Global Motion Models & Dense Optical Flow

21.05.2014

TECHNISCHE UNIVERSITÄT DARMSTADT

visual inference

Announcements

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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)

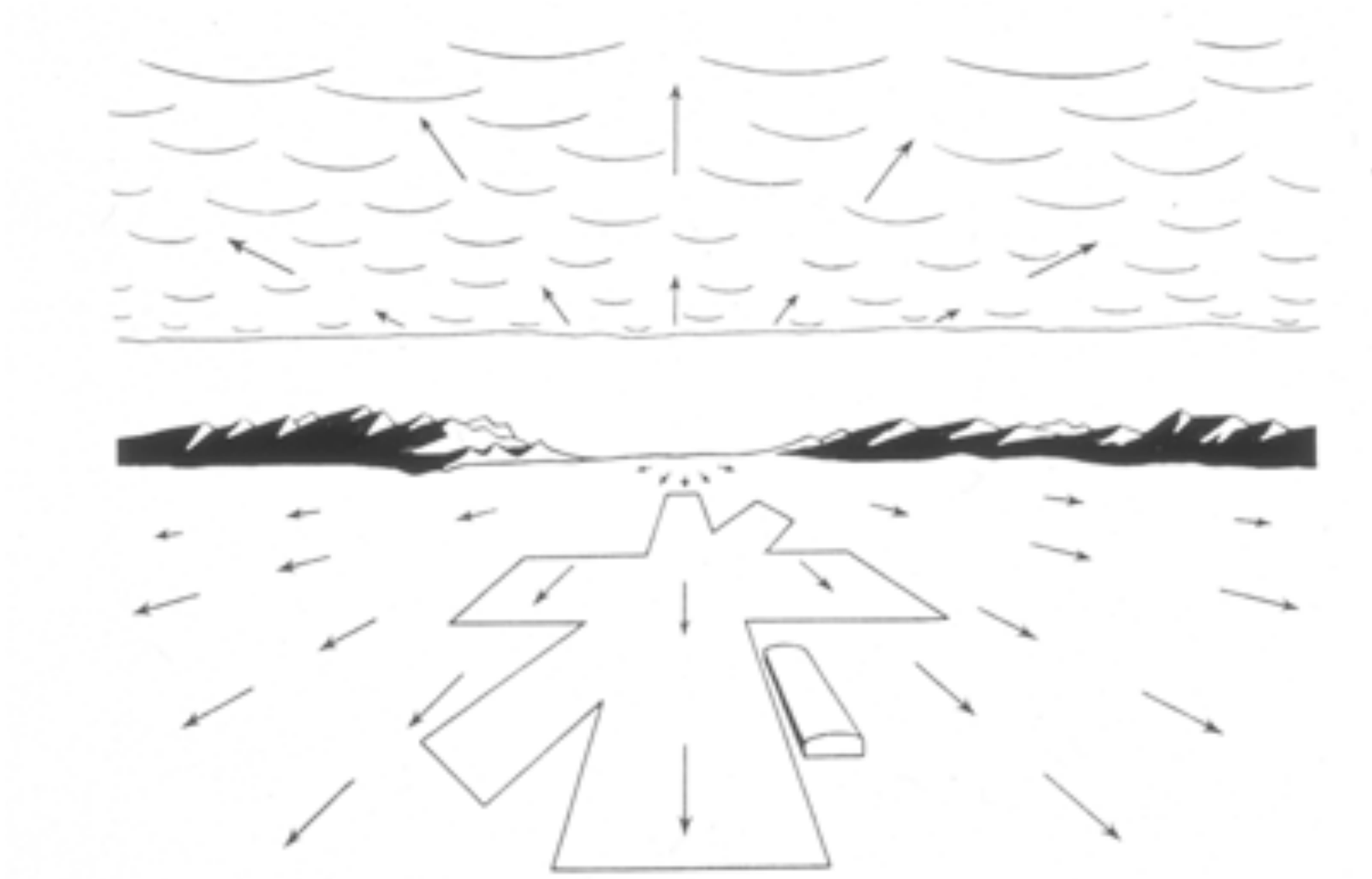
What is Image Motion?

TECHNISCHE UNIVERSITÄT DARMSTADT

Optical Flow

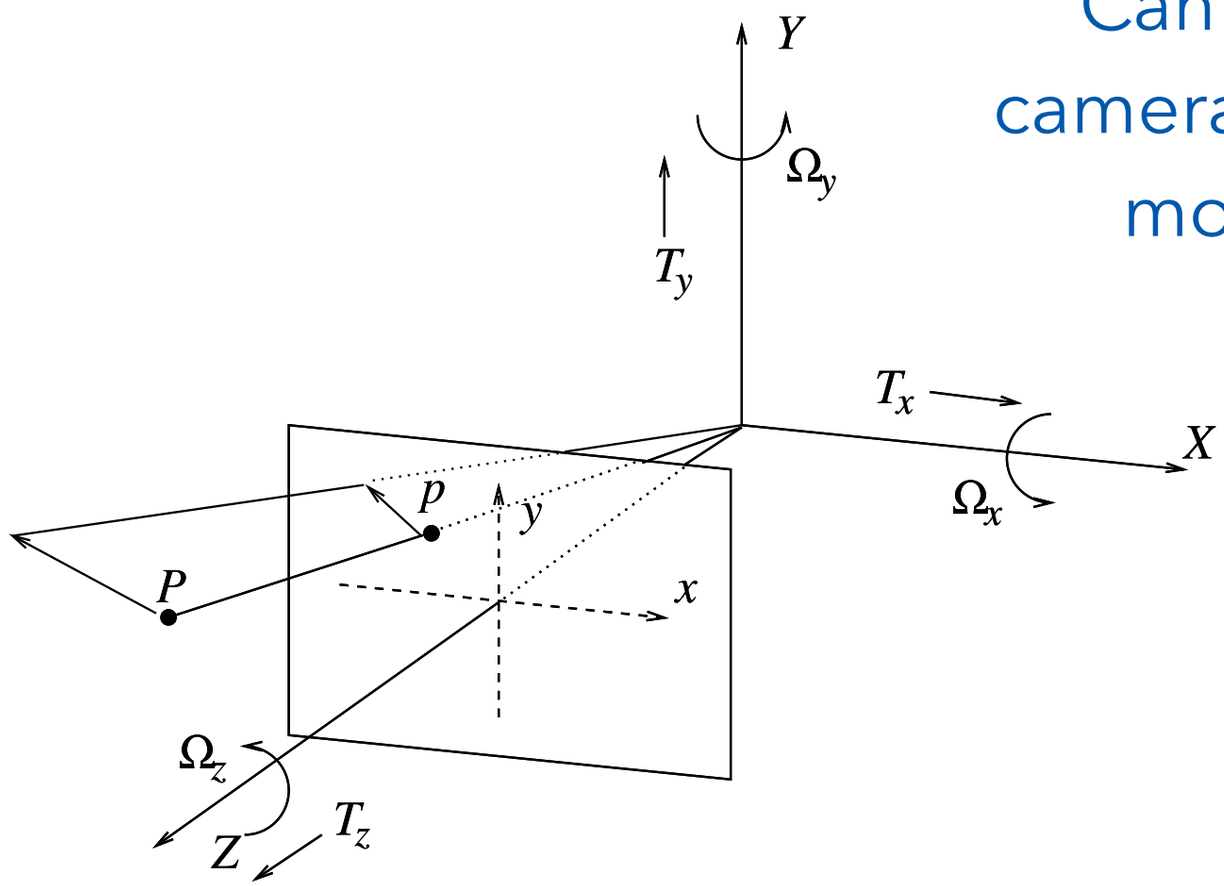
TECHNISCHE UNIVERSITÄT DARMSTADT

Note: Most slides borrowed from Michael Black.



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

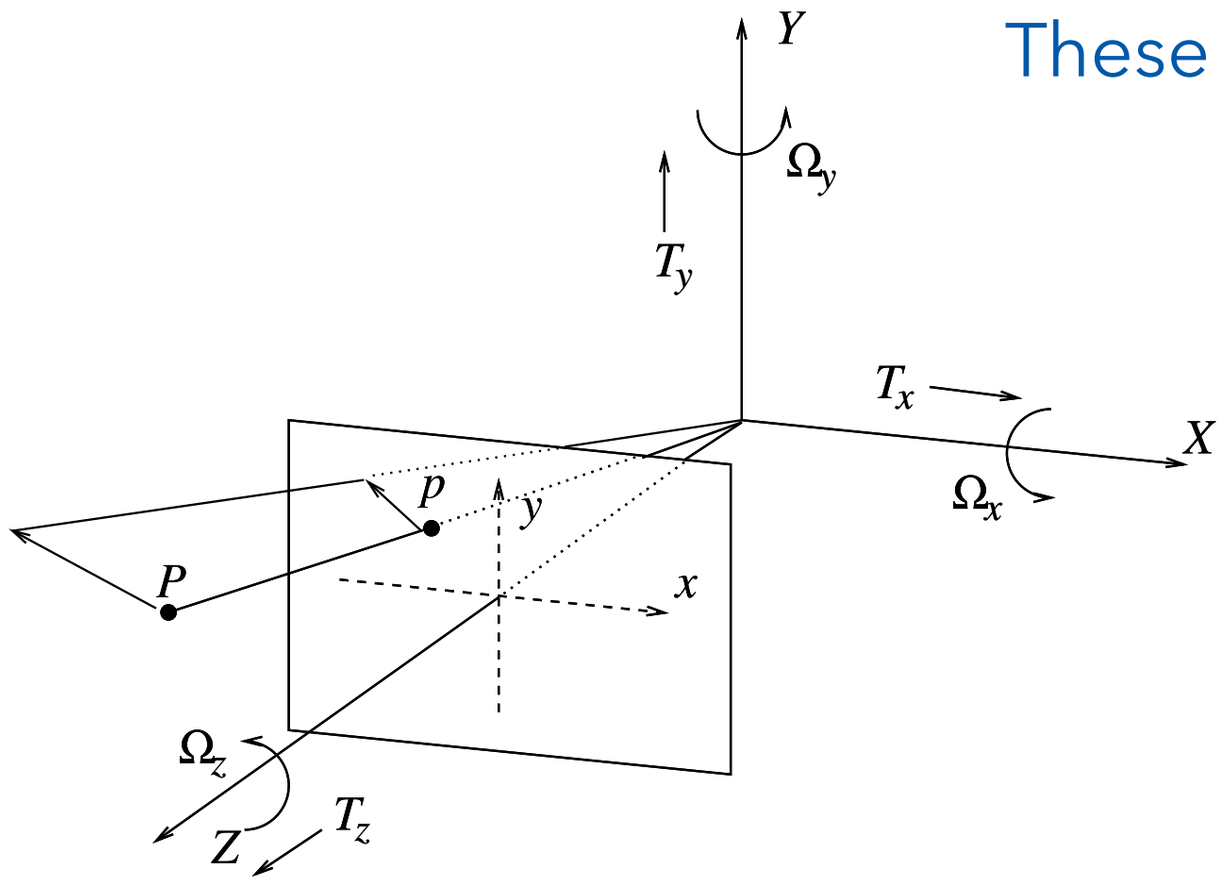
Motion Field



Can be the result of camera motion or object motion (or both)!

Motion field = 2D motion field representing the projection of the 3D motion of points in the scene onto the image plane.

Optical Flow



These are not the same!
Why?

Optical flow = 2D velocity field describing the apparent motion in the images.

Thought Experiment 1

TECHNISCHE UNIVERSITÄT DARMSTADT

Lambertian ball rotating in 3D

What does the 2D motion field look like?

What does the 2D optical flow field look like?

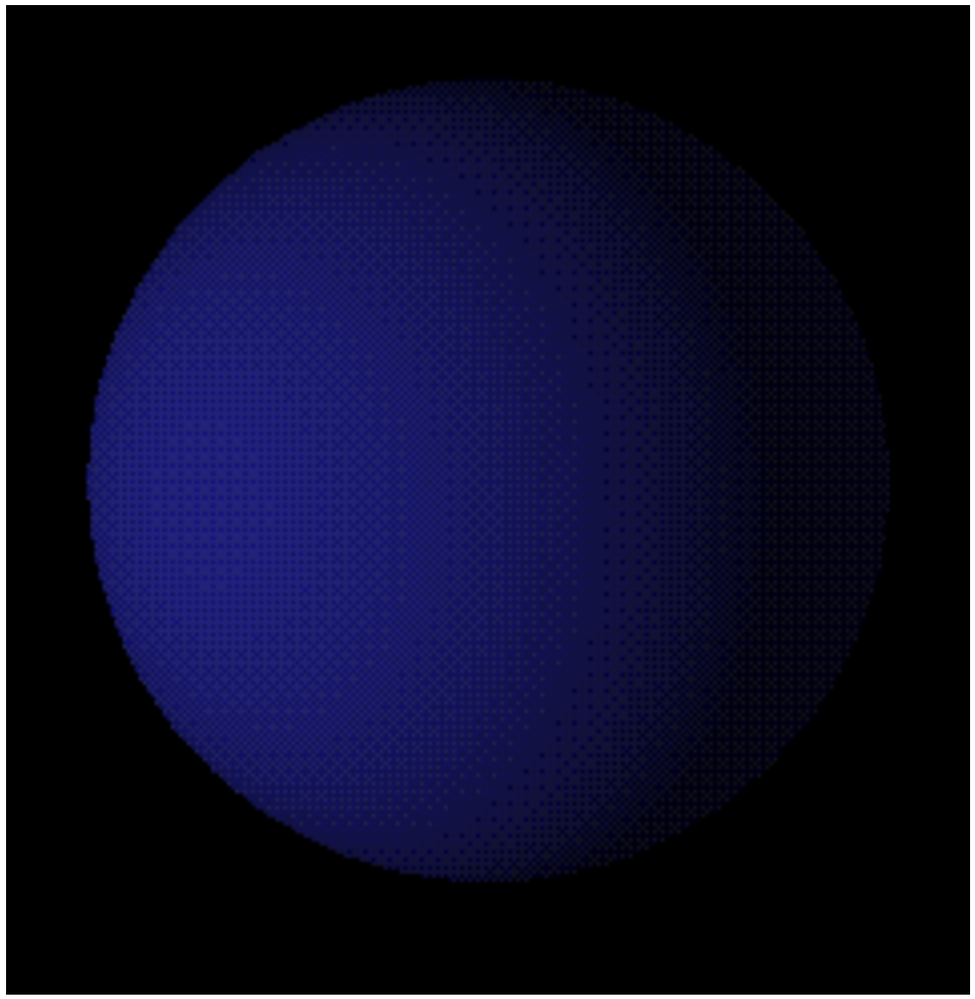


Image source: http://www.evl.uic.edu/aej/488/lecture12.html

Thought Experiment 2

TECHNISCHE UNIVERSITÄT DARMSTADT

Stationary specular ball, moving light source.

What does the 2D motion field look like?

What does the 2D optical flow field look like?

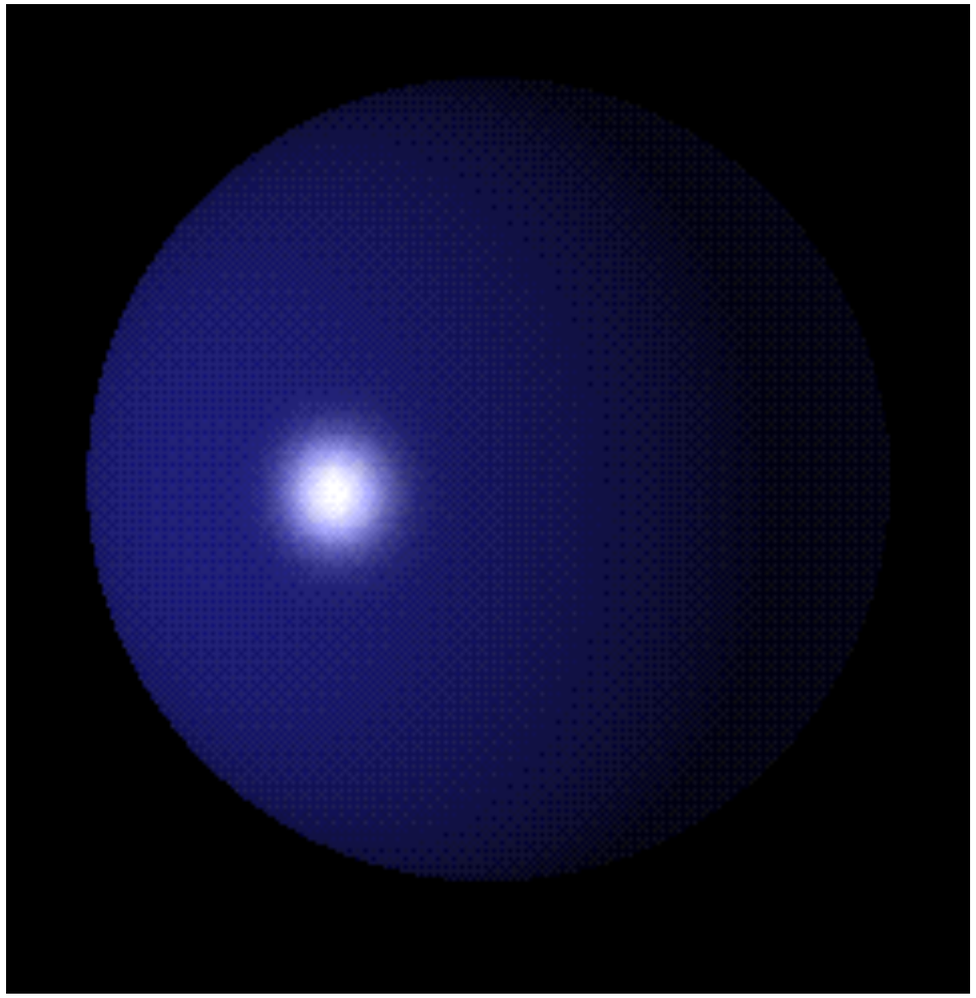


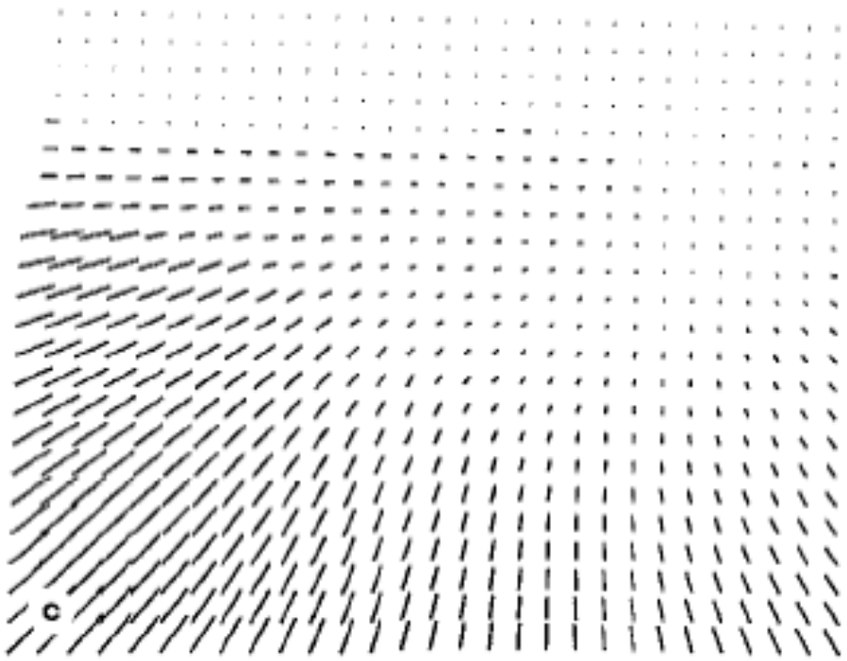
Image source: http://www.evl.uic.edu/aej/488/lecture12.html

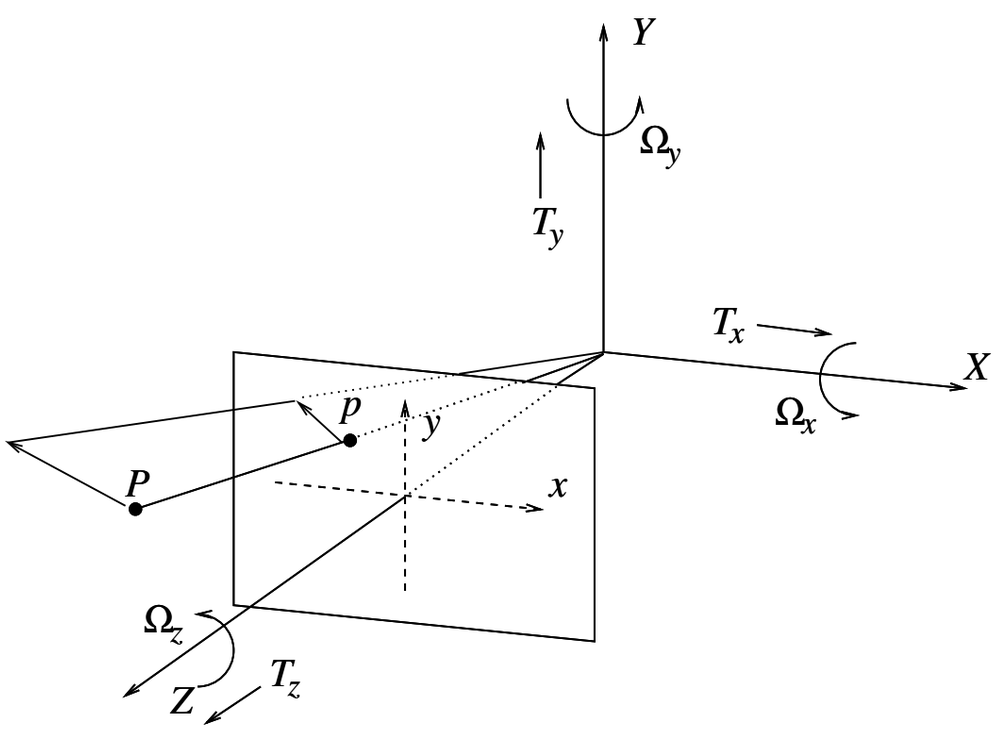
Optical Flow Field

TECHNISCHE UNIVERSITÄT DARMSTADT

Image irradiance at time t
and location $\mathbf{x} = (x, y)$:

$$I(x, y, t)$$



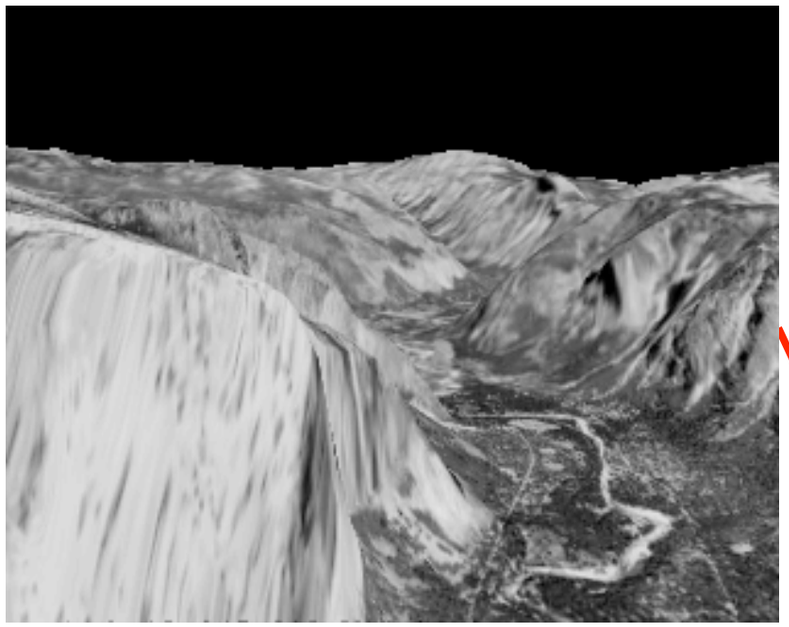


$u(x, y)$ Horizontal component

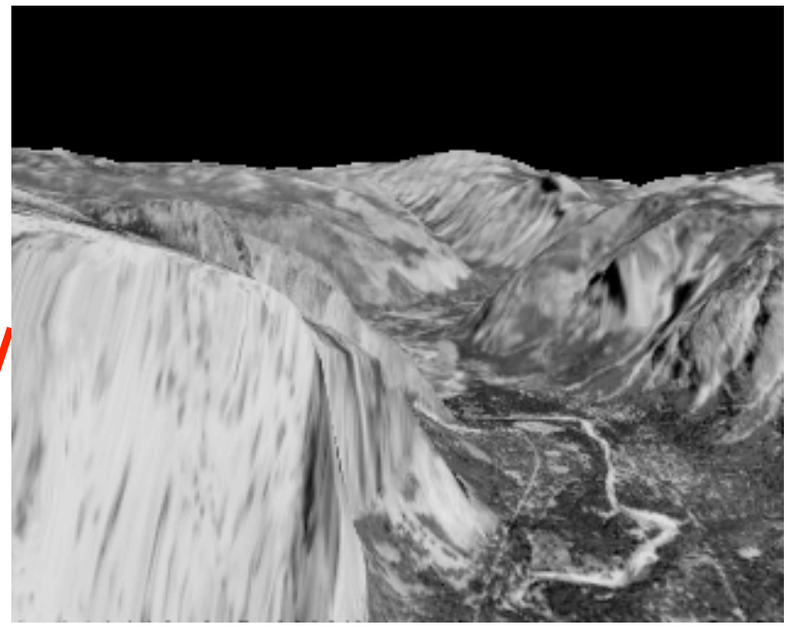
$v(x, y)$ Vertical component

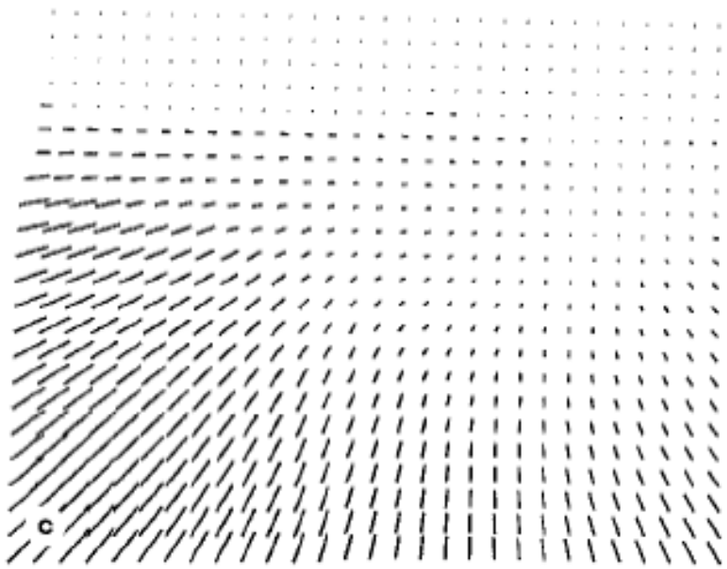
Optical Flow Estimation

time t



time t+1



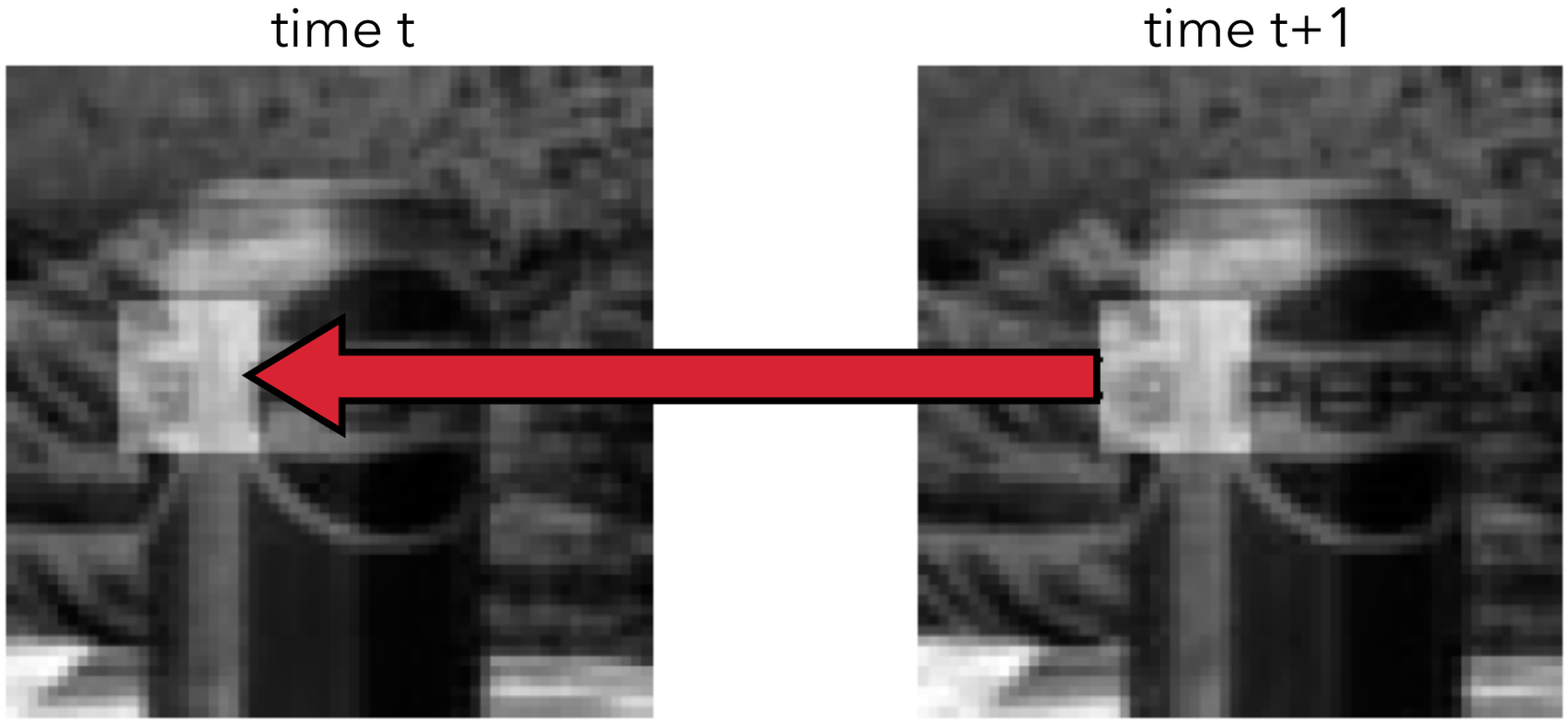


How do we compute flow?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Reminder from CV1:
  - Make 3 important assumptions first.
  - Then exploit those computationally.

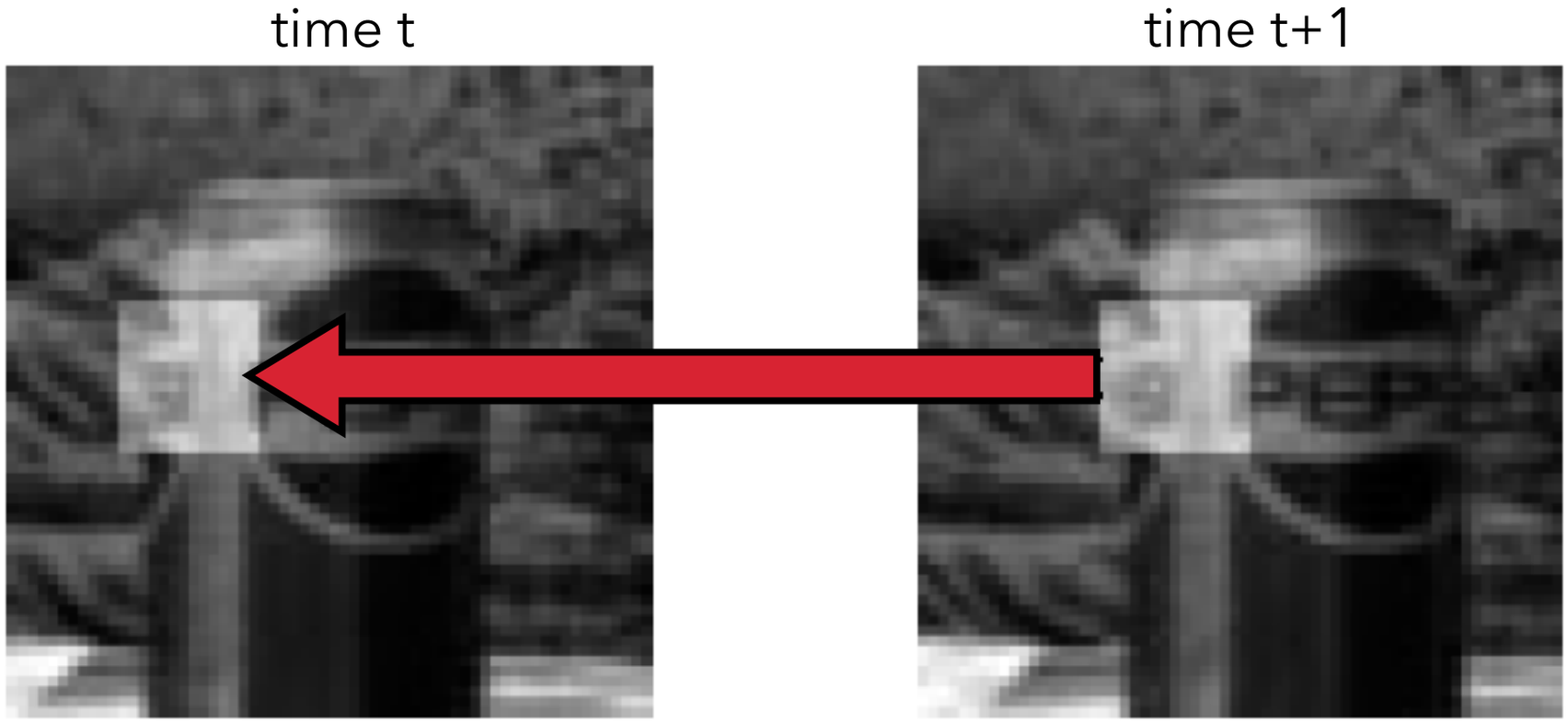
Brightness Constancy



Assumption 1:

Image measurements (e.g. brightness) in a small region remain the same although their location may change.

Brightness Constancy



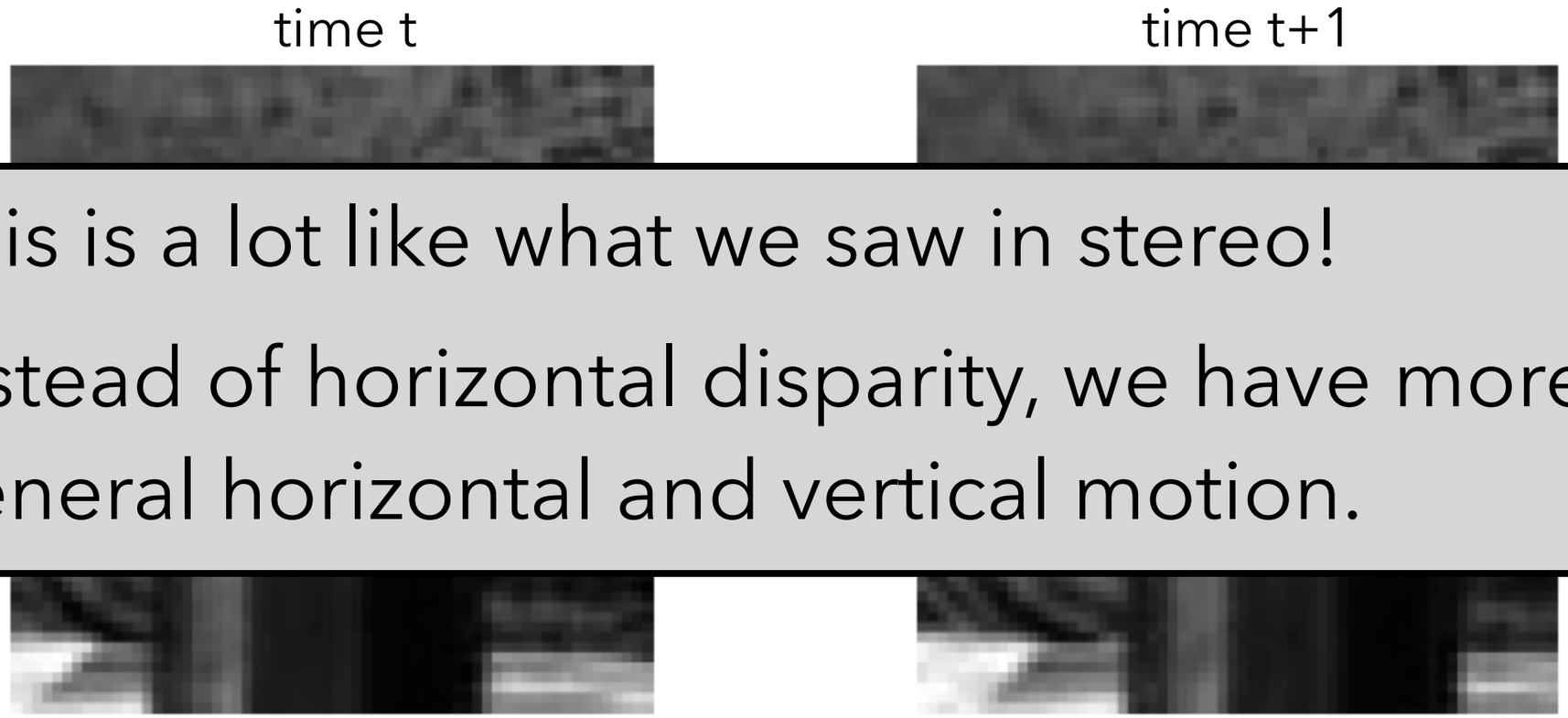
Assumption 1:

$$I(x + u(x,y), y + v(x,y), t + 1) = I(x, y, t)$$

shifted by horizontal flow

shifted by vertical flow

Brightness Constancy



- This is a lot like what we saw in stereo!
- Instead of horizontal disparity, we have more general horizontal and vertical motion.

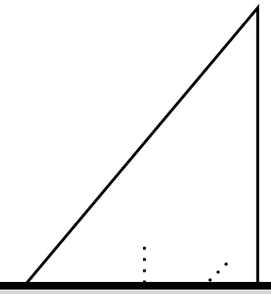
Assumption 1:

$$I(x + u(x, y), y + v(x, y), t + 1) = I(x, y, t)$$

shifted by horizontal flow

shifted by vertical flow

Spatial Coherence



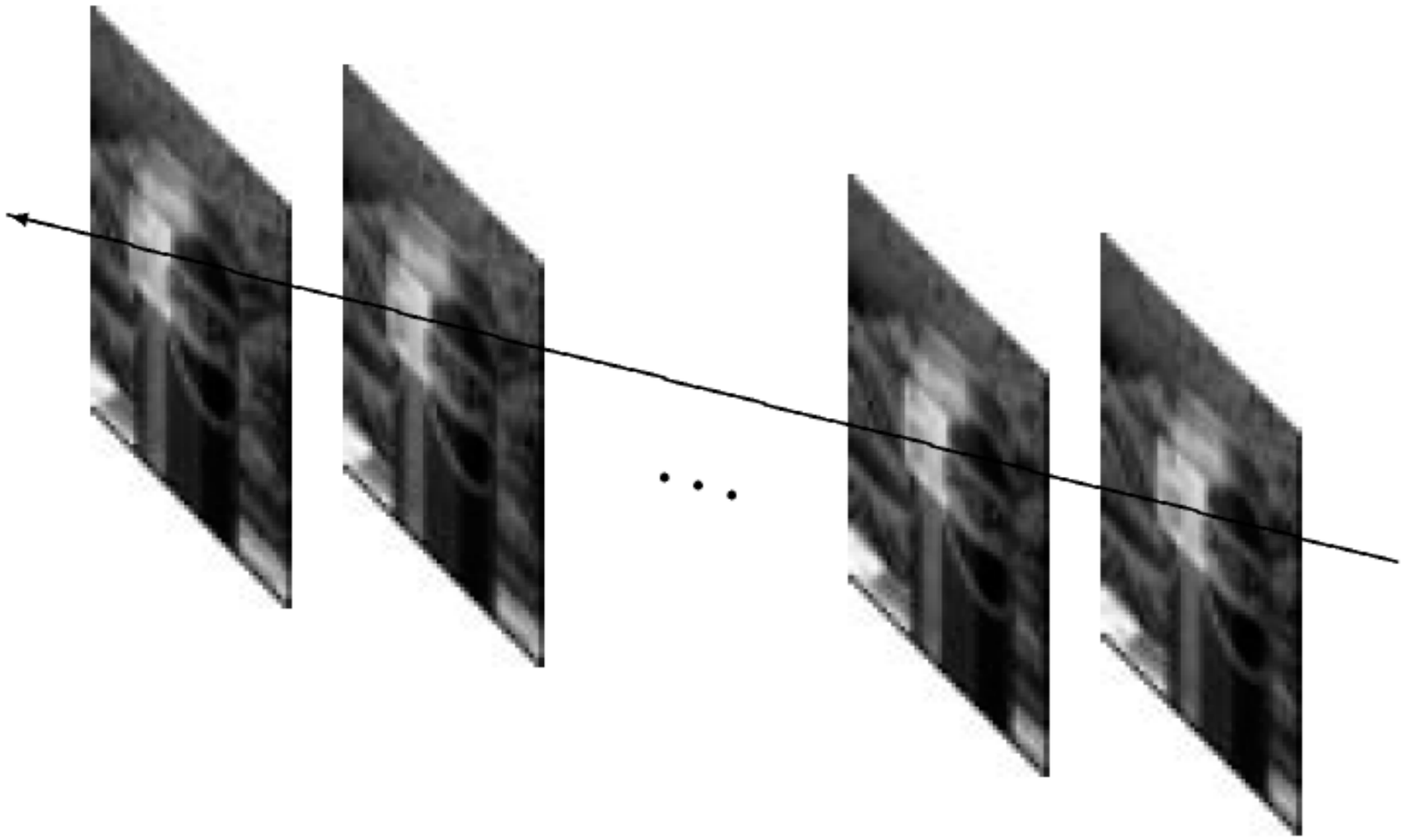
- This is a lot like what we saw in stereo!
- Corresponding notion of spatial regularity

Assumption 2:

- Neighboring points in the scene typically belong to the same surface and hence typically have similar 3D motions.
- Since they also project to nearby points in the image, we expect spatial coherence in the image flow.

Temporal Persistence

TECHNISCHE UNIVERSITÄT DARMSTADT



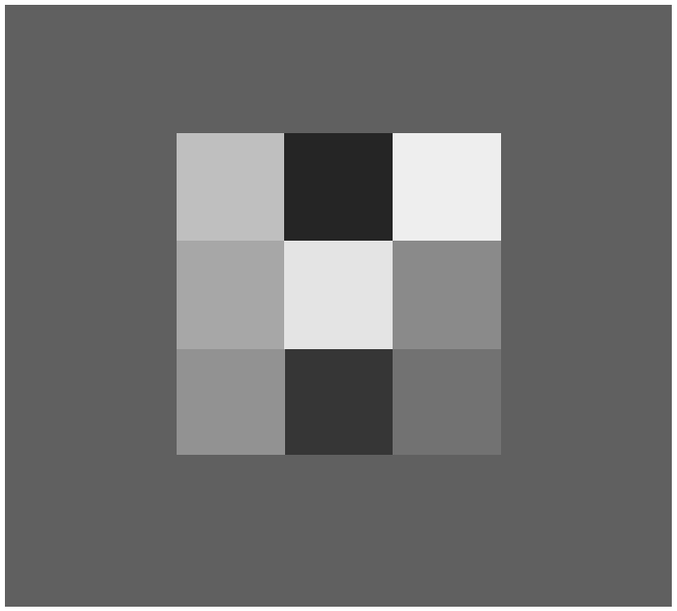
Assumption 3:

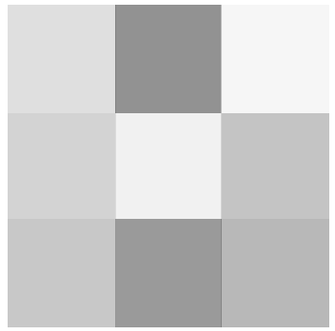
The image motion of a surface patch changes gradually over time.

Minimize Brightness Difference

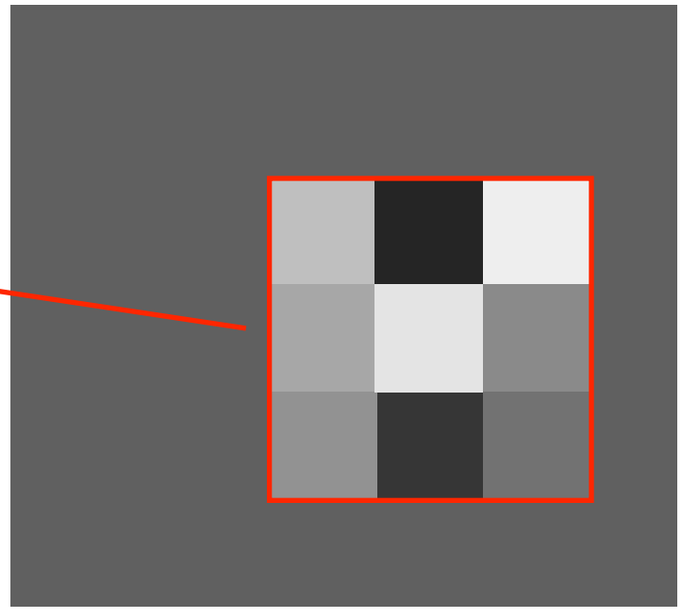
TECHNISCHE UNIVERSITÄT DARMSTADT

$I(x, y, t+1)$





$I(x, y, t)$



$$E_{SSD}(u, v) = \sum_{(x,y) \in R} \left(I(x+u, y+v, t+1) - I(x, y, t) \right)^2$$

Simple Flow Estimation Algorithm

TECHNISCHE UNIVERSITÄT DARMSTADT

$$E_{SSD}(u,v) = \sum_{(x,y)\in R} \big(I(x+u, y+v, t+1) - I(x,y,t)\big)^2$$

- Simple way of flow estimation:
  - Discretize the space of possible motions.
  - For each pixel, try all possible motions.
  - Take the one that minimizes the SSD.
- Problems with this approach:
  - Very inefficient.
  - The motions are discrete, but we may not want that.
- So how can we optimize this nonlinear and non-convex function?

Can we approximate this?

TECHNISCHE UNIVERSITÄT DARMSTADT

$$E_{SSD}(u,v) = \sum_{(x,y)\in R} \left(I(x+u, y+v, t+1) - I(x,y,t) \right)^2$$

Let us look at the general case:

$$I(x+\Delta_x, y+\Delta_y, t+\Delta_t)$$

Taylor series approximation:

$$I(x,y,t) + \Delta_x \frac{\partial}{\partial x} I(x,y,t) + \Delta_y \frac{\partial}{\partial y} I(x,y,t) + \Delta_t \frac{\partial}{\partial t} I(x,y,t) + \epsilon(\Delta_x^2, \Delta_y^2, \Delta_t^2)$$

Approximation error

Can we approximate this?

TECHNISCHE UNIVERSITÄT DARMSTADT

$$E_{SSD}(u,v) = \sum_{(x,y)\in R} \big(I(x+u,y+v,t+1) - I(x,y,t)\big)^2$$

Assume small motion

Brightness varies smoothly

$$I(x,y,t) + u\frac{\partial}{\partial x}I(x,y,t) + v\frac{\partial}{\partial y}I(x,y,t) + \frac{\partial}{\partial t}I(x,y,t) + \epsilon(u,v,1)$$

SSD approximation:

$$\begin{aligned}
E_{SSD}(u,v) &\approx \sum_{(x,y)\in R} \big(I(x,y,t) + u\frac{\partial}{\partial x}I(x,y,t) + v\frac{\partial}{\partial y}I(x,y,t) + \frac{\partial}{\partial t}I(x,y,t) - I(x,y,t)\big)^2 \\
&= \sum_{(x,y)\in R} \big(u\frac{\partial}{\partial x}I(x,y,t) + v\frac{\partial}{\partial y}I(x,y,t) + \frac{\partial}{\partial t}I(x,y,t)\big)^2 \\
&= \sum_{(x,y)\in R} \big(u\cdot I_x(x,y,t) + v\cdot I_y(x,y,t) + I_t(x,y,t)\big)^2
\end{aligned}$$

Optical Flow Constraint Equation

$$E_{SSD}(u,v) \approx \sum_{(x,y)\in R} \left(u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t)\right)^2$$

- This approximated SSD objective is a convex function of the motion u and v.
- It becomes much easier to optimize.
  - We will see that shortly.
- But, this only holds for small motions!

Optical Flow Constraint Equation

TECHNISCHE UNIVERSITÄT DARMSTADT

$$E_{SSD}(u,v) \approx \sum_{(x,y)\in R} \left(u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t)\right)^2$$

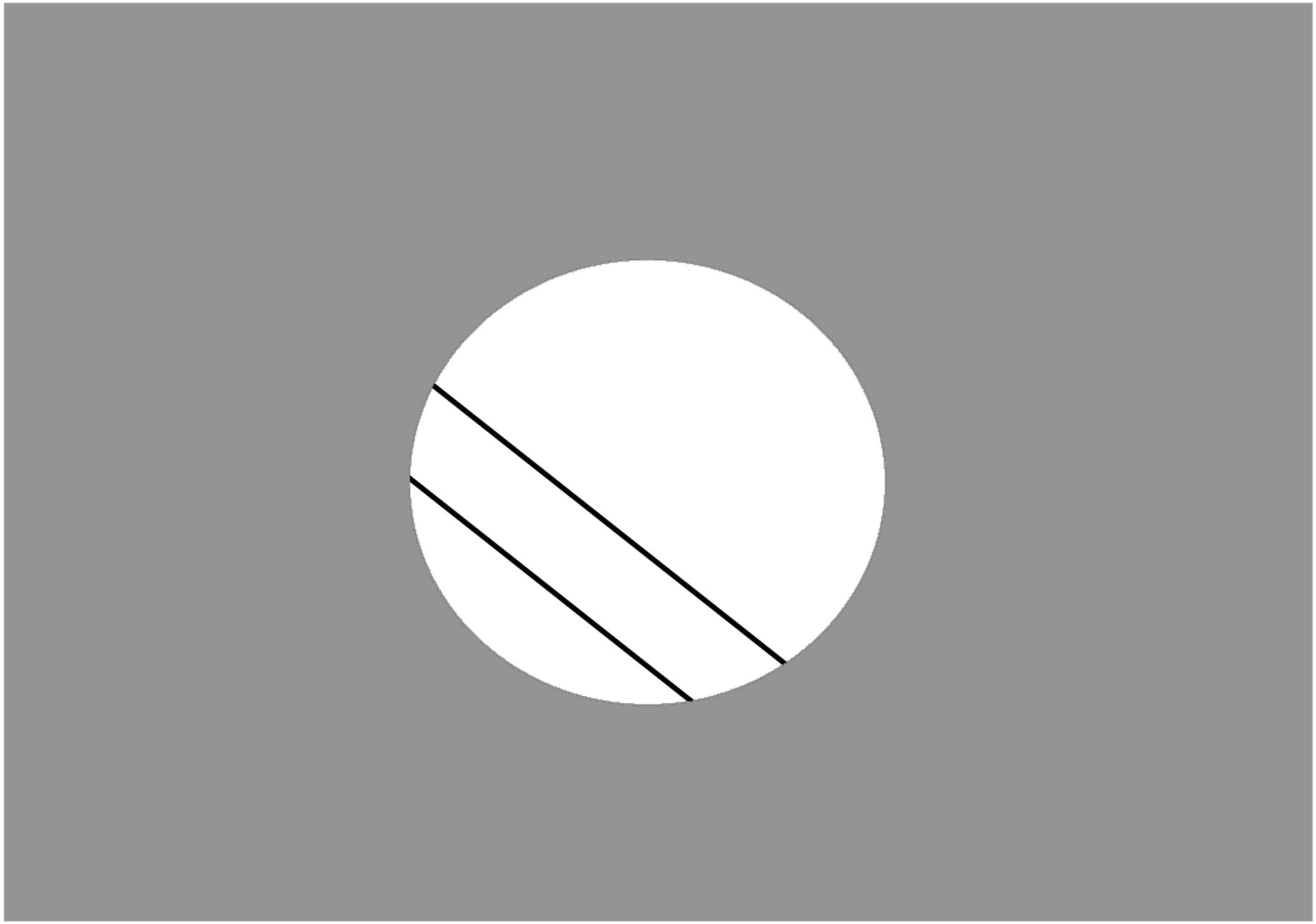
- By minimizing this Taylor series approximation to the SSD, we are trying to enforce the so-called optical flow constraint equation (OFCE):

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

- This is also called the linearized brightness constancy constraint.

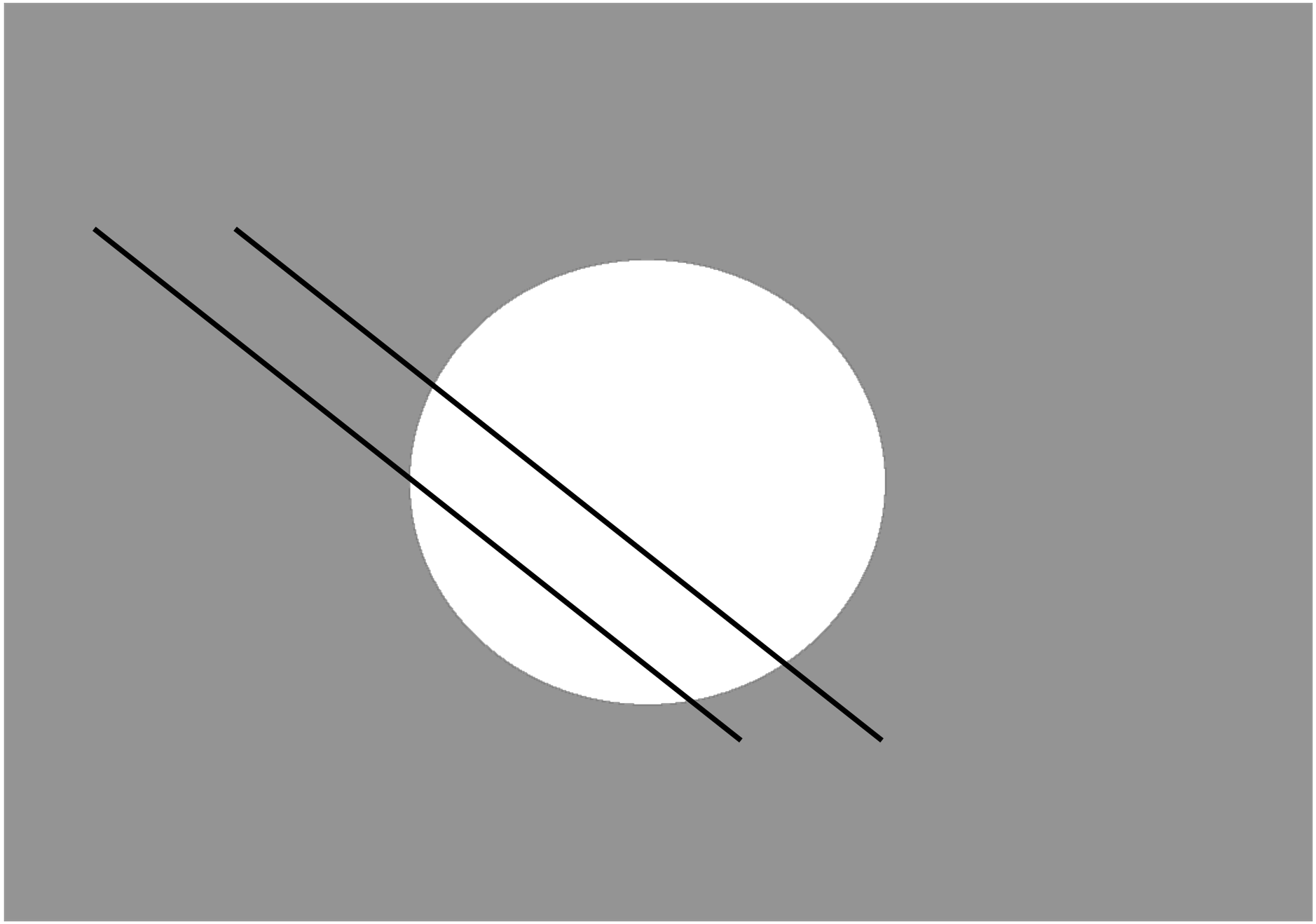
Aperture Problem

TECHNISCHE UNIVERSITÄT DARMSTADT



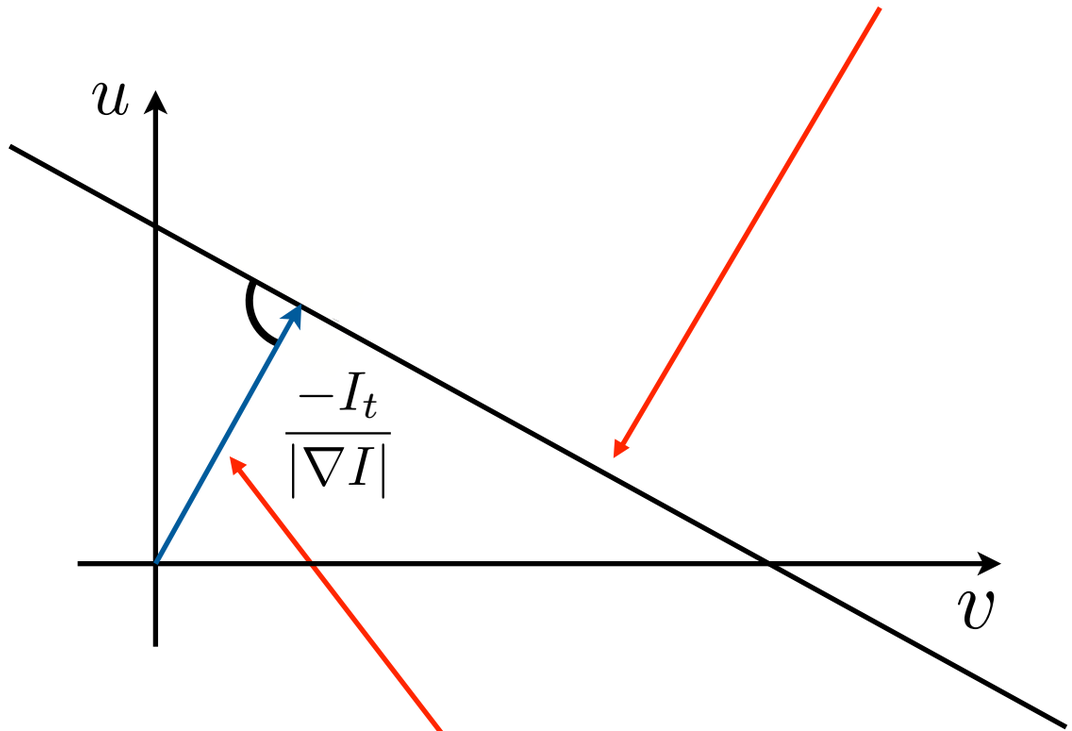
Aperture Problem

TECHNISCHE UNIVERSITÄT DARMSTADT



If we take a single image pixel, we get a constraint line:

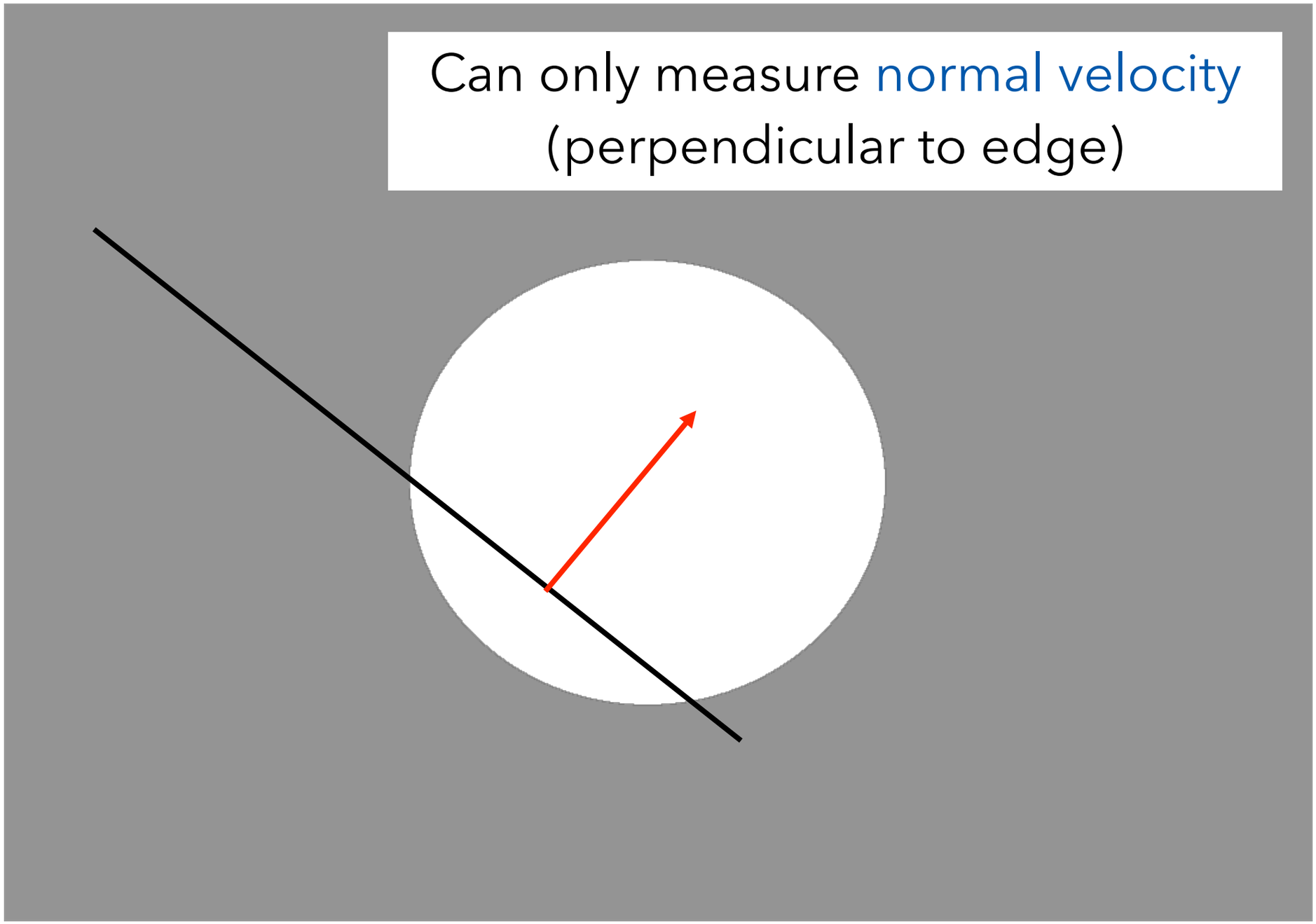
$$u \cdot I_x + v \cdot I_y + I_t = 0$$



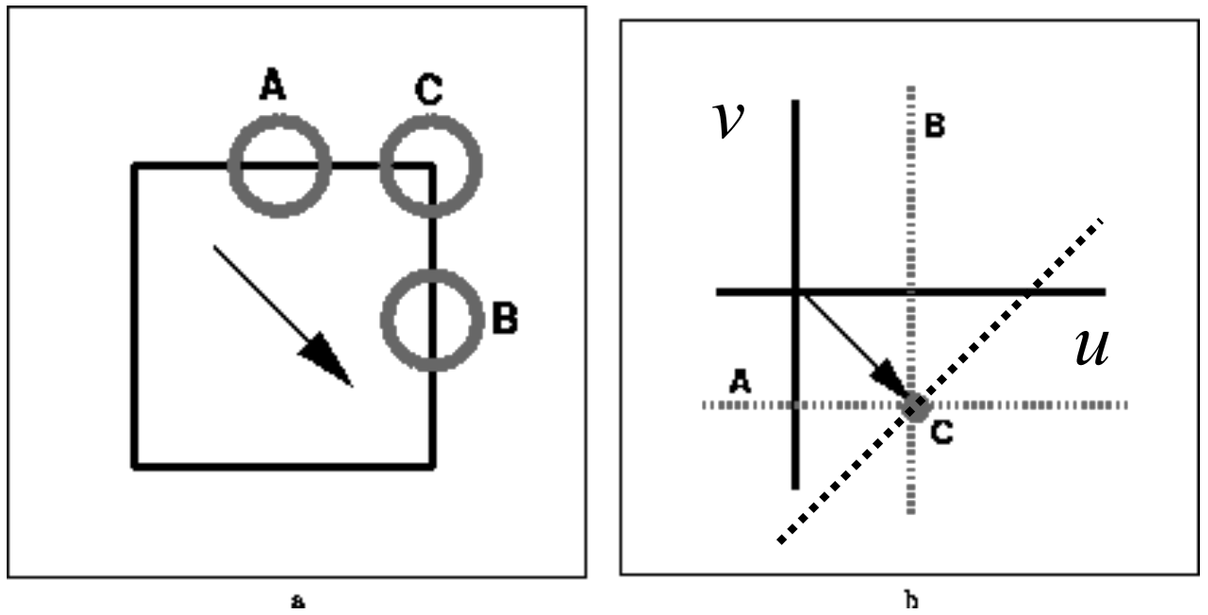
"Normal flow"

Aperture Problem

TECHNISCHE UNIVERSITÄT DARMSTADT



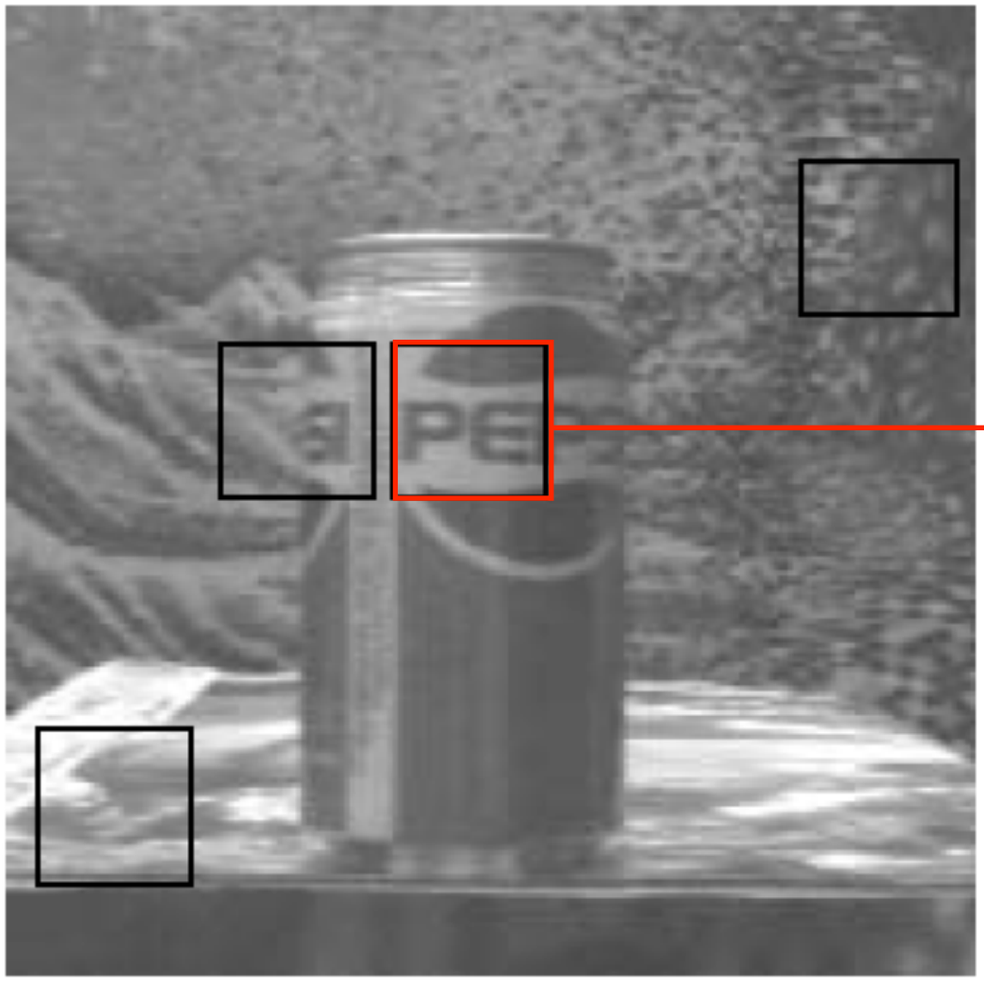
Multiple Constraints

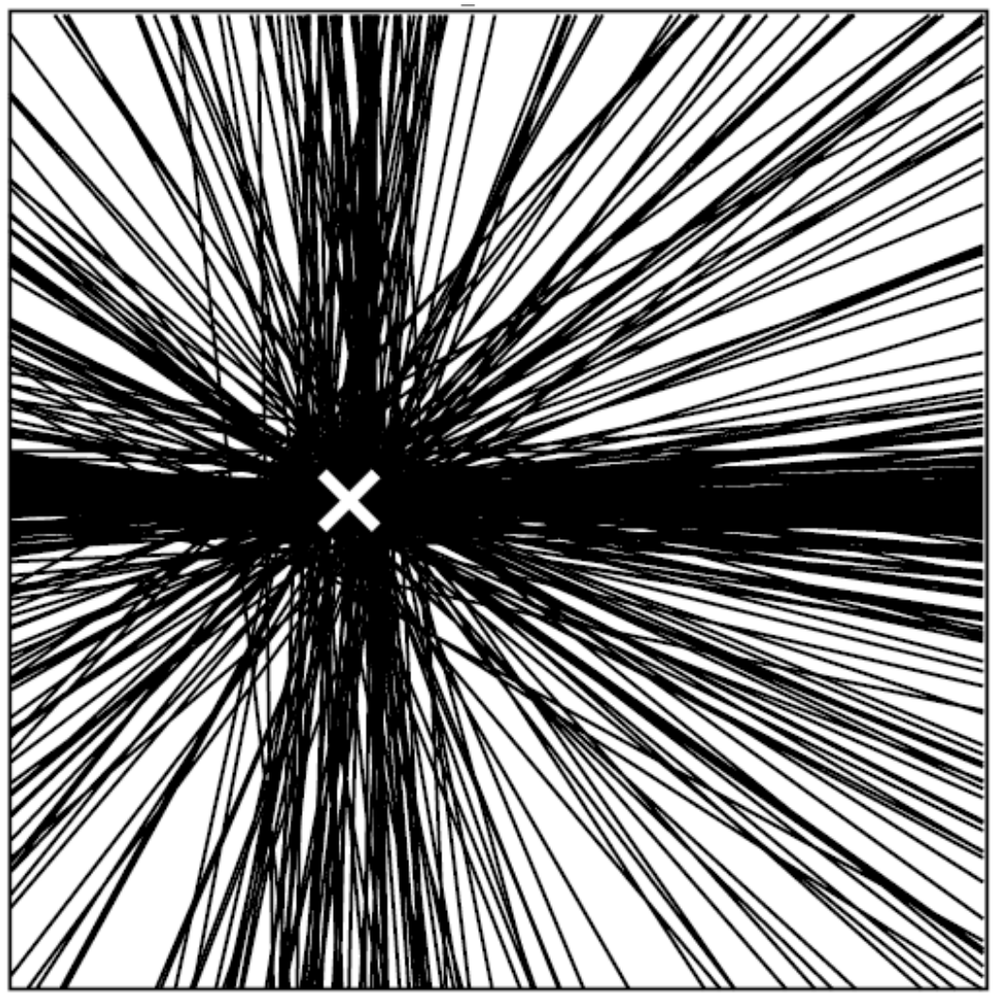


[Yair Weiss]

Combine multiple constraints to get an estimate of the velocity (not only the normal component).

Multiple Constraints





Area-Based Flow

TECHNISCHE UNIVERSITÄT DARMSTADT

- How do we combine multiple constraints?
  - Remember our assumptions.
  - We will assume spatial smoothness of the flow.
- More specifically:
  - Assume that the flow is constant in a region.

$$E_{SSD}(u,v) \approx \sum_{(x,y)\in R} \left(u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t)\right)^2$$

This is what we have been doing (sliding window).

- But how do we solve for the motion?

Optimization

TECHNISCHE UNIVERSITÄT DARMSTADT

$$E_{SSD}(u,v) \approx \sum_{(x,y)\in R} \big(u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t)\big)^2$$

Differentiate with respect to u and v and set this to zero.

$$\frac{\partial}{\partial u} E_{SSD}(u,v) \approx 2 \sum_{(x,y)\in R} \big(u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t)\big) I_x(x,y,t) = 0$$

$$\frac{\partial}{\partial v} E_{SSD}(u,v) \approx 2 \sum_{(x,y)\in R} \big(u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t)\big) I_y(x,y,t) = 0$$

Optimization

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\frac{\partial E_{SSD}}{\partial u} \approx 2\sum_R \big(u \cdot I_x + v \cdot I_y + I_t\big) I_x = 0$$

$$\frac{\partial E_{SSD}}{\partial v} \approx 2\sum_R \big(u \cdot I_x + v \cdot I_y + I_t\big) I_y = 0$$

Rearrange the terms and drop the constant:

$$\left[\sum_R I_x^2\right] u + \left[\sum_R I_x I_y\right] v = -\left[\sum_R I_x I_t\right]$$

$$\left[\sum_R I_x I_y\right] u + \left[\sum_R I_y^2\right] v = -\left[\sum_R I_y I_t\right]$$

Optimization

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\left[\sum_R I_x^2\right] u + \left[\sum_R I_x I_y\right] v = -\left[\sum_R I_x I_t\right]$$

$$\left[\sum_R I_x I_y\right] u + \left[\sum_R I_y^2\right] v = -\left[\sum_R I_y I_t\right]$$

System of 2 equations in 2 unknowns:

$$\begin{bmatrix} \sum_R I_x^2 & \sum_R I_x I_y \\ \sum_R I_x I_y & \sum_R I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum_R I_x I_t \\ -\sum_R I_y I_t \end{bmatrix}$$

Symmetric positive definite (invertible)

Structure tensor

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\begin{bmatrix} \sum_R I_x^2 & \sum_R I_x I_y \\ \sum_R I_x I_y & \sum_R I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum_R I_x I_t \\ -\sum_R I_y I_t \end{bmatrix}$$

Structure tensor

- The structure tensor is an important tool in vision.
  - The eigenvectors give the principal directions of the local image variation.
  - The eigenvalues indicate their strength.
- You have seen this in CV I when talking about feature detectors!

Optimization

$$\begin{bmatrix} \sum_R I_x^2 & \sum_R I_x I_y \\ \sum_R I_x I_y & \sum_R I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum_R I_x I_t \\ -\sum_R I_y I_t \end{bmatrix}$$

- Rewrite using the abbreviations from before:

$$\left(\sum_R \nabla I \nabla I^{\mathrm{T}} \right) \mathbf{u} = - \sum_R I_t \nabla I$$

- Invert structure tensor to obtain flow:

$$\mathbf{u} = - \left(\sum_R \nabla I \nabla I^{\mathrm{T}} \right)^{-1} \left(\sum_R I_t \nabla I \right)$$

Solving for u

$$\mathbf{u} = -\left(\sum_R \nabla I \nabla I^{\mathrm{T}}\right)^{-1} \left(\sum_R I_t \nabla I\right)$$

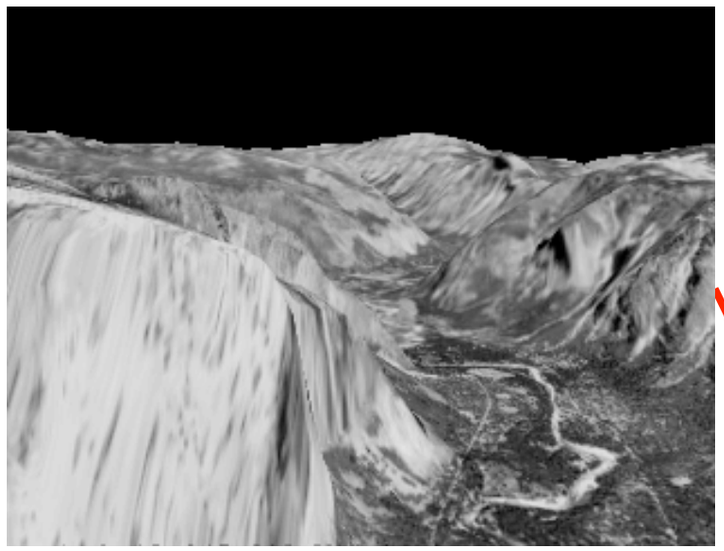
- This is a classical flow technique:

  B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. IJCAI, pp. 674–679, 1981.

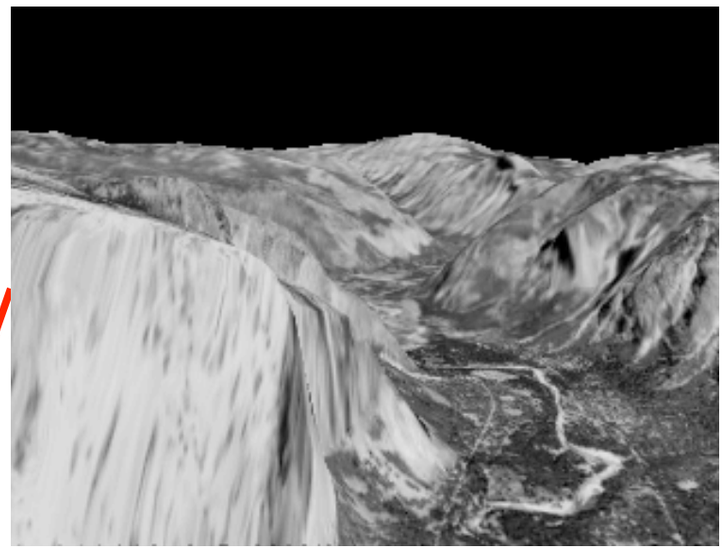
Optical Flow Estimation

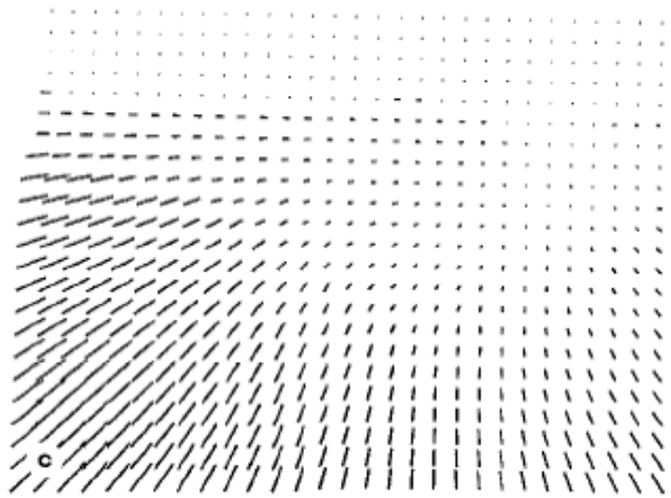
TECHNISCHE UNIVERSITÄT DARMSTADT

time t



time t+1





Our goal was to estimate a “dense” flow field with a motion vector for every pixel.

What have we achieved so far?

TECHNISCHE UNIVERSITÄT DARMSTADT

- We talked about the Lucas-Kanade method and estimated the flow using the following expression:

$$\mathbf{u} = -\left(\sum_R \nabla I \nabla I^{\mathrm{T}}\right)^{-1} \left(\sum_R I_t \nabla I\right)$$

- Initially we motivated this using a small window R.
- In the end we took the region to mean the entire image and computed a single motion vector for the image.
- This allows us to do image registration.
- We refined this based on an affine flow model.

Dense LK Flow

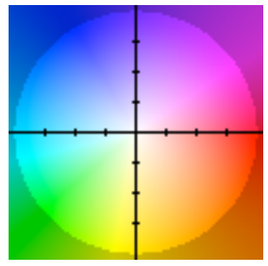
TECHNISCHE UNIVERSITÄT DARMSTADT

- What if we now want to estimate dense flow?
  - We can just take the region R to be a small region around every pixel and compute a flow vector for every pixel.
- But we face a crucial problem:
  - The LK method only works for small motions.
- Two workarounds (use both):
  - Iterative estimation.
  - Coarse-to-fine estimation.

Iterative Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Take image pair and compute LK flow (21x21 window):



Frame 1

Frame 2

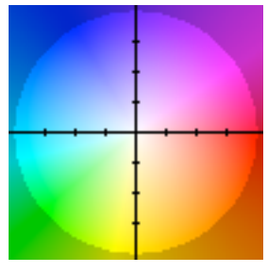
Frame 2 (warped)

Frame 1

Iterative Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

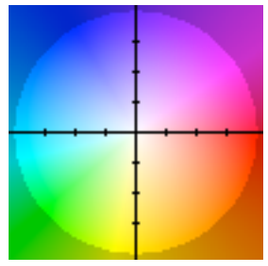
- Backward warp the second image toward the first:



Iterative Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

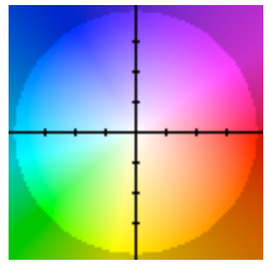
- Estimate incremental flow from warped image pair:



Iterative Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

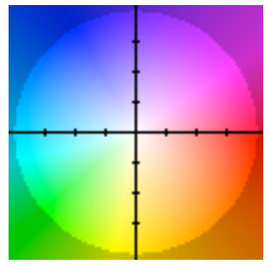
- Add incremental flow to previous estimate:



Iterative Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

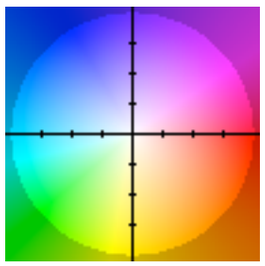
- Warp again and so on... until convergence.



Coarse-to-fine Estimation

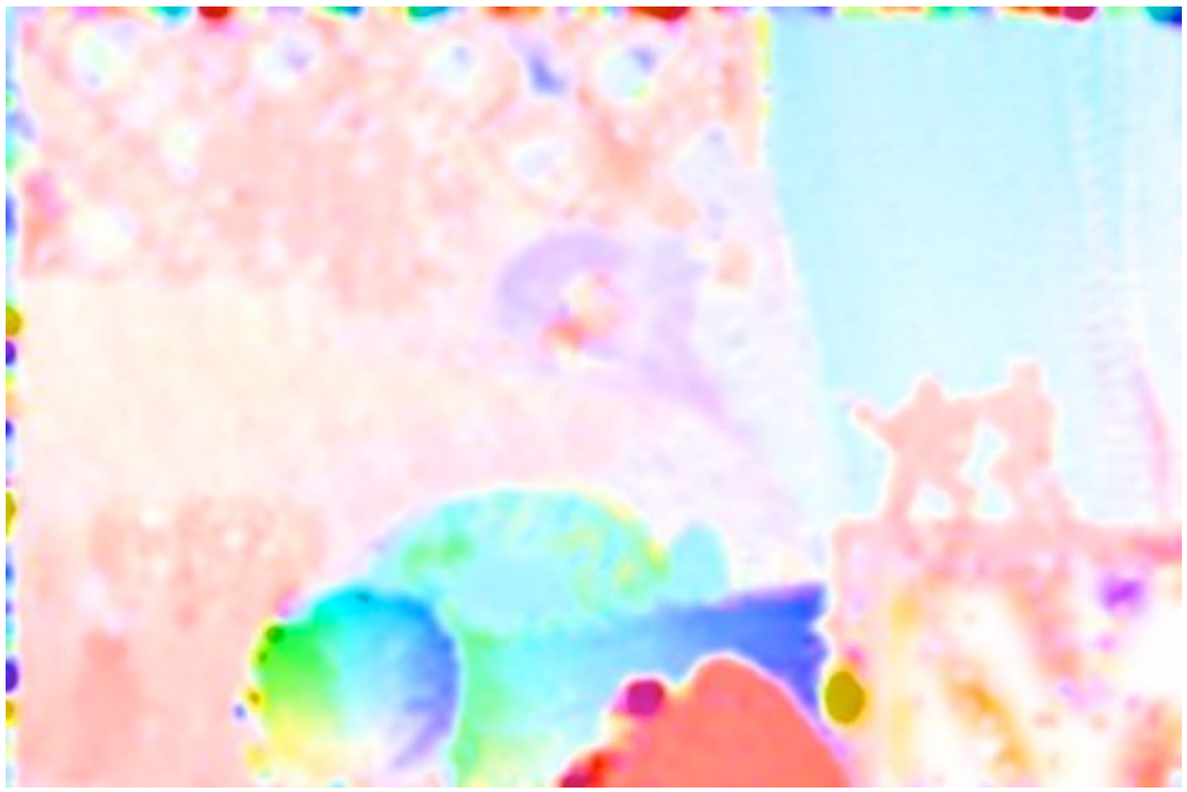
TECHNISCHE UNIVERSITÄT DARMSTADT

- Of coarse, we also apply the usual trick and use a Gaussian pyramid:
  - Initialize with the flow from a coarser level.
- If we do this on the previous image pair, we get this:

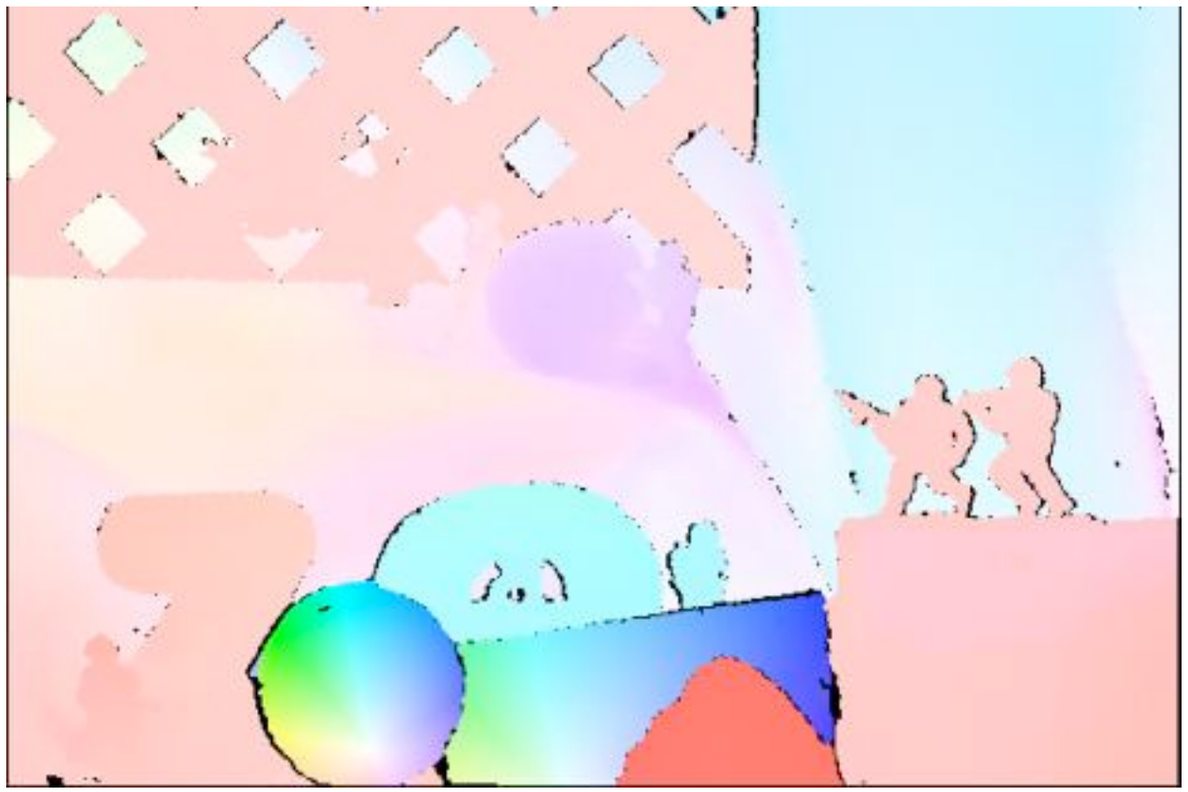


Is that a good result?

- Let's see:
  - Maybe not...



Pyramid LK method

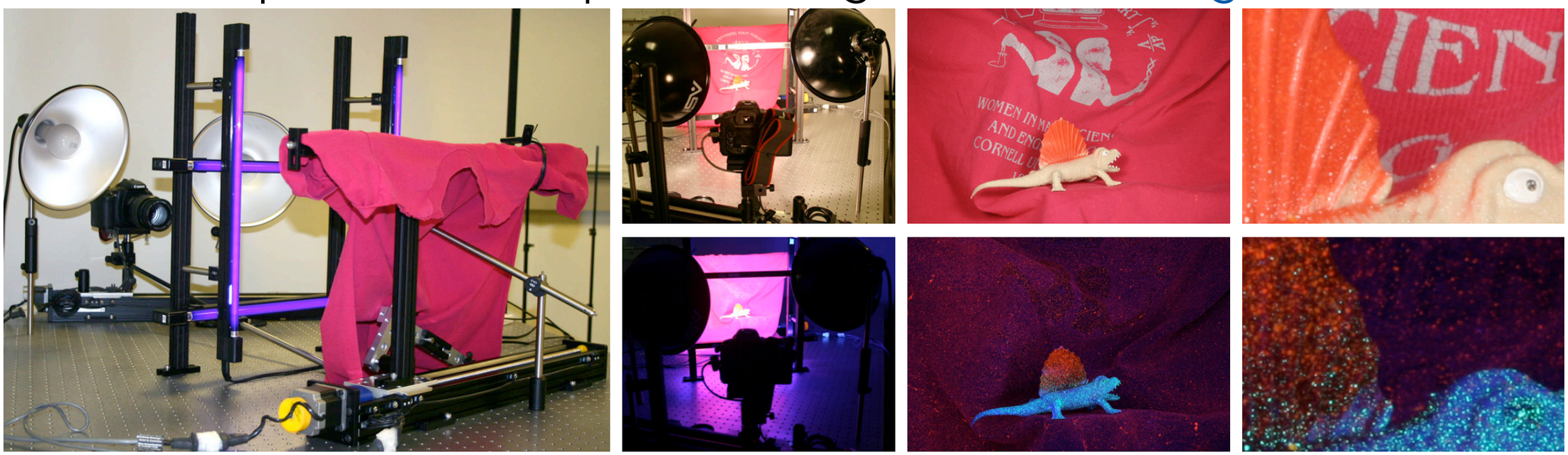


Ground truth

Aside: How to get the ground truth?

TECHNISCHE UNIVERSITÄT DARMSTADT

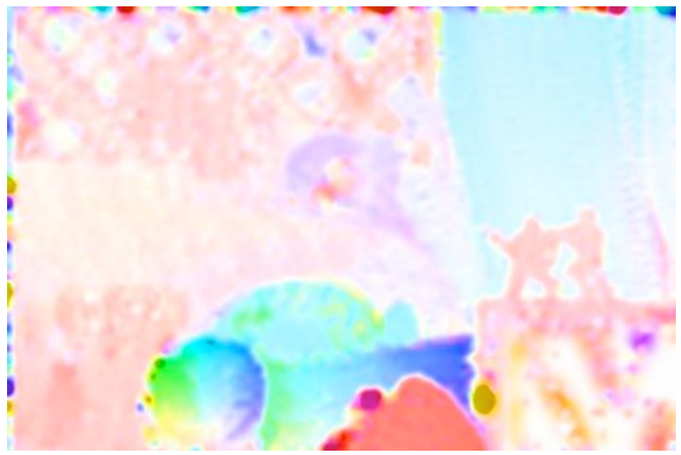
- Until recently, there were only very few image sequences with ground truth flow available.
  - All of them were synthetic and furthermore very simple.
- Middlebury optical flow benchmark [Baker et al., 07]:
  - Contains a number of complex synthetic and real sequences with ground truth.
- Real sequences are captured using normal + UV light:



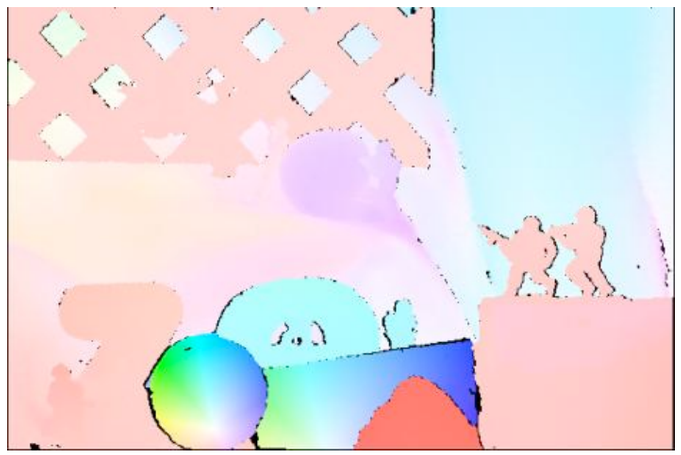
What is the problem?

TECHNISCHE UNIVERSITÄT DARMSTADT

- The window is **too big**!
  - All the discontinuities in the flow are smoothed over.
  - But discontinuities do exist, e.g., at motion boundaries.



- But: The window is also **too small**!
  - In some areas, the flow is really bad, because there is not enough image information in the window.
  - This happens in areas with little texture.



- LK is a local optical flow method.

First Option: Revisiting the assumptions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Before we move on to something more complex, revisit the assumptions:
  - brightness constancy: SSD error as objective
  - spatial coherence: motion within a patch is constant!

- Constant motion: Only translational motion within the image plane is allowed

Translational Model

What's wrong with the translational assumption (i.e. constant motion within a region R)?

How can we generalize it?

- Assume a parametric motion model within the region
  - Old idea
  - Recent renewed interest: efficient to implement

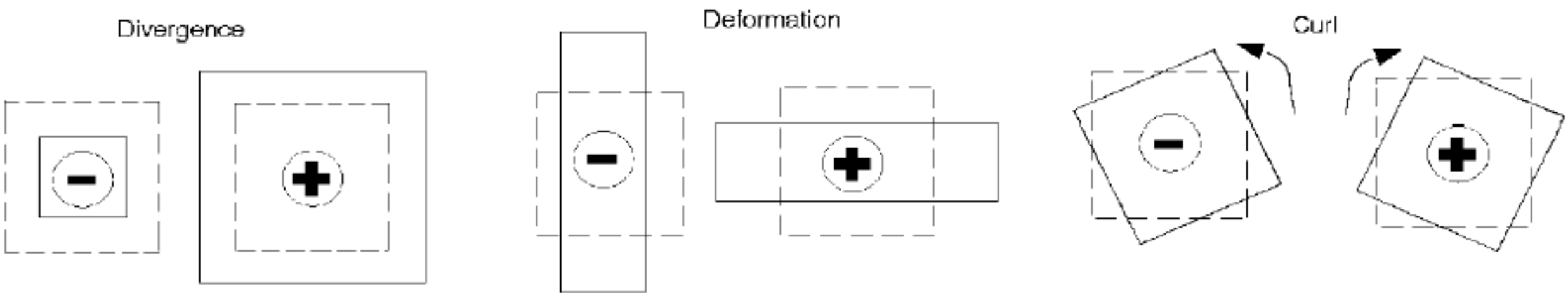
Affine Flow

TECHNISCHE UNIVERSITÄT DARMSTADT

- Parameterize the flow within the region:

$$\mathbf{u}(\mathbf{x};\, \boldsymbol{a}) = \begin{bmatrix} u(\mathbf{x};\, \boldsymbol{a}) \\ v(\mathbf{x};\, \boldsymbol{a}) \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix}$$

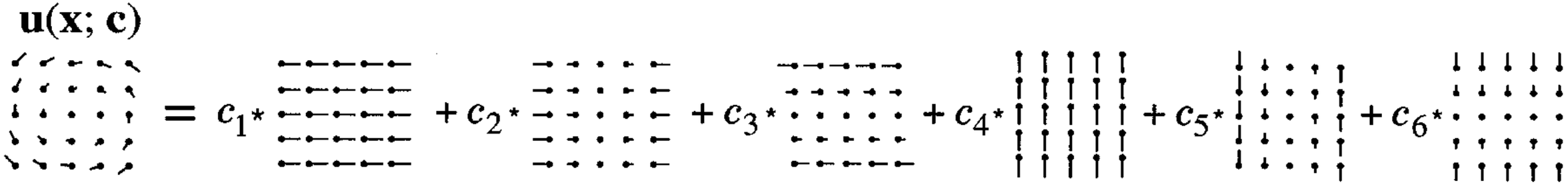
$$E(\boldsymbol{a}) = \sum_{\mathbf{x} \in R} \left(\nabla I(\mathbf{x})^{\mathrm{T}} \mathbf{u}(\mathbf{x};\, \boldsymbol{a}) + I_t(\mathbf{x}) \right)^2$$



Linear Basis

TECHNISCHE UNIVERSITÄT DARMSTADT

This can be thought of as a basis representation of the flow within the region.



$$\mathbf{u}(\mathbf{x};\, \boldsymbol{c}) = \sum_{j=1}^{n} c_j \boldsymbol{B}_j(\mathbf{x})$$

Optimization

$$E(\boldsymbol{A}) = \sum_{(x,y)\in R} \left(I_x a_1 + I_x a_2 x + I_x a_3 y + I_y a_4 + I_y a_5 x + I_y a_6 y + I_t\right)^2$$

Differentiate wrt the a_i and set equal to zero:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x^2 x & \sum I_x^2 y & \sum I_x I_y & \sum I_x I_y x & \sum I_x I_y y \\ \sum I_x^2 x & \sum I_x^2 x^2 & \sum I_x^2 xy & \sum I_x I_y x & \sum I_x I_y x^2 & \sum I_x I_y xy \\ & & & \vdots & & \\ & & & & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_x I_t x \\ \sum I_x I_t y \\ \sum I_y I_t \\ \sum I_y I_t x \\ \sum I_y I_t y \end{bmatrix}$$

Affine Flow Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- As before, we solve for the parameters (now only more) by inverting the matrix.
- We can use the iterative image registration (CV1)
  - help overcome small motion assumption
- Need warping in its full generality:

affine

Why Affine?

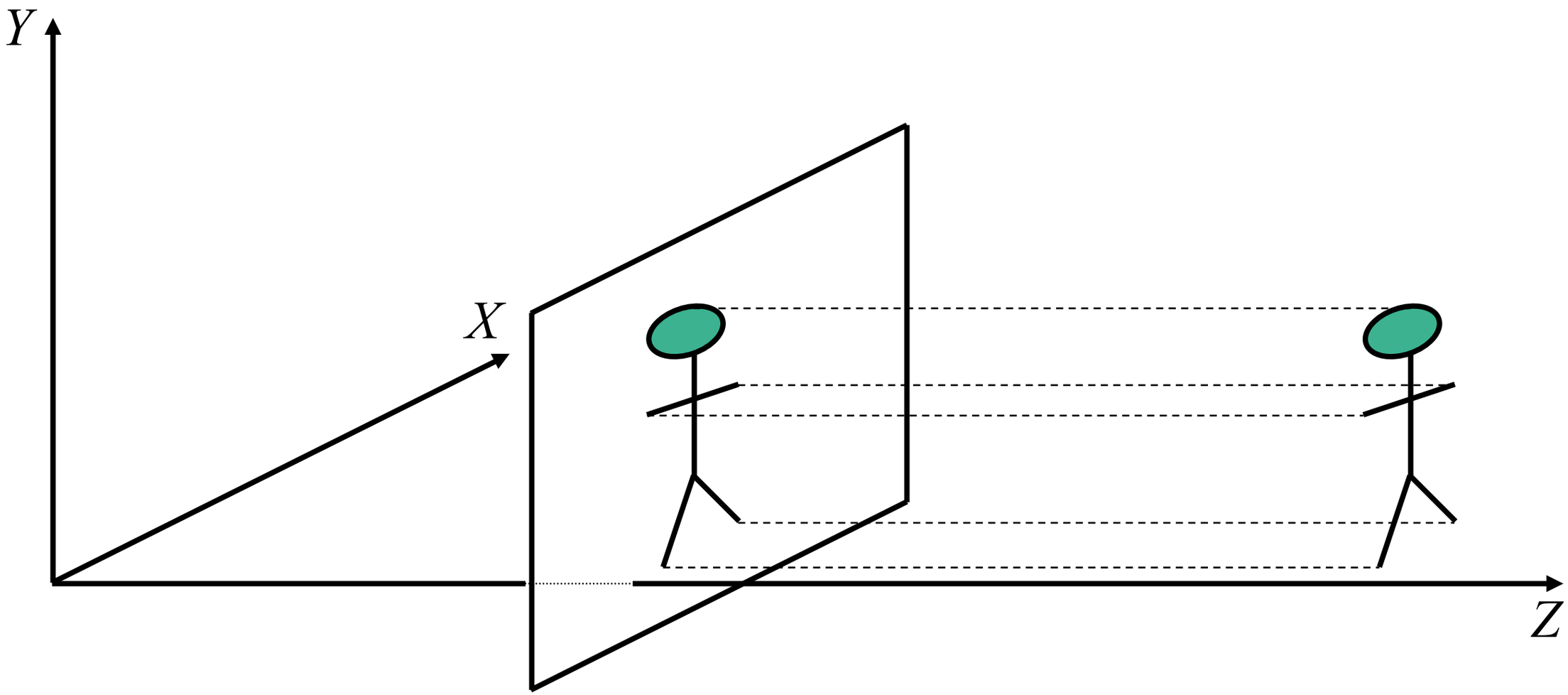
TECHNISCHE UNIVERSITÄT DARMSTADT

- Where does this affine approximation come from?

- All our models are approximations to the world. What are the assumptions in the affine approximation?

- For this we need some geometry.

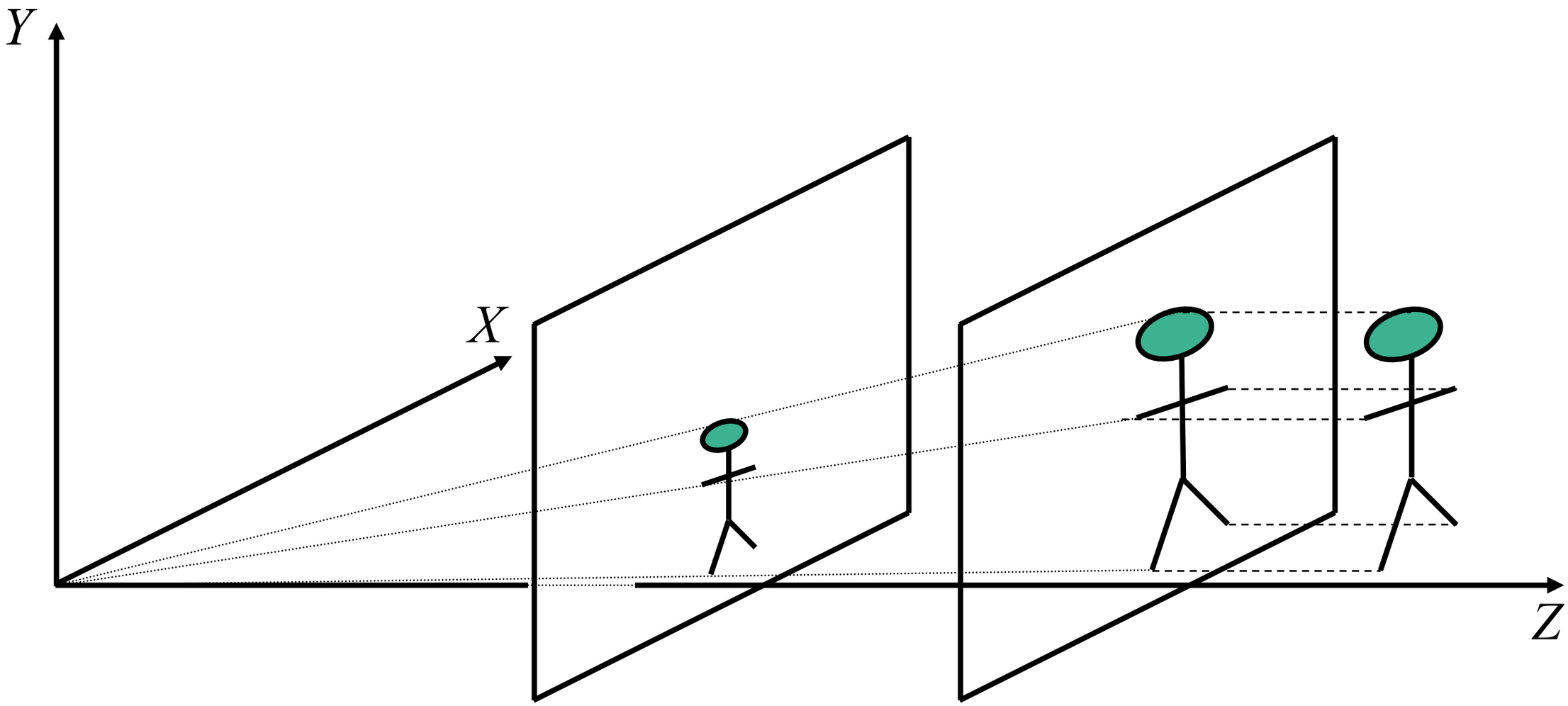
Recall: Orthographic Projection



Assume the world is infinitely far away.

$$(x, y, z) \rightarrow (x', y')$$

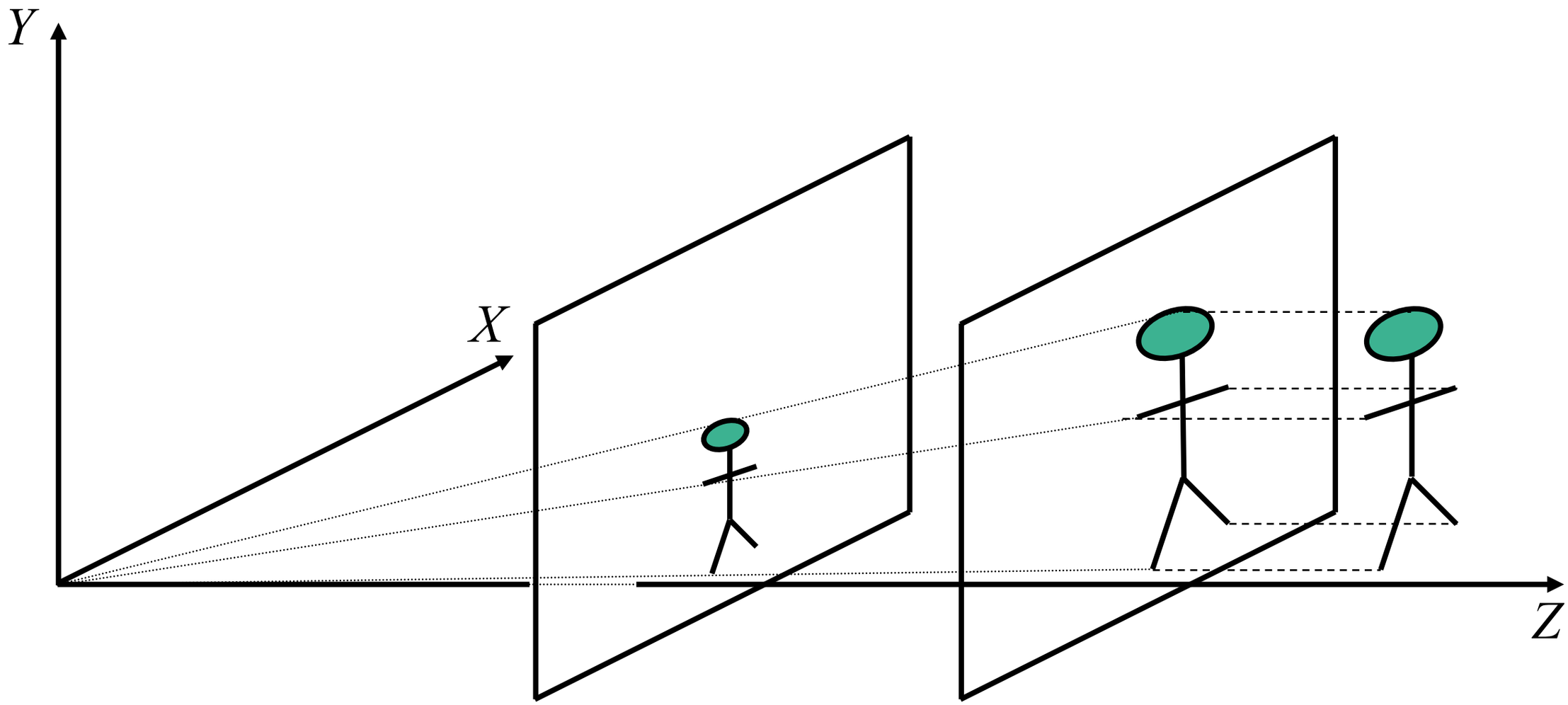
Weak Perspective (scaled orthography)



Assume variation in depth is small relative to the distance from the camera.

Approximate scene as a fronto-parallel plane

Weak Perspective (scaled orthography)



$$(x', y') = (sx, sy)$$

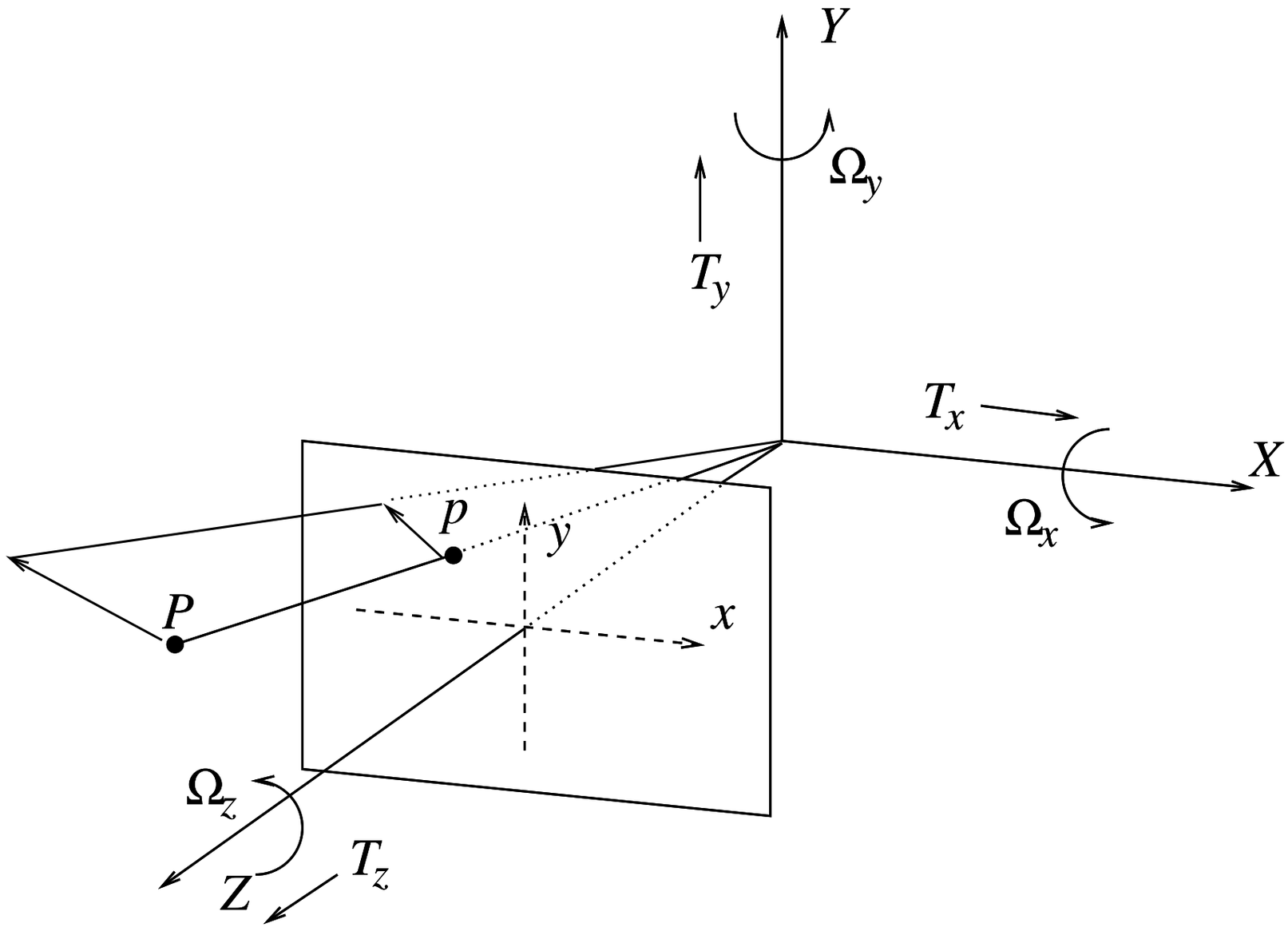
Claim

TECHNISCHE UNIVERSITÄT DARMSTADT

- For **small motions**, affine flow approximates the motion of a **plane** viewed under **orthographic projection**.

$$\mathbf{u}(\mathbf{x};\, \boldsymbol{a}) = \begin{bmatrix} u(\mathbf{x};\, \boldsymbol{a}) \\ v(\mathbf{x};\, \boldsymbol{a}) \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix}$$

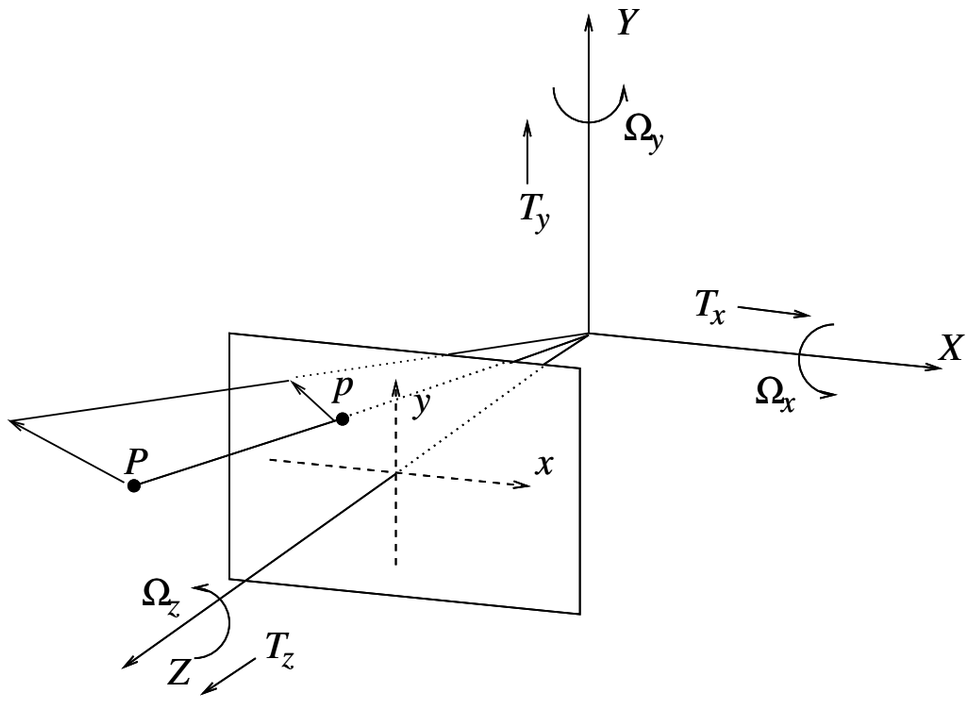
Recall: 3D motion



Motion Models

TECHNISCHE UNIVERSITÄT DARMSTADT

3D Rigid Motion



rotation translation

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \boldsymbol{T} = \mathbf{R}_{\mathbf{Z}}^{\boldsymbol{\Omega}_\mathbf{Z}} \mathbf{R}_{\mathbf{Y}}^{\boldsymbol{\Omega}_\mathbf{Y}} \mathbf{R}_{\mathbf{X}}^{\boldsymbol{\Omega}_\mathbf{X}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \boldsymbol{T}$$

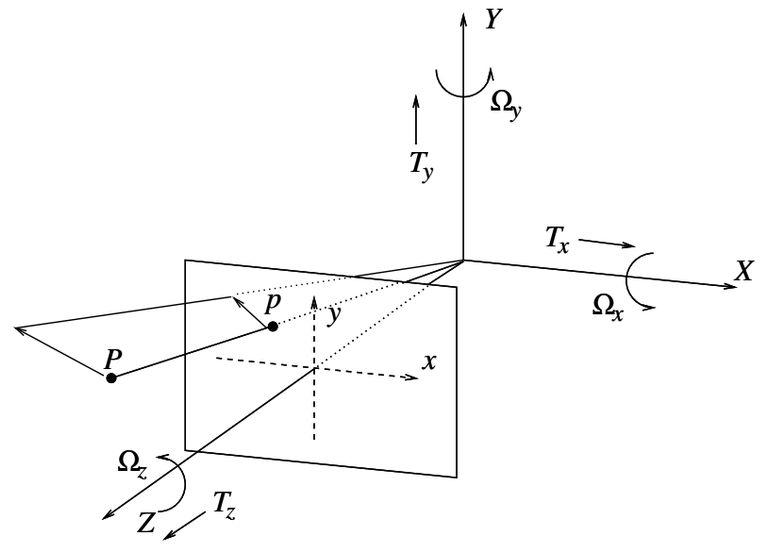
Euler angles

Rotation

TECHNISCHE UNIVERSITÄT DARMSTADT

Review: FP Ch 2.1, Euclidean Geometry

$$\mathbf{R}_Z^{\Omega_Z} = \begin{bmatrix} \cos\Omega_Z & -\sin\Omega_Z & 0 \\ \sin\Omega_Z & \cos\Omega_Z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\mathbf{R}_Y^{\Omega_Y} = \begin{bmatrix} \cos\Omega_Y & 0 & \sin\Omega_Y \\ 0 & 1 & 0 \\ -\sin\Omega_Y & 0 & \cos\Omega_Y \end{bmatrix} \qquad \mathbf{R}_X^{\Omega_X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Omega_X & -\sin\Omega_X \\ 0 & \sin\Omega_X & \cos\Omega_X \end{bmatrix}$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos\Omega_Y\cos\Omega_Z & \sin\Omega_X\sin\Omega_Y\cos\Omega_Z - \cos\Omega_X\sin\Omega_Z & \cos\Omega_X\sin\Omega_Y\cos\Omega_Z + \sin\Omega_X\sin\Omega_Z \\ \cos\Omega_Y\sin\Omega_Z & \sin\Omega_X\sin\Omega_Y\sin\Omega_Z + \cos\Omega_X\cos\Omega_Z & \cos\Omega_X\sin\Omega_Y\sin\Omega_Z - \sin\Omega_X\cos\Omega_Z \\ -\sin\Omega_Y & \sin\Omega_X\cos\Omega_Y & \cos\Omega_X\cos\Omega_Y \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

Assumption: Small Motion

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos\Omega_Y \cos\Omega_Z & \sin\Omega_X \sin\Omega_Y \cos\Omega_Z - \cos\Omega_X \sin\Omega_Z & \cos\Omega_X \sin\Omega_Y \cos\Omega_Z + \sin\Omega_X \sin\Omega_Z \\ \cos\Omega_Y \sin\Omega_Z & \sin\Omega_X \sin\Omega_Y \sin\Omega_Z + \cos\Omega_X \cos\Omega_Z & \cos\Omega_X \sin\Omega_Y \sin\Omega_Z - \sin\Omega_X \cos\Omega_Z \\ -\sin\Omega_Y & \sin\Omega_X \cos\Omega_Y & \cos\Omega_X \cos\Omega_Y \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

Assume that rotation is small

$$\cos\theta \approx 1 \qquad \sin\theta \approx \theta$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & -\Omega_Z & \Omega_Y \\ \Omega_Z & 1 & -\Omega_X \\ -\Omega_Y & \Omega_X & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

3D Motion

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \left(\begin{bmatrix} 0 & -\Omega_Z & \Omega_Y \\ \Omega_Z & 0 & -\Omega_X \\ -\Omega_Y & \Omega_X & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} = \begin{bmatrix} X' - X \\ Y' - Y \\ Z' - Z \end{bmatrix} = \begin{bmatrix} 0 & -\Omega_Z & \Omega_Y \\ \Omega_Z & 0 & -\Omega_X \\ -\Omega_Y & \Omega_X & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

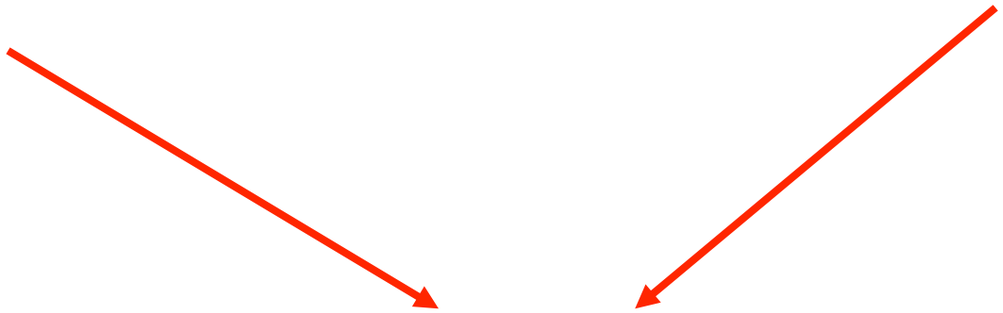
$$\begin{aligned} V_X &= -\Omega_Z Y + \Omega_Y Z + T_X \\ V_Y &= \Omega_Z X - \Omega_X Z + T_Y \\ V_Z &= -\Omega_Y X + \Omega_X Y + T_Z \end{aligned}$$

Assumption:
Planar World + Orthographic Projection

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\begin{aligned}
x &= X \\
y &= Y \\
Z &= a + bX + cY
\end{aligned}
$$

$$
\begin{aligned}
u &= V_X = -\Omega_Z y + \Omega_Y Z + T_X \\
v &= V_Y = \Omega_Z x - \Omega_X Z + T_Y
\end{aligned}
$$



$$
\begin{aligned}
u &= V_X = -\Omega_Z y + \Omega_Y (a + bx + cy) + T_X \\
v &= V_Y = \Omega_Z x - \Omega_X (a + bx + cy) + T_Y
\end{aligned}
$$

Assumption: Planar World + Orthographic Projection

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\begin{aligned}
u &= -\Omega_Z y + \Omega_Y (a + bx + cy) + T_X \\
v &= \Omega_Z x - \Omega_X (a + bx + cy) + T_Y
\end{aligned}
$$

$$
\begin{aligned}
u &= \Omega_Y b x + (\Omega_Y c - \Omega_Z) y + (\Omega_Y a + T_X) \\
v &= (\Omega_Z - \Omega_X b) x - \Omega_X c y + (T_Y - \Omega_X a)
\end{aligned}
$$

$$
\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix}
$$

Substitute:

$$
\begin{aligned}
a_1 &= \Omega_Y a + T_X \\
a_2 &= \Omega_Y b \\
a_3 &= \Omega_Y c - \Omega_Z \\
a_4 &= T_Y - \Omega_X a \\
a_5 &= \Omega_Z - \Omega_X b \\
a_6 &= -\Omega_X c
\end{aligned}
$$

Affine Flow

TECHNISCHE UNIVERSITÄT DARMSTADT

- Small motion assumption
  - e.g. at video frame rate
- Planar surface
  - look at only a small region of the scene
- Orthographic projection
  - surface distant from camera
  - long focal length

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix}$$

Second Option: Global regularization

TECHNISCHE UNIVERSITÄT DARMSTADT

- Remember: We were faced with challenges similar to those in LK in stereo.
  - Window-based approaches turned out to be problematic.
  - In stereo, we introduced prior knowledge which supplied global regularity on the disparity map.
- In optical flow, we can do the same:
  - We impose global regularity of the flow field, and by that impose prior knowledge.

Horn & Schunk

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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?

TECHNISCHE UNIVERSITÄT DARMSTADT

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

TECHNISCHE UNIVERSITÄT DARMSTADT

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

- 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

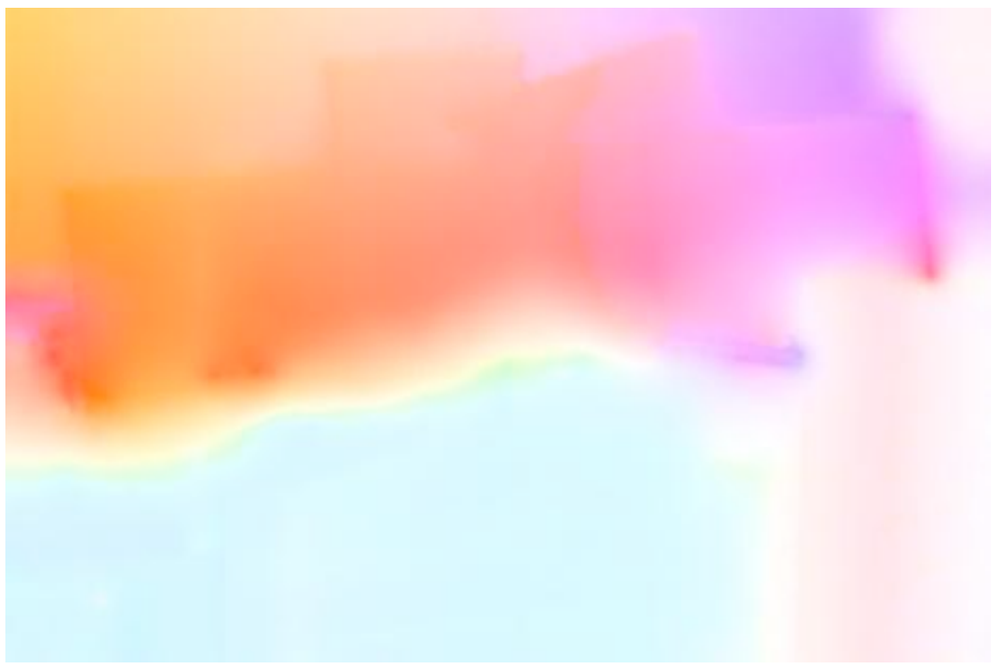
TECHNISCHE UNIVERSITÄT DARMSTADT

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

- 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.

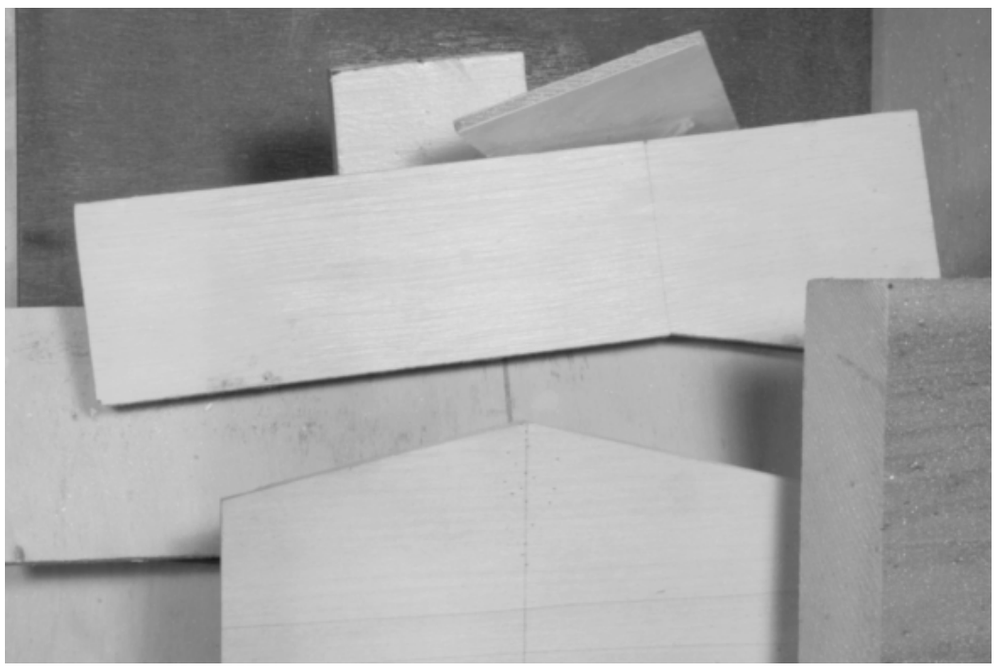
Some H&S results

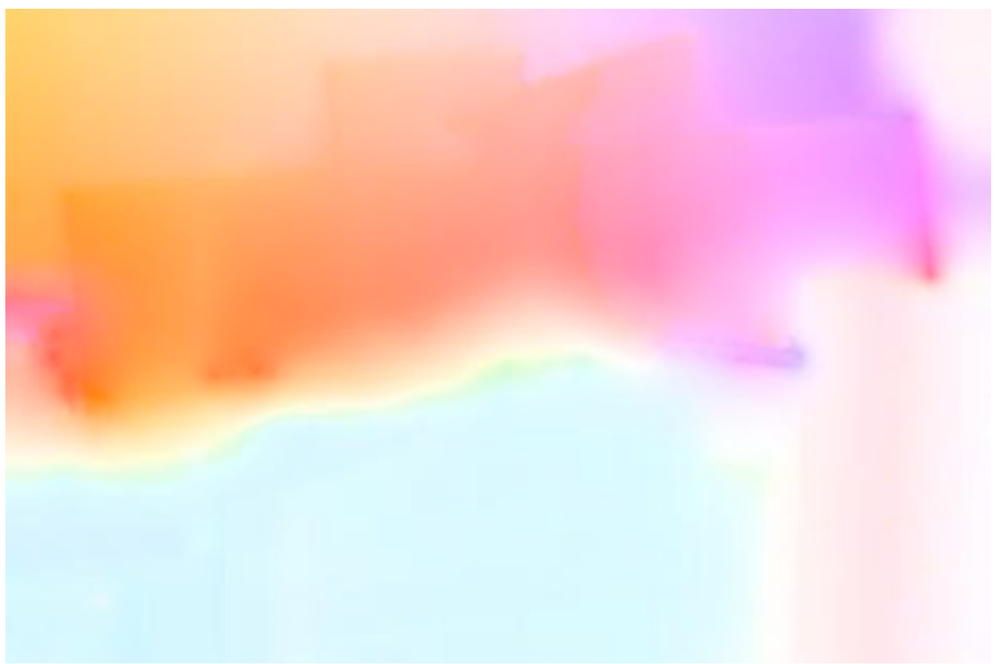
TECHNISCHE UNIVERSITÄT DARMSTADT



Some H&S results

TECHNISCHE UNIVERSITÄT DARMSTADT





Problem with H&S

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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:

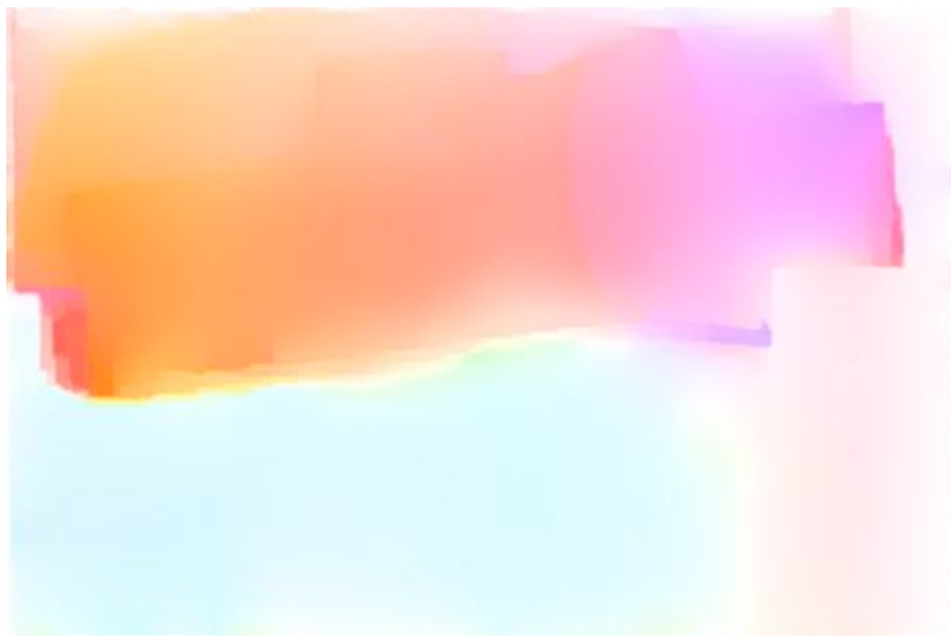
$$p(\boldsymbol{u}, \boldsymbol{v} | I^0, I^1) \propto p(I^1 | \boldsymbol{u}, \boldsymbol{v}, I^0) \cdot p(\boldsymbol{u}, \boldsymbol{v} | I^0)$$

Observation model
embodies brightness constancy

"Prior" on the flow field

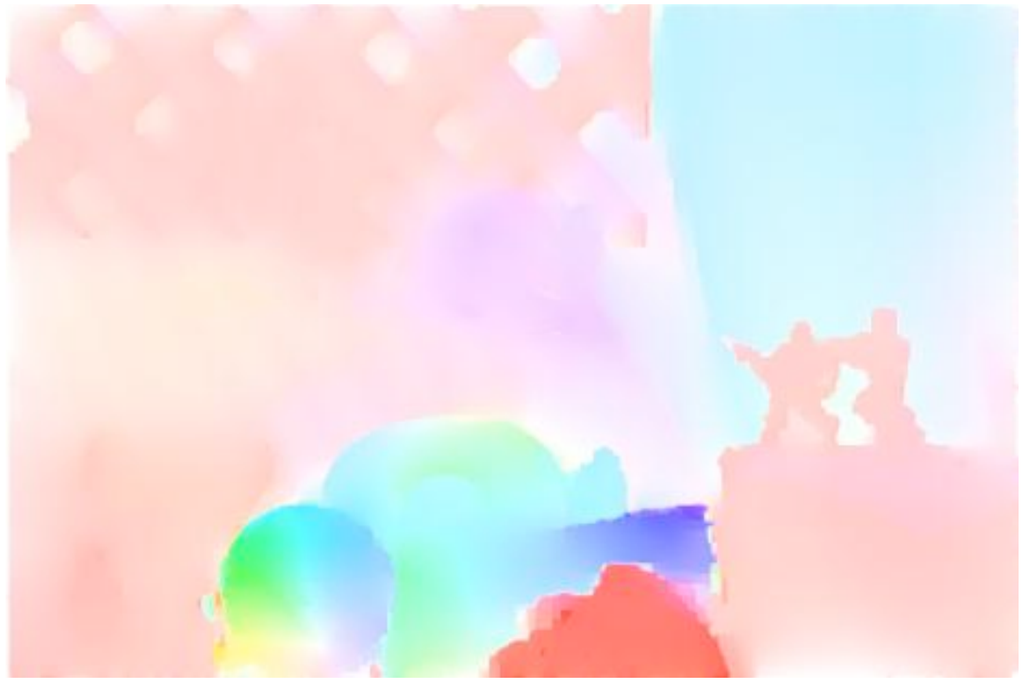
Some More Results

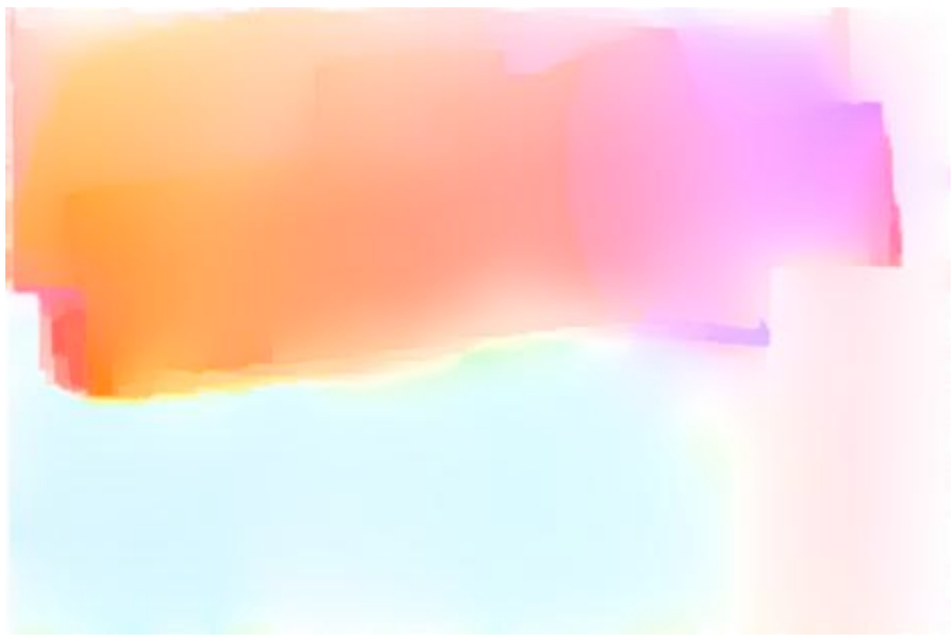
TECHNISCHE UNIVERSITÄT DARMSTADT



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

TECHNISCHE UNIVERSITÄT DARMSTADT

Method by [Sun et al. 2010]