Dense Optical Flow Wrapup

28.05.2014

TECHNISCHE UNIVERSITÄT DARMSTADT

visual inference

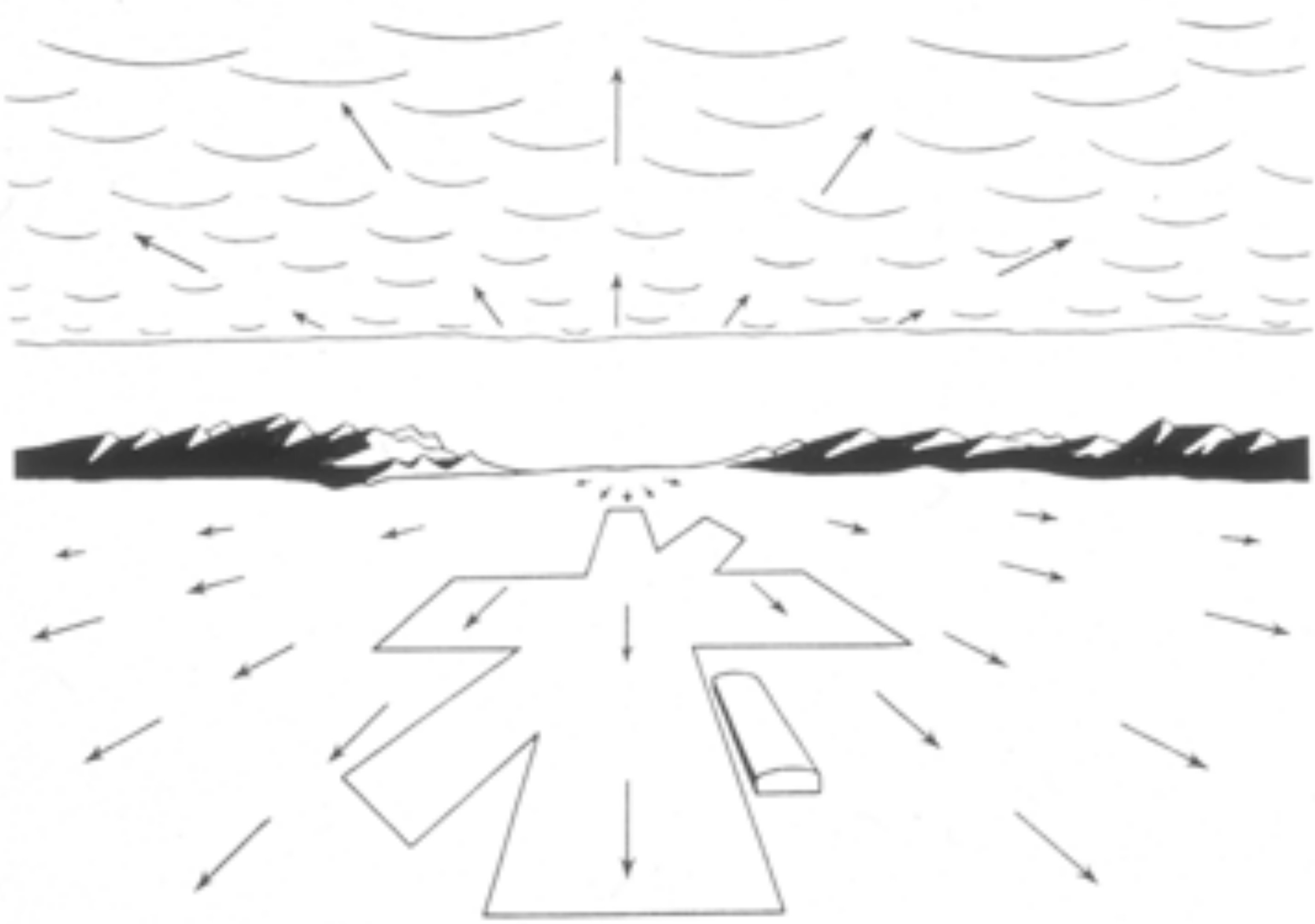
Announcements (Repeat)

- Exam:
  - Written exam on Wednesday, 6. Aug. 2014, 10:00-12:00
  - Room S2|02 C205
  - See TUCaN
  - Please make sure that you are registered!

- TU meet and move:
  - 4. June 2014
  - We will have our lecture and finish in time (11:30)

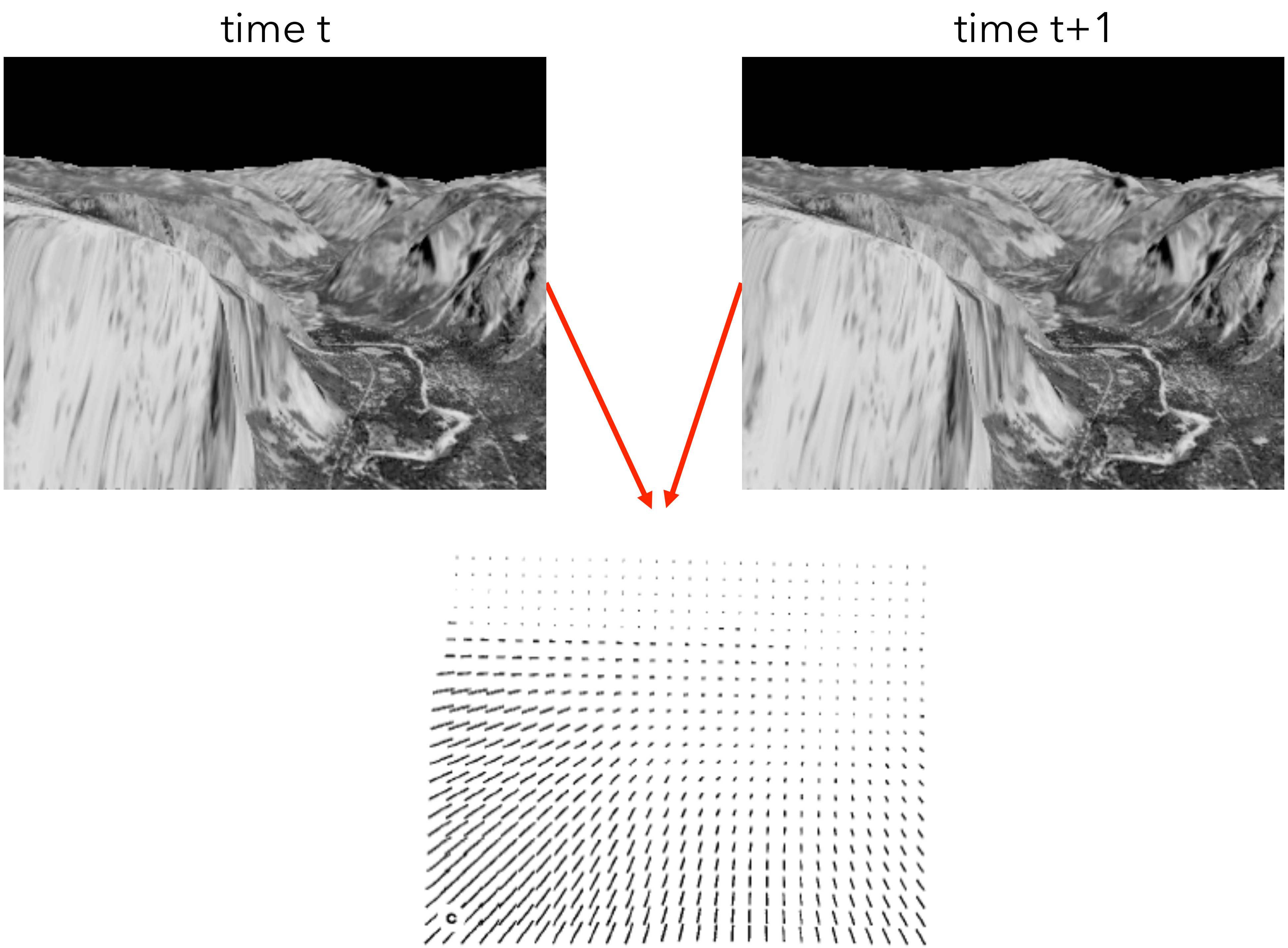
Optical Flow

TECHNISCHE UNIVERSITÄT DARMSTADT



J. J. Gibson, The Ecological Approach to Visual Perception

Optical Flow Estimation



Horn & Schunk

- The classical global optical flow technique was developed by Horn & Schunck:
  - B. K. P. Horn and B. G. Schunck. Determining optical flow. Artificial Intelligence, 17(1–3):185–203, 1981.
- It formulates optical flow as a problem of energy minimization.
  - Ideally they would like to minimize the following energy:

$$E(u,v) = \int \big(I(x+u(x,y), y+v(x,y), t+1) - I(x,y,t)\big)^2 +$$
$$\lambda \cdot \left(||\nabla u(x,y)||^2 + ||\nabla v(x,y)||^2\right) \, \mathrm{d}x\mathrm{d}y$$

Where does this energy come from?

Brightness difference between corresponding pixels

Quadratic penalty for brightness changes

$$E(u, v) = \int \left(I(x + u(x, y), y + v(x, y), t + 1) - I(x, y, t)\right)^2 +$$
$$\lambda \cdot \left(||\nabla u(x, y)||^2 + ||\nabla v(x, y)||^2\right) \, \mathrm{d}x\mathrm{d}y$$

Regularization parameter

Gradient magnitude of horz. flow
Penalizes changes in flow

Same for vertical flow

Linearizing the Brightness Constancy Assumption

$$E(u,v) = \int \big(I(x+u(x,y), y+v(x,y), t+1) - I(x,y,t)\big)^2 + \lambda \cdot \left(||\nabla u(x,y)||^2 + ||\nabla v(x,y)||^2\right) \, \mathrm{d}x\mathrm{d}y$$

- If we tried to minimize this directly to find the flow, we would run into problems, because the energy is non-convex and has many local optima.
  - We've had this problem before.
  - We linearized the brightness constancy assumption.
  - Will do the same here:

$$E(u,v) = \int \big(I_x(x,y,t)u(x,y) + I_y(x,y,t)v(x,y) + I_t(x,y,t)\big)^2 + \lambda \cdot \left(||\nabla u(x,y)||^2 + ||\nabla v(x,y)||^2\right) \, \mathrm{d}x\mathrm{d}y$$

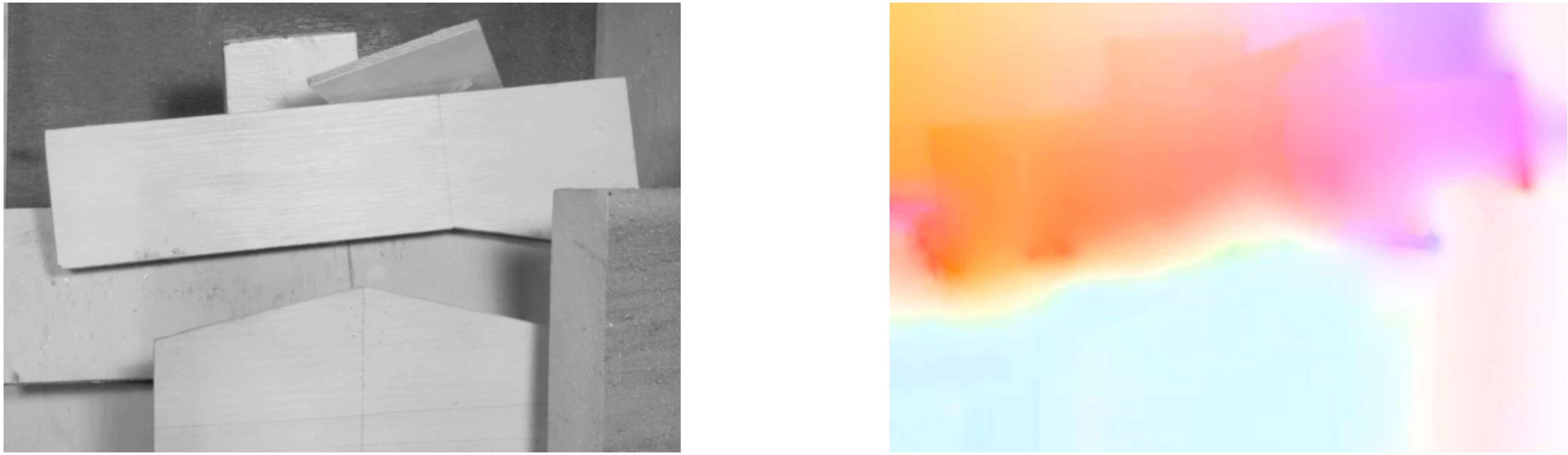
Actual H&S Energy

TECHNISCHE UNIVERSITÄT DARMSTADT

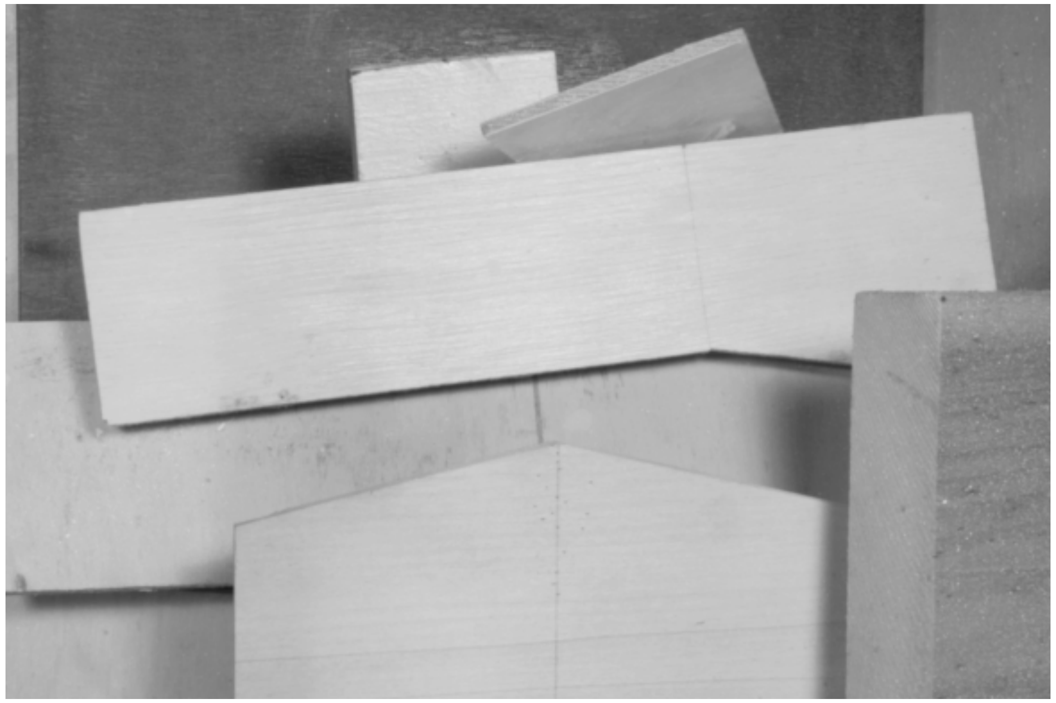
$$E(u,v) = \int \left(I_x(x,y,t)u(x,y) + I_y(x,y,t)v(x,y) + I_t(x,y,t) \right)^2 + \lambda \cdot \left(||\nabla u(x,y)||^2 + ||\nabla v(x,y)||^2 \right) \,\mathrm{d}x\mathrm{d}y$$

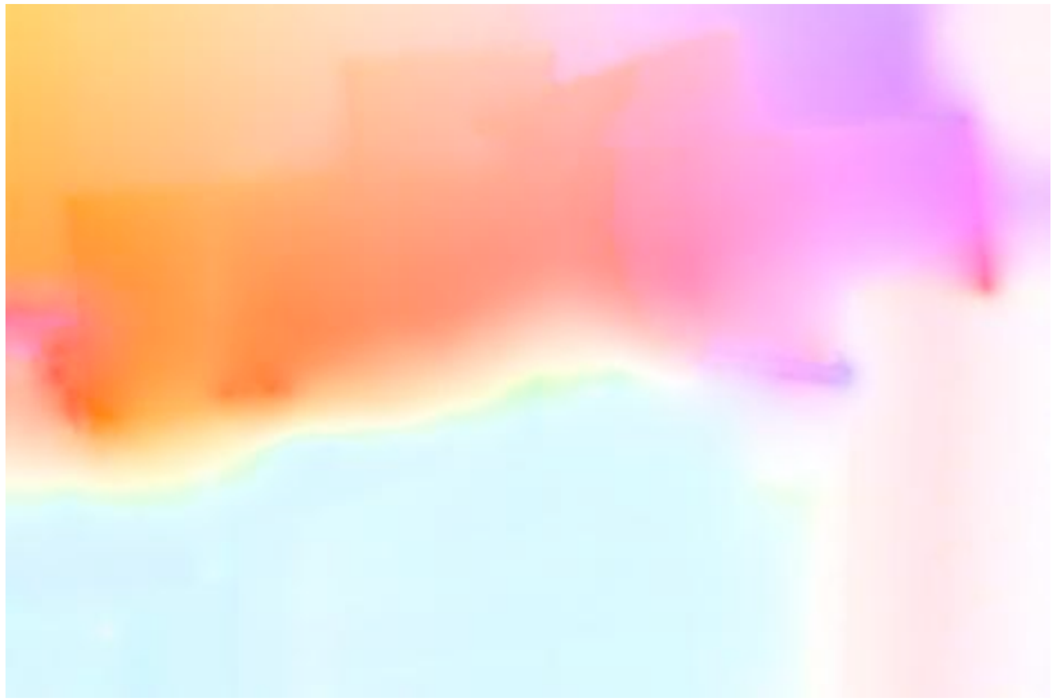
- This energy imposes a quadratic penalty on the optical flow constraint (OFC).
- It is convex and thus has a unique optimum.
- We estimate flow by discretizing it spatially and doing gradient descent.
- Old problem:
  - The linearization only works for small motions.
  - Hence we estimate the flow using coarse-to-fine refinement with warping.

Some H&S results



Some H&S results





Local vs. Global Methods

Q: What is the conceptual difference?

Q: What are the trade-offs?

Problem with H&S

- These results are quite a bit better than what we can get with the LK method, but one problem remains.
- The flow is **very smooth**:
  - In particular, the method smoothes over the discontinuities that are in the flow.
- Why?
  - We use a quadratic penalty to penalize changes in the flow.
  - We have seen that this does not allow for discontinuities.
  - It penalizes large changes too much.

Probabilistic Formulation

TECHNISCHE UNIVERSITÄT DARMSTADT

- We will do what we have already done several times:
  - We model the problem of optical flow estimation as one of probabilistic inference.
  - This allows us to motivate the formulation from observations on the data.
- Posterior for optical flow estimation:

$$p(\boldsymbol{u}, \boldsymbol{v} | I^0, I^1) \propto \underbrace{p(I^1 | \boldsymbol{u}, \boldsymbol{v}, I^0)}_{\text{Observation model embodies brightness constancy}} \cdot \underbrace{p(\boldsymbol{u}, \boldsymbol{v} | I^0)}_{\text{"Prior" on the flow field}}$$

Observation model
embodies brightness constancy

"Prior" on the flow field

Probabilistic Formulation

$$p(\boldsymbol{u}, \boldsymbol{v} | I^0, I^1) \propto p(I^1 | \boldsymbol{u}, \boldsymbol{v}, I^0) \cdot p(\boldsymbol{u}, \boldsymbol{v} | I^0)$$

- We simplify this further to obtain a model based on a real prior on optical flow:

$$p(\boldsymbol{u}, \boldsymbol{v} | I^0, I^1) \,\tilde{\propto}\, p(I^1 | \boldsymbol{u}, \boldsymbol{v}, I^0) \cdot p(\boldsymbol{u}, \boldsymbol{v})$$

- Modeling the observation model:
  - As usual, we assume conditional independence of the pixel sites:

$$\begin{aligned} p(I^1 | \boldsymbol{u}, \boldsymbol{v}, I^0) &= \prod_{i,j} p(I^1 | u_{ij}, v_{ij}, I^0_{ij}) \\ &= \prod_{i,j} p(I^1(i + u_{ij}, j + v_{ij}) | I^0_{ij}) \end{aligned}$$

Observation Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

- Model the likelihood based on brightness differences:

$$\begin{aligned} p(I^1|\boldsymbol{u}, \boldsymbol{v}, I^0) &= \prod_{i,j} p(I^1(i + u_{ij}, j + v_{ij})|I^0_{ij}) \\ &= \prod_{i,j} f_D(I^1(i + u_{ij}, j + v_{ij}) - I^0_{ij}) \end{aligned}$$

  - Compute $I^1(i + u_{ij}, j + v_{ij})$ using interpolation (bilinear or bicubic).
  - We could simply assume a Gaussian distribution on the brightness differences.
  - From stereo, we know that this is not a good assumption, because of occlusions, etc.
  - We need to use a robust observation model.

Flow Prior

- From what we saw in stereo and image processing, we know that we can model such priors as MRFs:

$$
\begin{aligned}
p(\boldsymbol{u}, \boldsymbol{v}) &= p(\boldsymbol{u}) \cdot p(\boldsymbol{v}) \\
&\propto \prod_{ij} f_H(u_{ij} - u_{i+1,j}) \cdot f_V(u_{ij} - u_{i,j+1}) \cdot \\
&\qquad f_H(v_{ij} - v_{i+1,j}) \cdot f_V(v_{ij} - v_{i,j+1})
\end{aligned}
$$

- This assumes that the horizontal and vertical flow are independent, and that this simple 4 neighborhood is sufficient to model flow.
- If we choose the potentials as Gaussians and take the negative log, we get back the Horn & Schunck energy, e.g.:

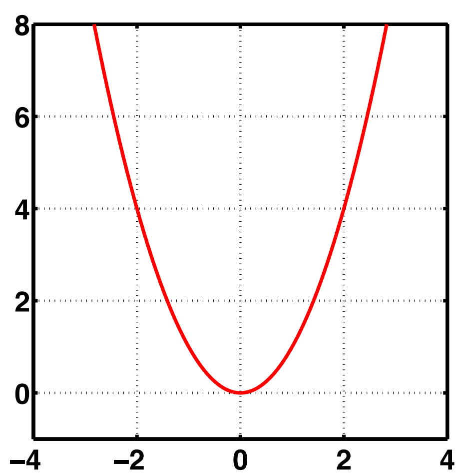
$$
-\log\left(\mathcal{N}(u_{ij} - u_{i+1,j}; 0, 1)\mathcal{N}(u_{ij} - u_{i,j+1}; 0, 1)\right) + C \approx \left(\frac{\partial u(i,j)}{\partial x}\right)^2 + \left(\frac{\partial u(i,j)}{\partial y}\right)^2 = ||\nabla u(i,j)||^2
$$

Flow Prior

- Of course we know that this didn't work.
  - We wanted a prior that allows for discontinuities in the flow field.
  - We need something more heavy-tailed than a Gaussian, for example a Student-t distribution:

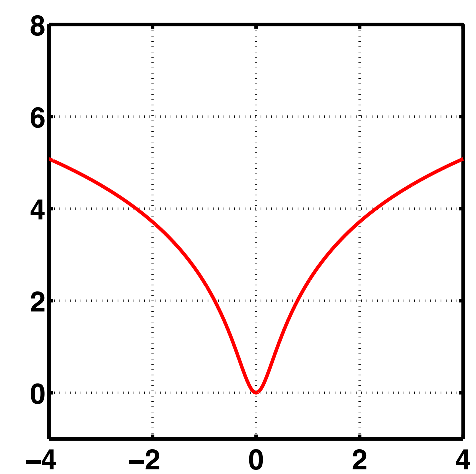
$$f_H(u_{ij} - u_{i+1,j}) = \left(1 + \frac{1}{2\sigma^2}(u_{ij} - u_{i+1,j})^2\right)^{-\alpha}$$

  - This has been proposed by [Black & Anandan, 92].



Gaussian

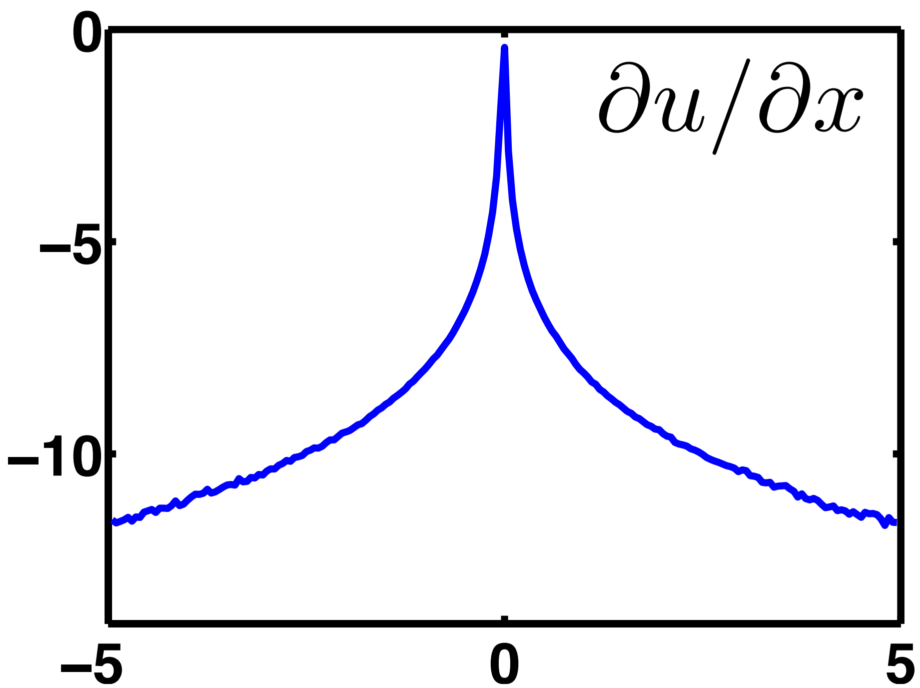
negative log-density (i.e. energy)

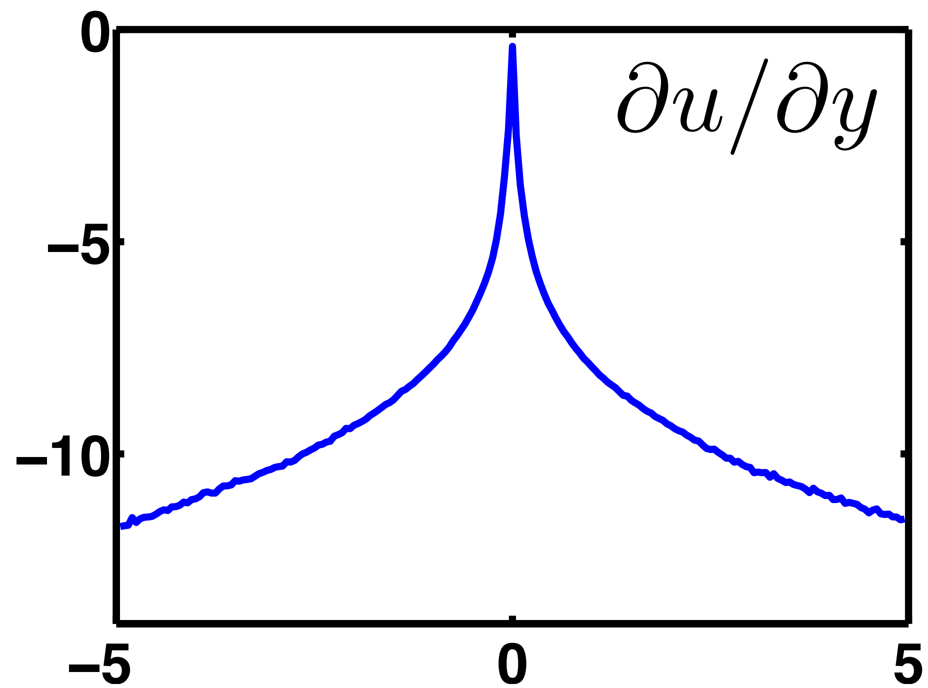


Student-t

Flow Prior

- This can be motivated by looking at the statistics of optical flow [Roth & Black, 05]:





  - Note: These are log-densities, i.e. the sign is reversed.
  - Very similar to a Student-t distribution!
  - But where did we get these statistics from?

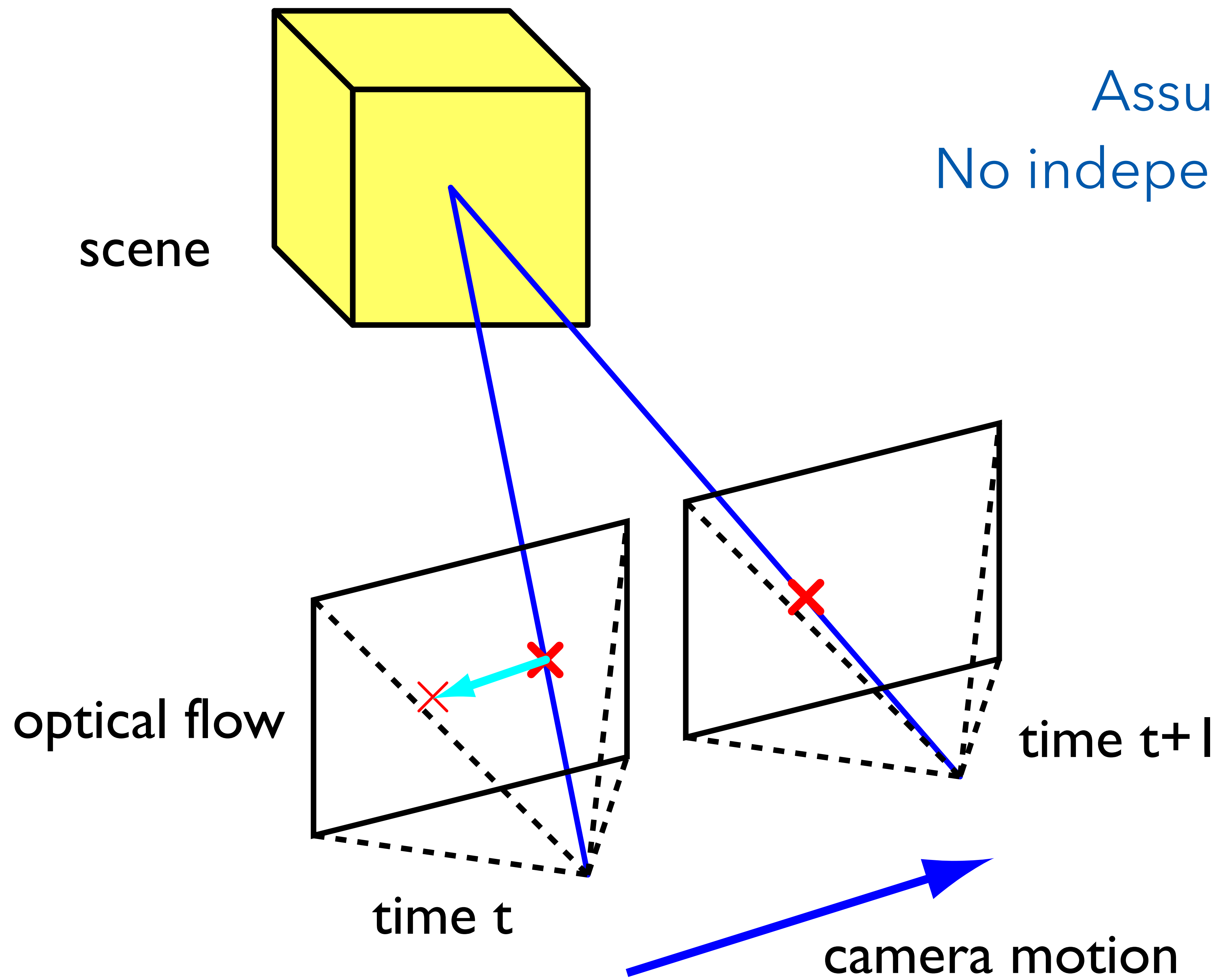
Aside: Flow Database

TECHNISCHE UNIVERSITÄT DARMSTADT

- We need an extensive database of optical flow fields.
- How do we obtain this?
  - Unlike natural images or scene depth:
    We cannot measure natural flow directly.
  - Using the UV light method is too labor intensive.
- Other Approach:
  Synthesize optical flow
  - Use set of “natural” scene geometries.
  - Use set of “natural” camera motions.
  - Database of 800 flow fields.

Synthesizing Flow

TECHNISCHE UNIVERSITÄT DARMSTADT



Assumption:
No independent motion.

Synthesizing Flow

- Need “natural” scene geometry
  ➔ Range images of natural scenes

Brown Range Image Database [Huang et al. 2000]: ~200 range images

Camera Motion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Also need “natural” 3D camera motions:
  - We used a database of 3D camera motions .
  - Extracted from video clips by 2d3 using boujou.
- Properties:
  - Hand-held and car-mounted cameras.
  - Various motion types: Moving into the scene, active fixation of objects, car driving on residential streets
  - ~100 clips

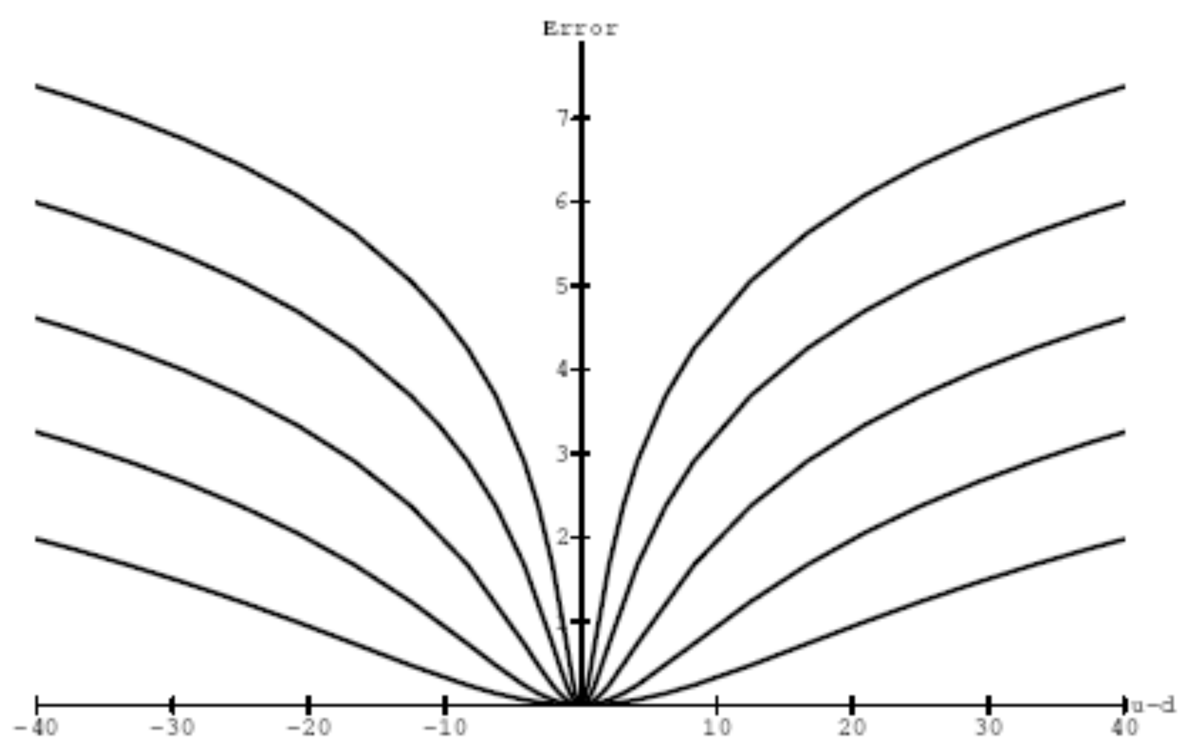
Example Clip

Flow Estimation

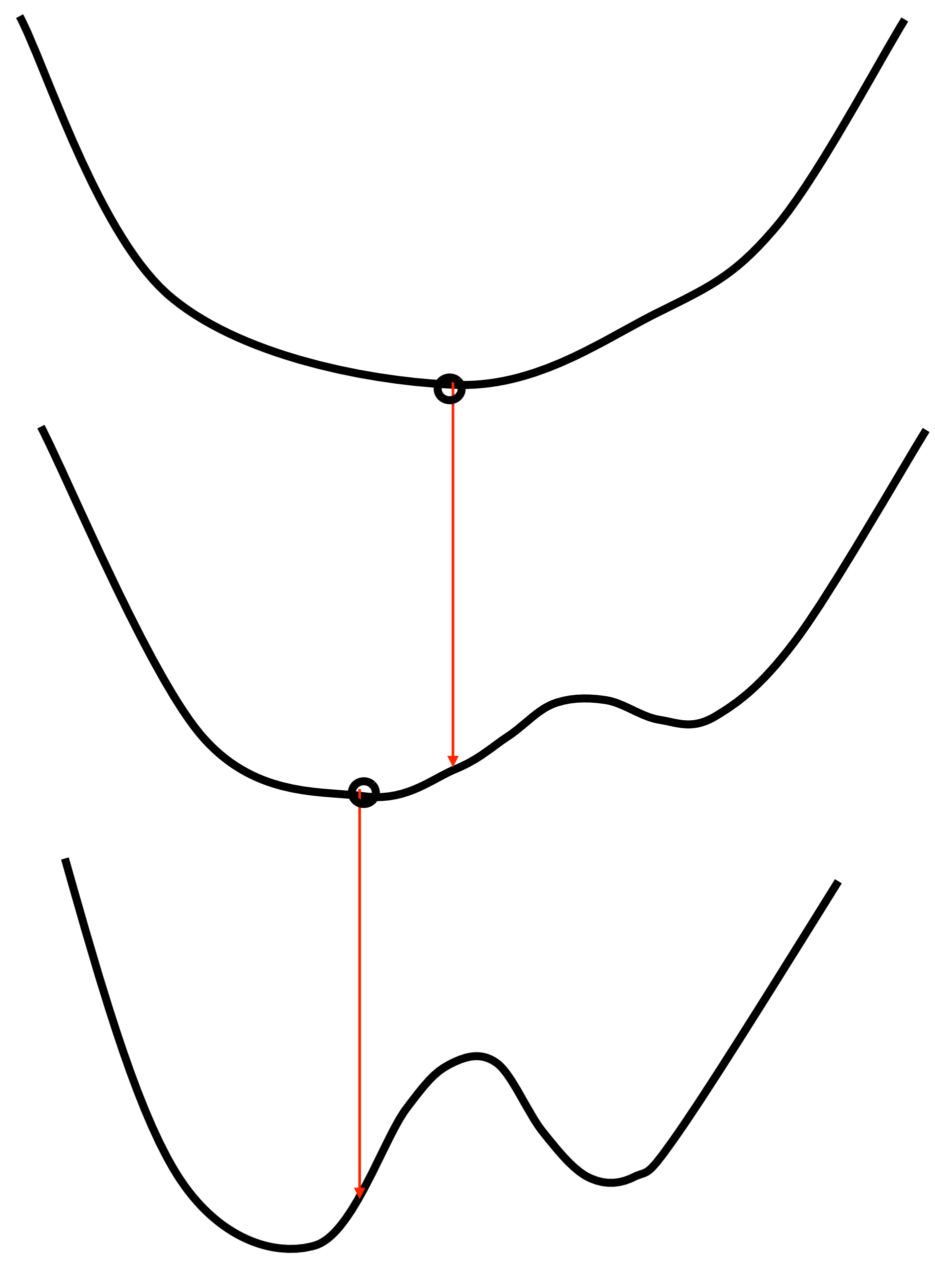
- Now we can put likelihood and prior together.
- But how do we actually estimate flow?
  - We need to use probabilistic inference, but...
- Discrete optimization is difficult:
  - At each pixel, we have a 2D flow vector, which is difficult to represent discretely (too many labels).
- Continuous optimization is also difficult:
  - Both data term and spatial term are non-convex in this formulation, which makes it difficult to apply simple gradient descent techniques.

Deterministic Annealing

- Start with a "quadratic" (i.e. convex) optimization problem, and optimize (locally).
- Then gradually increase the difficulty, i.e. non-convexity of the problem:



Continuation Method



GNC: Graduated Non-Convexity

Simple Synthetic Example

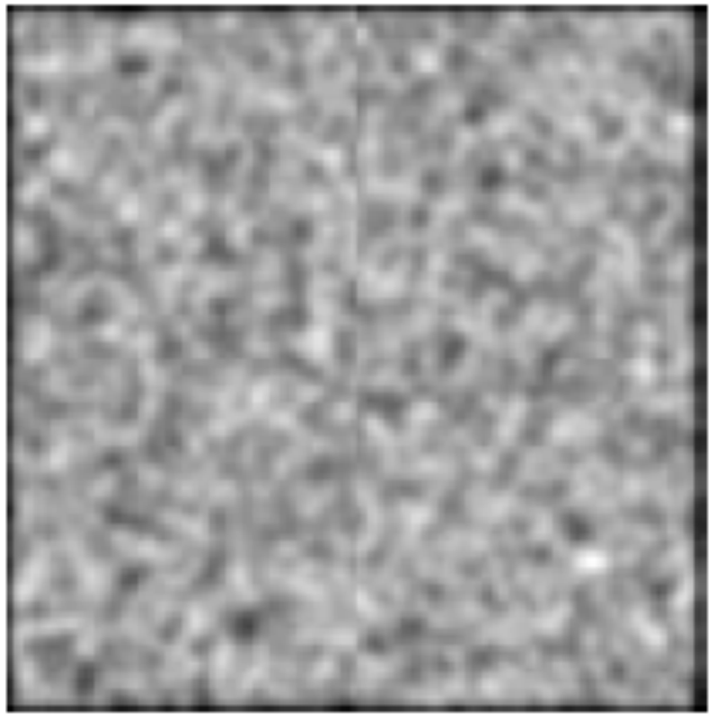


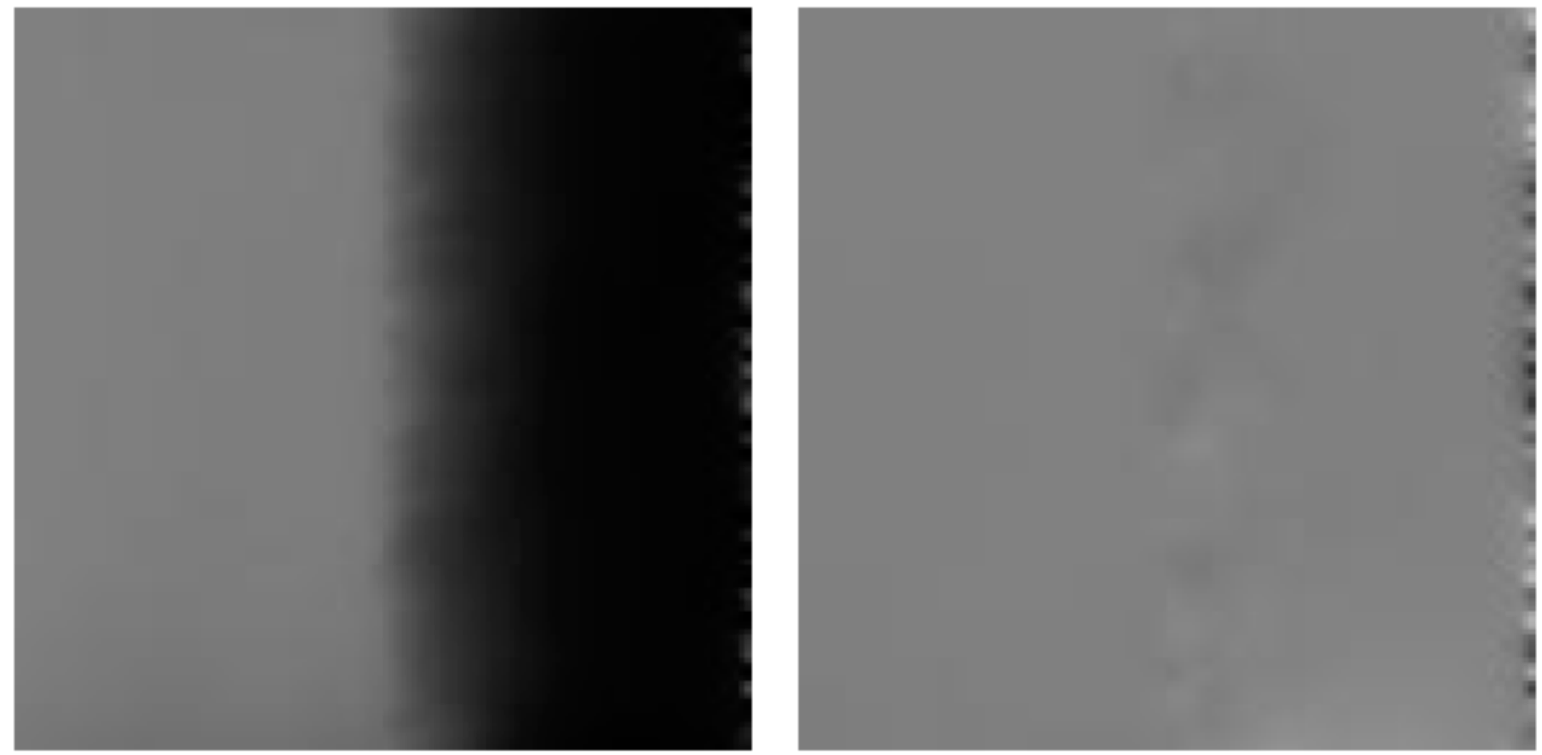
Image from sequence

horizontal flow

vertical flow

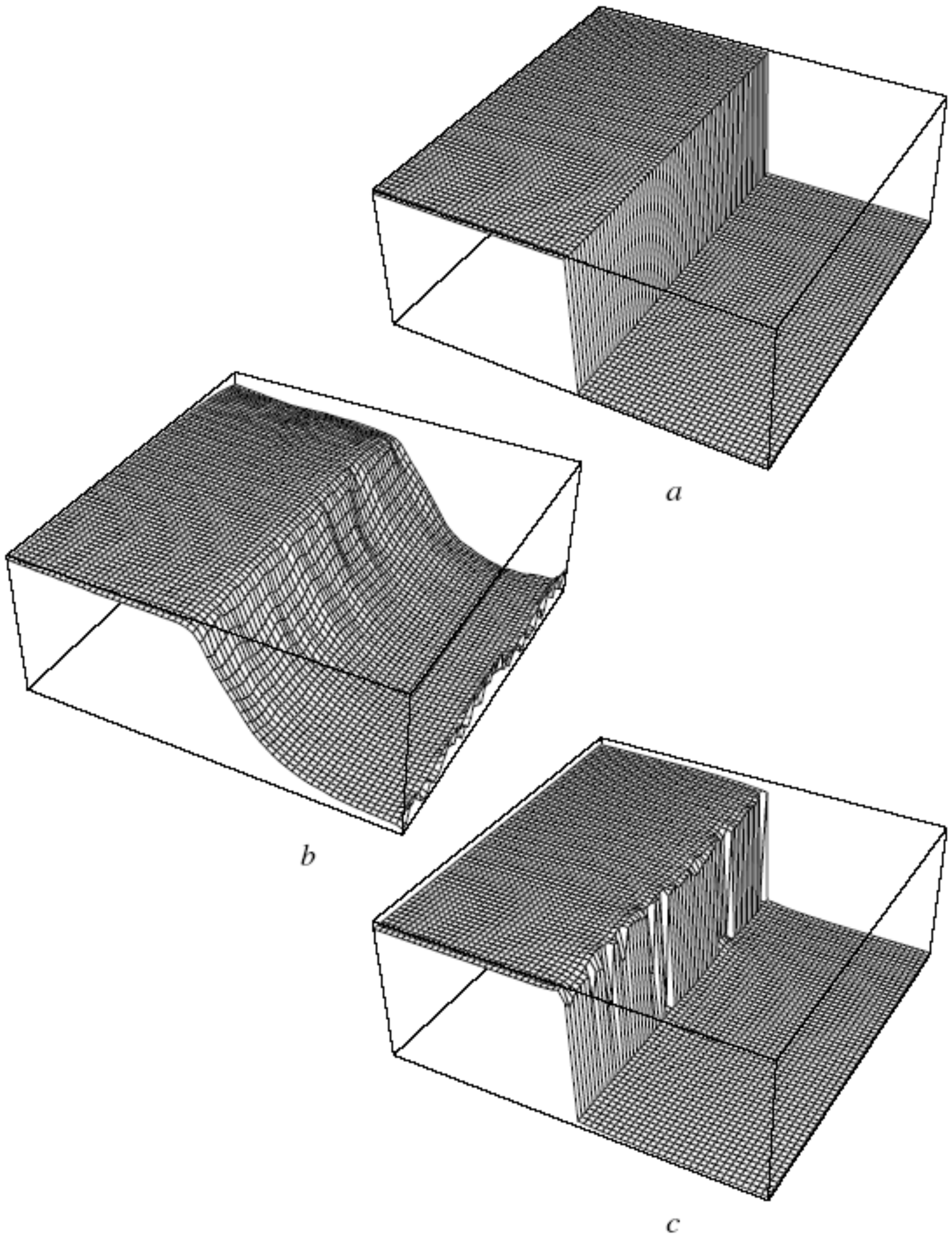
Simple Synthetic Example

Estimated flow with Gaussian spatial term:



Estimated flow with robust (Student-t) spatial term:

Horizontal Flow

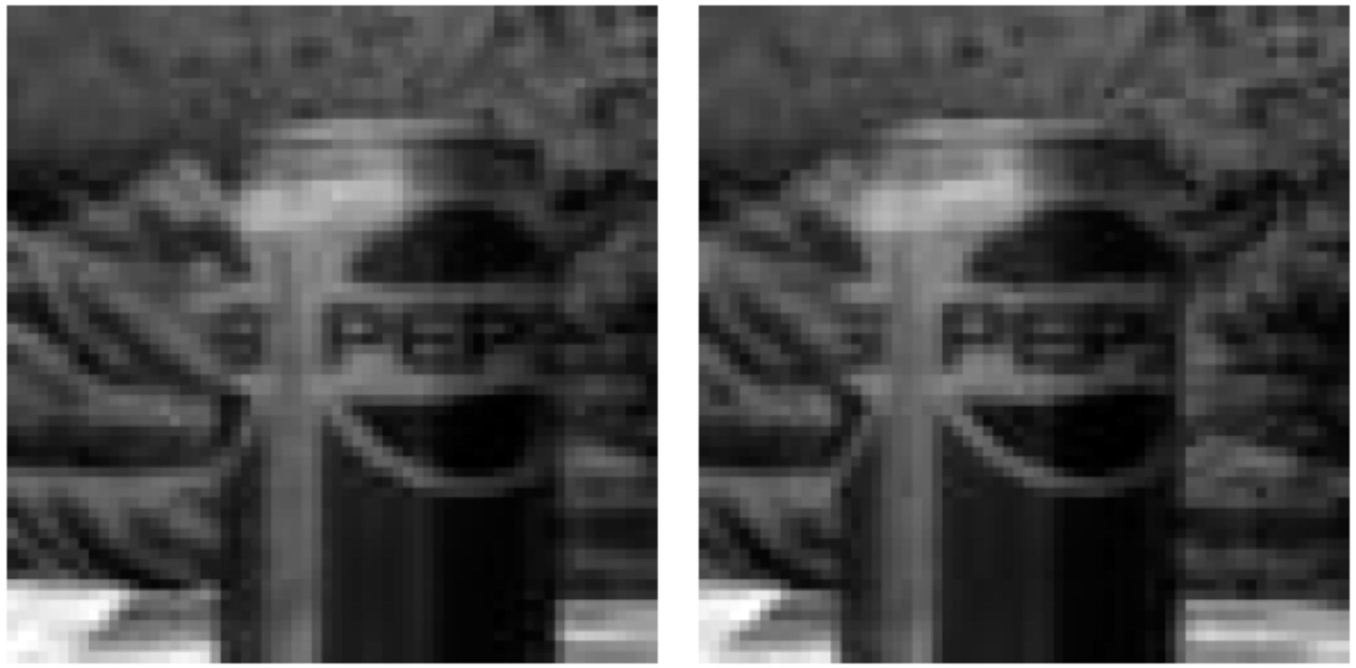


True horizontal flow

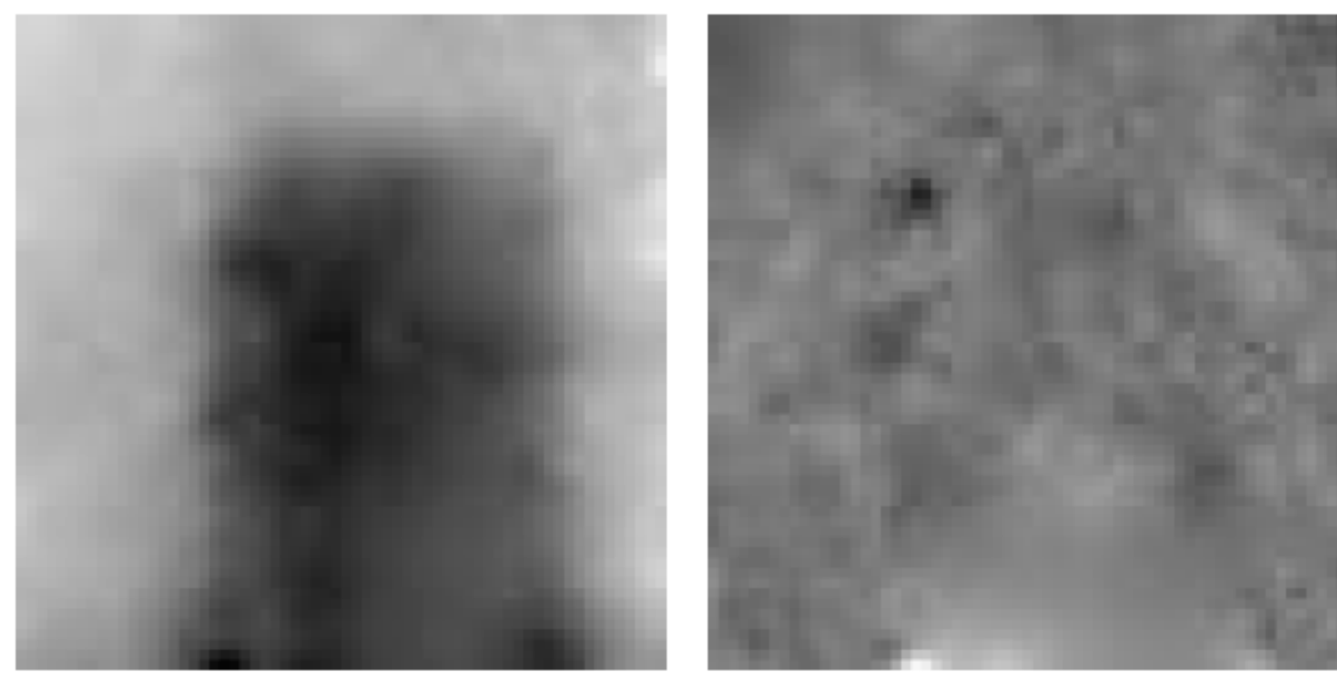
Estimated with Gaussian spatial term

Estimated flow with robust (Student-t) spatial term

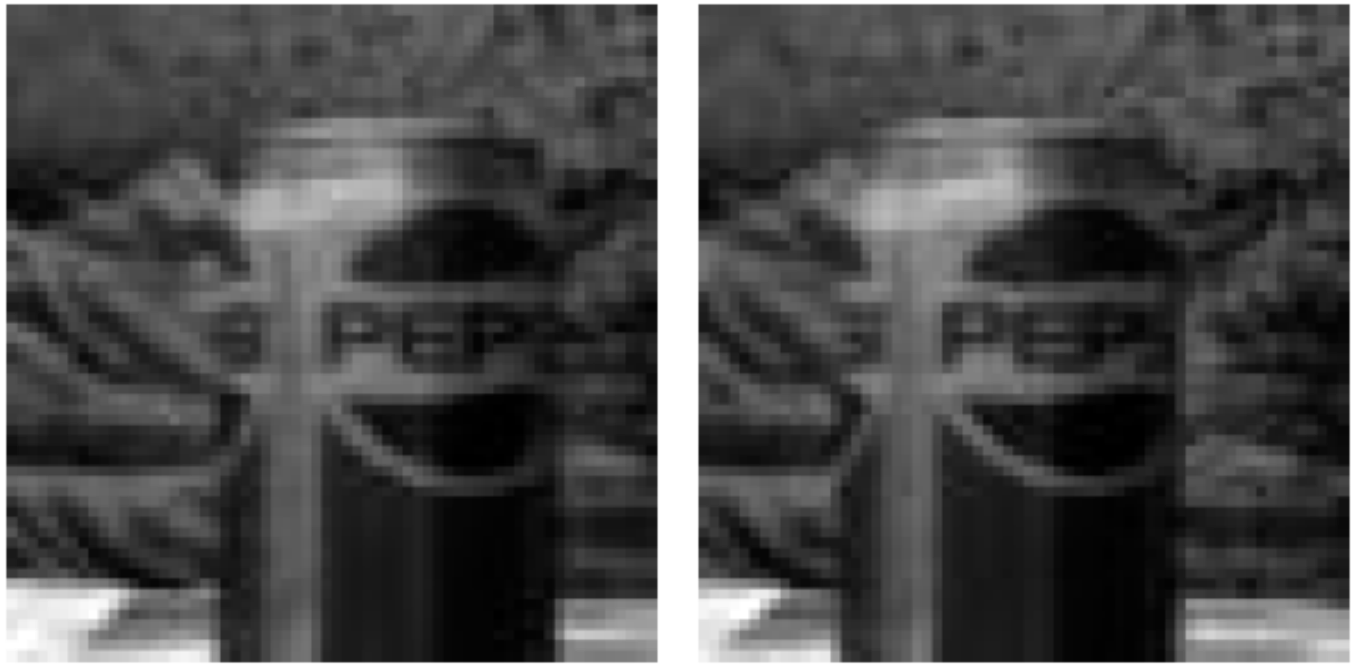
Real Sequence



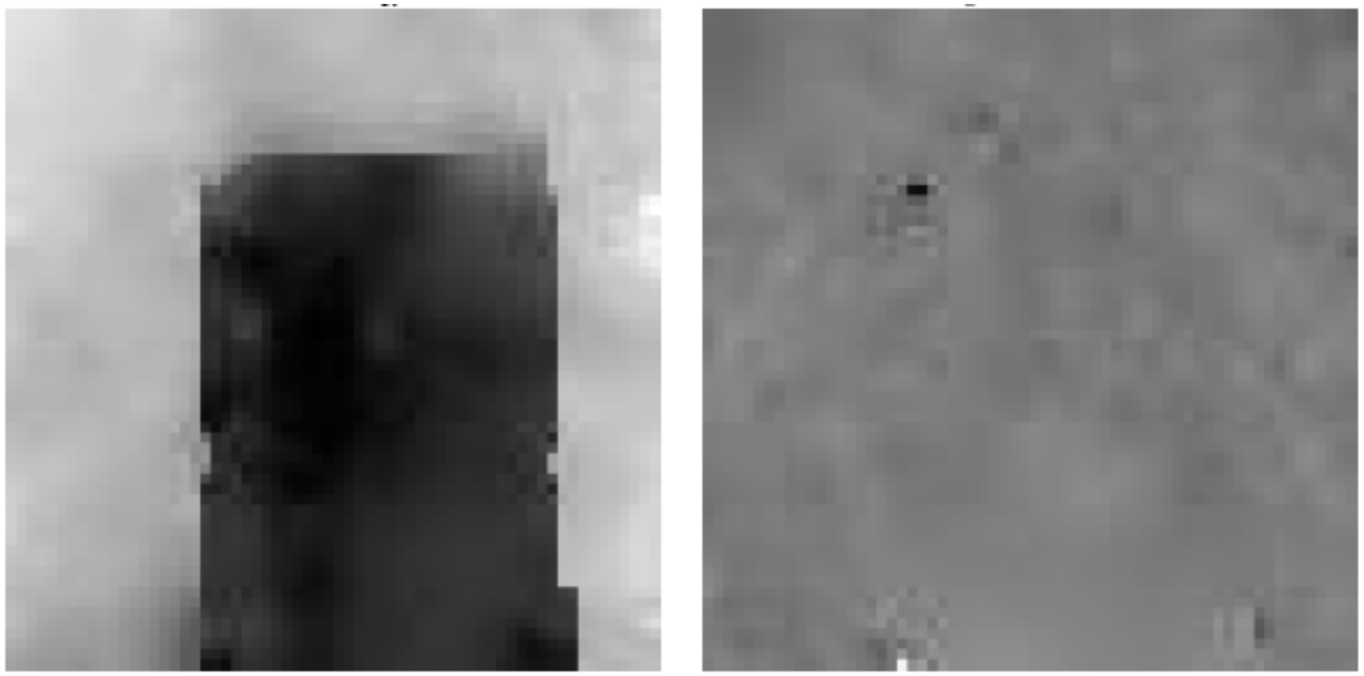
- Deterministic Annealing
- First stage:
  - (Almost) convex
  - Large σ



Real Sequence



- Deterministic Annealing
- Last stage:
  - Non-convex
  - Small σ



Current Methods

- Many current state-of-the-art methods sidestep these optimization issues (to some extent):
  - They use slightly robust, but still convex penalties.
  - This is not really justified by the statistics of the data, but it does make the optimization problem a whole lot easier.
  - Then the problem becomes convex if we linearize the brightness constancy assumption.
    - Otherwise, it is still non-convex, but nevertheless easier.
  - They make up for the loss in fidelity by extending the formulation in other ways.

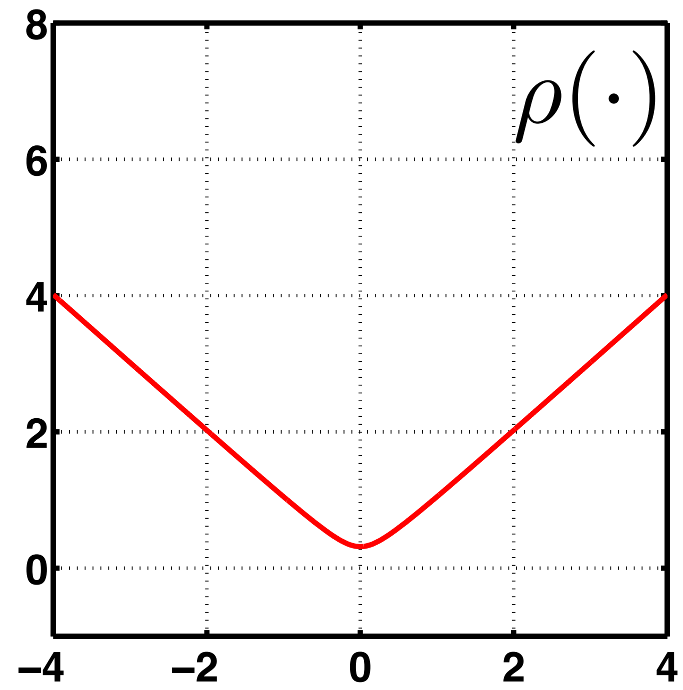
CLG Approach

- The CLG approach (combined local-global) combines the LK and the HS approaches in a single framework [Bruhn et al., 2005]:
  - The data term is modeled using the structure tensor, similar to the Lukas-Kanade method.
  - The spatial term is similar to what we've seen, and uses a convex penalty function.

$$E_{\mathrm{data}}(\mathbf{w}) = \int \rho\left(\sqrt{\mathbf{w}(x,y)^{\mathrm{T}}\mathbf{S}_\sigma(x,y)\mathbf{w}(x,y)}\right)\,\mathrm{d}x\,\mathrm{d}y$$

where

$$w = (u, v, 1)$$
$$\mathbf{S}_\sigma = G_\sigma * (\nabla_3 f \ \nabla_3 f^{\mathrm{T}})$$
$$\nabla_3 f = (f_x, f_y, f_t)^{\mathrm{T}}$$



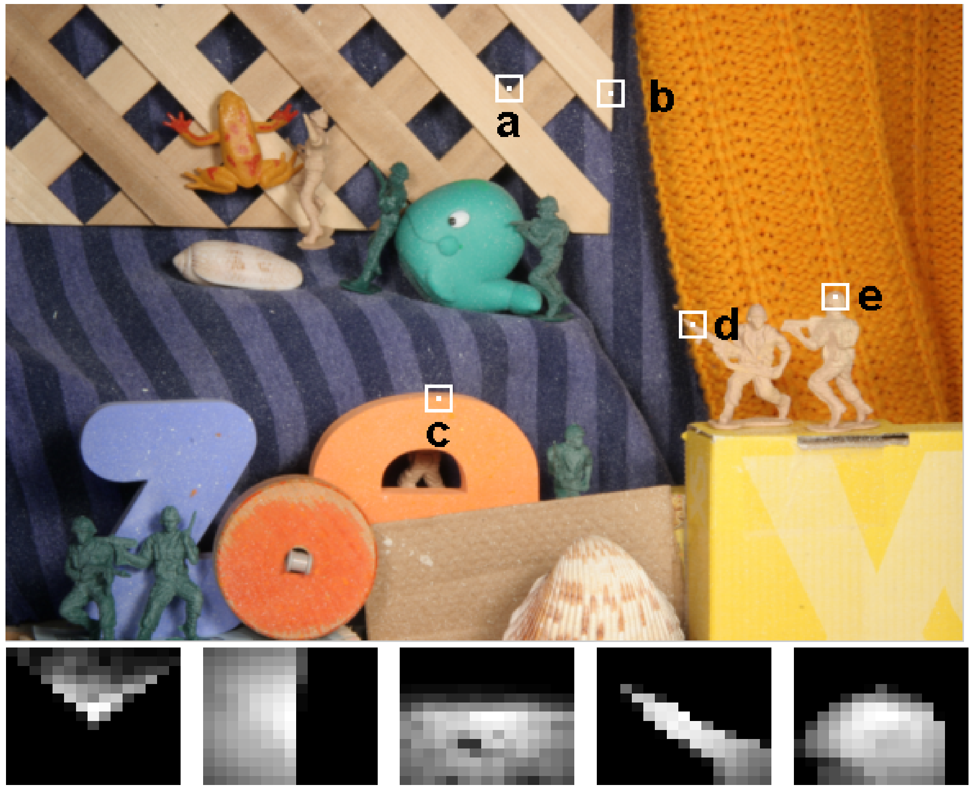
Non-Local Regularizers

- Using only pairwise regularization is too limited
    - cannot really express more than robust smoothness
- Idea: Use larger neighborhoods in the regularizer (i.e. non-local)

$$\sum_{i,j} \sum_{(i',j') \in N_{i,j}} w_{i,j,i',j'} \left(|\hat{u}_{i,j} - \hat{u}_{i',j'}| + |\hat{v}_{i,j} - \hat{v}_{i',j'}| \right)$$

- Problem: Not all pixels in the neighborhood are equally meaningful
    - Introduce adaptive weights

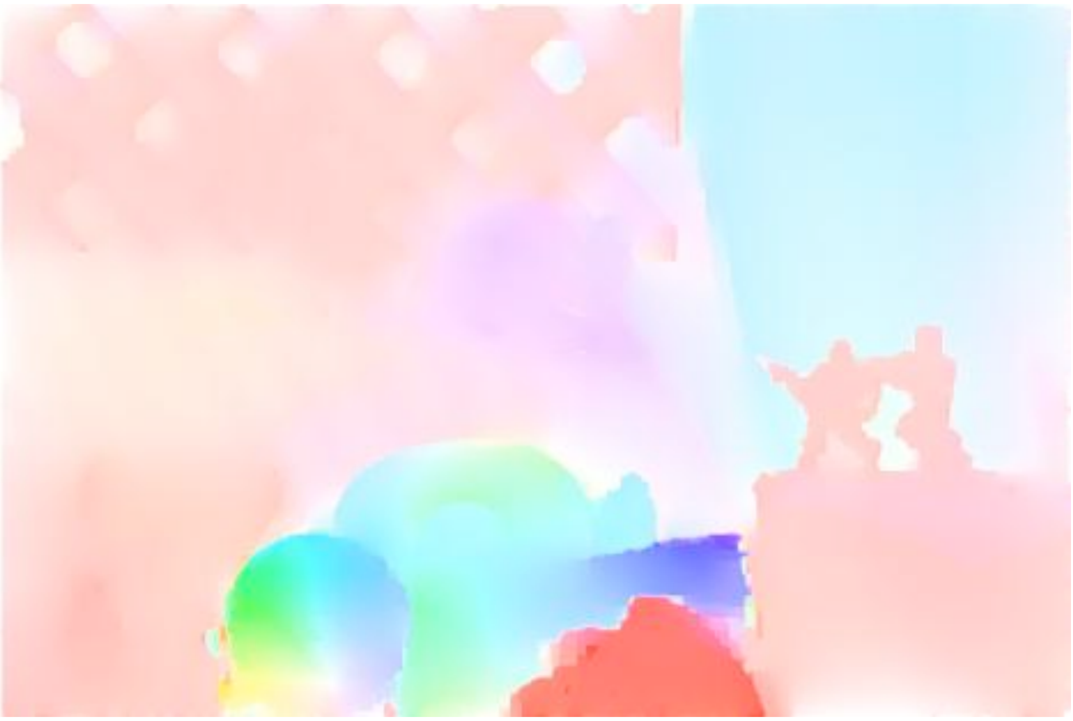
$$\exp\left\{ -\frac{|i-i'|^2 + |j-j'|^2}{2\sigma_1^2} - \frac{|\mathbf{I}(i,j) - \mathbf{I}(i',j')|^2}{2\sigma_2^2} \right\} \frac{o(i',j')}{o(i,j)}$$

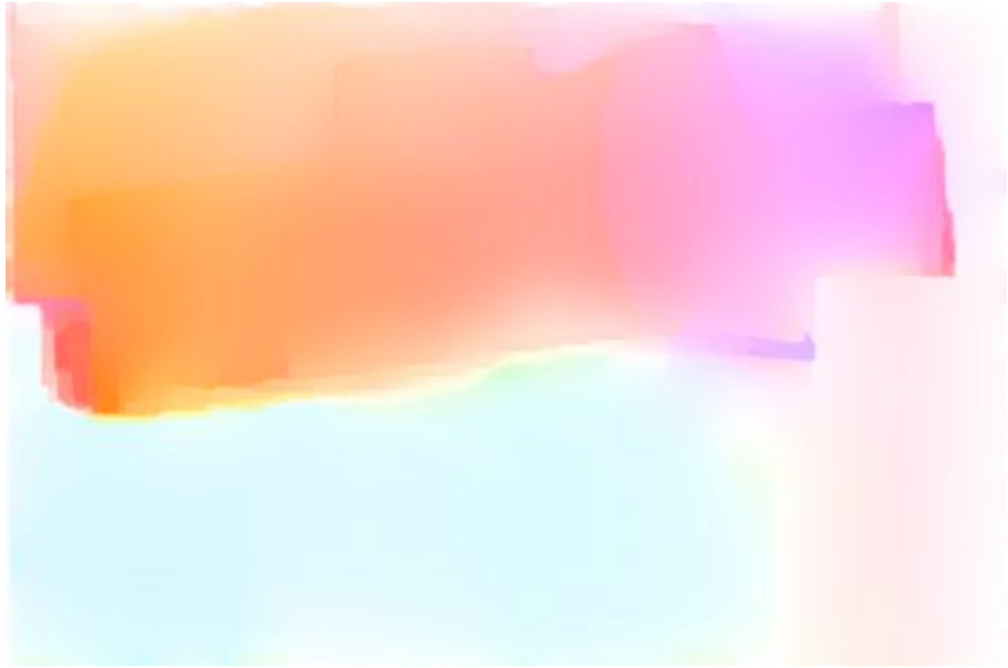


Some More Results

Method by [Black & Anandan, 96]

Some More Results





Method by [Black & Anandan, 96]

Some More Results

Method by [Sun et al. 2010]

Some More Results

Method by [Sun et al. 2010]

Next Topic

- Now we will shift gears again and turn to the related problem of tracking in image sequences.

Introduction to Tracking

28.05.2014

TECHNISCHE UNIVERSITÄT DARMSTADT

visual inference

New Topic: Tracking

TECHNISCHE UNIVERSITÄT DARMSTADT

- Tracking is the problem of finding the motion of an object in an image sequence.
- Useful for:
  - Animation & Interaction
  - Navigation
  - Recognition of objects
  - Video surveillance
  - Medical applications
  - Computer assisted living
  - Etc.

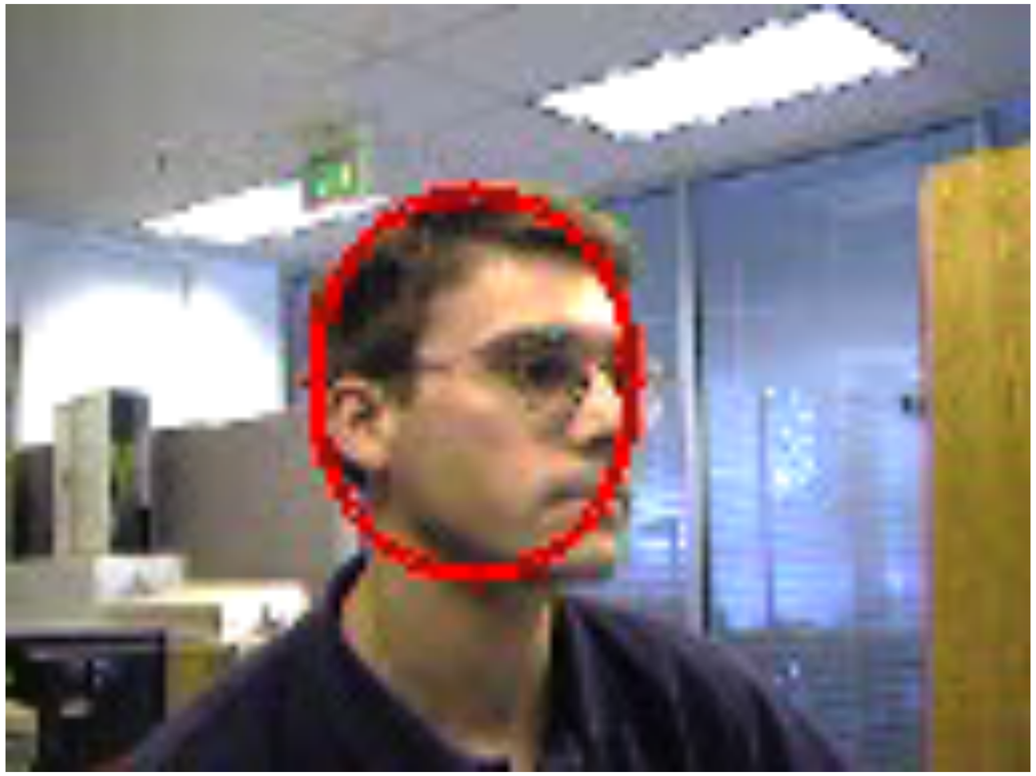
Tracking

- We typically distinguish 3 cases:
  - Tracking rigid objects
  - Tracking articulated objects, e.g. humans or animals
  - Tracking fully non-rigid objects
- Today:
  - Rigid objects
- Next time:
  - Tracking articulated objects, specifically humans

Face Tracking

TECHNISCHE UNIVERSITÄT DARMSTADT

- Face tracking using color histograms and image gradients along contour:



- http://robotics.stanford.edu/~birch/headtracker/

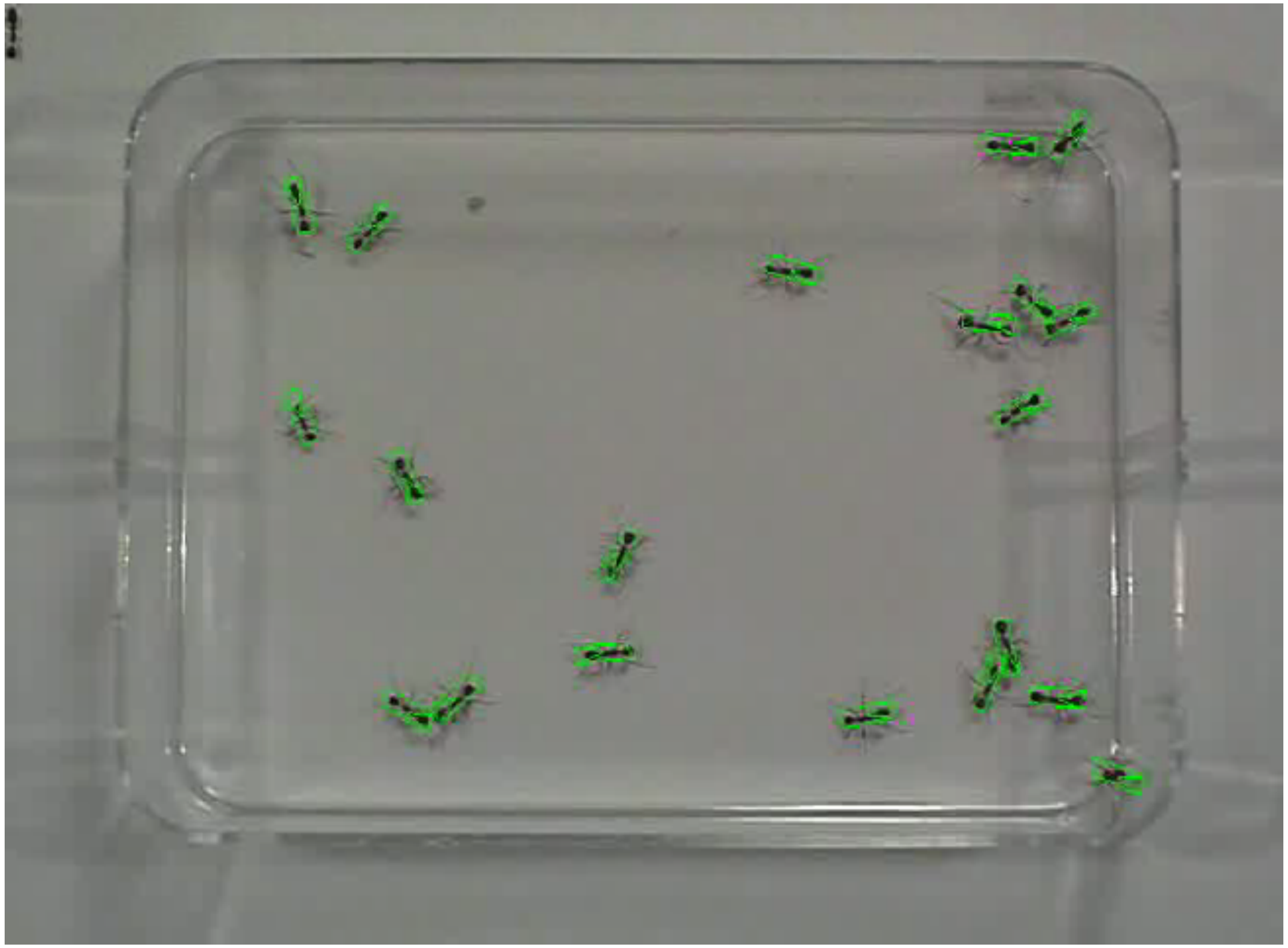
Lane Tracking

- Lane tracking, e.g. for car navigation:

- http://path.berkeley.edu/~zuwhan/lanedetection/index.html

"Ant Tracking"

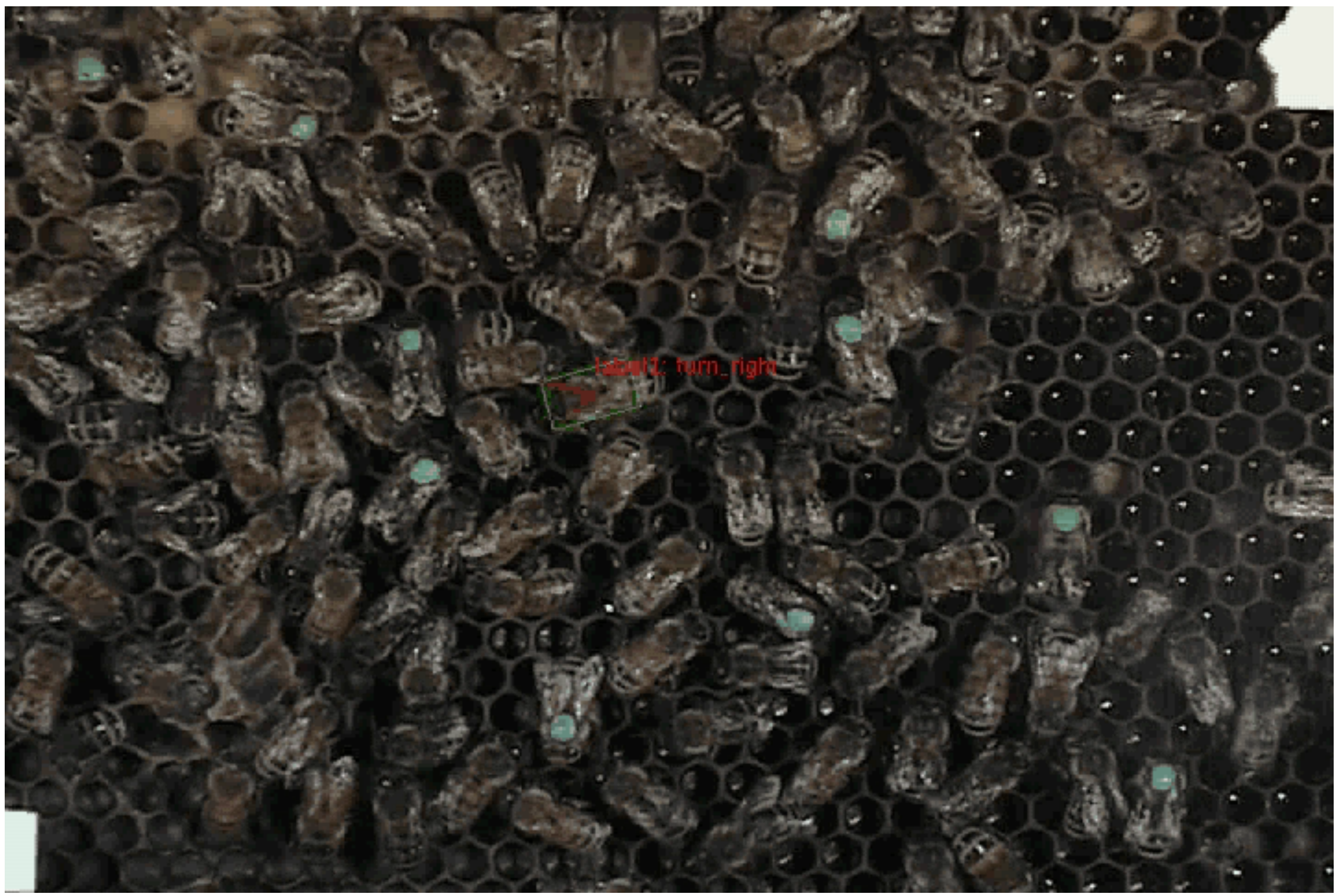
- Tracking is also very useful for facilitating behavioral research in animals.



http://www.cc.gatech.edu/~borg/biotracking/recent-results.html

“Bee Tracking”

- Tracking is also very useful for facilitating behavioral research in animals.



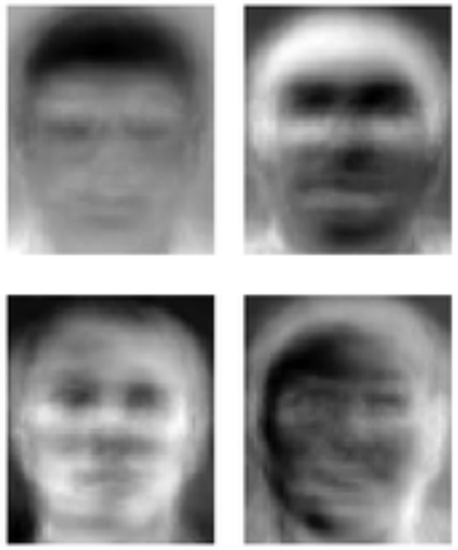
http://www.cc.gatech.edu/~borg/biotracking/recent-results.html

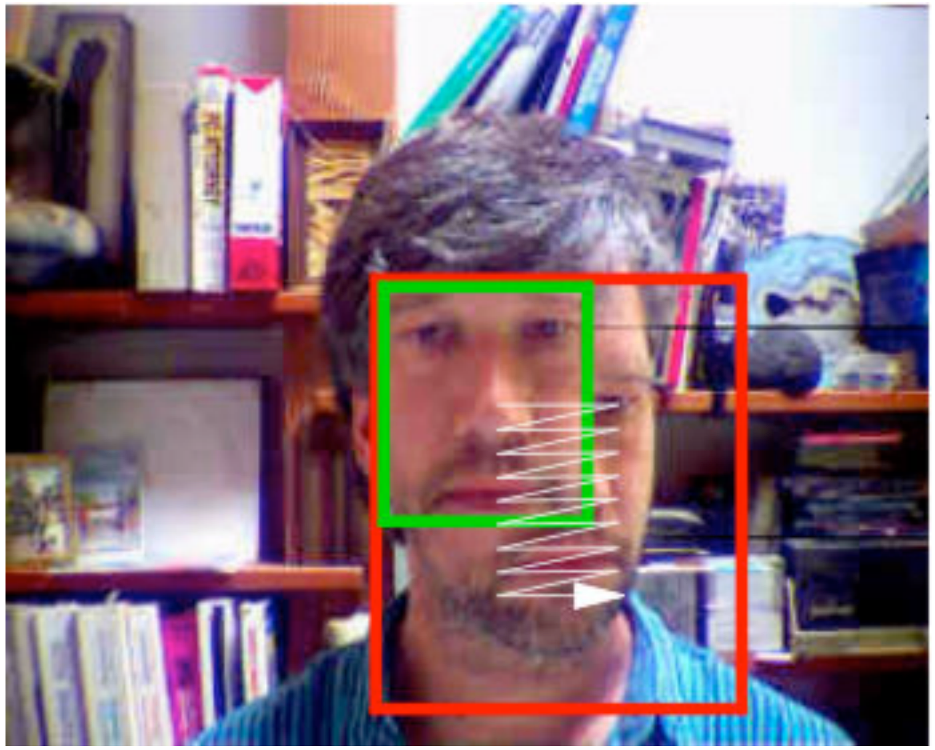
Challenges

- Fast motions
- Changing appearance
- Changing object pose
- Dynamic backgrounds
- Lots of clutter
- Occlusions
- Poor image quality
- ...

Simple Solution

- If the object we want to track is easy to find, we can detect it in every frame ("Tracking by detection").
- Example: Face tracking
  - If there is only one person in the scene and is always in a frontal view, we could use a PCA model of the face to find it.
  - But we may not want to search the entire image, because that is inefficient.
  - Use the position from the previous frame to constrain search.





[IBM Vision Group]

Simple (?) Solution

- If the object we want to track is easy to find, we can just detect it in every frame ("Tracking by detection").
- Problem: Need data association!
  - Many objects may be present & the detector may misfire

  - How do we know which detection is which?
  - In general, attempt data association
    - For now, sidestep this issue

Gradient-based tracking

- To avoid the explicit search, we can also take our matching function (e.g. from PCA) and compute the gradient w.r.t. the object motion between frames.
  - For example, SSD-matching against a simple template:

$$E_{SSD}(x,y) = \sum_{r,c} \big(I(x+c, y+r) - T(c,r)\big)^2$$

- Now we compute $\partial/\partial x \; E_{SSD}(x,y)$ and $\partial/\partial y \; E_{SSD}(x,y)$ by linearizing the brightness constancy assumption (around the previous position).
- This is almost like what we did when we derived the LK algorithm in optical flow.

Gradient-based tracking

TECHNISCHE UNIVERSITÄT DARMSTADT

- Then we can either solve for the motion explicitly, or we can just go a step along the gradient.
- What is the hidden assumption behind all this?
  - The object motion between frames is small.
  - That is not a bad assumption in itself, but we don't have any explicit control over how large the motion may be.
  - We're constrained by the working range of the linearized brightness constancy constraint.
- We need a more general way of incorporating the **dynamics** of the object!

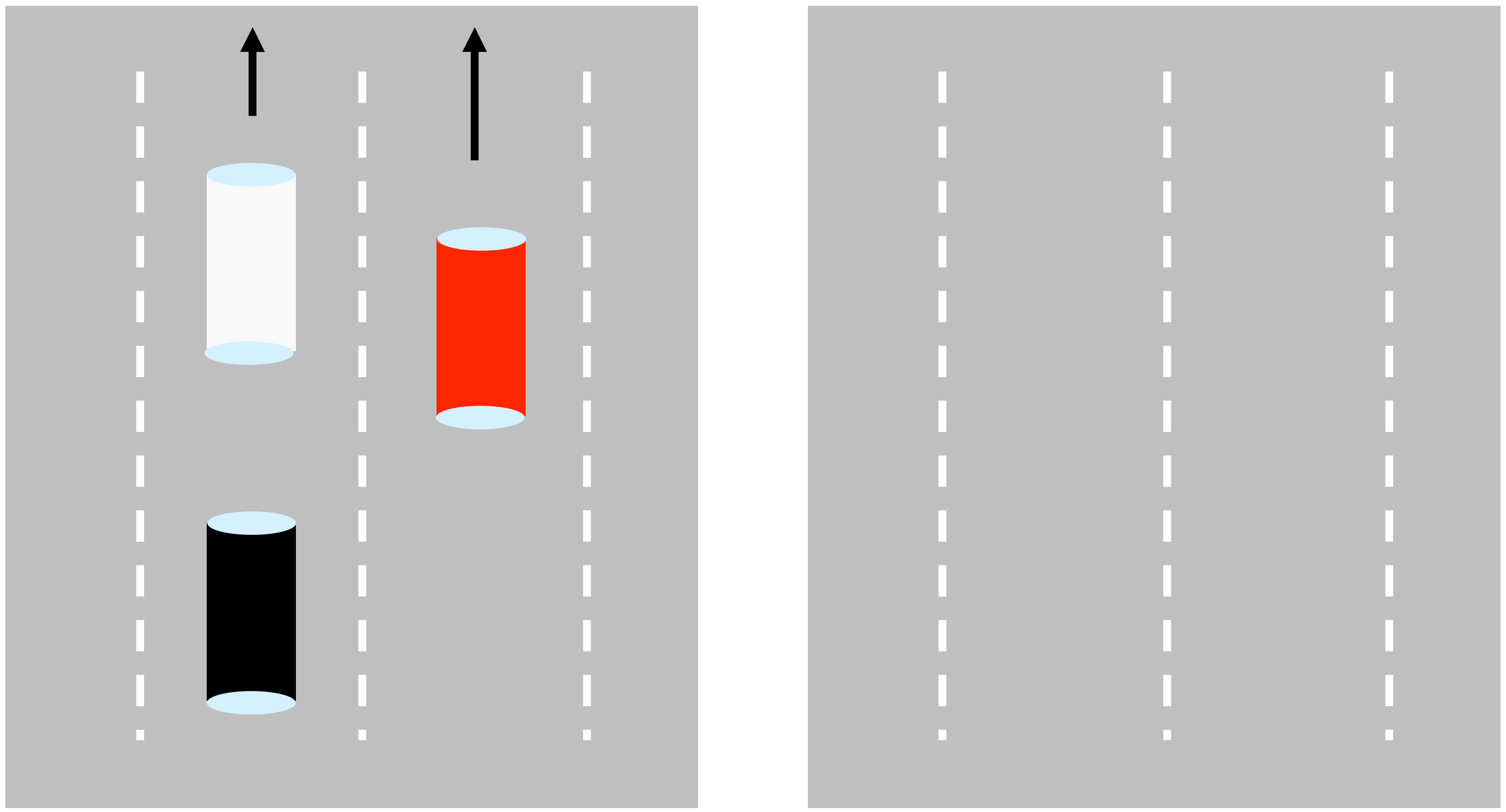
Typical Models of Dynamics

TECHNISCHE UNIVERSITÄT DARMSTADT

- Constant position:
  - I.e. no real dynamics, but if the velocity of the object is sufficiently small, this can work.
- Constant velocity (possibly unknown):
  - We assume that the velocity does not change over time.
  - As long as the object does not quickly change velocity or direction, this is a quite reasonable model.
- Constant acceleration (possibly unknown):
  - Also captures the acceleration of the object.
  - This may include both the velocity, but also the directional acceleration.

Illustration

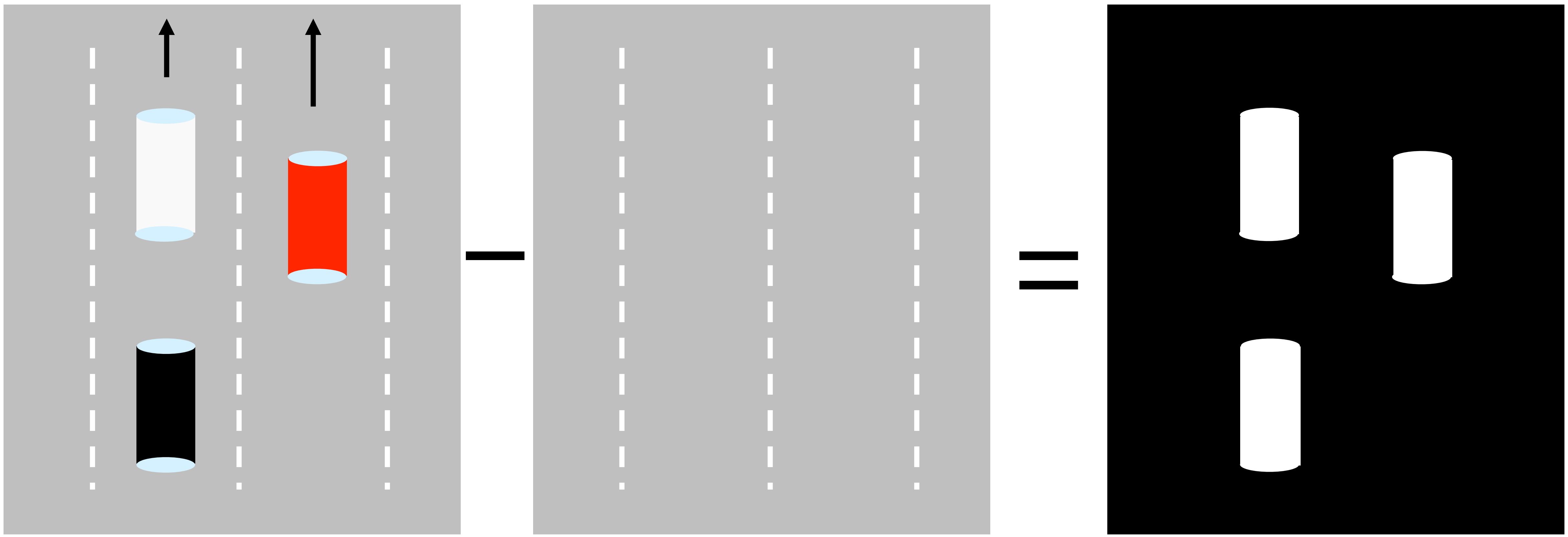
TECHNISCHE UNIVERSITÄT DARMSTADT



- Goal: Estimate car position at each time instant (say, of the red car).
- Observations: Image sequence and known background.

[Michael Black]

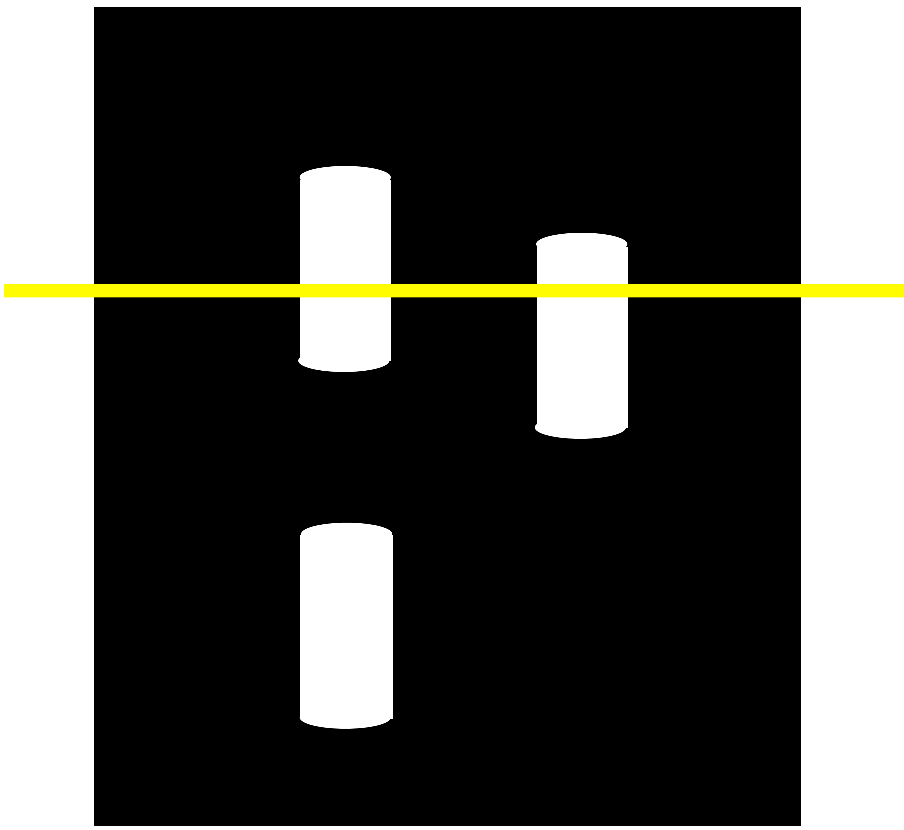
Illustration

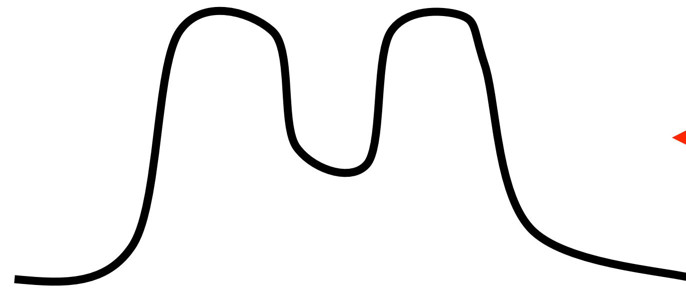


- Perform background subtraction.
- Obtain binary map of possible cars.
- But which one is the one we want to track?

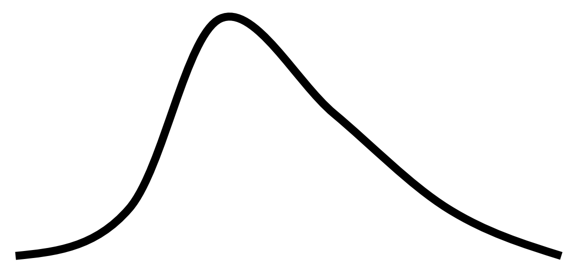
[Michael Black]

Bayesian Tracking

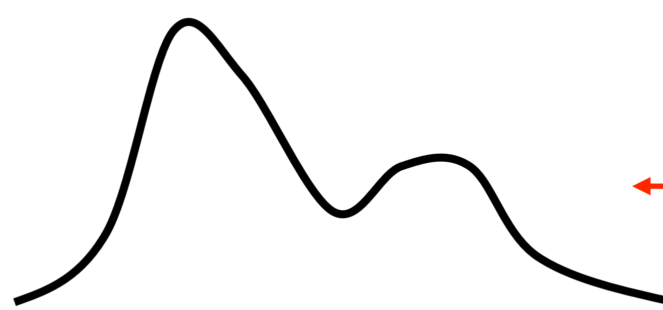




Likelihood:
noisy observation
$p(FG|\mathrm{car} = (x, y))$



Prior:
$p(\mathrm{car} = (x, y))$



Posterior:
Bayesian update
$p(\mathrm{car} = (x, y)|FG)$

system state: car position

observations: images

[Michael Black]

Notation

- $\boldsymbol{x}_k \in \mathbb{R}^d$: internal state at k-th frame (hidden random variable, e.g., position of the object in the image).

  $\mathbf{X}_k = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k]^{\mathrm{T}}$: history up to time step k

- $\boldsymbol{z}_k \in \mathbb{R}^c$: measurement at k-th frame (observable random variable, e.g. the given image).

  $\mathbf{Z}_k = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k]^{\mathrm{T}}$: history up to time step k

[Michael Black]

Goal

Estimating the posterior probability $p(\boldsymbol{x}_k|\mathbf{Z}_k)$

How ???

One idea:
Recursion

$$p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1}) \quad \Rightarrow \quad p(\boldsymbol{x}_k|\mathbf{Z}_k)$$

- How to realize the recursion ?
- What assumptions are necessary ?

[Michael Black]

Recursive Estimation

$$
\begin{aligned}
& p(\boldsymbol{x}_k|\mathbf{Z}_k) \\
&= p(\boldsymbol{x}_k|\boldsymbol{z}_k, \mathbf{Z}_{k-1}) \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathbf{Z}_{k-1}) \cdot p(\boldsymbol{x}_k|\mathbf{Z}_{k-1}) \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot p(\boldsymbol{x}_k|\mathbf{Z}_{k-1}) \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1} \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathbf{Z}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1} \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}
\end{aligned}
$$

Bayes rule:

$$p(a|b) = p(b|a)p(a)/p(b)$$

Assumption:

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathbf{Z}_{k-1}) = p(\boldsymbol{z}_k|\boldsymbol{x}_k)$$

Marginalization:

$$p(a) = \int p(a, b)\, \mathrm{d}b$$

Assumption:

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathbf{Z}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

Bayesian Formulation

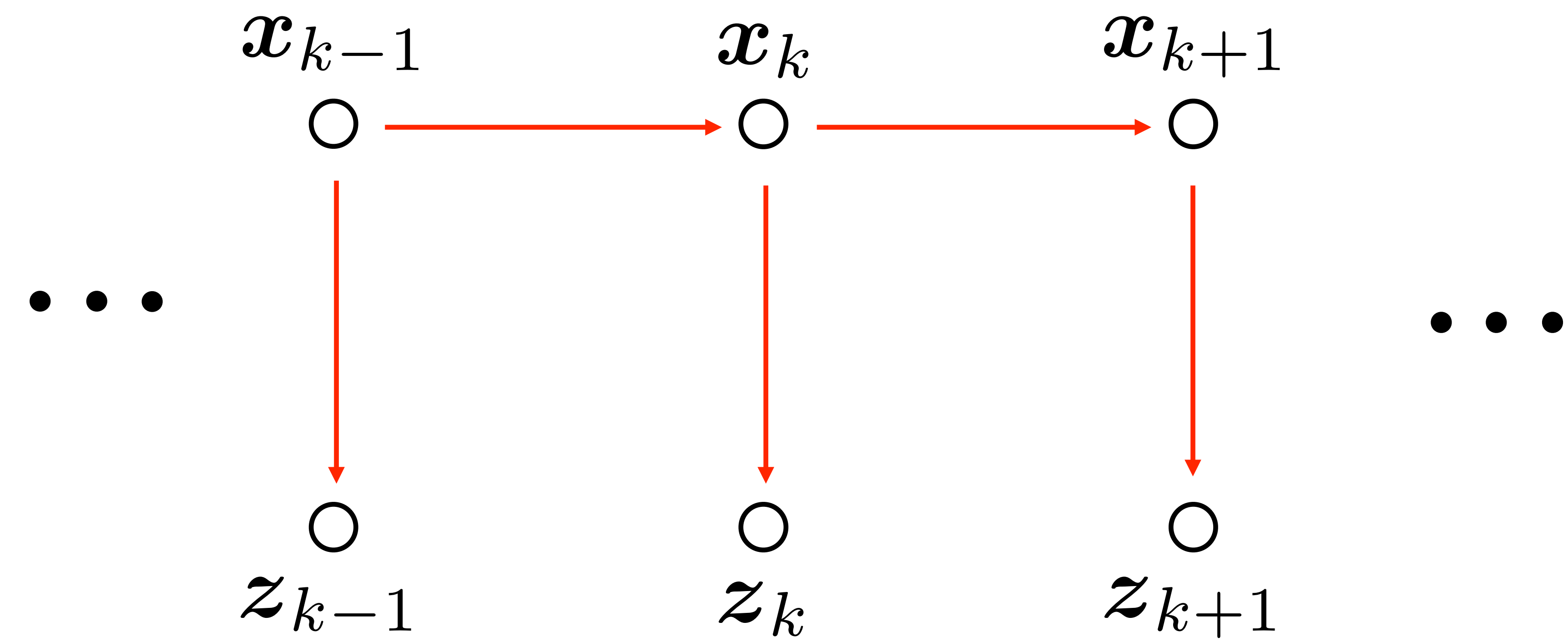
TECHNISCHE UNIVERSITÄT DARMSTADT

$$\boxed{p(\boldsymbol{x}_k|\mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1}}$$

$p(\boldsymbol{x}_k\|\mathbf{Z}_k)$	posterior probability at current time step
$p(\boldsymbol{z}_k\|\boldsymbol{x}_k)$	likelihood
$p(\boldsymbol{x}_k\|\boldsymbol{x}_{k-1})$	temporal prior
$p(\boldsymbol{x}_{k-1}\|\mathbf{Z}_{k-1})$	posterior probability at previous time step
κ	normalizing term

Bayesian Graphical Model

- Hidden Markov model:



Assumptions:

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathbf{Z}_{k-1}) = p(\boldsymbol{z}_k|\boldsymbol{x}_k) \qquad p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathbf{Z}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

$$p(\boldsymbol{x}_k|\mathbf{X}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$