Univerza v Ljubljani

Advanced CV methods
What have we learned and what’s next?

Matej Kristan

Laboratorij za Umetne Vizualne Spoznavne Sisteme,
Fakulteta za računalništvo in informatiko,
Univerza v Ljubljani

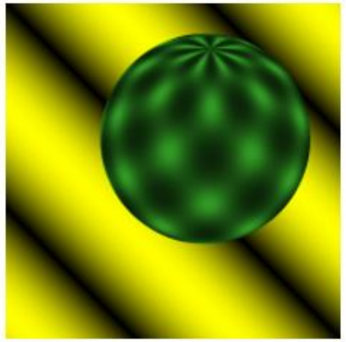
Topics covered

1. Pixel-level motion estimation
2. Patch tracking I: Deterministic gradient descent
   (intensity-based visual model)
3. Patch tracking II: Deterministic gradient ascent
   (histogram-based visual model)
4. Discriminative tracking (tracking by classifiers)
5. Recursive Bayes Filters
6. Fully-trainable trackers (deep learning tracking)
7. Long-term tracking
8. Performance evaluation

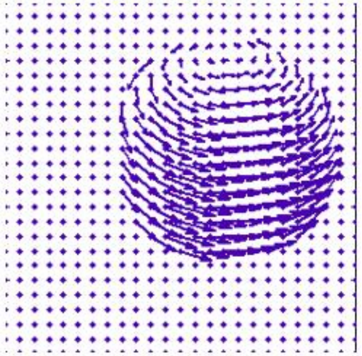
Followed the "engineering principle"

ACVM 1: Pixel-level motion estimation

- Ideally: Projection of 3D motion to image


frame #1

+


flow field

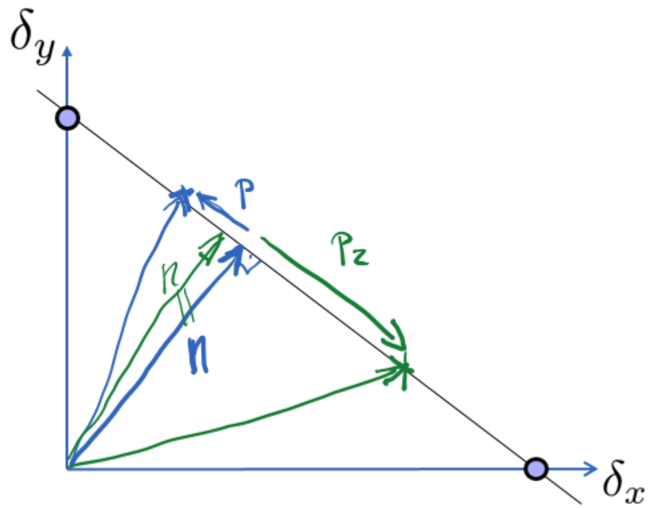
=


frame #2

- Optical flow constraint

$$I_x u + I_y v + I_t = 0$$

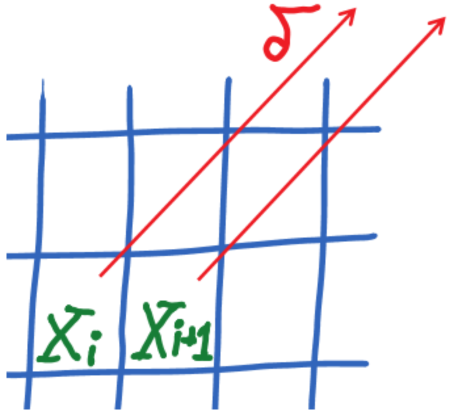
- The aperture problem



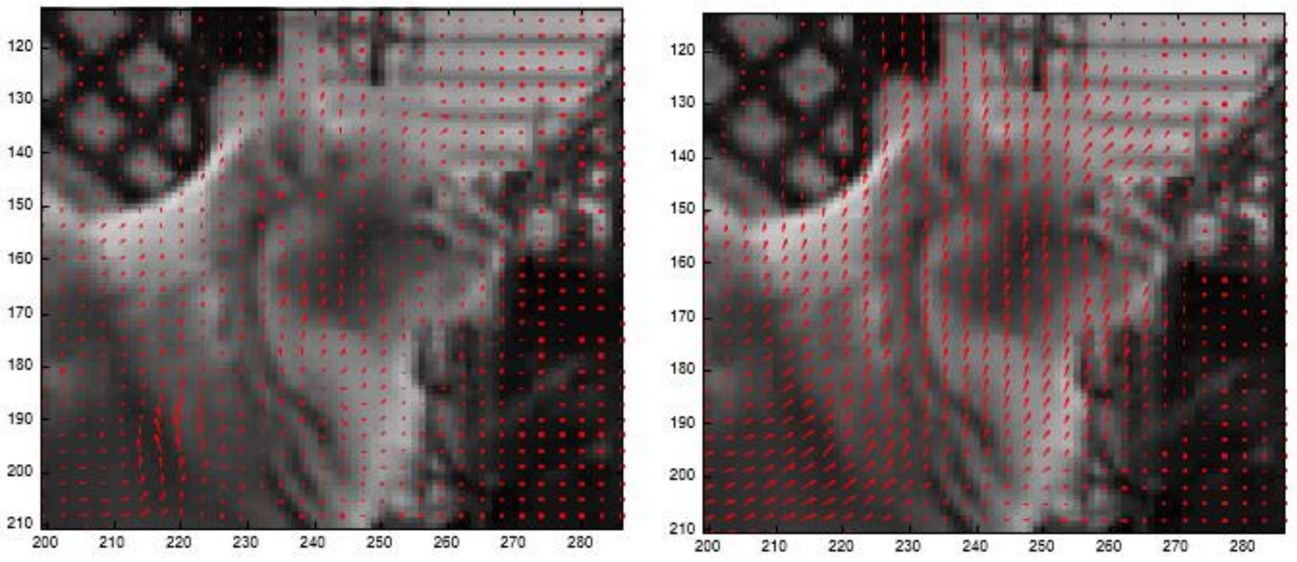
ACVM 1: Pixel-level motion estimation

- Lucas Kanade flow estimation
  - Least squares solution

$$\begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = -\begin{bmatrix} \sum I_x^{\ 2} & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^{\ 2} \end{bmatrix}^{-1} \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$



- Improve by Pyramids



ACVM 1: Pixel-level motion estimation

- Horn and Schunck minimize the functional:

$$E = \iint_D \left(I_x u + I_y v + I_t\right)^2 + \alpha\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) dx\,dy$$

FLOW FIELD

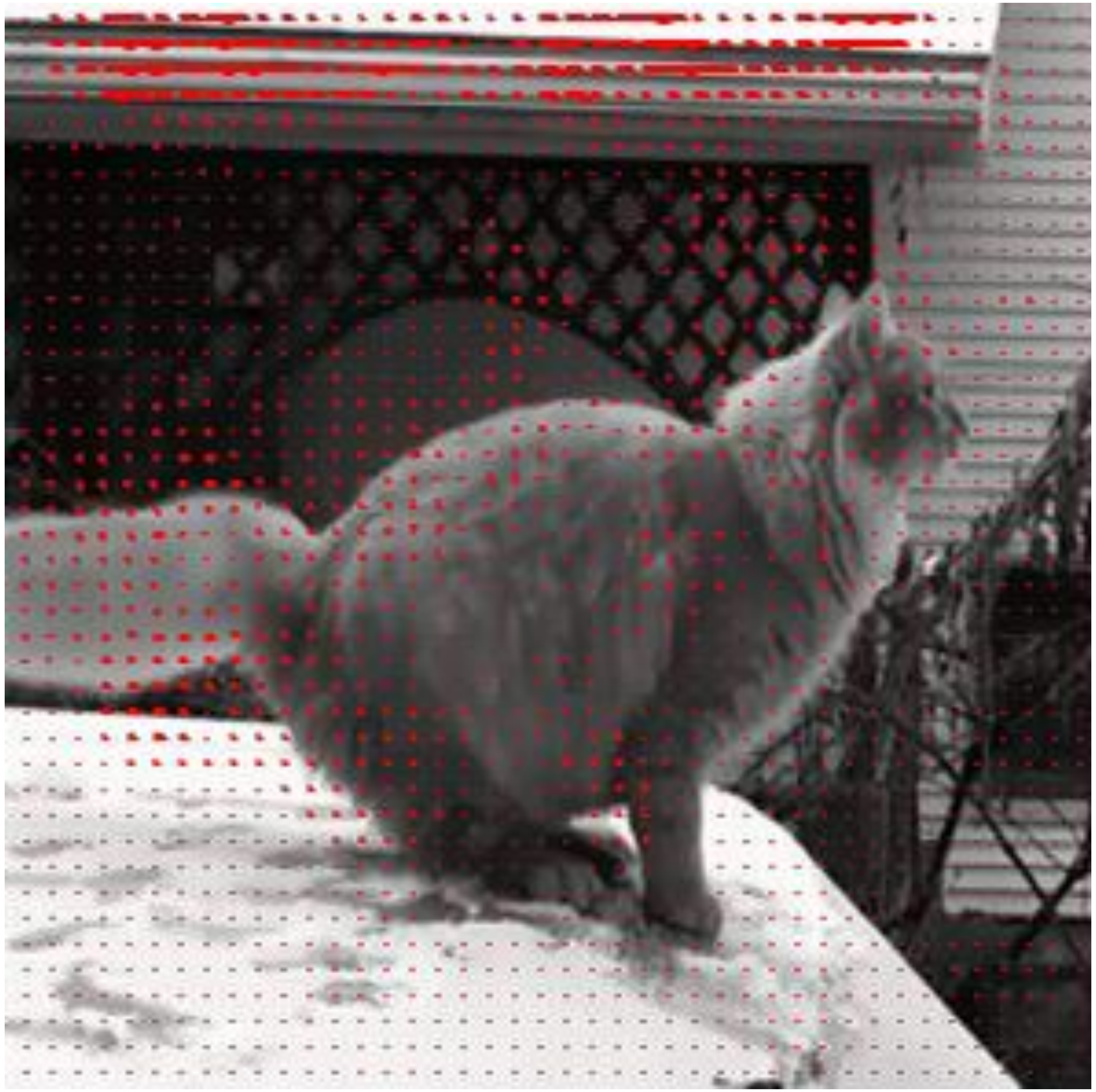
- Solve by Euler-Lagrange, discretize, iterate

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} (I_x{}^2 + \alpha) & I_x I_y \\ I_x I_y & (I_y{}^2 + \alpha) \end{bmatrix}^{-1} \begin{bmatrix} \alpha\bar{u} - I_x I_t \\ \alpha\bar{v} - I_y I_t \end{bmatrix}$$

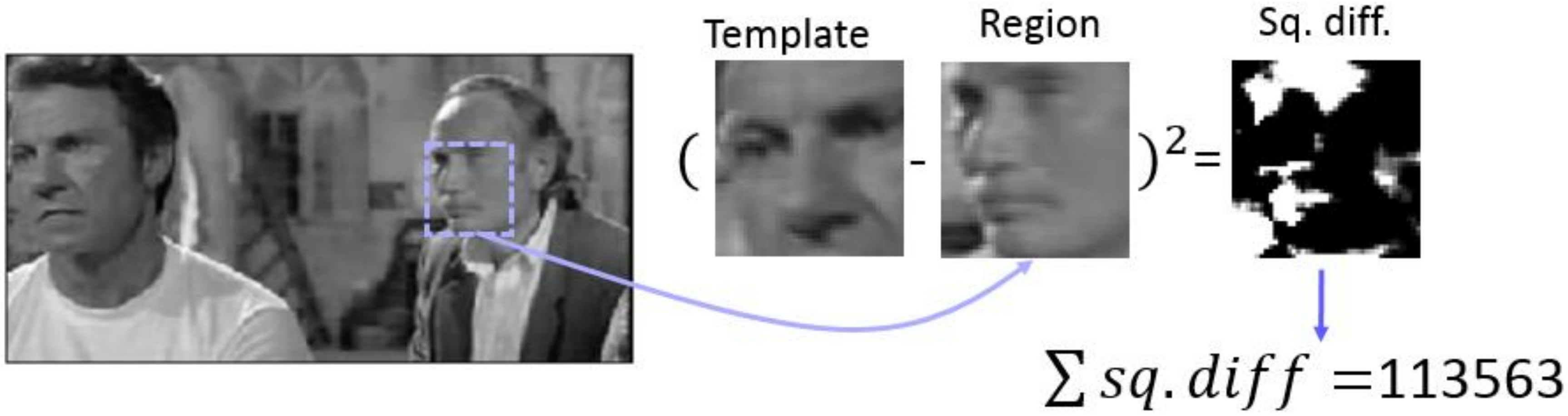
- Interesting to compare to Lucas-Kanade

$$\begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = -\begin{bmatrix} \sum I_x{}^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y{}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

ACVM 1: Pixel-level motion estimation

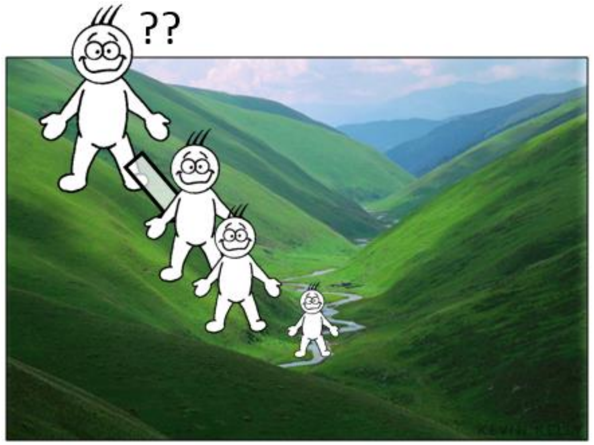


ACVM 2: Patch tracking I



- Patch tracking as a gradient descent on SSD
  - Parametric deformation model $W(x;p)$
  - The cost function:

$$E(\Delta p) = \sum_{x} \left(I(W(x;p+\Delta p)) - T(x) \right)^2$$

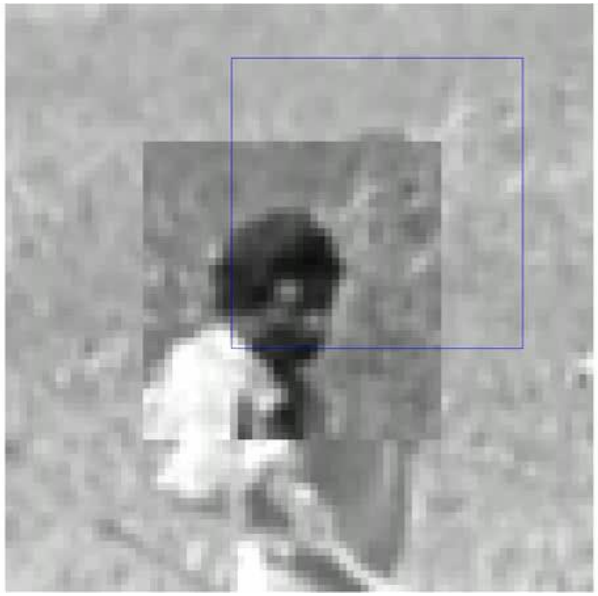


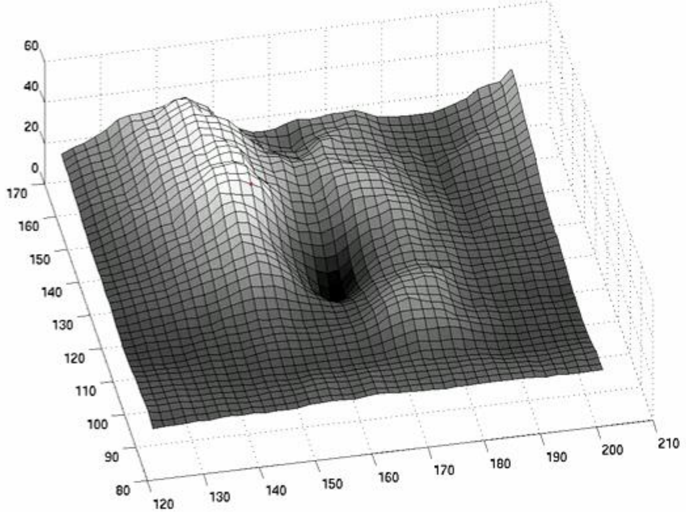
ACVM 2: Patch tracking I

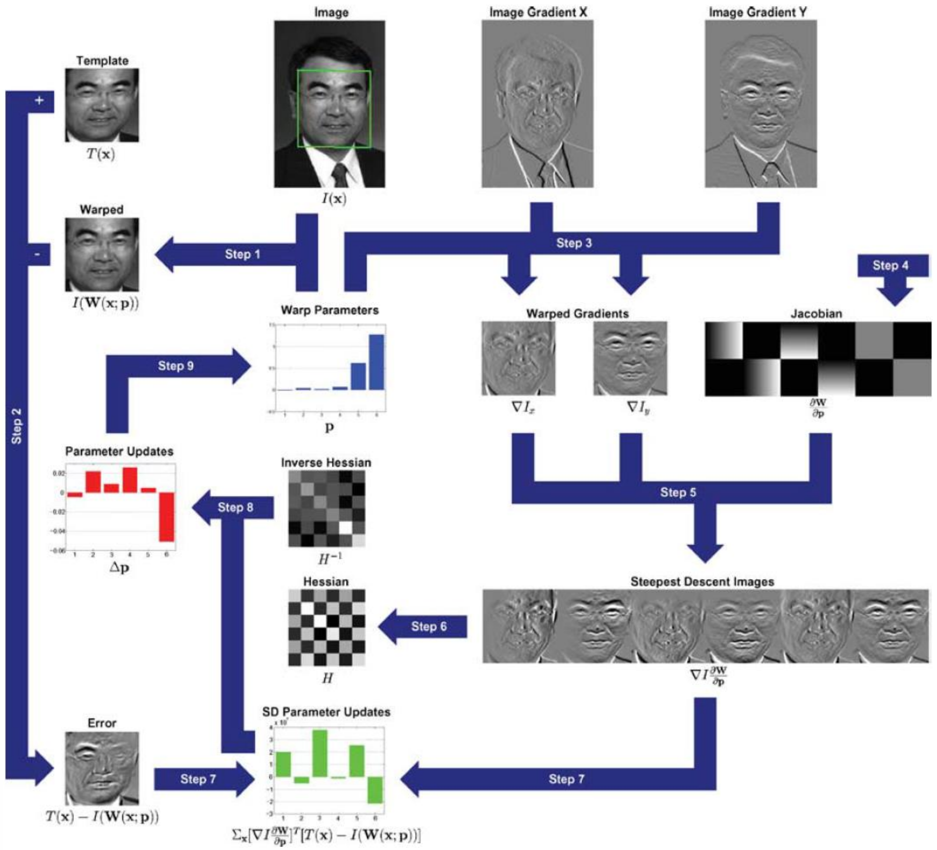
Lukas Kanade:

- Linearize deformation
- The Gradient:

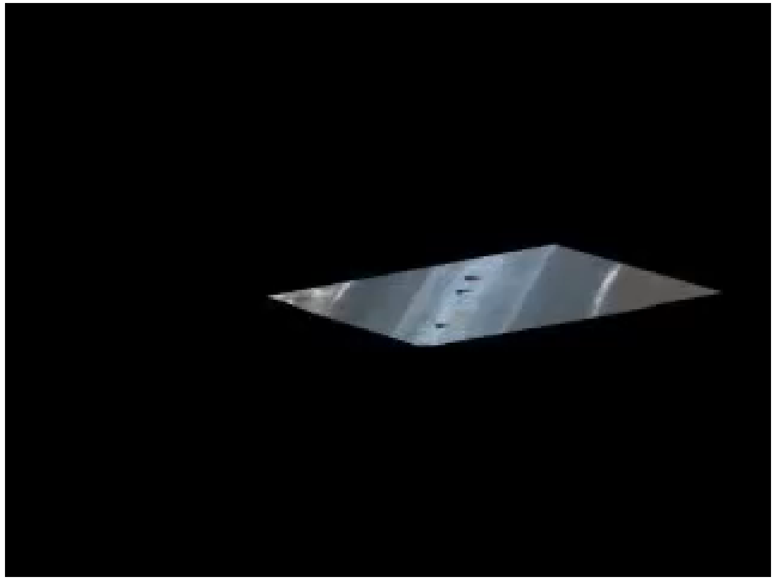
$$\Delta p = H^{-1} \sum_x \left[\nabla I^T \frac{\partial W}{\partial p} \right]^T \left[T(\mathrm{x}) - \mathrm{I}(\mathrm{W}(\mathrm{x};\mathrm{p})) \right]$$

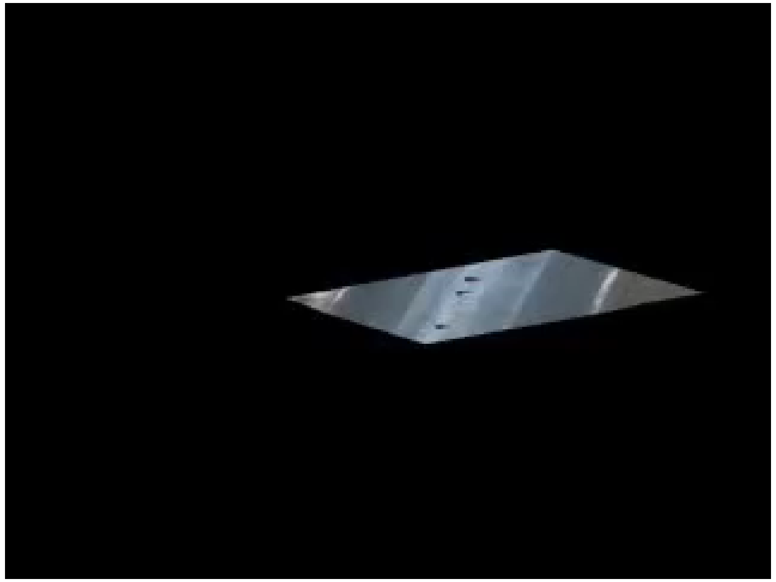






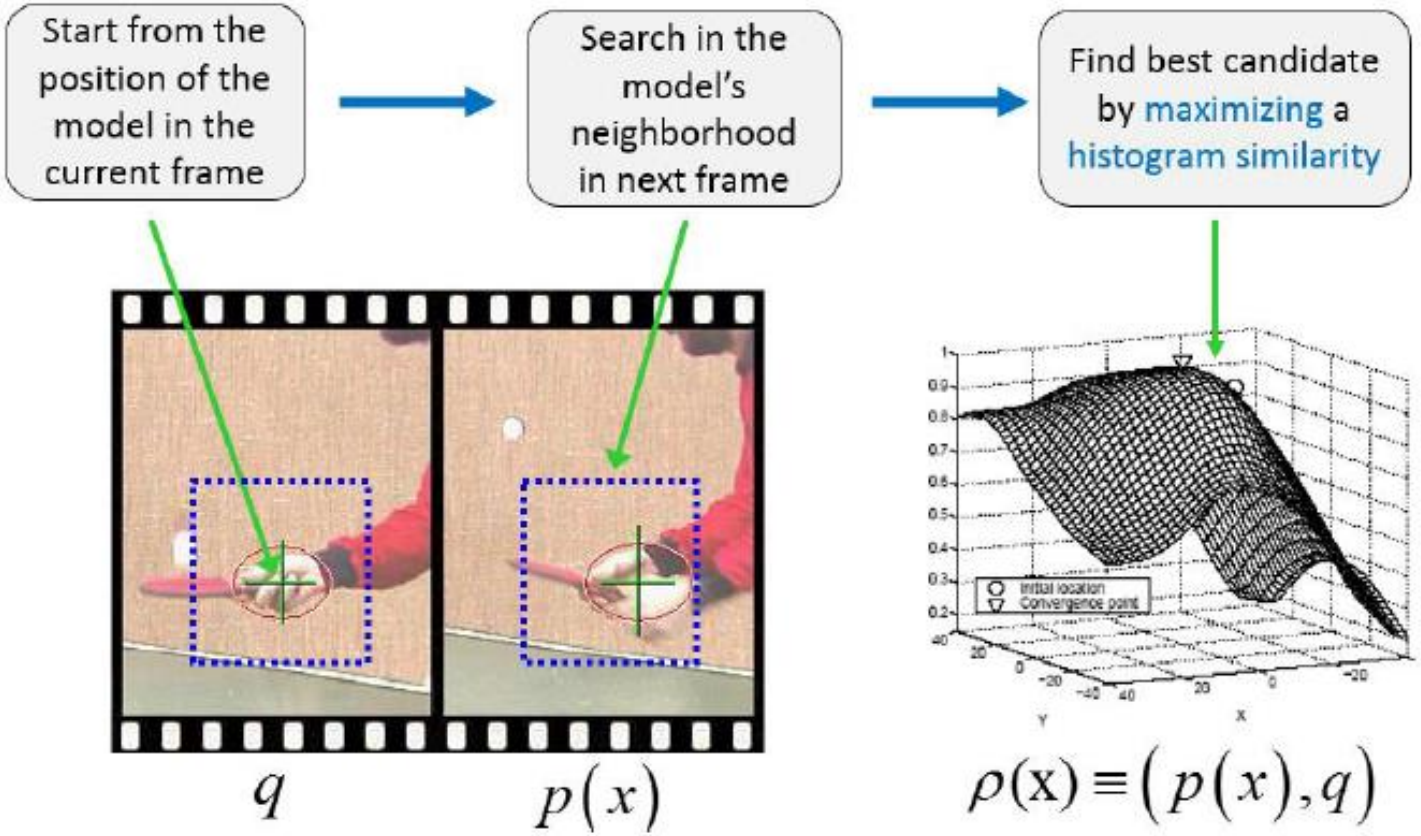
ACVM 2: Patch tracking I

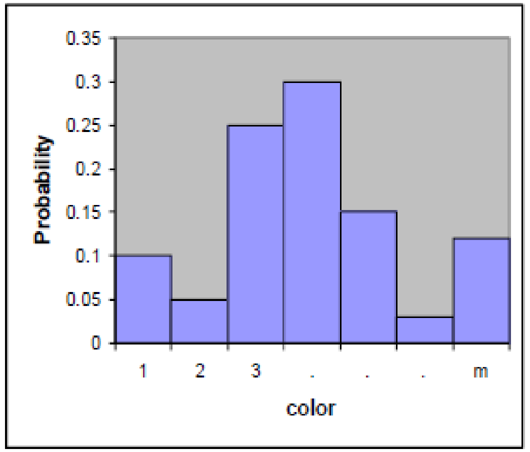




ACVM 3: Patch tracking II

- Template is a histogram!





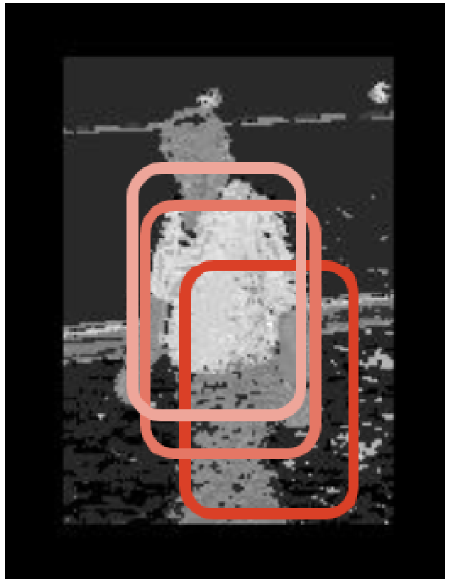
- Gradient ascent on Bahttacharrya distance is carried out by Mean Shift!

ACVM 3: Patch tracking II

- Deterministic gradient ascent using a histogram is achieved by Mean Shift!
- Implemented via backprojection

Mean Shift is simple: calculate the weighted mean position!

Weight image (backprojection)

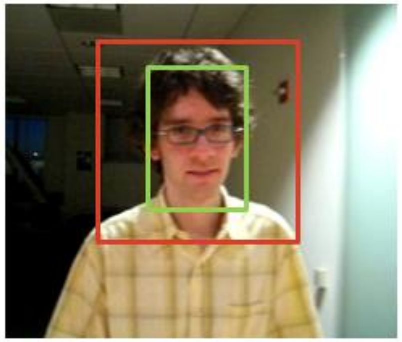


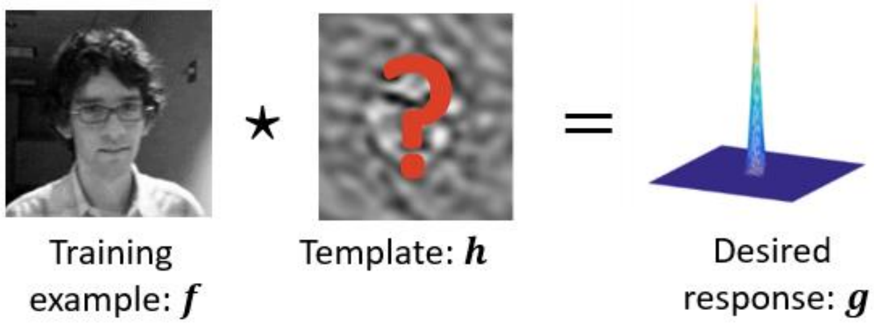
$$w_i = \sqrt{\frac{q_{b(x_i)}}{p_{b(x_i)}(x_0)}}$$

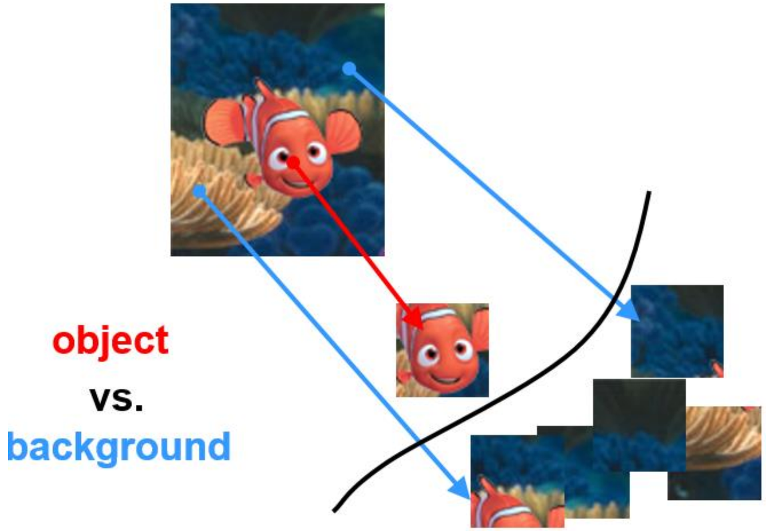
$$x^{(k+1)} = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i}$$

ACVM 4: Discriminative tracking

- Classification-based tracking







Correlation-based tracking

$$\arg\min_{\mathbf{h}} |\mathbf{f} \star \mathbf{h} - \mathbf{g}|^2$$

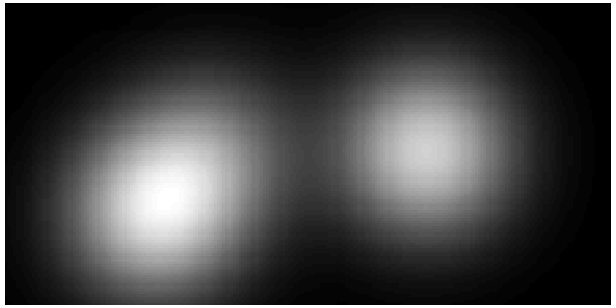
ACVM 5: Recursive Bayes Filtering (RBF)

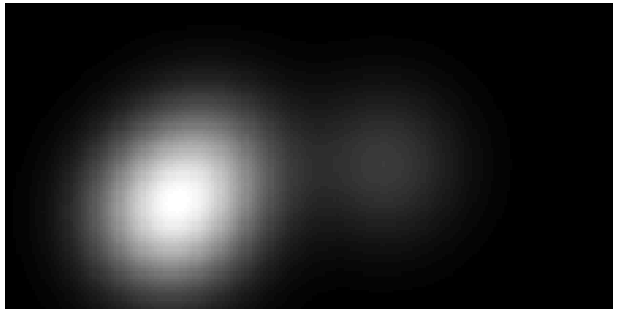
- Combine prior knowledge, dynamic model and observations
- Reason about the target in terms of pdfs

Observe a scene at *t*

$$p(\mathrm{B}\,|\,\mathrm{A}) = \frac{p(\mathrm{A}\,|\,\mathrm{B})\,\mathrm{p}(\mathrm{B})}{p(\mathrm{A})} \propto p(\mathrm{A}\,|\,\mathrm{B})\,\mathrm{p}(\mathrm{B})$$

Measurement pdf $*$ Prior knowledge (pdf) $\propto$ Posterior pdf





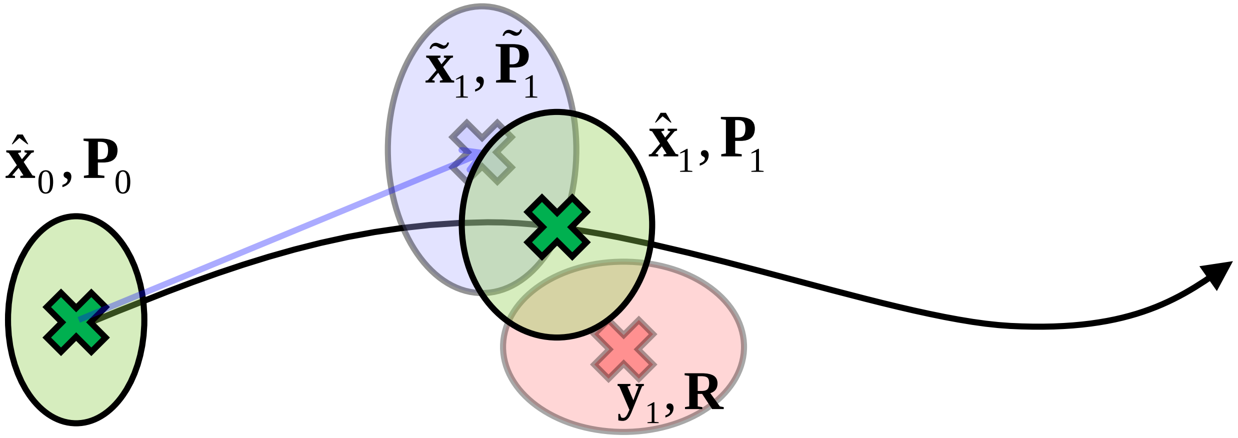
ACVM 5: RBF – Kalman filter

- Assume all pdfs are Gaussians

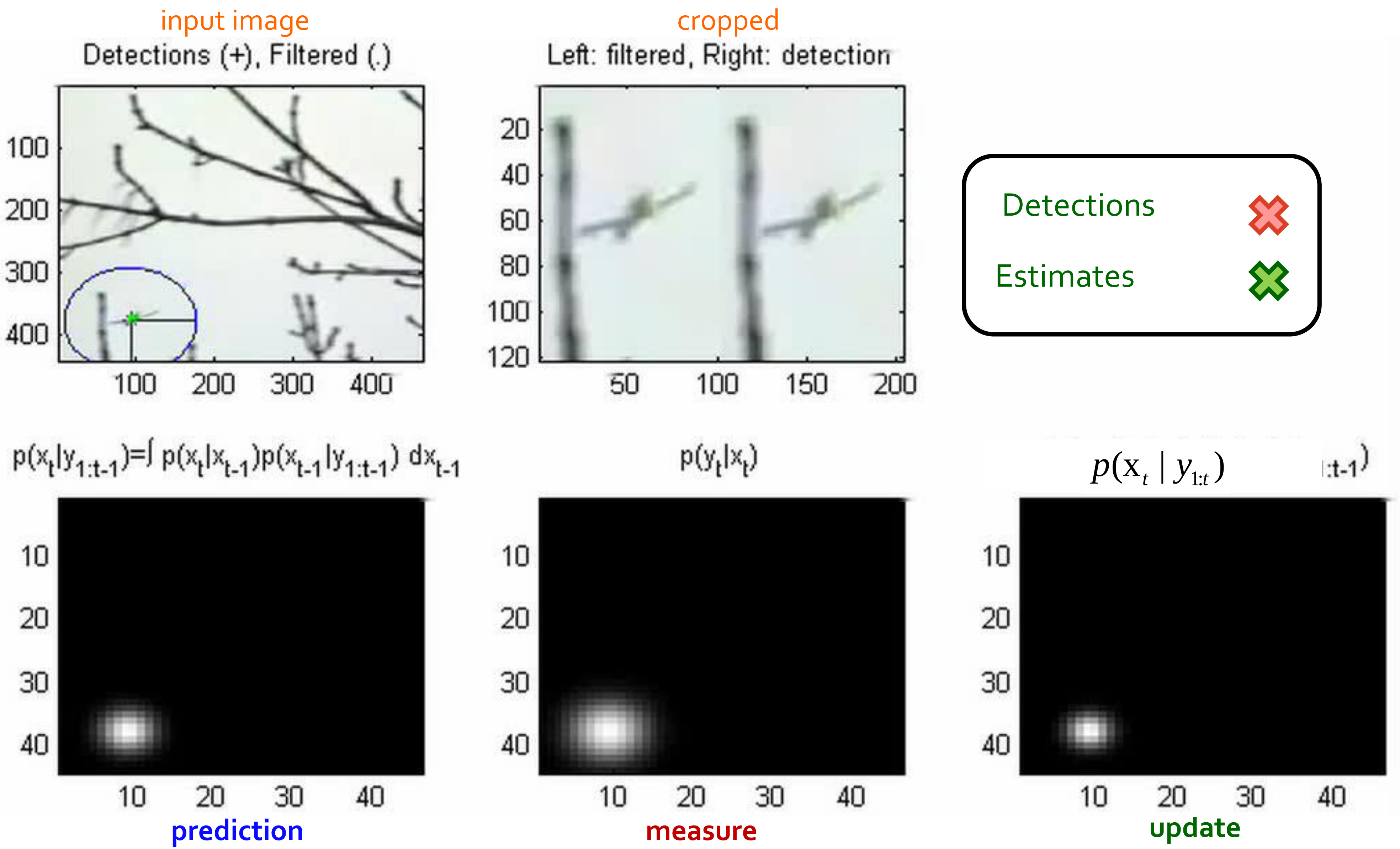
$$\underbrace{p(\mathbf{x}_k|\mathbf{y}_{1:k})}_{\text{posterior estimate}} \propto \underbrace{p(\mathbf{y}_k|\mathbf{x}_k)}_{\text{Observation model}} \int \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1})}_{\text{motion model}} \underbrace{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}_{\text{posterior at k-1}} d\mathbf{x}_{k-1}$$

- Predict: $\tilde{\mathbf{x}}_1 = \mathbf{\Phi}\hat{\mathbf{x}}_0 \;;\; \tilde{\mathbf{P}}_1 = \mathbf{\Phi}\mathbf{P}_0\mathbf{\Phi}^T + \mathbf{Q}$
- Update: $\hat{\mathbf{x}}_1 = \tilde{\mathbf{x}}_1 + \mathbf{K}(\mathbf{y}_1 - \mathbf{H}\tilde{\mathbf{x}}_1)$ $\quad\mathbf{K} = \tilde{\mathbf{P}}_1\mathbf{H}^T(\mathbf{H}\tilde{\mathbf{P}}_1\mathbf{H}^T + \mathbf{R})^{-1}$

$$\mathbf{P}_1 = (\mathbf{I} - \mathbf{K}\mathbf{H})\tilde{\mathbf{P}}_1$$



ACVM 5: RBF – Kalman filter

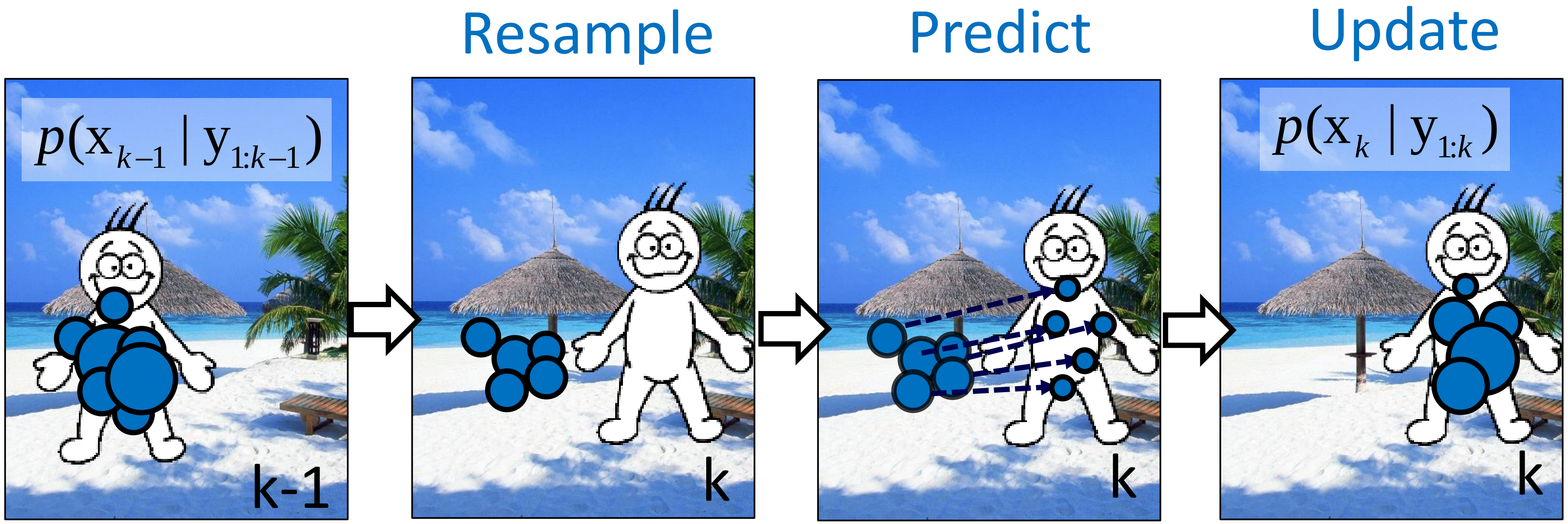


ACVM 5: RBF – Particle filter

- Assume arbitrary pdfs – particle representation

$$\underbrace{p(\mathbf{x}_k|\mathbf{y}_{1:k})}_{\text{posterior estimate}} \propto \underbrace{p(\mathbf{y}_k|\mathbf{x}_k)}_{\text{Observation model}} \int \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1})}_{\text{motion model}} \underbrace{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}_{\text{posterior at k-1}} d\mathbf{x}_{k-1}$$

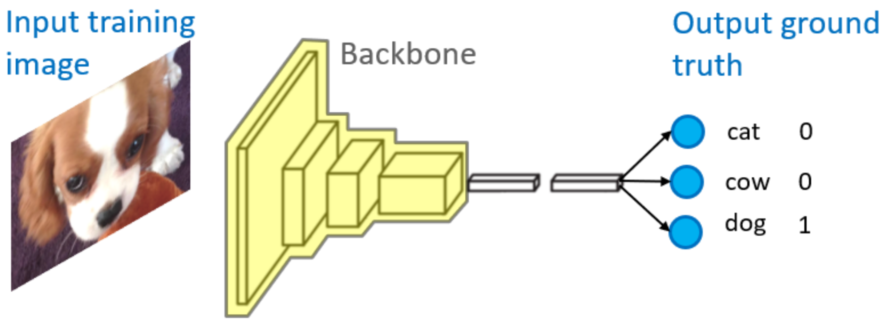
- Stochastic implementation by Monte Carlo



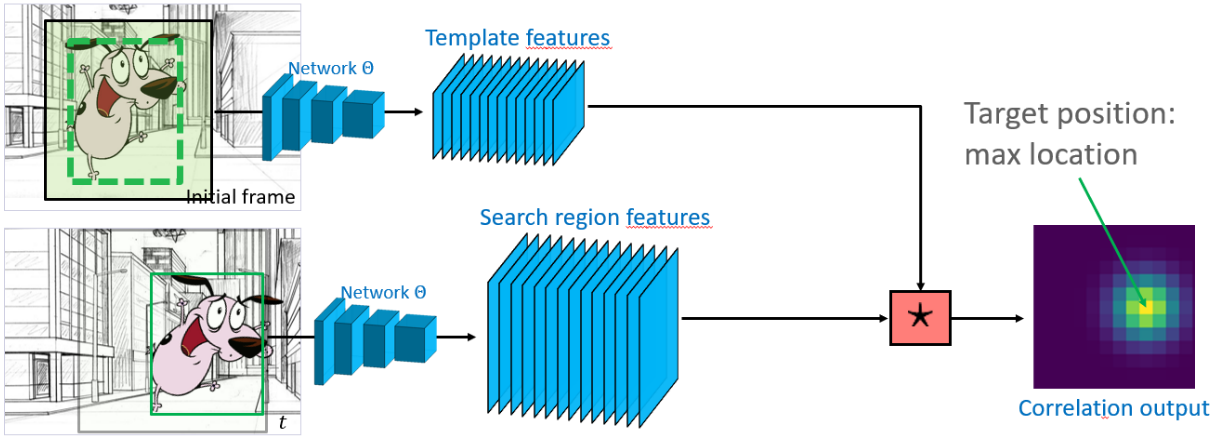
ACVM 5: RBF – Particle filter

ACVM 6: Deep learning trackers

- Convolutional neural networks
- Various tracker architectures
- Train features/parameters specific for tracking
- Target bounding box scale/aspect estimation



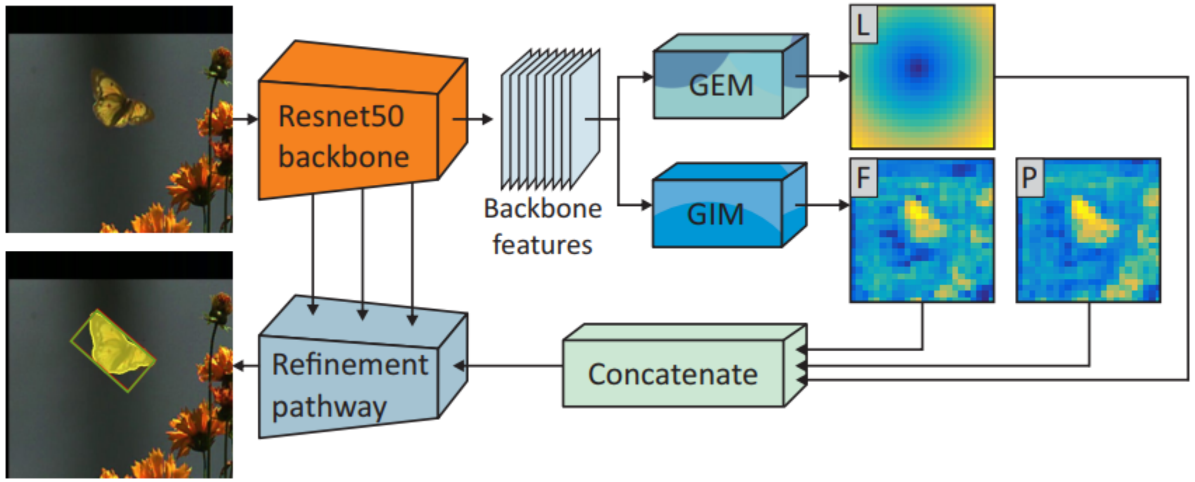
SiamFc

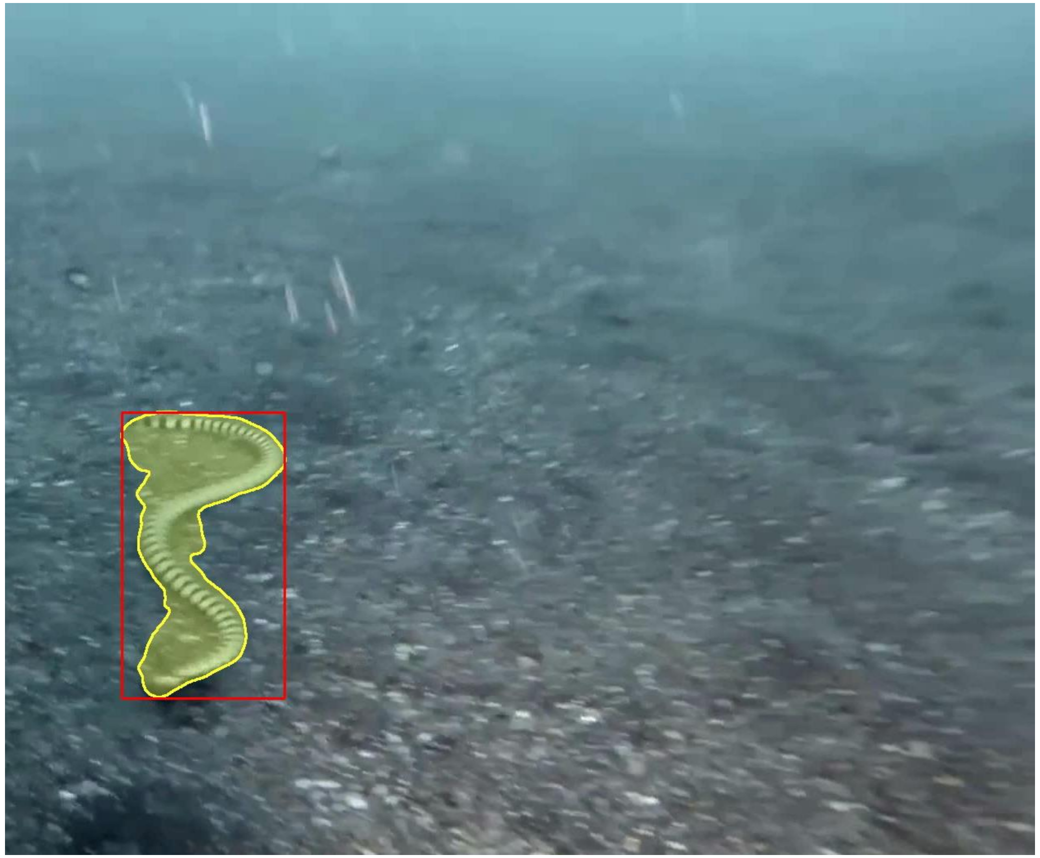


ACVM 6: Deep learning trackers

- Beyond bounding boxes: estimate target position by segmentation

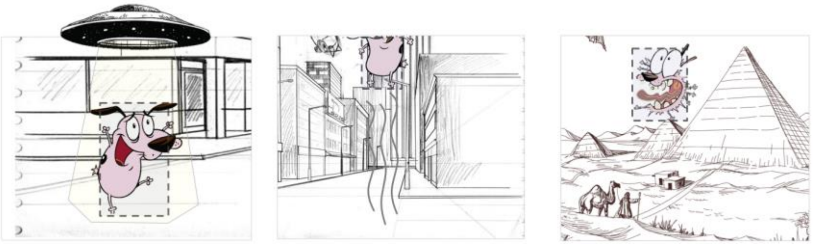
D3S



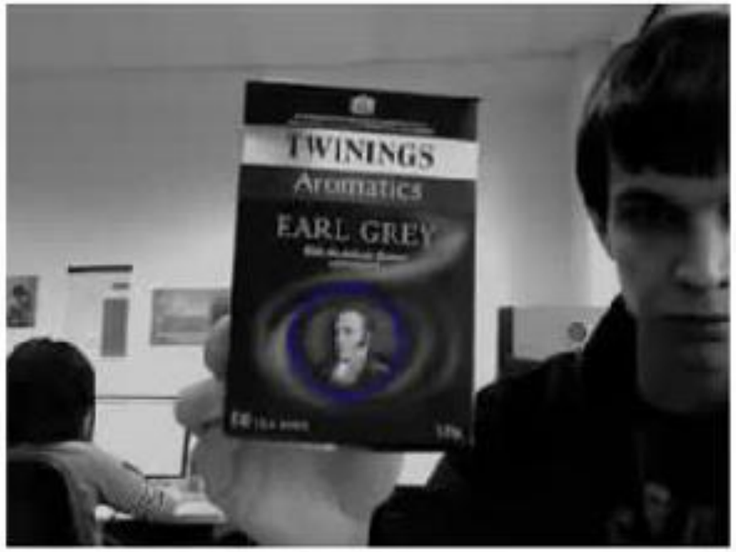


ACVM 7: Long-term tracking

- Carefully decide when to update the visual model
- Explicitly address reasons why tracking might fail
- Include re-detection



Predator (TLD)

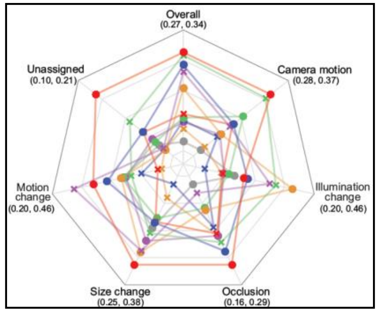


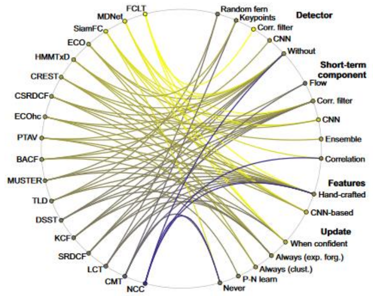
ALIEN

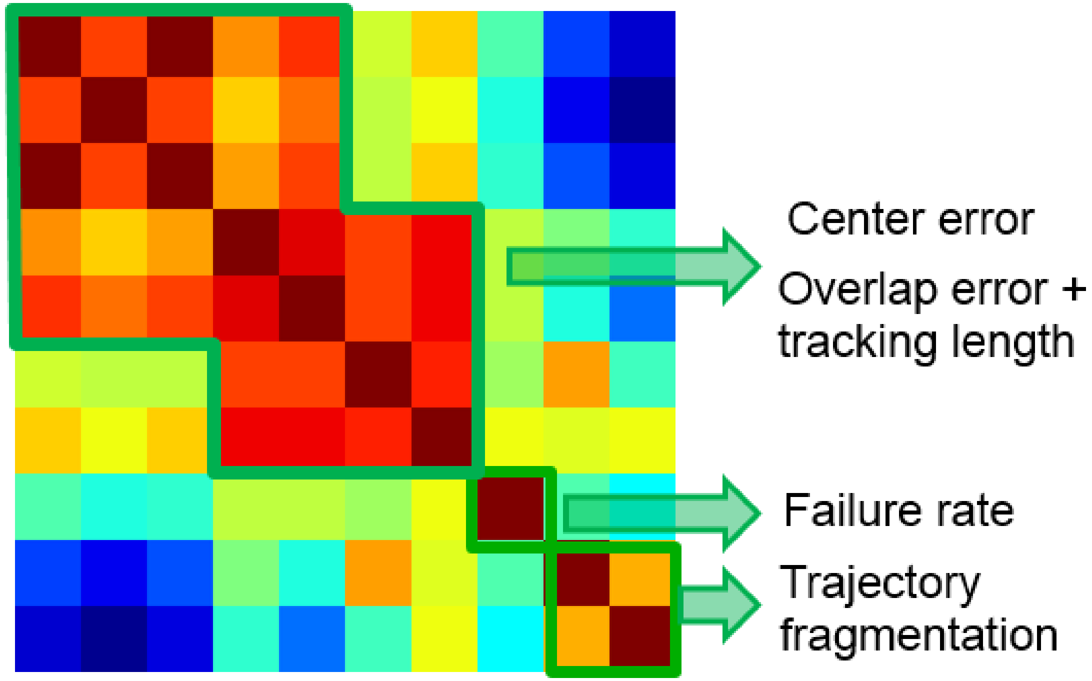
FCLT

ACVM 7: Performance evaluation

- Presented popular ST/LT performance measures







- Overviewed the VOT2022 results and pointed out the current state-of-the-art in short- and long-term tracking

The final steps…

ADVANCED COMPUTER VISION METHODS

The final grade structure

- Final grade: A*0.6 + C*0.4

NOTES:

- Positively pass all lab assignments (A) –required
- Homework (B) – not required, but desired
- Pass the written exam (C) –required

Written exam (C): Content

- Writing time: 45min
- A high-level overview of the topics we have covered
- E.g., will not have to derive a linearization of a motion model
- But will have to demonstrate that you understand how a particular flow-estimation/tracking algorithm works
- Will have to know the steps of the algorithms
- Not only the tracker from the lab, but also what we have covered exclusively in lectures.
- Require at least 50% @ written exam to pass!

I hope ACVM has been a fruitful learning experience

- Alan and I will be available in the lab (ViCoS)/mail for consultations!
- This course was modified in *real-time where required !*

- There is a lot of room for improvement.
- Please leave constructive remarks at the Studis polls.
- Always welcome to our lab for a tracking project
  (general computer vision as well, of course – detection, robotics, etc.)
  - For the 1st year students: Master's theses topics available

A list of other Computer-vision-oriented courses

- Bachelor’s level:
  - Machine Perception (Matej Kristan, Vicos)
  - Multimedia Systems (Luka Čehovin, Vicos)
  - Development of Intelligent Systems (Danijel Skočaj, Vicos)
  - Hand’s-on embedded computer vision
    (Luka Čehovin, Vicos – but might not be available for master’s level students. Perhaps as RVP(?))
- Master’s level
  - Deep learning (Danijel Skočaj, Vicos)
  - Image based biometry (Peter Peer, CVLab)
  - Biomedical Signal and image Processing (Franc Jager, LBCSI)