Announcement

We will have Quiz 3 in class, Monday, March 20th.

Final Project Presentation Schedule

Presentation days:

April 19 and 24

A team project needs a combined presentation.

Each individual presentation has 12 + 3 minutes Q&A
Each team presentation has 18 + 3 minutes Q&A

Send me an email (tongy@cse.sc.edu) that includes:

- Your name and the dates starting from the most preferred
- Earlier email has higher priority in choosing the date

Materials Covered in the Presentation

For a research/hands-on project

- An introduction of the background
- A brief literature review
- Methodology of your proposed method
- Experimental results if any
- Conclusion and future work

For a survey project

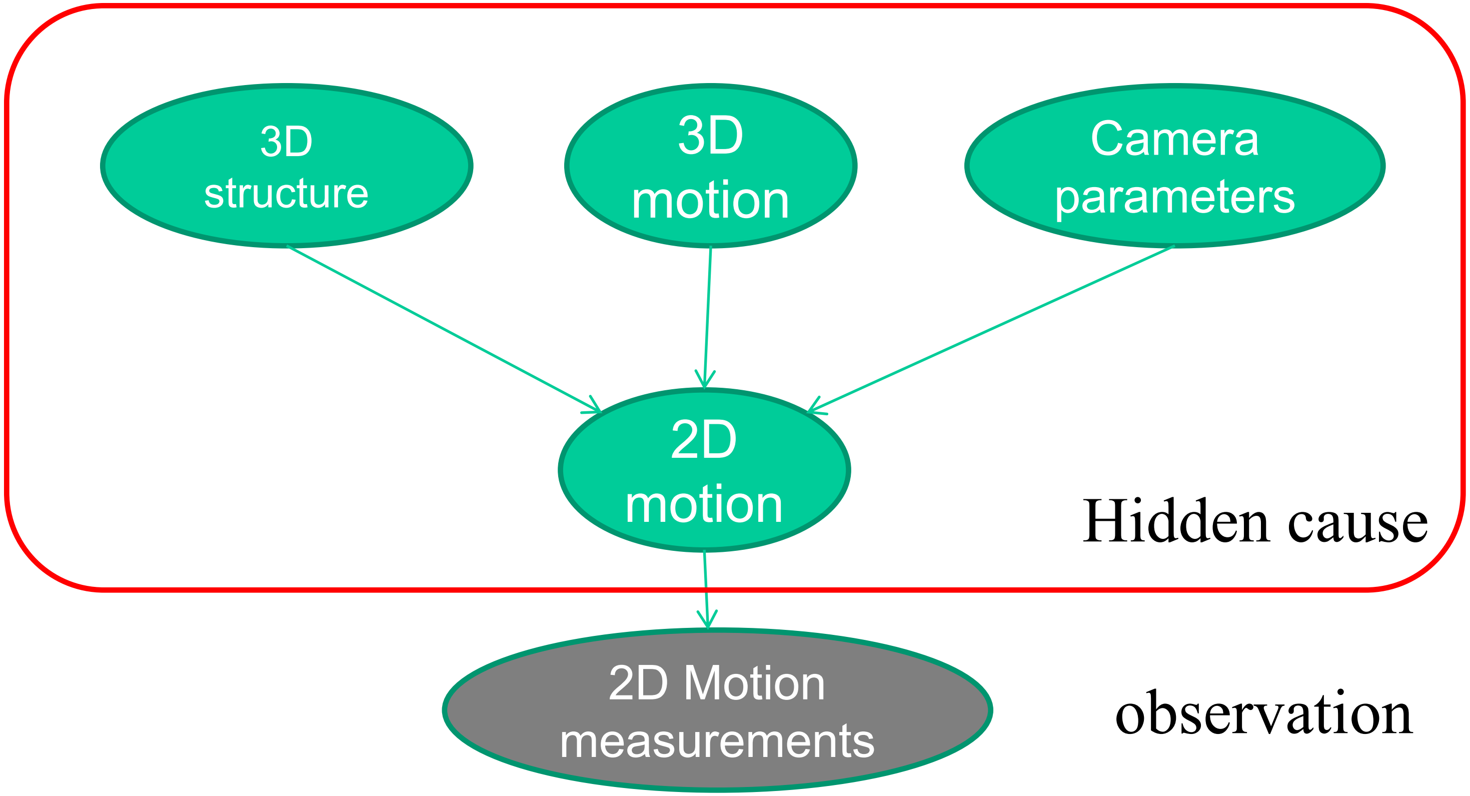
- An introduction of the background
- Discussion and critical comments on the papers you reviewed
- A comparison on the benchmark datasets is preferred
- Conclusion

Today

Early vision on multiple images

- Object tracking

Motion Modeling and Analysis



Tasks:

- Estimate 2D motion represented by ***motion field***
- Infer the 3D motion and/or 3D structure

Optical-Flow Equation

$$\nabla I \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} + I_t = 0$$

∇I: Gradient on the first/reference image

I_t: Difference across images

Lukas-Kanade Flow

Problem: we have more equations than unknowns

$$\underset{25\times2}{\mathbf{A}}\ \underset{2\times1}{\mathbf{v}} = \underset{25\times1}{\mathbf{b}} \longrightarrow \text{minimize} \quad \|\mathbf{Av}-\mathbf{b}\|^2$$

Solution: solve least squares problem

- minimum least squares solution given by solution of:

$$\underset{2\times2}{(A^T A)}\ \underset{2\times1}{\mathbf{v}} = \underset{2\times1}{A^T b}$$

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

- The summations are over all pixels in the *K* x *K* window
- **A^TA** should be invertible
- **A^TA** should not be too small due to noise
- **A^TA** should be well-conditioned

Summary of Motion Analysis by Optical Flow

Motion field:

- Measured on 2D images
- The projection of 3D velocity field on the image plane

Optical flow: the apparent motion of brightness patterns

- Optical flow $\neq$ Motion field
- Most of the time, optical flow corresponds to motion field
- Important assumptions/constraints
  - Brightness constancy
  - Small motion
  - Constant motion in a neighborhood

Dense – estimating motion for every pixel!

Object Tracking

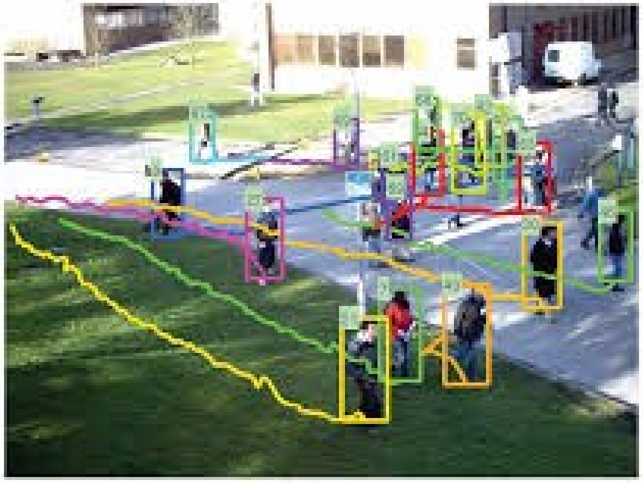
Tracking rigid object with rigid motion (scaling and translation)

- Global motion
- Examples: tracking vehicles, buildings, etc. from the same view angle

Tracking object with rigid/nonrigid motion

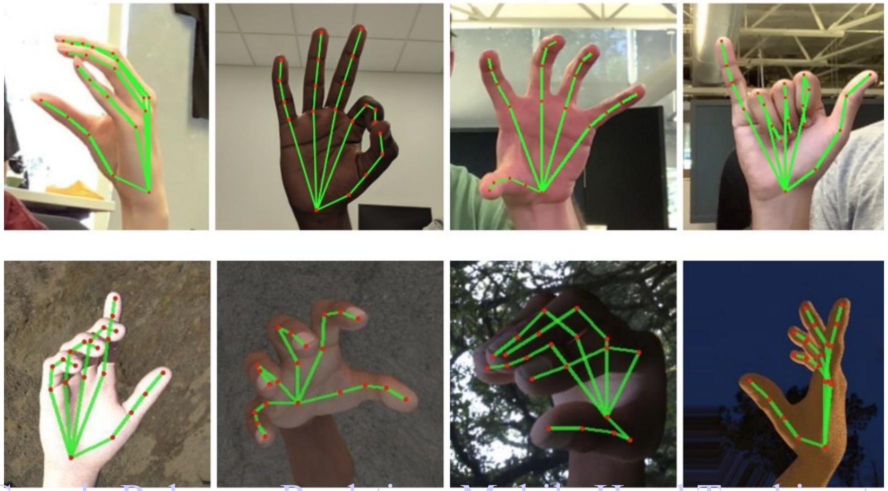
- Global motion + view change + local deformation
- Examples: tracking animals, persons, faces, etc.

Examples



xiang_iccv15.pdf (stanford.edu)

Object Tracking in Deep Learning – Deep Machine Learning AI



Google Releases Real-time Mobile Hand Tracking to R&D Community – Road to VR

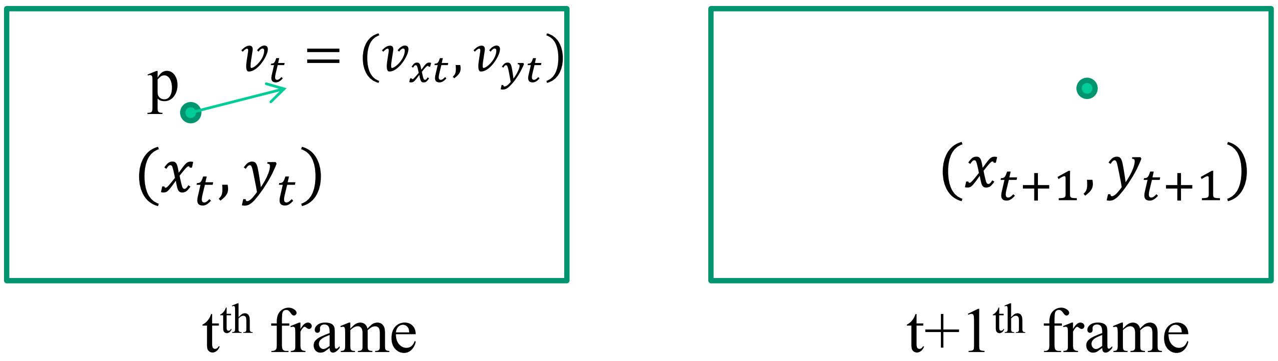
Facial Landmark Tracking | Facetrace | AlgoFace

Object Tracking – Motion Analysis from Multiple Frames

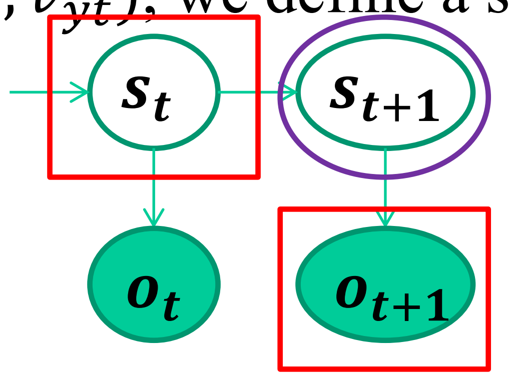
Tracking vs optical flow:

- Tracking is often applied for a few specified targets in the video
- Optical flow is applied for any points on the image – dense optical flow
- Both of them need to compute the correspondence between images

Object Tracking – Motion Analysis from Multiple Frames



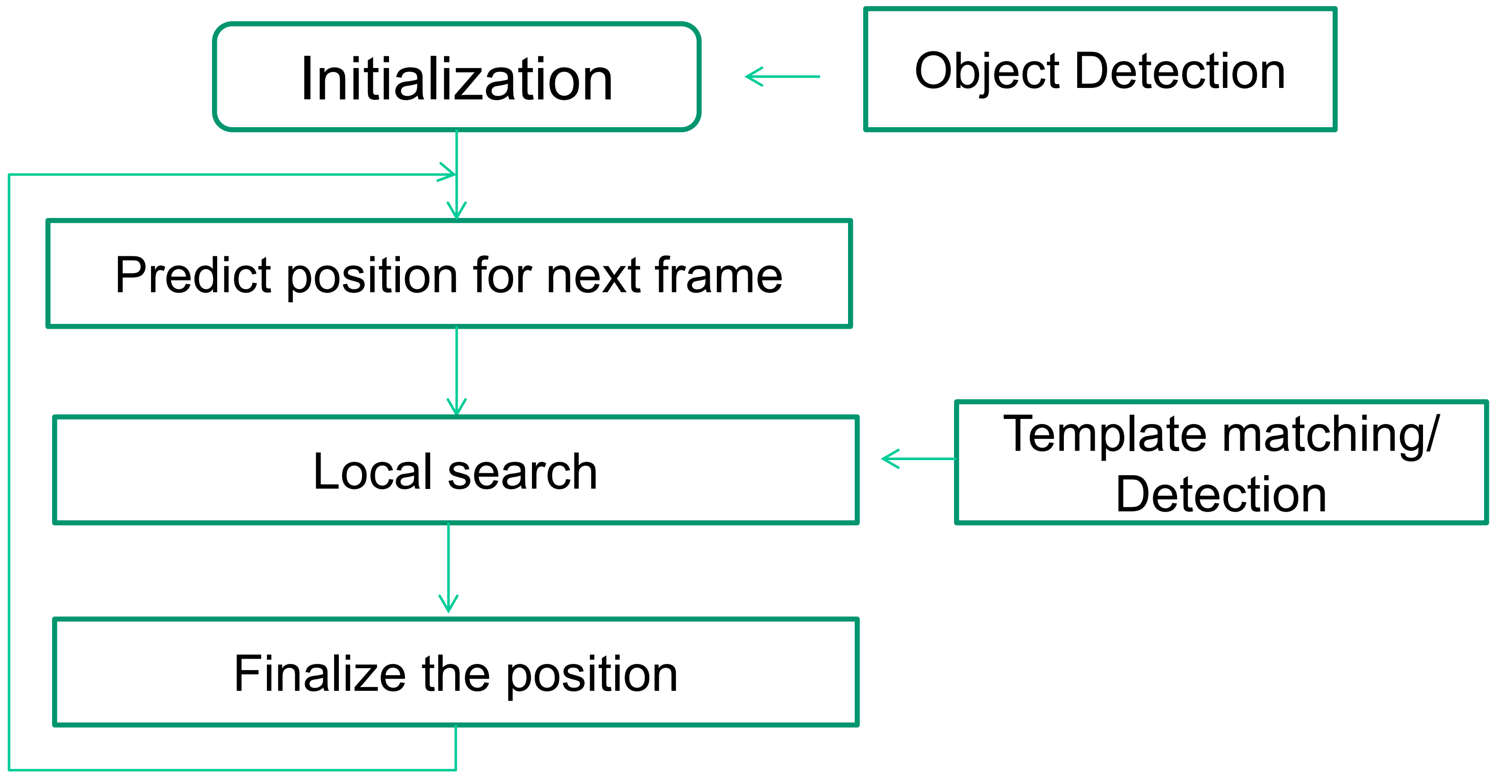
For a point $p = (x_t, y_t)$ at the t^{th} time frame with a velocity of $v_t = (v_{xt}, v_{yt})$, we define a state vector $\mathbf{s}_t = [x_t, y_t, v_{xt}, v_{yt}]^T$



Question:

Given $\mathbf{s}_t$ and $\mathbf{o}_{t+1} = (x_{t+1}, y_{t+1})$, how to estimate $\mathbf{s}_{t+1}$?

General Strategy for Object Tracking



A well-known algorithm following the strategy -- Kalman filter

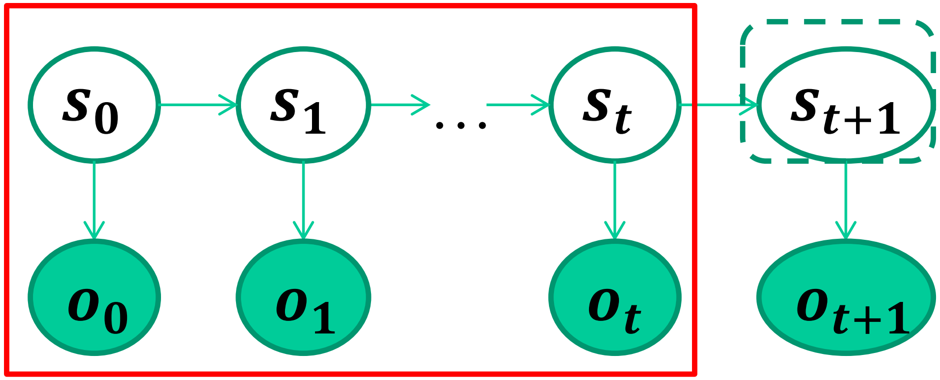
Tracking – General Probabilistic Formulation

Given

- $P(\mathbf{s}_t|\mathbf{o}_0,\cdots,\mathbf{o}_t)$ - "Prior"

We should like to know

- $P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)$ - "Predictive distribution"
- $P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t,\mathbf{o}_{t+1})$ – "Posterior"



How to compute them?

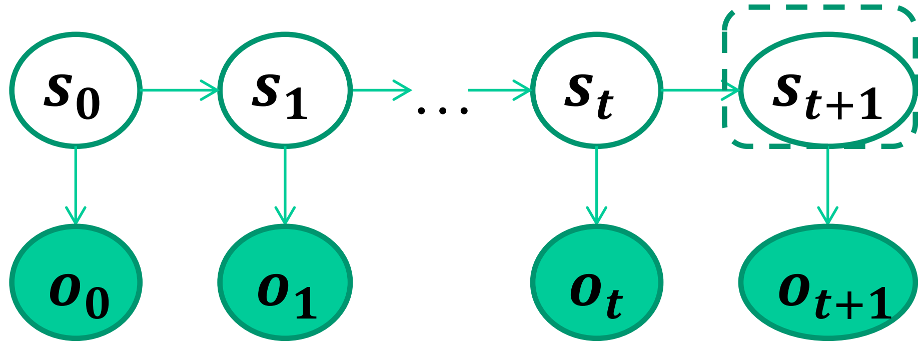
$$P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)=\sum_{\mathbf{s}_t}P(\mathbf{s}_{t+1},\mathbf{s}_t|\mathbf{o}_0,\cdots,\mathbf{o}_t)$$

$$=\sum_{\mathbf{s}_t}P(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{o}_0,\cdots,\mathbf{o}_t)*P(\mathbf{s}_t|\mathbf{o}_0,\cdots,\mathbf{o}_t)$$

Tracking – General Probabilistic Formulation

Given

- $P(\mathbf{s}_t|\mathbf{o}_0,\cdots,\mathbf{o}_t)$ - "Prior"



We should like to know

- $P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)$ - "Predictive distribution"
- $P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t,\mathbf{o}_{t+1})$ – "Posterior"

How to compute them?

$$P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t,\mathbf{o}_{t+1}) = \frac{P(\mathbf{o}_{t+1}|\mathbf{s}_{t+1},\mathbf{o}_0,\cdots,\mathbf{o}_t) * P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)}{\sum_{\mathbf{s}_{t+1}} P(\mathbf{o}_{t+1}|\mathbf{s}_{t+1},\mathbf{o}_0,\cdots,\mathbf{o}_t) * P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)}$$

Tracking – General Assumptions

- **First-order markov**
  - The current state only depends on the state of the previous time step
  - $P(\mathbf{s}_t|\mathbf{s}_0,\cdots,\mathbf{s}_{t-1}) = P(\mathbf{s}_t|\mathbf{s}_{t-1})$
- **Given the current state, the measurement at current time step is independent of the previous measurements**
  - $P(\mathbf{o}_t|\boldsymbol{s}_t,\mathbf{o}_0,\cdots,\mathbf{o}_{t-1}) = P(\mathbf{o}_t|\mathbf{s}_t)$

Tracking – General Probabilistic Formulation

Prediction: $$P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t) = \sum_{\mathbf{s}_t} P(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{o}_0,\cdots,\mathbf{o}_t) * P(\mathbf{s}_t|\mathbf{o}_0,\cdots,\mathbf{o}_t)$$

$$= \sum_{\mathbf{s}_t} \boxed{P(\mathbf{s}_{t+1}|\mathbf{s}_t)} * P(\mathbf{s}_t|\mathbf{o}_0,\cdots,\mathbf{o}_t)$$

System model

Update:

$$P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t,\mathbf{o}_{t+1}) = \frac{P(\mathbf{o}_{t+1}|\mathbf{s}_{t+1},\mathbf{o}_0,\cdots,\mathbf{o}_t) * P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)}{\sum_{\mathbf{s}_{t+1}} P(\mathbf{o}_{t+1}|\mathbf{s}_{t+1},\mathbf{o}_0,\cdots,\mathbf{o}_t) * P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)}$$

$$= \frac{\boxed{P(\mathbf{o}_{t+1}|\mathbf{s}_{t+1})} * P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)}{\sum_{\mathbf{s}_{t+1}} P(\mathbf{o}_{t+1}|\mathbf{s}_{t+1}) * P(\mathbf{s}_{t+1}|\mathbf{o}_0,\cdots,\mathbf{o}_t)}$$

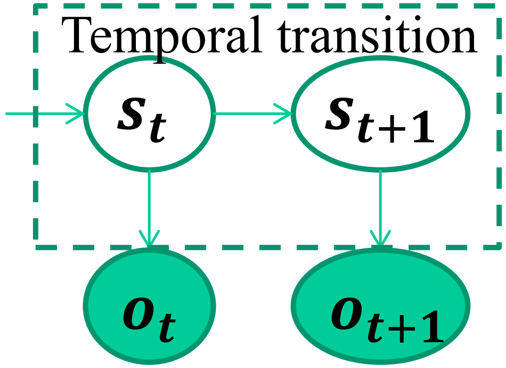
Measurement model

Kalman Filtering

Assumptions

- **Linear state model**
  - The next state s_{t+1} is linearly related to the current state s_t
- **Uncertainty satisfy a Gaussian**

Linear State Model



$$\mathbf{s}_{t+1} = \Phi \mathbf{s}_t + \mathbf{w}_t$$

Φ: state transition matrix

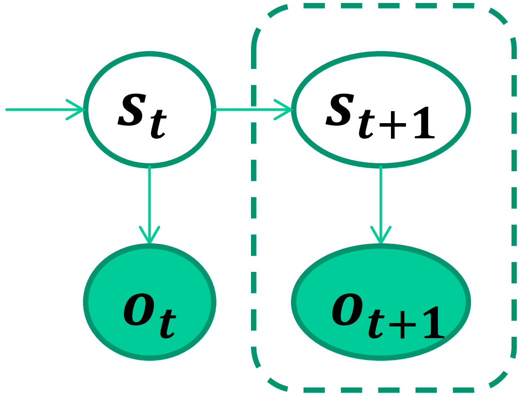
$\mathbf{w}_t$: system perturbation satisfying a Gaussian distribution $\mathbf{w}_t \sim N(0, \boldsymbol{W})$

If motion between two frames is small,

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Measurement Model

$$\mathbf{o}_t = \mathbf{H}\mathbf{s}_t + \mathbf{r}_t$$

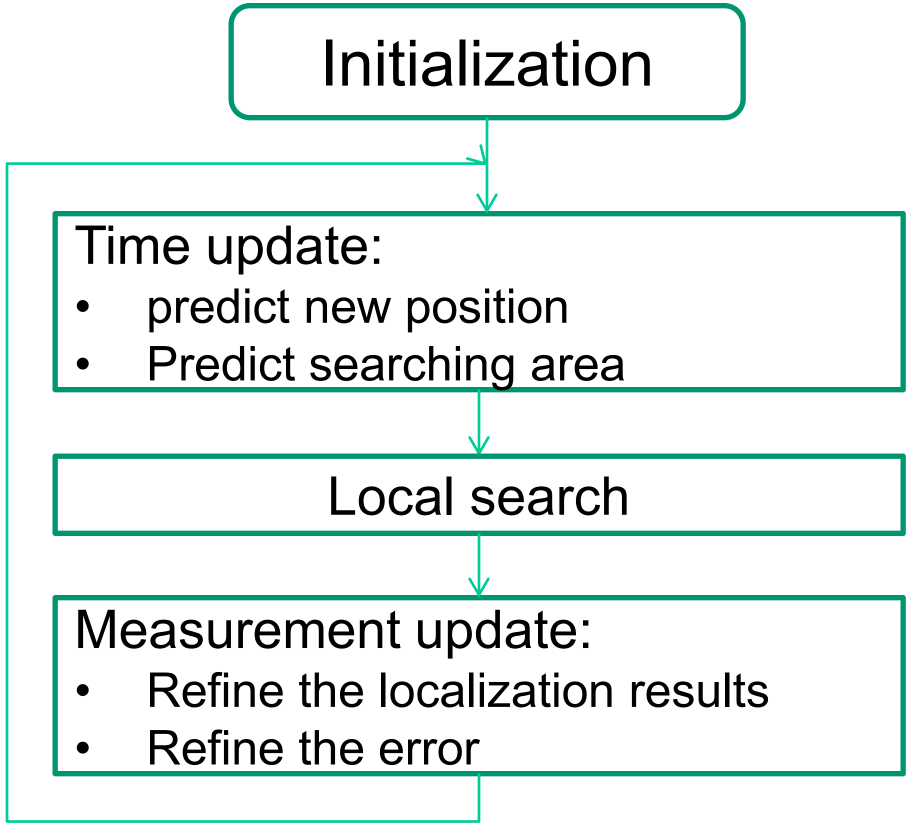


- **H** describes how the measurement relates to the state vector
- **For example,** $\mathbf{o}_t = (x_t, y_t)$ is the measured feature position

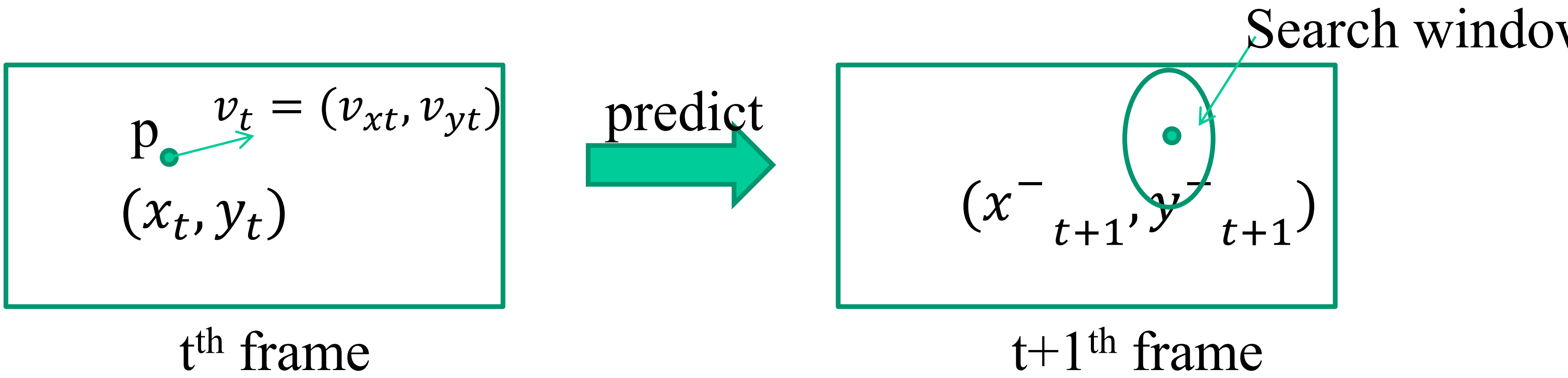
  The simplest case $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

- $\mathbf{r}_t$ is the measurement uncertainty satisfying a Gaussian distribution $\mathbf{r}_t \sim N(0, \boldsymbol{R})$

Kalman Filtering



Kalman Filtering – Prediction



State prediction: $\mathbf{s}^-_{t+1} = \Phi \mathbf{s}_t$

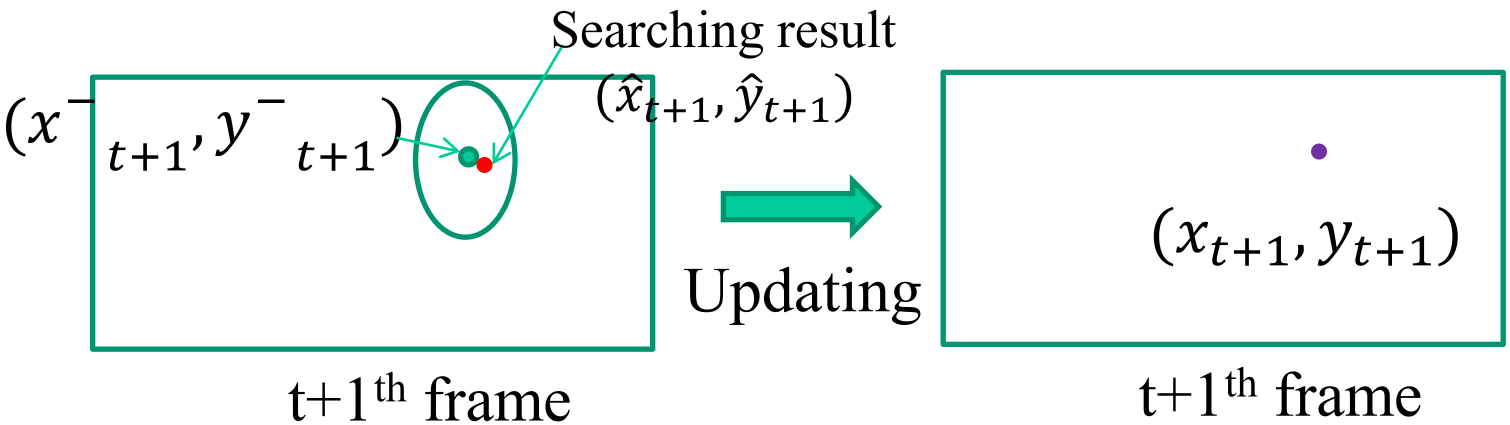
Confidence of the prediction: $\mathbf{\Sigma}^-_{t+1} = \Phi \mathbf{\Sigma}_t \Phi^T + W$

↓

Determine the size of search window

$3\lambda_x \times 3\lambda_y$

Kalman Filtering – Updating

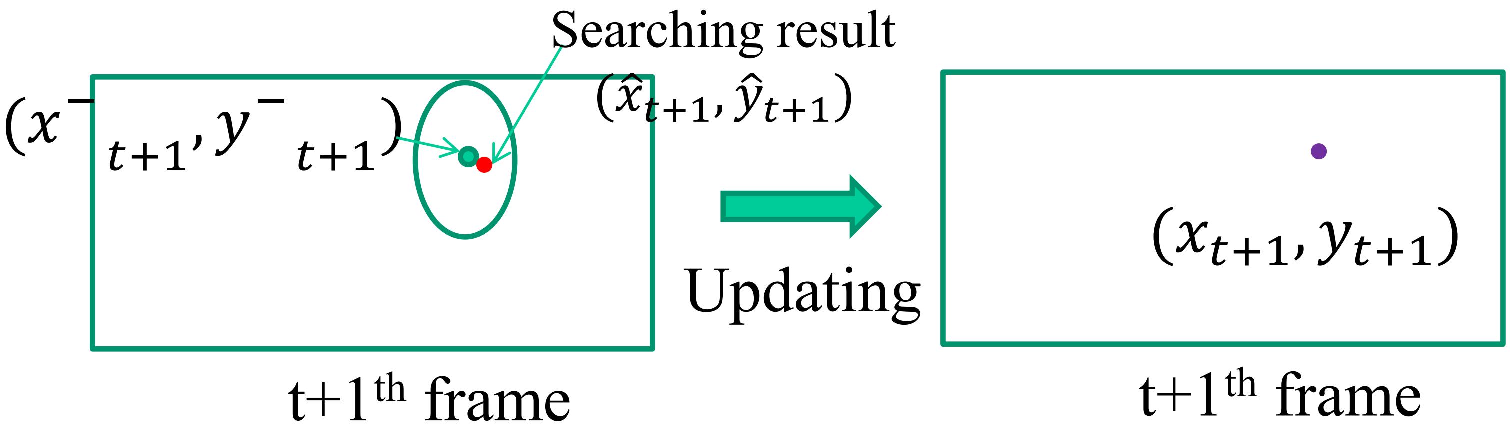


Kalman gain: $K_{t+1} = \sum_{t+1}^{-} H^T \left(H \sum_{t+1}^{-} H^T + R\right)^{-1}$

A weighting factor determines

- the contribution of the measurement and the prediction in the final estimation -- which one is more reliable

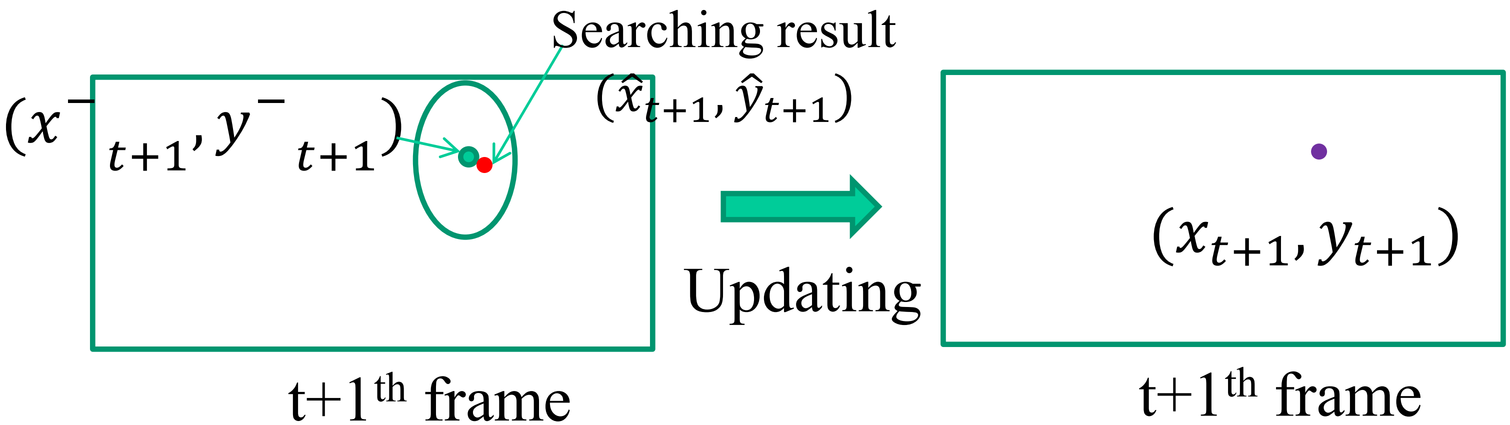
Kalman Filtering – Updating



State update: $\mathbf{s}_{t+1} = \mathbf{s}^-_{t+1} + K_{t+1}(\mathbf{o}_{t+1} - \mathbf{H}\mathbf{s}^-_{t+1})$

Measurement residual: difference between the measurement and the prediction

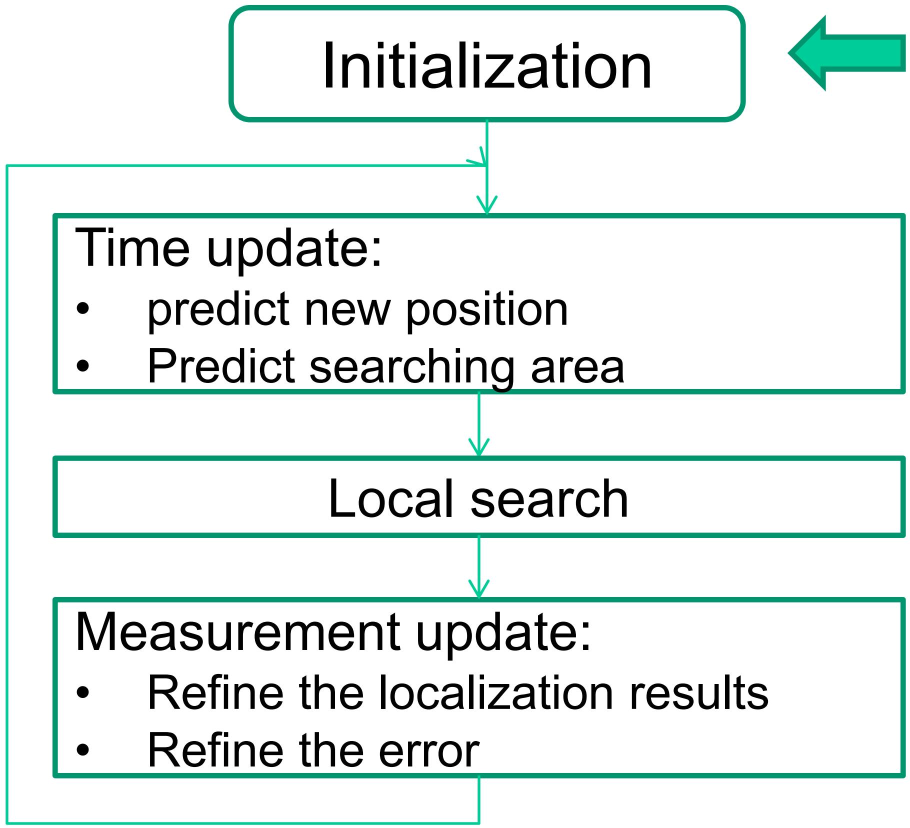
Kalman Filtering – Updating



State update: $\mathbf{s}_{t+1} = \mathbf{s}^-_{t+1} + K_{t+1}(\mathbf{o}_{t+1} - \mathbf{H}\mathbf{s}^-_{t+1})$

Error covariance update: $\Sigma_{t+1} = (I - K_{t+1}H)\,\Sigma^-_{t+1}$

Kalman Filtering



$\mathbf{s}_0$

$$W = \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

$$R = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 36 & 0 \\ 0 & 0 & 0 & 36 \end{bmatrix}$$

Limitation of Kalman Filtering

Failed cases:

- **State transition is not linear**
  - Sudden motion direction/velocity changes

→ Extended Kalman filtering

- **Uncertainty is not Gaussian**

→ Unscented Kalman filtering

Reading Assignments

An Introduction to the Kalman Filter by Welch and Bishop

http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf