Continuous Edge Gradient-Based Template Matching for Articulated Objects

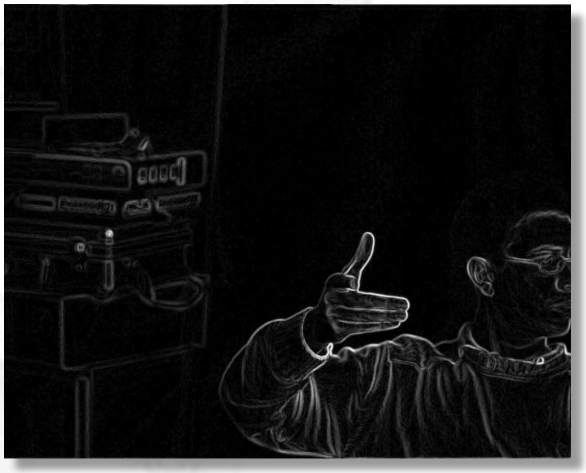
Daniel Mohr

Clausthal University, Germany

dmoh@tu-clausthal.de

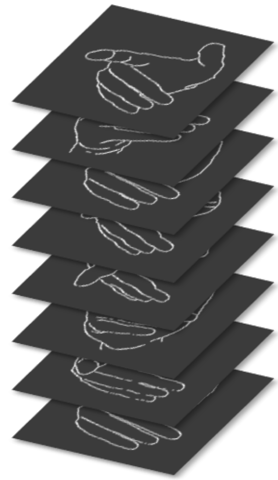
VISAPP '09, Feb 2009, Lisboa, Portugal

Goal: Compare Edge Images



Query edge image

+



Set of template edge images

=

Greatest similarity value for all templates

- *for each* position in the query image

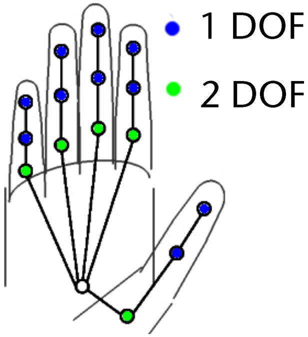
    for each template

        compute similarity value

Application: Hand Tracking

- Given an image, 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 $\theta_1, \cdots, \theta_N, \quad \theta_i \in \mathbb{R}^{26}$
  - Project hand models onto 2D and compare with query image
  - Estimate global position by position/scale of the hand in the query image and orientation/joint angles by different templates

Related Work

- Extract binary edges from template T and query image Q

  → Edge pixel sets E_T and E_Q

- Compare edge images by set distance measure
  - Hausdorff distance

$$\mathcal{H}(E_T, E_Q) = \max_{a_i \in E_T} \{ \min_{b_j \in E_Q} \{ d(a_i, b_j) \} \}.$$

  - Directed chamfer distance [Barrow et al, IJCAI1977]

$$\mathcal{C}(E_T, E_Q) = \frac{1}{|E_T|} \sum_{a_i \in E_T} \min_{b_j \in E_Q} \{ d(a_i, b_j) \}$$

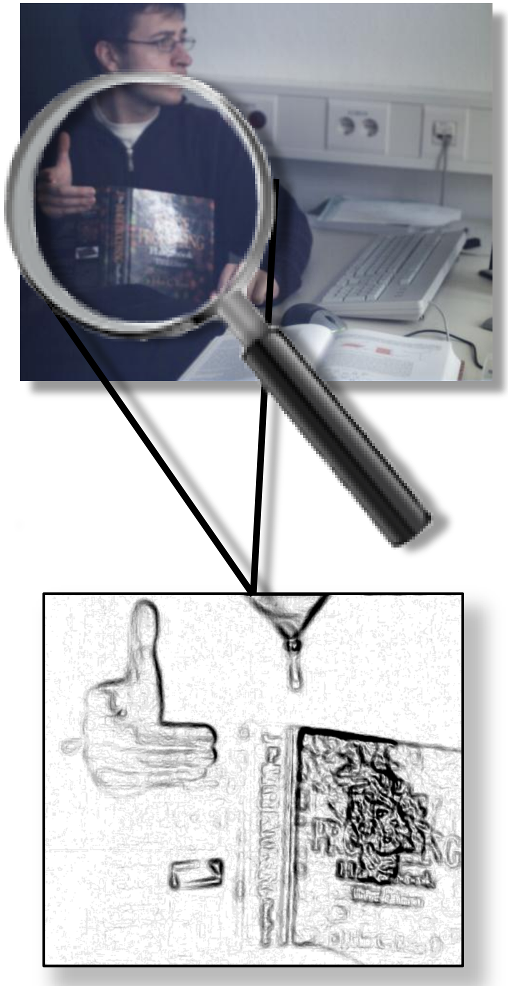
For example used by [Athitsos et al CVPR 2004] and [Gavrila et al, ICCV1999]

- Edge orientation significantly improves measure
- Two options are:
  - Split edge sets according to edge orientation and compare sets separately
    - Integrated in chamfer distance by [Stenger et al, PAMI 2006]
  - Replace distance between edge pixel coordinates by sum of the edge coordinate distance and the edge orientation difference
    - Integrated in Hausdorff distance by [Olson et al, ICIP1997]

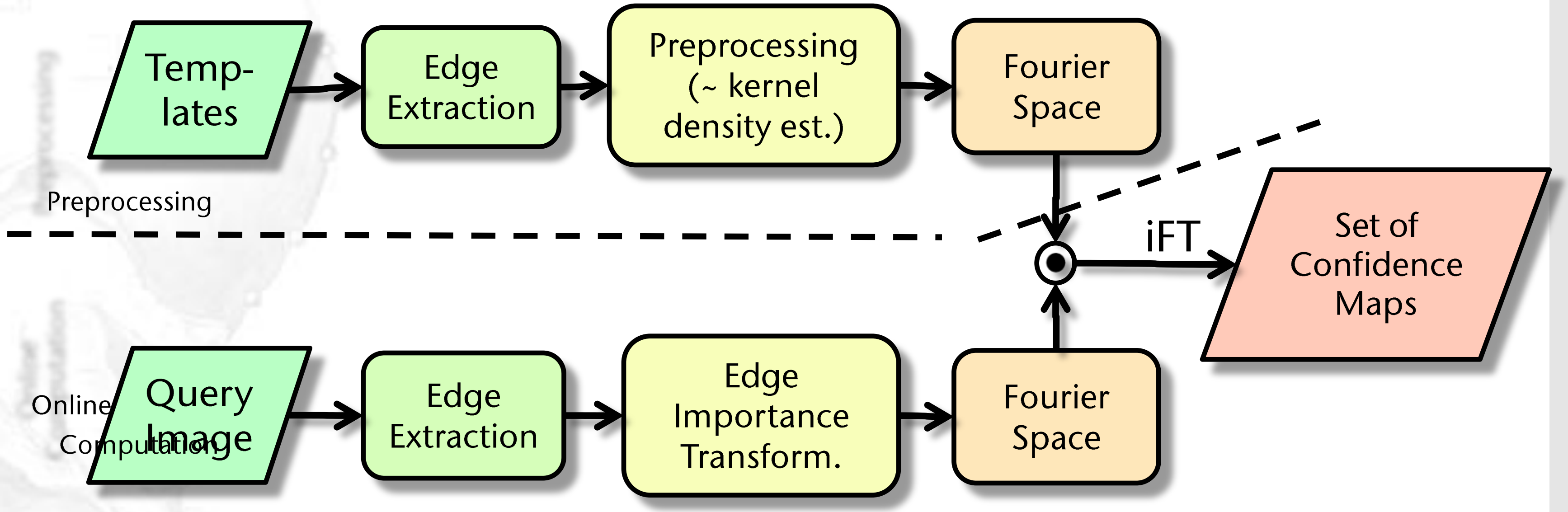
Problems of previous distance measures

- Query image binary edge extraction
  - Uncontrolled conditions
    - Illumination
    - Object and background color
    - Camera parameters

  → Hard to choose good thresholds for binary edge extraction
- Include edge orientation:
  - Previous algorithms that use exact orientation are inefficient in computation
  - Discretization introduces additional inaccuracy



Overview



- Our Approach
  - No query image related parameters
  - Use continuous orientation (no discretization)
  - Still fast computation

Our New Similarity Measure

1. Generate Templates by a synthetic model
   - Conditions known
   - Can find good thresholds for binary edge extraction


color buffer

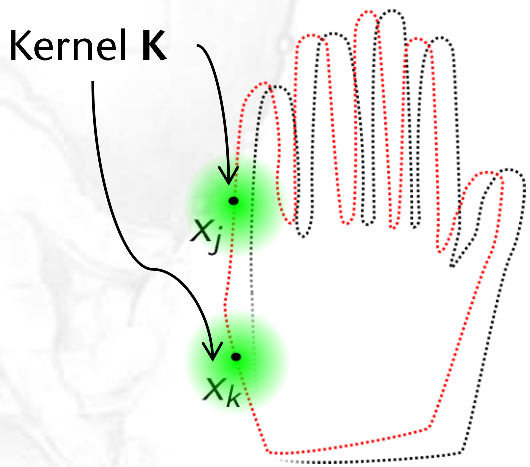

depth buffer

2. Query image
   - Captured under uncontrolled conditions
   - extract edge intensities and orientations

- We need a similarity value at each query image pixel between
  - a binary edge image (template) and
  - a scalar edge image (query)

Chamfer vs Our Approach

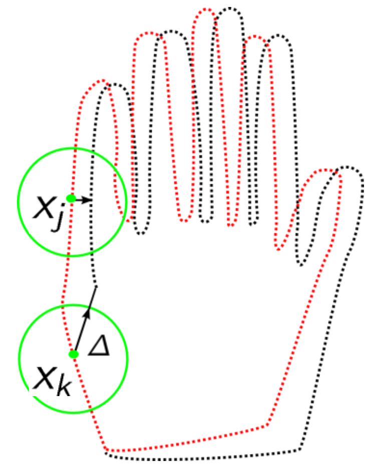
Our matching approach



$$P(\mathbf{x}_i, I_Q) = \frac{1}{N} \sum_{\mathbf{p} \in \mathcal{N}(\mathbf{x}_i)} K\left(\frac{\mathbf{p} - \mathbf{x}_i}{h}\right) I_Q(\mathbf{p})$$

+ Robust against outliers

Truncated chamfer distance



$$\mathcal{C}(x_i, E_Q) = \min\left(\min_{b_j \in E_Q} d(x_i, b_j), \Delta\right)$$

+ Robust against outliers

- Needs difficult-to-adjust parameters

- Orientation discretized

How to Incorporate Edge Orientation

- Replace scalar multiplication of edge intensities by dot product of edge gradient vector

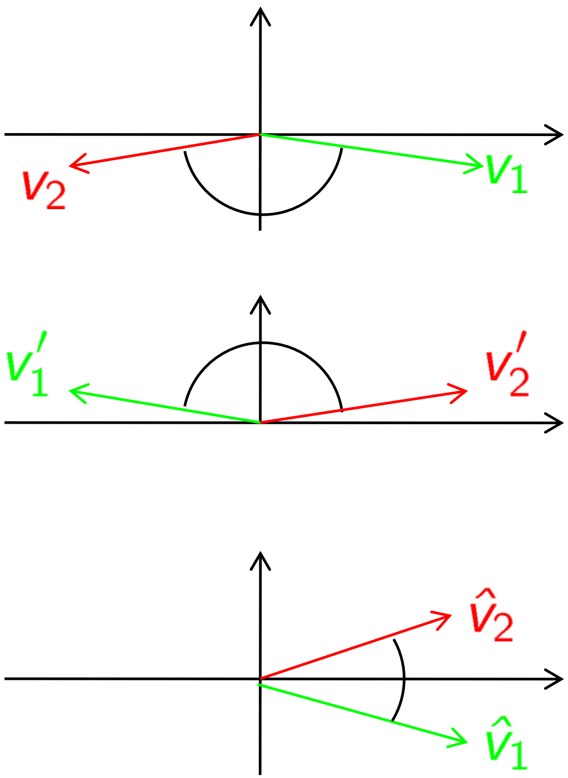
$$P(\mathbf{x}_i, I_Q) = \frac{1}{N} \sum_{\mathbf{p} \in \mathcal{N}(\mathbf{x}_i)} K\left(\frac{\mathbf{p} - \mathbf{x}_i}{h}\right) G_T(\mathbf{x}_i) G_Q(\mathbf{p})$$

- Map edge gradient $v = (v_x, v_y)$:

$$\theta = \arctan \frac{v_y}{v_x}$$

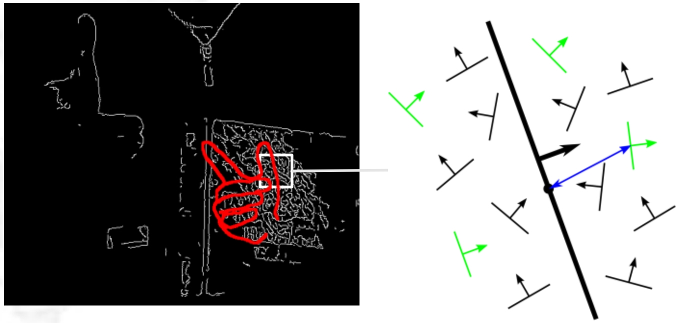
$$\theta' = \begin{cases} \theta & \theta \geq 0 \\ \theta + \pi & \theta < 0 \end{cases}$$

$$\hat{\mathbf{v}} = (\hat{v}_x, \hat{v}_y) = \|\mathbf{v}\|_2 \cdot (\cos(2\theta'), \sin(2\theta'))$$



Distance in noisy regions

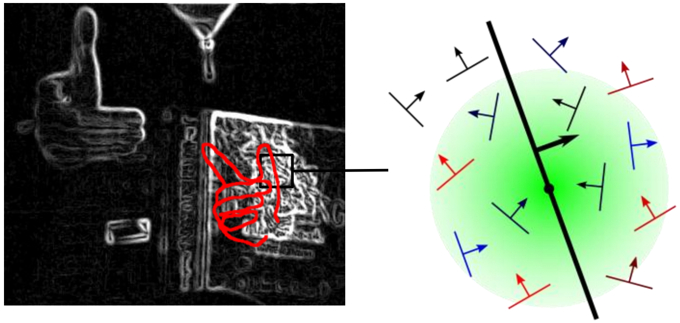
- Chamfer distance strongly underestimates distance near image noise



- Chamfer approach matches to the closest noise edge with similar edge orientation

→ low distance

→ wrong



- Our approach sums over the neighborhood

→ low similarity

→ high distance

→ correct

Efficient Computation of Our New Measure

- Overall similarity between query image I_Q and template E_T at position $\mathbf{O}$ on the query image

$$S_{E_T,I_Q}(\mathbf{O}) = \frac{1}{|E_T|} \sum_{\mathbf{x}\in E_T} \frac{1}{N} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} K\left(\frac{\mathbf{y}-\mathbf{x}}{h}\right) G_T(\mathbf{x}) G_Q(\mathbf{O}+\mathbf{y})$$

  - K has limited support →

$$= \sum_{\mathbf{y}\in D}\left(G_Q(\mathbf{O}+\mathbf{y}) \underbrace{\frac{1}{|E_T|N} \sum_{\mathbf{x}\in D} I_T(\mathbf{y}) K\left(\frac{\mathbf{y}-\mathbf{x}}{h}\right) G_T(\mathbf{x})}_{\tilde{I}_T(\mathbf{y})} \right)$$

$$D = [1, \mathrm{width}(I_T)] \times [1, \mathrm{height}(I_T)]$$

  - $\tilde{I}_T$ depends only on the template and can be precalculated

Efficient Computation (cont.)

- Finally, we arrive at

$$S_{E_T, I_Q}(\mathbf{O}) = \sum_{\mathbf{y} \in D} G_Q(\mathbf{O} + \mathbf{y}) \tilde{I}_T(\mathbf{y})$$

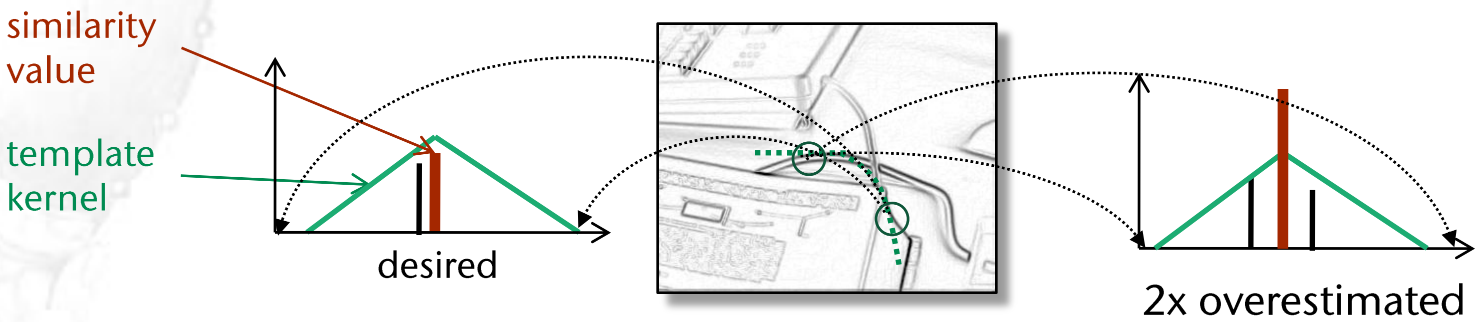
- Computing S for all positions ***O*** of the query image is a correlation
  - We convert it into a convolution
- Calculate convolution in Fourier space
  - Each pixel of G_Q and $\tilde{I}_T$ is a 2D-vectors
  - each component is Fourier transformed separately
  - Linearity of the Fourier transform with respect to addition

    → components added in Fourier space

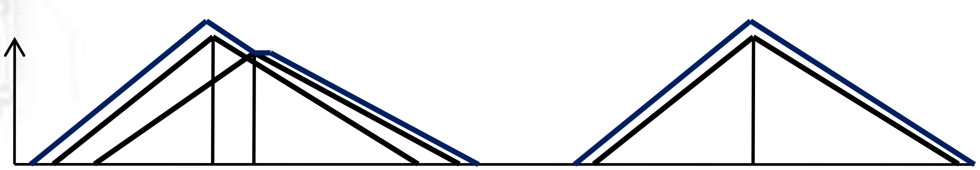
    → saves one inverse Fourier transform

Correct measure with close parallel edges

- Multiple parallel edges very close to each other



- To overcome this problem we preprocess the query image
  - Apply kernel function to edge intensities
  - Edge intensity ← max of neighborhood intensities



two close strong edges after preprocessing

Single strong edge after preprocessing

Implementation Details

- Our approach is well suited for implementation in the stream processing model
- We use the Cuda library from Nvidia
- Query image uploaded to GPU as delivered by frame grabber
  - Image stored as texture, because texture access optimized for consecutive access to neighboring pixels
  - Edge gradient calculation and all preprocessing done efficient on GPU
- Templates are preprocessed and Fourier transformed offline
  - uploaded to graphics hardware at initialization

Results: Evaluation of our Method

- Given ***m*** templates, generate for each template a confidence map
  - Generate at each position in the query image a similarity value between template T_k and I_Q

    → Confidence Map $\mathcal{S}(k, x, y)$
- At each position in query image, find the template best matching at the appropriate window

  → Combined Confidence Map

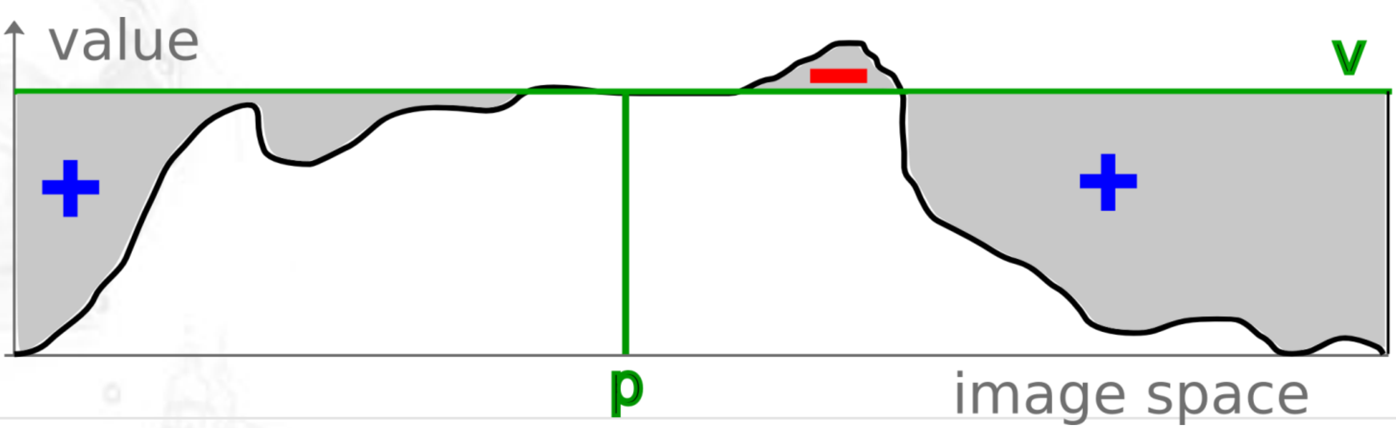
$$\mathcal{S}(x, y) = \max_{i \in [1,m]} \{\mathcal{S}(i, x, y)\} .$$

- Measure the quality of the template matching approach by evaluating the combined confidence map

Quality Measure

- Good similarity measure is characterized by
  - As little values as possible are higher then the value at the correct hand position $\mathbf{P}$
  - Difference between the value at the correct hand position $\mathbf{P}$ and values at other positions should be as high as possible

$$\mathcal{Q} = \frac{1}{N} \sum_{\substack{0 \le x < W_Q \\ 0 \le y < H_Q}} \left(\mathcal{S}(\mathbf{P}) - \mathcal{S}(x, y) \right)$$



Datasets

- Pointing hand
  (2 translational, 2 rotational)
  - 300 templates: different viewpoints

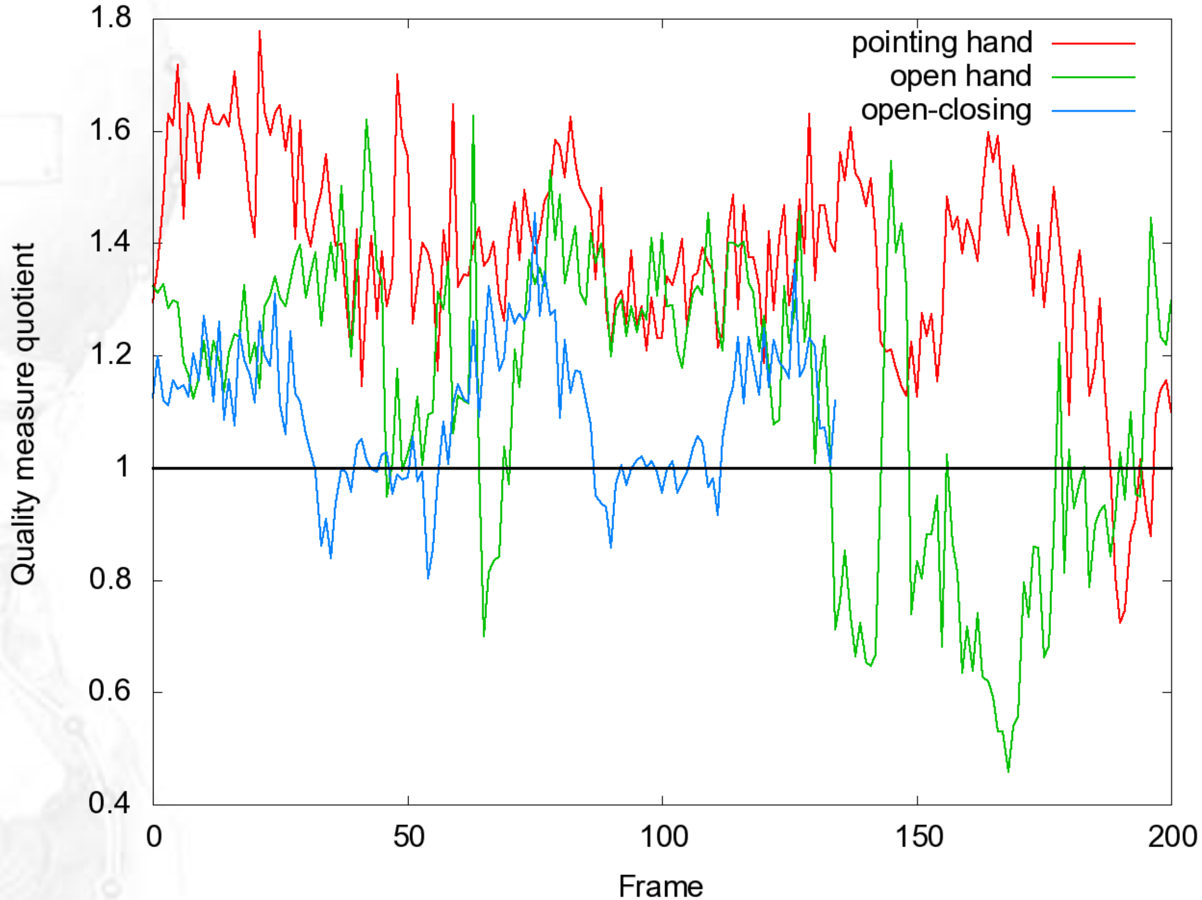
- Open hand
  (2 translational, 2 rotational)
  - 300 templates: different viewpoints

- Hand open-closing
  (2 translational, abduction, flexion)
  - 300 templates: different joint angles

Performance

- Hardware setup
  - AMD Athlon X2 Dual, 2GB RAM
  - Geforce 8800 GTX, 768 MB RAM
- Input
  - Query image resolution 320x256
  - Average template resolution 80x80
- We can generate 330 confidence maps/second (full compare 330 templates with a query image)

Comparison with Chamfer matching



- quality measure quotient between our approach and truncated chamfer w/ 6 orientation channels
- *we manually optimized threshold values for chamfer matching*

Videos

Original image	Best matching template calculated by our approach
Combined Confidence Map: chamfer matching	Combined Confidence Map: our approach

Video panel contents

Pointing hand

Open hand

Open-closing hand

Conclusions

- Robust and fast method to compare edge images
  - No query image related parameters
  - Incorporate continuous edge orientation
- Well suited for stream processing model
  - Implemented on graphics hardware
  - 330 confidence maps per second
- Our method can easily be combined with other template matching approaches such as color region overlapping
- Our application: tracking of the human hand

Future work

- Anisotropic kernel functions
  - Higher variance along edge gradient
- Asymmetric kernel functions
  - Different weights at hand inner/outer regions
- Automatically determine kernel parameters

Acknowledgment

- Gabriel Zachmann, Clausthal University
- DFG grant DFG grant ZA292/1-1