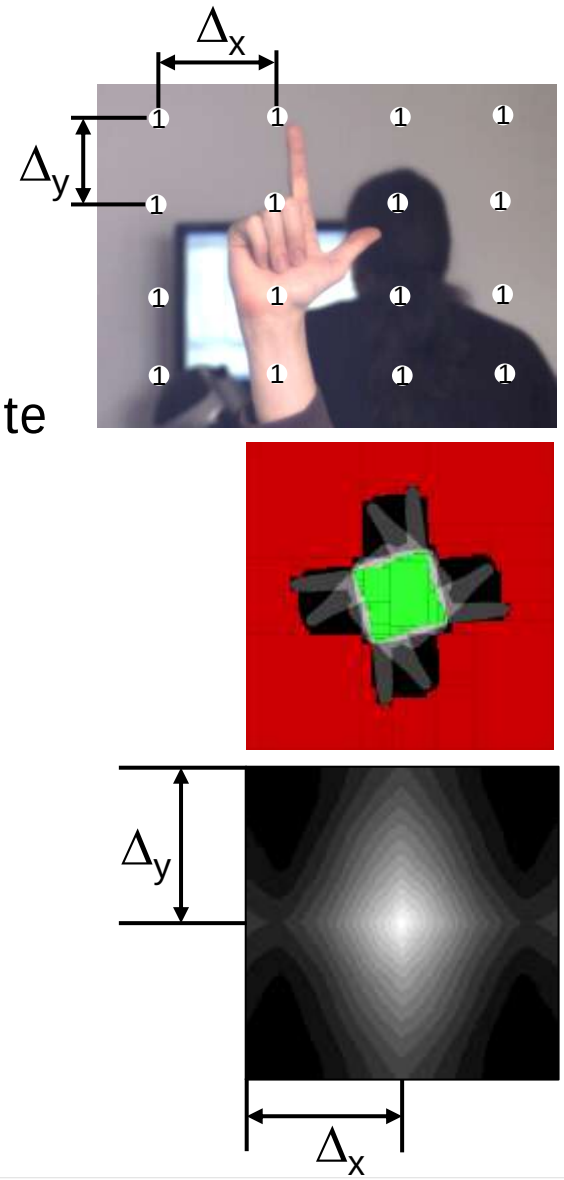
→ No knowledge about input image

→ Estimate based only on template silhouette

- Idea
  - Consider likelihood map as a function
  - Sampling theorem:
    sample at least with 2 × *highest freq.*
  - What is the highest freq. ?

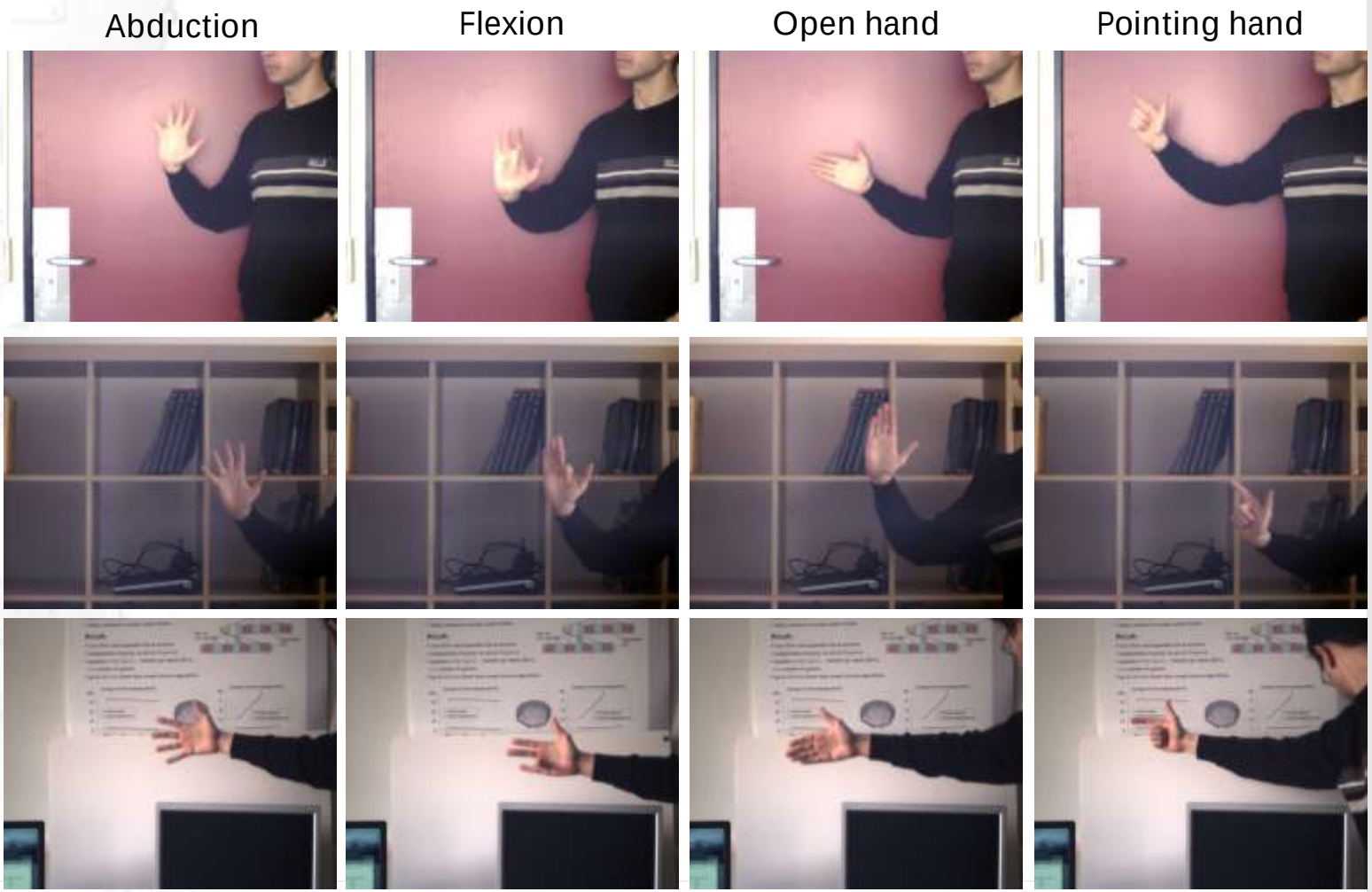
→ Intersection of template with itself

→ Autocorrelation of template



Experimental Evaluation

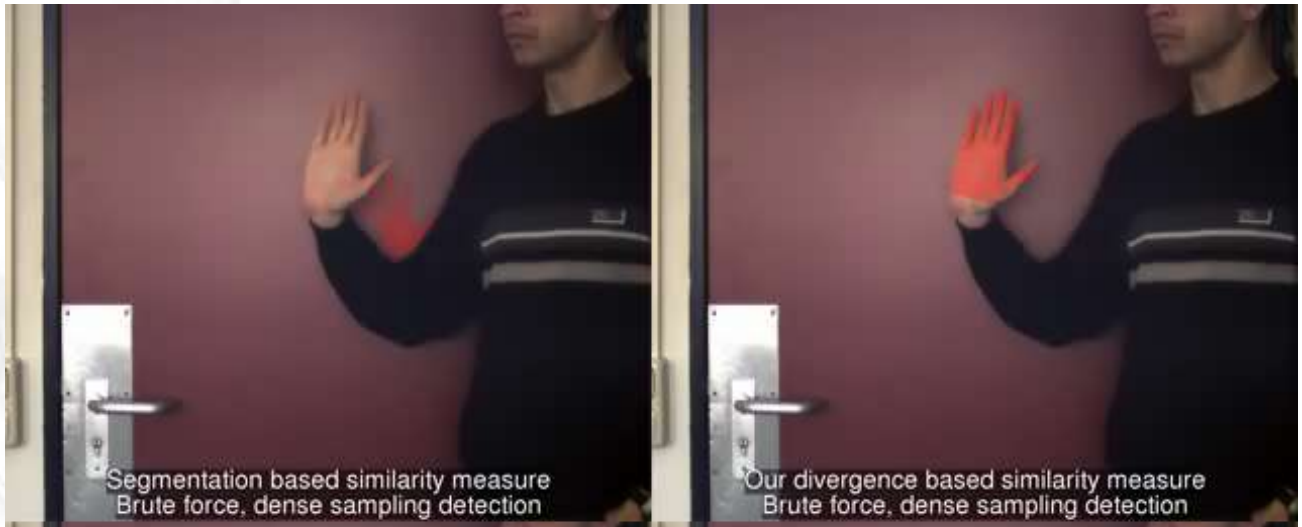
- 12 different test configurations



Qualitative Evaluation

Skin segmentation | Our method

Brute-force

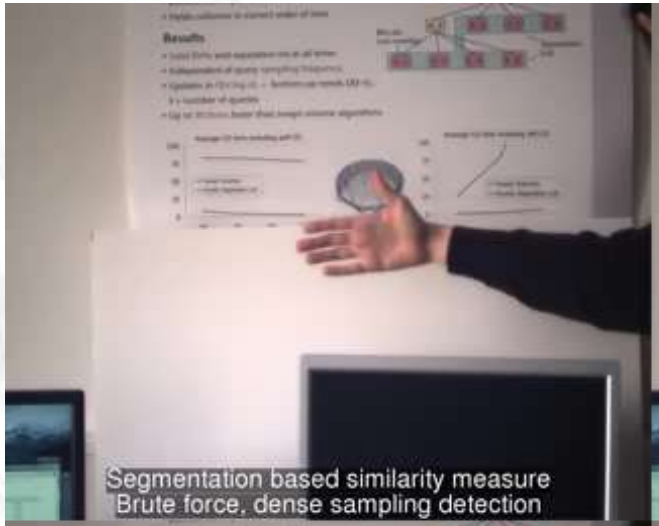


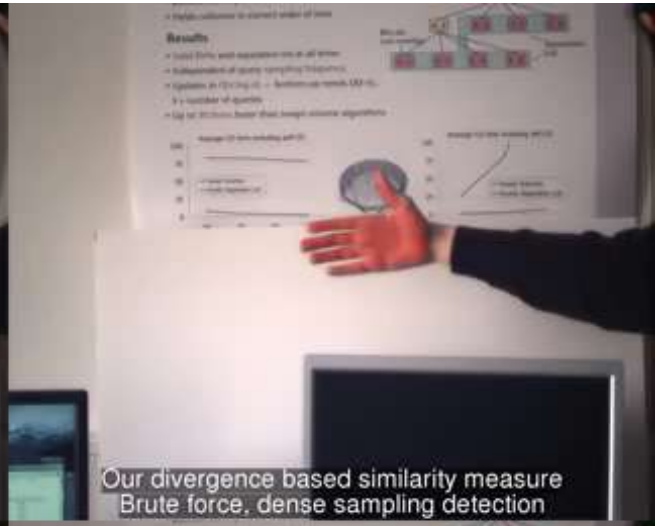
Coarse-to-fine

Qualitative Evaluation

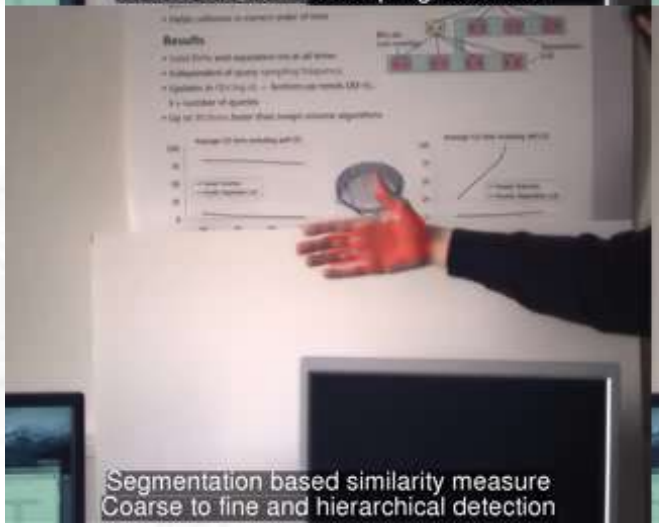
Skin segmentation | Our method

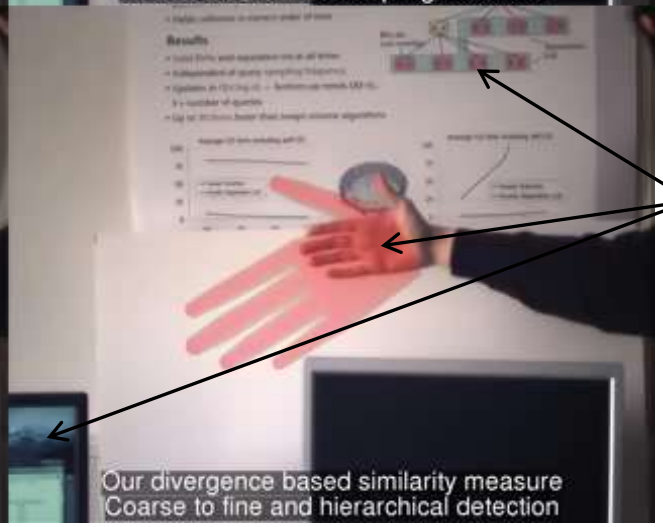
Brute-force



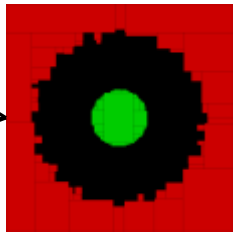


Coarse-to-fine





root node

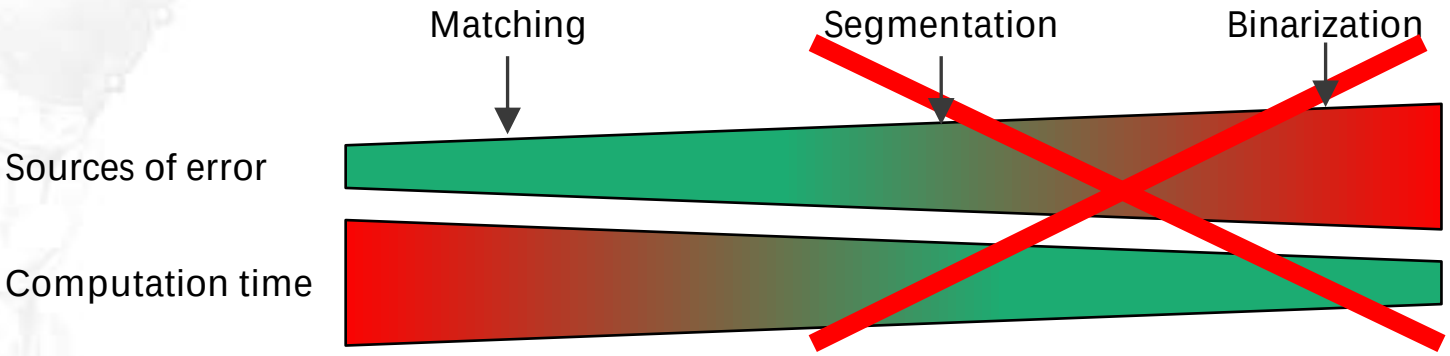


Computation Time

- On 2.6 GHz CPU, in C++, not optimized
- Our coarse-to-fine approach: 0.6 fps
- Compared to brute-force:
  - Several minutes per frame
  - Hand tuned matching threshold, step size 12 pixel: 0.3 fps
- On a Geforce GTX 480 GPU
  - 30 fps

Conclusions

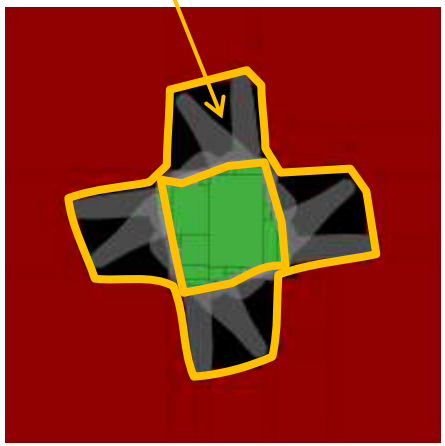
- Novel color divergence-based detection approach
  - No segmentation, no edge extraction, required



- Tracking by detection approach, very fast on GPU
- Extension from colors to other input modalities (e.g. depth values) straight forward
- Not limited to hand tracking

Future Work

- Extend color distribution to
  - Mixture of Gaussians or
  - Non-parametric representation (higher computation cost)
- Try to extract information from in-between region



Thanks for your attention!

Questions?