Computer Vision 2 — Exercise 1

Generalized Lucas-Kanade Tracking & Kalman Filter

M.Sc. Francis Engelmann, Dr. Jörg Stückler
engelmann@vision.rwth-aachen.de, stueckler@vision.rwth-aachen.de

RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de

Content Exercise 1

- **Organisation:**
  - Exercises are not mandatory for exam participation
  - Complete solutions will be provided on L2P
- **Question 1: Generalized Lucas-Kanade Tracking**
  - Lucas-Kanade Optical Flow
  - Basic LK Template Tracking
  - Generalized LK Template Tracking
- **Question 2: Kalman Filter**
  - Constant velocity model

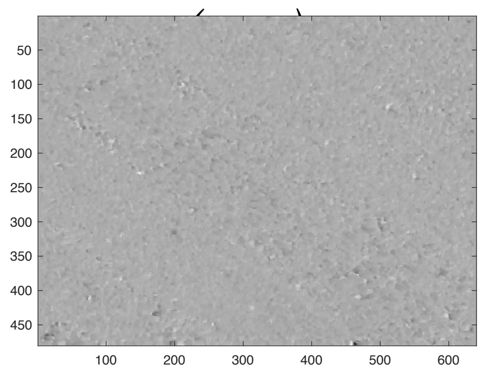
Q1 a) Lucas-Kanade Optical Flow

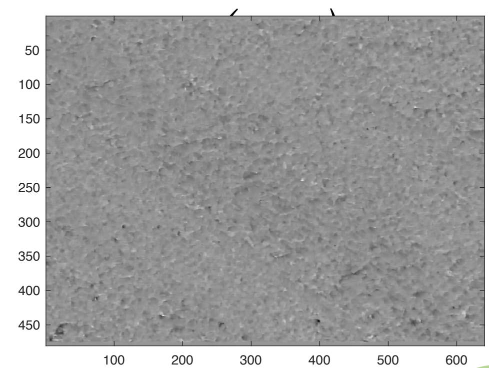
- **Given:** two sequential input images $\boldsymbol{I}$ at different time steps $\boldsymbol{t}$ and $\boldsymbol{t\text{-}1}$

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

$I(x,y,t)$

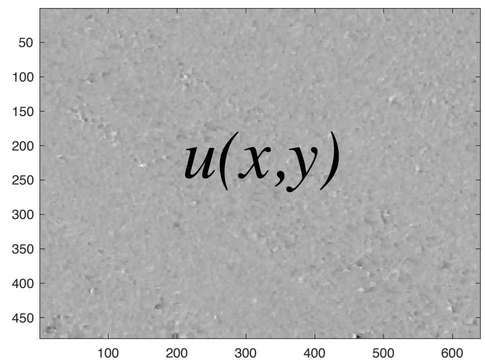
- **Goal:** estimate apparent motion $\boldsymbol{u}$ in horizontal and $\boldsymbol{v}$ vertical direction

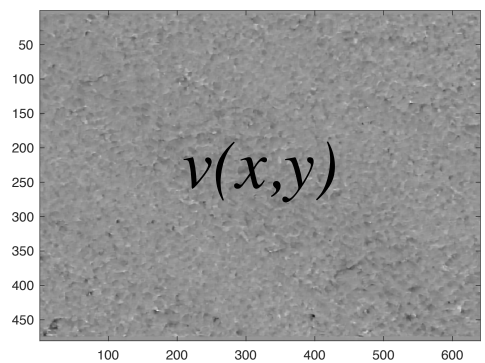


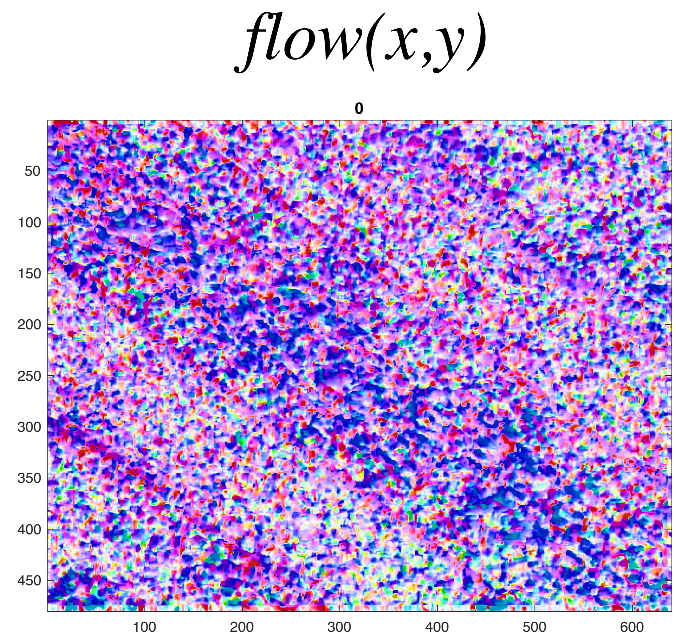


Q1 a) Lucas-Kanade Optical Flow

- Display flow as color map:







Q1 a) Lucas-Kanade Optical Flow

- Brightness Constancy Assumption:

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

- First order Taylor approximation:

$$I(x, y, t-1) \approx I(x, y, t) + I_x \cdot u(x, y) + I_y \cdot v(x, y)$$

- Rearranging:

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

Spatial derivatives (I_x, I_y) — Temporal derivative (I_t)

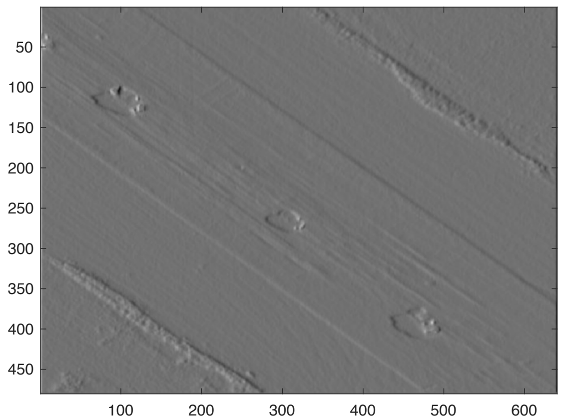
Q1 a) Lucas-Kanade Optical Flow

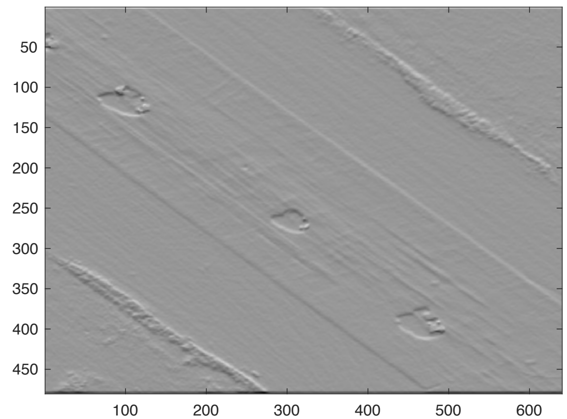
- Solve for ***u*** and ***v***:

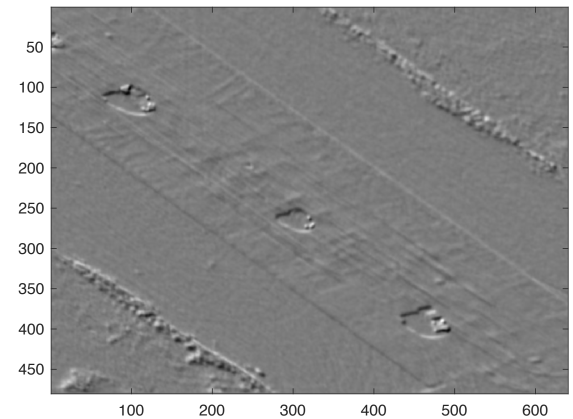
$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

Spatial derivatives (I_x, I_y) — Temporal derivative (I_t)

$$I_x = \frac{\partial I(x,y)}{\partial x} \qquad I_y = \frac{\partial I(x,y)}{\partial y} \qquad I_t = I(x,y,t) - I(x,y,t-1)$$







Q1 a) Lucas-Kanade Optical Flow

- Spatial coherence constraint:
  - Pixels in n x n neighborhood move the same. Here n=5.

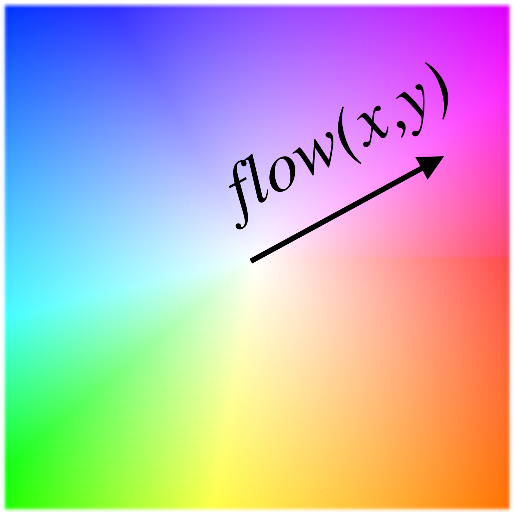
$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

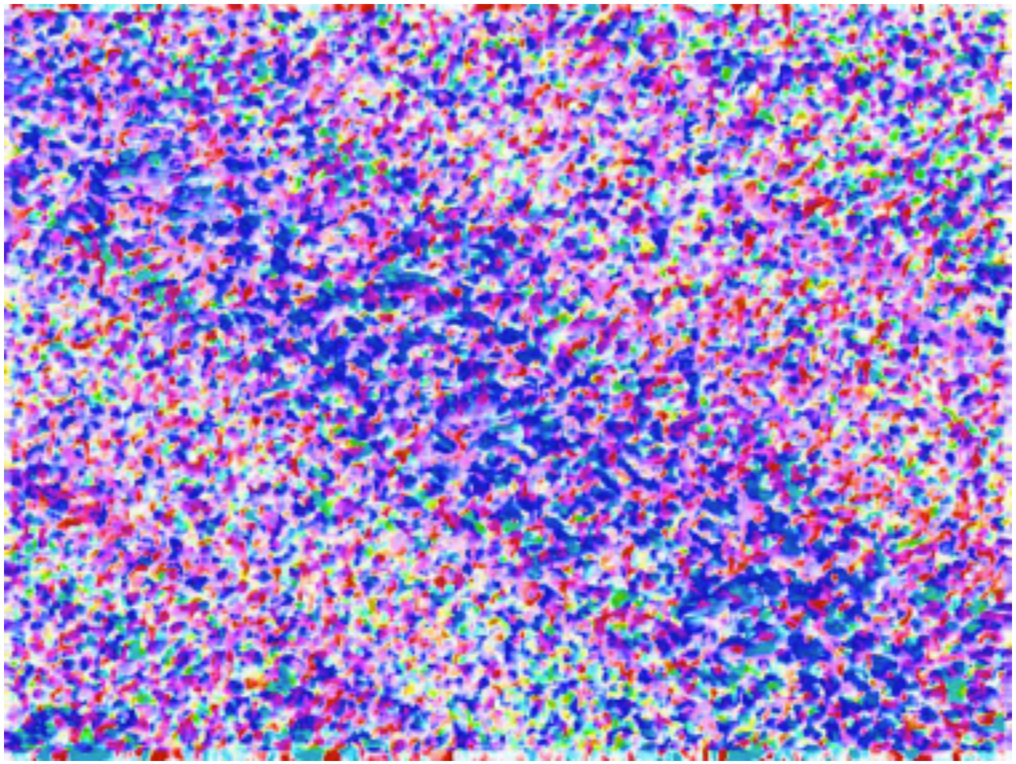
$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

$$\underset{25\text{x}2}{A} \;\; \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Q1 b) Lucas-Kanade Optical Flow

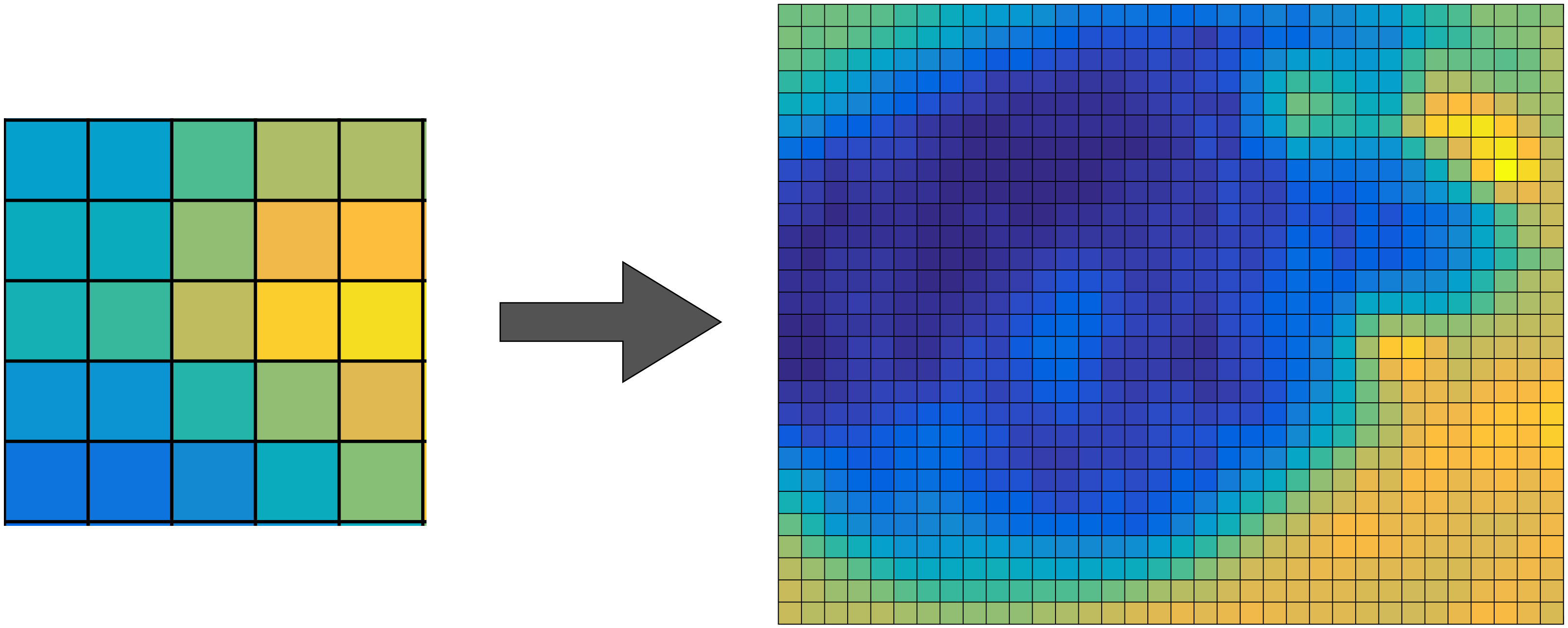
- Apply color map to estimated optical flow vector





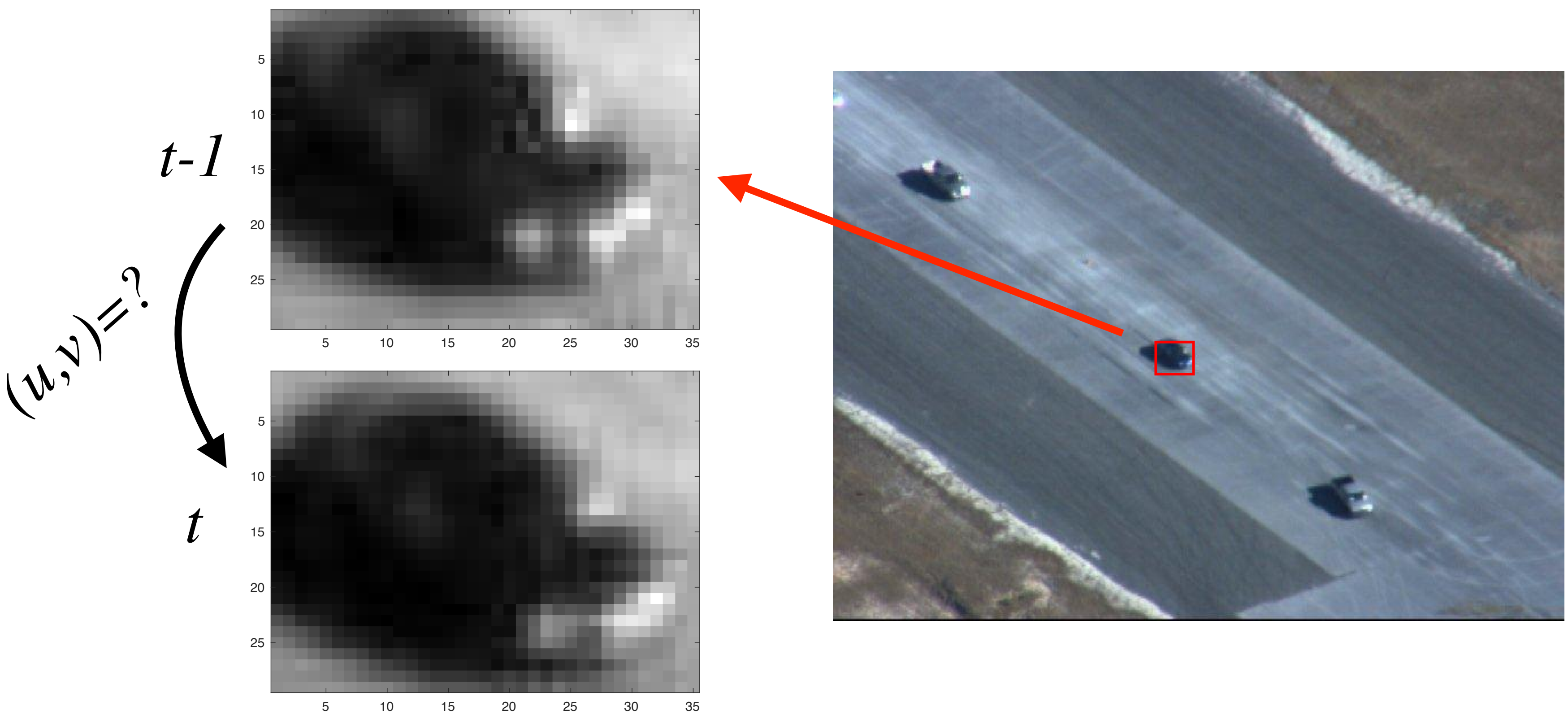
Q1 c) Lucas-Kanade Template Tracking

- Replace 5x5 window with user-specified template window
- Compute flow-vector per template, not per pixel!



Q1 c) Lucas-Kanade Template Tracking

- Replace <u>5x5</u> window with user-specified <u>template</u> window
- Compute flow-vector per <u>template</u>, not per <u>pixel</u>!



Q1 c) Lucas-Kanade Template Tracking

- Sum over pixels inside template-window
  - Taylor expansion

$$E(u,v) = \sum_{\mathbf{x}} \left[I(x+u, y+v) - T(x,y) \right]^2$$

$$\approx \sum_{\mathbf{x}} \left[I(x,y) + uI_x(x,y) + vI_y(x,y) - T(x,y) \right]^2$$

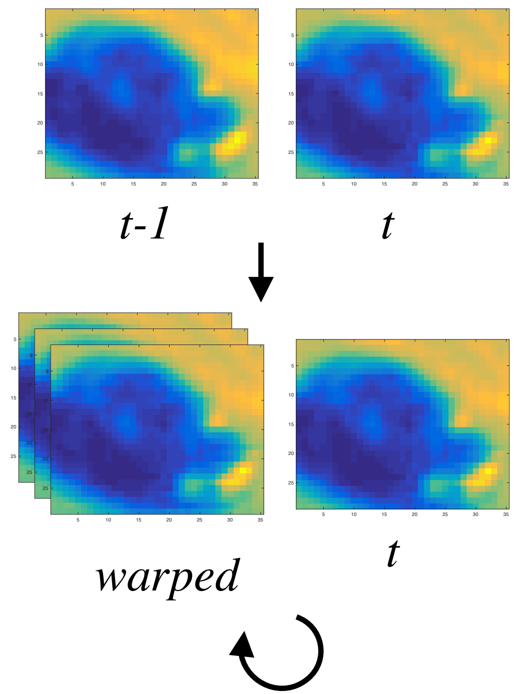
$$= \sum_{\mathbf{x}} \left[uI_x(x,y) + vI_y(x,y) + D(x,y) \right]^2 \quad \text{with} \quad D = I - T$$

$$\sum_{\mathbf{x}} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \sum_{\mathbf{x}} \begin{bmatrix} I_x D \\ I_y D \end{bmatrix}$$

$\Rightarrow$ **Solve via least-squares**

Q1 c) Lucas-Kanade Template Tracking

- Iterative LK Refinement:
  - Compute flow between previous template position and current template position.
  - Refined by recomputing flow between on warped previous template position and current template position.
  - Repeat until convergence.



Q1 c) Lucas-Kanade Template Tracking

- **Problem:** Assumption of pure translation for all pixels in a larger window is unreasonable for long periods of time.

Q1 d) Generalized LK Template Tracking

- **Problem:** Assumption of pure translation for all pixels in a larger window is unreasonable for long periods of time.

- **Solution:** Allow arbitrary template transformation-model instead of only pure translation.

Only translation (u,v)

$$E(u,v) = \sum_{\mathbf{x}} \left[I(x+u, y+v) - T(x,y) \right]^2$$

$$\downarrow$$

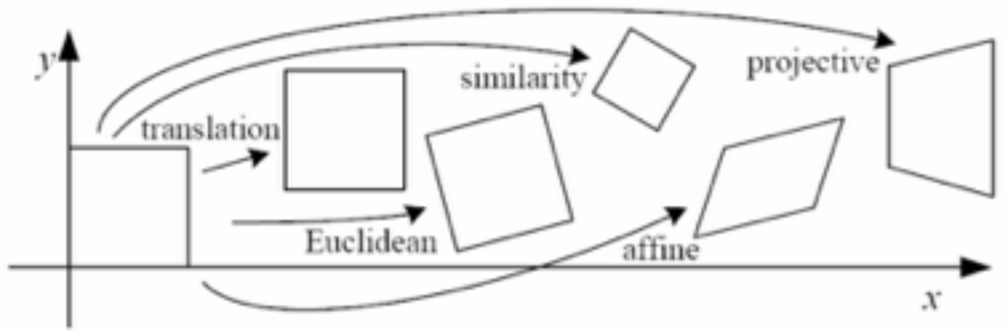
$$E(\mathbf{p}) = \sum_{\mathbf{x}} \left[I(\mathbf{W}([x,y];\mathbf{p})) - T([x,y]) \right]^2$$

General transformation

Q1 d) Generalized LK Template Tracking

$$E(\mathbf{p}) = \sum_{\mathbf{x}} \left[I(\mathbf{W}([x,y];\mathbf{p})) - T([x,y]) \right]^2$$

parameters p of transformation W are unknown



Q1 d) Generalized LK Template Tracking

- Here, motion model: Rotation Θ + Translation $\mathbf{t} = (t_x, t_y)$
  - 3 parameters

$$\mathbf{W}([x,y];\boldsymbol{\Theta},\mathbf{t_x},\mathbf{t_y}) = \begin{bmatrix} \cos\Theta & -\sin\Theta & \mathbf{t_x} \\ \sin\Theta & \cos\Theta & \mathbf{t_y} \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ 1 \end{bmatrix}$$

- Jacobian

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} 1 & 0 & -\mathrm{x}\sin\Theta - \mathrm{y}\cos\Theta \\ 0 & 1 & \mathrm{x}\cos\Theta - \mathrm{y}\sin\Theta \end{bmatrix}$$

Details: S. Baker, I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. In IJCV, Vol. 56(3), pp. 221-255, 2004. **“The Inverse Compositional Algorithm”**

Q1 d) Generalized LK Template Tracking

Q1 e) Generalized LK Template Tracking

Q1 e) Generalized LK Template Tracking

Q2: Kalman Filter Tracking

- **Constant Velocity Model**
  - State vector $\mathbf{x} = [p_x, p_y, v_x, v_y]'$

Visible in frame

Hidden, not visible in single frame

- **Dynamic model matrix D**
  - Predict next state $\mathbf{x}_{t+1}$ based on current state $\mathbf{x}_t$

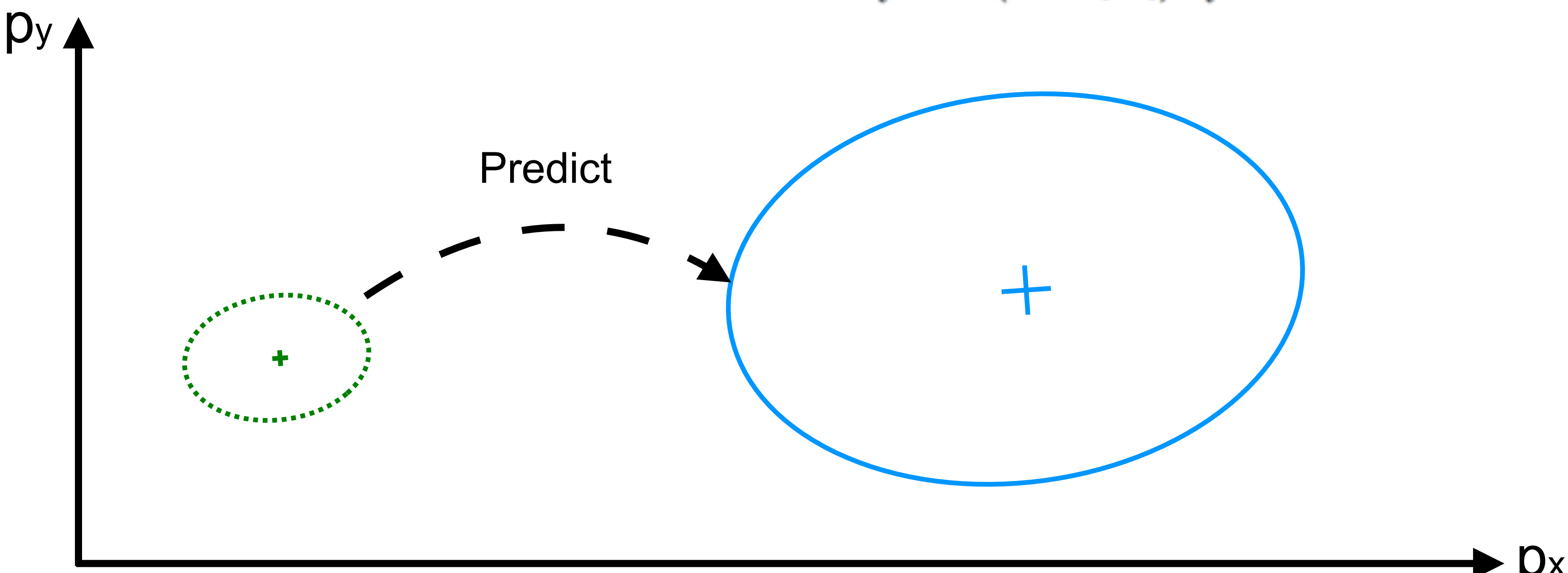
- **Measurement matrix M**
  - Extract the visible components from the state vector $\mathbf{x}_t$

Q2: Kalman Filter Tracking - Prediction

- **General Idea**: in each frame we have 2 steps
  - 1) **Prediction** 2) **Correction**

$$x_t^- = D_t x_{t-1}^+$$
$$\Sigma_t^- = D_t \Sigma_{t-1}^+ D_t^T + \Sigma_{d_t}$$

$$K_t = \Sigma_t^- M_t^T \left(M_t \Sigma_t^- M_t^T + \Sigma_{m_t}\right)^{-1}$$
$$x_t^+ = x_t^- + K_t \left(y_t - M_t x_t^-\right)$$
$$\Sigma_t^+ = \left(I - K_t M_t\right) \Sigma_t^-$$



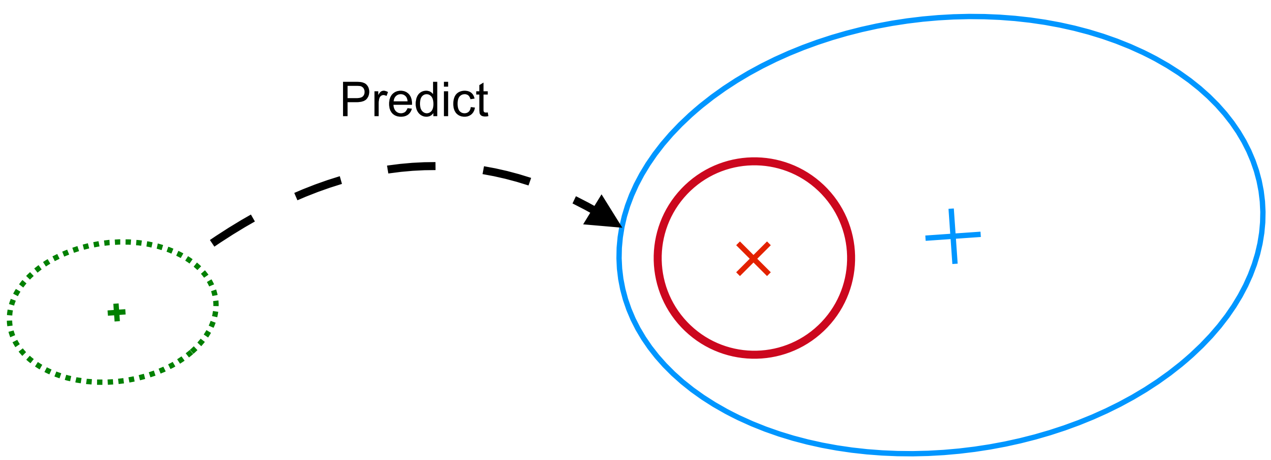
Q2: Kalman Filter Tracking - Measurement

- **General Idea**: in each frame we have 2 steps
  - 1) **Prediction**

$$\begin{aligned} x_t^- &= D_t x_{t-1}^+ \\ \Sigma_t^- &= D_t \Sigma_{t-1}^+ D_t^T + \Sigma_{d_t} \end{aligned}$$

2) **Correction**

$$\begin{aligned} K_t &= \Sigma_t^- M_t^T \left(M_t \Sigma_t^- M_t^T + \Sigma_{m_t} \right)^{-1} \\ x_t^+ &= x_t^- + K_t \left(y_t - M_t x_t^- \right) \\ \Sigma_t^+ &= \left(I - K_t M_t \right) \Sigma_t^- \end{aligned}$$

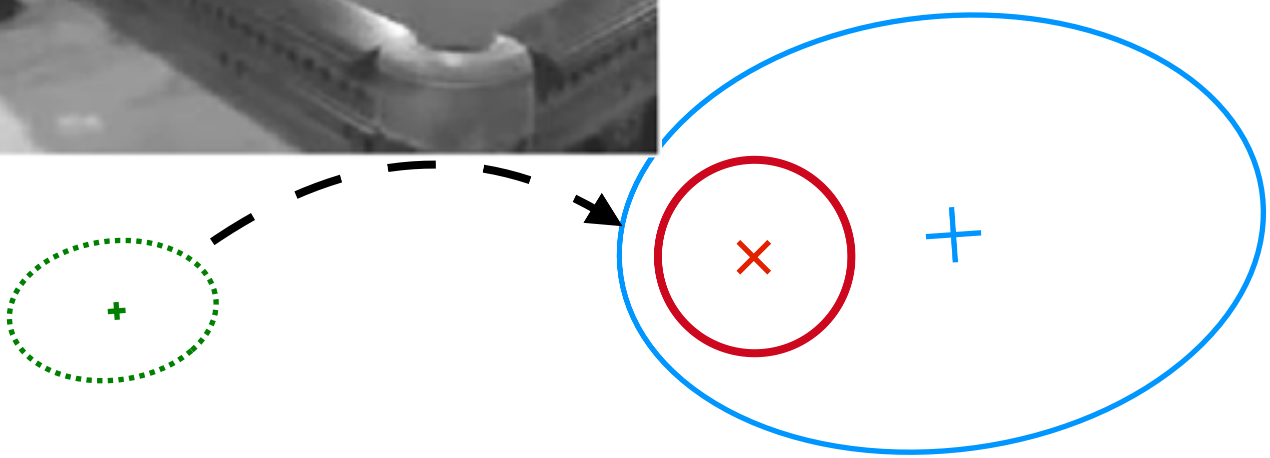


Q2: Kalman Filter Tracking - Measurement

ve 2 steps

2) **Correction**

$$K_t = \Sigma_t^- M_t^T \left(M_t \Sigma_t^- M_t^T + \Sigma_{m_t}\right)^{-1}$$
$$x_t^+ = x_t^- + K_t \left(y_t - M_t x_t^-\right)$$
$$\Sigma_t^+ = \left(I - K_t M_t\right) \Sigma_t^-$$

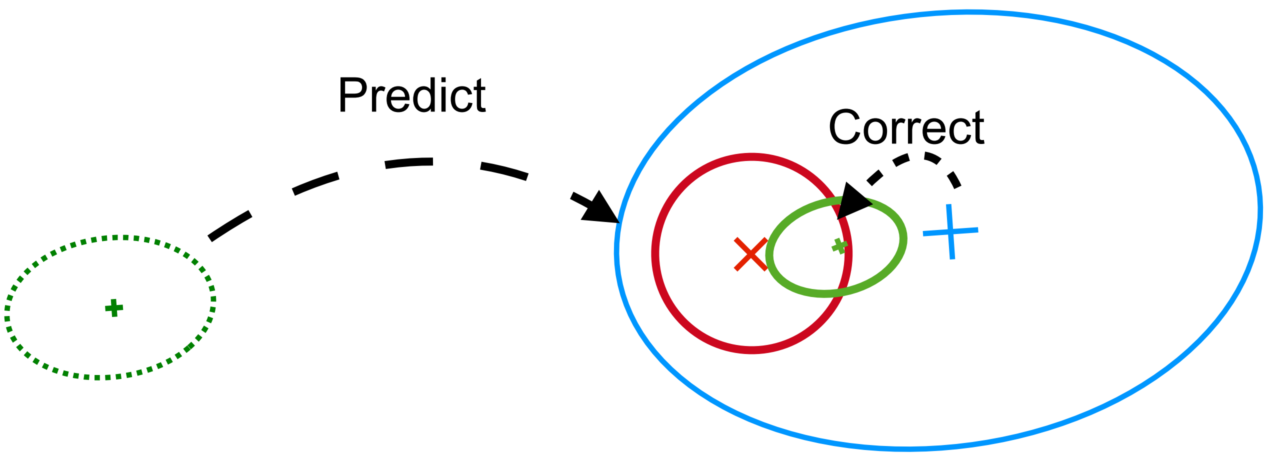


Q2: Kalman Filter Tracking - Correction

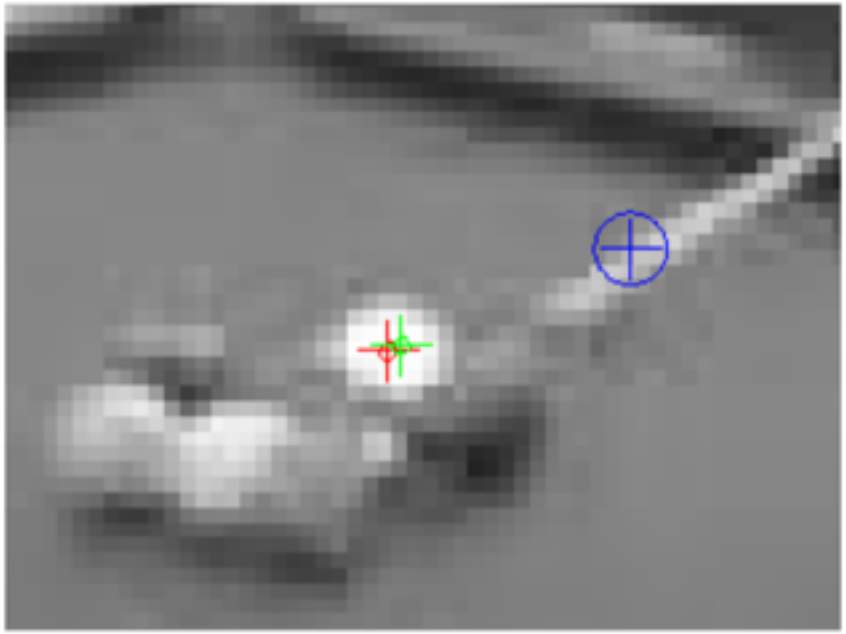
- **General Idea**: in each frame we have 2 steps
  - 1) **Prediction** 2) **Correction**

$$x_t^- = D_t x_{t-1}^+$$
$$\Sigma_t^- = D_t \Sigma_{t-1}^+ D_t^T + \Sigma_{d_t}$$

$$K_t = \Sigma_t^- M_t^T \left(M_t \Sigma_t^- M_t^T + \Sigma_{m_t} \right)^{-1}$$
$$x_t^+ = x_t^- + K_t \left(y_t - M_t x_t^- \right)$$
$$\Sigma_t^+ = \left(I - K_t M_t \right) \Sigma_t^-$$



Q2: Kalman Filter Tracking - Correction



ve 2 steps

2) **Correction**

$$
\begin{aligned}
K_t &= \Sigma_t^- M_t^T \left(M_t \Sigma_t^- M_t^T + \Sigma_{m_t} \right)^{-1} \\
x_t^+ &= x_t^- + K_t \left(y_t - M_t x_t^- \right) \\
\Sigma_t^+ &= \left(I - K_t M_t \right) \Sigma_t^-
\end{aligned}
$$