OPTIMAL SENSOR SELECTION FOR VIDEO-BASED TARGET TRACKING IN A WIRELESS SENSOR NETWORK

ICIP 2004
Presented in class by
Arslan Basharat (arslan@cs.ucf.edu)

Goal

- Target tracking in a network of multiple video cameras (sensors)
- Choose the sensors such that the information utility is maximized
- Average energy used is constrained
Approach

- Use multiple video cameras
- Overlapping / Non-overlapping Cameras
- Cameras are calibrated
- Active cameras detect target
- Tracker estimates position in 3D world
- Use UKF to obtain confidence in prediction
- Confidence measure as sensor utility measure

Approach

- Information Utility is maximized
- Set of sensor that maximize the sum information utility are enabled
Main Steps

- Camera Calibration
- Unscented Kalman Filter (UKF)
- Sensor utility measure
- Maximize Information Utility

Outline

- Camera Calibration
- Filtering Stage
  - Bayesian Formulation
  - Kalman Filter
  - Extended Kalman Filter
  - Unscented Transformation
  - Unscented Kalman Filter
- Sensor Utility Measure
- Information utility maximization
- Results
Camera Calibration

\[
\begin{bmatrix}
\frac{u_i S}{v_i S} \\
\frac{v_i S}{S}
\end{bmatrix} = Pr_i \begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
\]

- Projective camera model \( Pr_i = A_i [R_i | t_i] \)
- Intrinsic parameters
  - \( A \) Matrix (f, aspect ratio, principal point and skew)
- Extrinsic parameters
  - \( R \) rotation Matrix
  - \( t \) translation vector

Useful Filters for Tracking

- Bayesian Approach
  - Recursive probabilistic model (predict & update)
- Kalman Filter
  - Approximation of Bayesian Model
  - Different variants for linear & non-linear cases
- Particle Filter
  - Monte-Carlo Approach
- Joint Probabilistic Data Association Filter
- Multiple Hypothesis Tracker
Bayesian Tracking Approach

- **Recursive Model**
  - Linear case
  - Non-Linear case

State Vector: \( \mathbf{x} = [x, v_x, y, v_y, z, v_z]^T \)

Observation vector: \( \mathbf{z}_i = [u_i, v_i]^T \)

Observations up to time \( k \): \( \mathbf{Z}^k \)

Prior distribution (predict stage):
\[
p(x(k) | Z^{k-1}) = \int p(x(k) | x(k-1))p(x(k-1) | Z^{k-1})dx(k-1)
\]

Posteriori distribution (update stage):
\[
p(x(k) | Z^k) = \frac{p(z(k) | x(k))p(x(k) | Z^{k-1})}{p(z(k) | Z^{k-1})}
\]

Normalizing factor:
\[
p(z(k) | Z^{k-1}) = \int p(z(k) | x(k))p(x(k) | Z^{k-1})dx(k)
\]
Kalman Filter

- Kalman filter can be used to approximate optimal linear Bayesian solution.
- Kalman filter is recursive
  - Predict (Time Update)
  - Correct (Measurement Update)

State vector: \( \mathbf{x} = [x, x_v, y, y_v, z, z_v]^T \)

Measurement vector: \( \mathbf{z_i} = [u_i, v_i]^T \)

State update equation (A Linear Stochastic Difference Equation)

\[
x(k) = A x(k-1) + w(k-1)
\]

Measurement Equation

\[
z(k) = H x(k) + v(k)
\]

where,

- random variables \( w(k-1) \) and \( v(k) \) represent process and measurement noise with zero mean Normal distribution
  - \( p(w) \sim N(0, Q) \),
  - \( p(v) \sim N(0, R) \).

\( H \) relates the state to the measurement and \( A \) relates the state at time \( k \) and \( k-1 \).
Kalman Filter - Example

- Multi-frame feature tracking
- For each feature Kalman Filter can estimate
  - Position
  - Confidence
- State Vector
  - $x(k) = [x_k, y_k, v_x, v_y]^T$
- Measurement/Observation
  - $z(k) = [x_k, y_k]$

Kalman Filter - Example

- State update equation
  - $x_k = x_{k-1} + v_{x,k} + \alpha_{k-1}$
  - $v_k = v_{k-1} + \beta_{k-1}$
  - $x(k) = Ax(k-1) + w(k-1)$

- Measurement Equation
  - $z(k) = Hx(k) + v(k)$

---

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

$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
Kalman Filter - Example

Kalman Filter

Time Update ("Predict")

1. Project the state ahead
   \[ \hat{x}_k = A \hat{x}_{k-1} + Bu_{k-1} \]
2. Project the error covariance ahead
   \[ P_k = AP_{k-1}A^T + Q \]

Measurement Update ("Correct")

1. Compute the Kalman gain
   \[ K_k = P_kH^T(HP_kH^T + R)^{-1} \]
2. Update estimate with measurement \( z_k \)
   \[ \hat{x}_k = \hat{x}_k + K_k(z_k - H\hat{x}_k) \]
3. Update the error covariance
   \[ P_k = (I - K_kH)P_k \]

Initial estimates for \( \hat{x}_{k-1} \) and \( P_{k-1} \)

\[ e_k' = x_k' - \hat{x}_k' \]
\[ e_k = x_k - \hat{x}_k \]
\[ P_k' = E[e_k'e_k'^T] \]
\[ P_k = E[e_k'e_k^T] \]
Problem with Kalman Filter

- Linear case (Kalman Filter)
  - State update equation
    \[ x(k) = Ax(k-1) + w(k-1) \]
  - Measurement Equation
    \[ z(k) = Hx(k) + v(k) \]
- Non-Linear case (Extended Kalman Filter)
  \[ x(k) = f(x(k-1), w(k-1)) \]
  \[ z(k) = h(x(k), v(k)) \]
  where, \( f \) and \( h \) are non-linear functions

Extended Kalman Filter (EKF)

- Linearizes about current mean and covariance
- Using first order terms from Taylor series expansion of non-linear functions
- Linearize around current estimate through partial derivates of
  - State update function
  - Measurement function
**EKF**

\[
\begin{align*}
    x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}) \\
    z_k &= h(x_k, v_k)
\end{align*}
\]

Approximation of original model \((w, v \text{ unavailable})\)

\[
\begin{align*}
    \tilde{x}_k &= f(\tilde{x}_{k-1}, u_{k-1}, 0) \\
    \tilde{z}_k &= h(\tilde{x}_k, 0)
\end{align*}
\]

Linearizing through Taylor series expansion

\[
\begin{align*}
    x_k &= \tilde{x}_k + A(x_{k-1} - \tilde{x}_{k-1}) + Ww_{k-1} \\
    z_k &= \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k.
\end{align*}
\]

A, W, H, V are Jacobians and \(u_{k-1}\) is known input

**Problems with EKF**

- Only first order terms from Taylor expansion
- Probability densities of various random variables are no longer Normal after nonlinear transformation
- EKF is an ad hoc state estimator
- Use Unscented Kalman Filter (UKF)
Unscented Kalman Filter (UKF)

- A minimal set of carefully chosen points (sigma vectors)
- Captured accurately to the 3\textsuperscript{rd} order of Taylor expansion
- Based on Unscented Transformation

Unscented Transformation (UT)

- A method for calculating the statistics of a random variable which undergoes a nonlinear transformation
  - Random Variable $x$ (dimension $L$, mean $\hat{x}$, covariance $P_x$)
  - $y = g(x)$ where $g$ is non-linear transformation function
- For statistics of $y$
  - Form matrix $X$ of $2L+1$ sigma vectors $X_i$ (with corresponding weights $W_i$)
Unscented Transformation (UT)

\[
X_0 = x
\]
\[
X_i = x + \sqrt{(\beta + \alpha^2 - 1)} \Lambda \sqrt{P}
\]
\[
X_{2i-1} = x - \sqrt{(\beta + \alpha^2 - 1)} \Lambda \sqrt{P}
\]

Scaling Parameter \( \beta = \alpha^2 + \gamma - 1 \)

\( \alpha \) determines the spread of the sigma points around \( x \)

\[
\gamma_i = \sigma_i(x) \quad i = 1, \ldots, 2\gamma
\]

\[
\sum_{i=0}^{2\gamma} w_{i}^D \gamma_i = \frac{1}{\gamma}
\]
\[
\sum_{i=0}^{2\gamma} w_{i}^C (x - \gamma_i) (x - \gamma_i)^T = \frac{1}{\gamma}
\]

Example

Gaussian prior is propagated through an arbitrary highly nonlinear function. Monte-Carlo sampling, Extended Kalman Filter (EKF) and Unscented Transformation (UT) results are shown in the figure above.
Unscented Kalman Filter (UKF)

- UKF is extension of UT to the recursive Kalman Filter approach.
- State random variable is defined with augmented state vector and augmented covariance matrix

\[
\begin{bmatrix}
    x^a_k \\
    v_k \\
    u_k
\end{bmatrix} = \begin{bmatrix}
    x_k^T \\
    v_k^T \\
    u_k^T
\end{bmatrix}
\]

\[
P^a = \begin{bmatrix}
P_x & 0 & 0 \\
0 & P_v & 0 \\
0 & 0 & P_u
\end{bmatrix}
\]

UKF Algorithm

Initialize with:

\[
\begin{align*}
    \hat{x}_0 &= E[x_0] \\
    P_0 &= E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \\
    \hat{x}_0^a &= E[p^a] = [\hat{x}_0^a \ 0 \ 0]^T \\
    P_0^a &= E[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix}
P_0 & 0 & 0 \\
0 & P_v & 0 \\
0 & 0 & P_u
\end{bmatrix}
\end{align*}
\]
UKF Algorithm

For $k \in \{1, \ldots, \infty\}$,

- Calculate sigma points:
  \[ \mathbf{x}_k^{\pm} = \mathbf{x}_{k-1}^* \pm \sqrt{(L + \lambda)P_k^{-1}} \]

- Time update:
  \[
  \begin{align*}
  \mathbf{x}_{k|k-1} &= \mathbf{F} \mathbf{x}_{k-1|k-1} \\
  \mathbf{s}_k &= \sum_{i=0}^{2L} \mathbf{w}_i \mathbf{x}_{k|k-1}^{\pm} \\
  \mathbf{P}_k &= \sum_{i=0}^{2L} \mathbf{w}_i \left( \mathbf{x}_{k|k-1}^{\pm} - \mathbf{s}_k \right) \left( \mathbf{x}_{k|k-1}^{\pm} - \mathbf{s}_k \right)^T \\
  \mathbf{y}_{k|k-1} &= \mathbf{H} \mathbf{x}_{k|k-1} \\
  \mathbf{\hat{y}}_k &= \sum_{i=0}^{2L} \mathbf{w}_i \mathbf{y}_{k|k-1}^{\pm} \\
  \lambda &= \text{composite scaling parameter}, \quad L = \text{dimension of augmented state vector}, \quad P_k = \text{process noise covariance}, \quad P_n = \text{measurement noise covariance}
  
- Measurement update equations:
  \[
  \begin{align*}
  \mathbf{P}_{k|k} &= \sum_{i=0}^{2L} \mathbf{w}_i \left[ \mathbf{H} \mathbf{x}_{k|k-1}^{\pm} - \mathbf{y}_{k|k-1} \right] \left[ \mathbf{H} \mathbf{x}_{k|k-1}^{\pm} - \mathbf{y}_{k|k-1} \right]^T \\
  \mathbf{P}_{m} &= \sum_{i=0}^{2L} \mathbf{w}_i \left[ \mathbf{h}(m) \mathbf{x}_{k|k-1}^{\pm/m} - \mathbf{y}_{k|k-1} \right] \left[ \mathbf{h}(m) \mathbf{x}_{k|k-1}^{\pm/m} - \mathbf{y}_{k|k-1} \right]^T \\
  \mathbf{K} &= \mathbf{P}_{m} \mathbf{P}_{x|k}^{-1} \\
  \mathbf{s}_k &= \mathbf{s}_k + \mathbf{K} \left( \mathbf{z}_k - \mathbf{y}_k \right) \\
  \mathbf{P}_k &= \mathbf{P}_k - \mathbf{K} \mathbf{P}_{m} \mathbf{K}^T
  
Sensor Utility Measure

\[
\psi_c(k + 1) = -\text{trace}[\mathbf{P}_c(k + 1)]
\]

- $\mathbf{P}_c(k+1)$ is the covariance matrix for error
- Larger the error small the sensor utility
Information Utility (IU)

- Sensor state $\pi_i[k]$
  - Off/Initializing/On
- Energy used by sensor $i$: $E_i(\pi_i[k])$
- Information Utility
  - $U(\pi[k], k) \Rightarrow \psi_c(k)$

\[
\begin{align*}
\min_{\pi_N} & \quad \sum_{k=1}^{K} -U(\pi[k], k) \\
\text{s.t.} & \quad \sum_{k=1}^{K} \sum_{i \in N} E_i(\pi_i[k]) \leq K \cdot E_{ave}
\end{align*}
\]

Information Utility (IU)

- $W$ future time window
- Sensors in range $R$
- $\hat{U}$ is the estimated IU for future

\[
\begin{align*}
\min_{\pi_R[k]} & \quad -U(\pi_R[k], k) - \sum_{j=k+1}^{W} \hat{U}(\pi_R[j], j) \\
\text{s.t.} & \quad \sum_{j=k-W+1}^{k} \sum_{i \in N} E_i(\pi_i[j]) \leq W \cdot E_{ave} \\
& \quad \sum_{i \in N} E_i(\pi_i[j]) \leq E_{ph}
\end{align*}
\]

- IU maximization problem solved using tree pruning algorithm
Optimal Selection is turning on predicted suitable sensors in advance.

<table>
<thead>
<tr>
<th></th>
<th>Optimal Selection</th>
<th>Random Sensor Selection</th>
<th>Closest Sensor Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ave. E</td>
<td>4.93</td>
<td>2.41</td>
<td>4.41</td>
</tr>
<tr>
<td>rmse</td>
<td>0.40</td>
<td>0.52</td>
<td>0.52</td>
</tr>
<tr>
<td>% lost tracks</td>
<td>6.3</td>
<td>58.0</td>
<td>11.3</td>
</tr>
</tbody>
</table>

rmse: Root Mean Square Error
Issues

- Calibration of all cameras
- Multiple target tracking
- Goodness of avg. energy constraint on ‘information utility’
  - $E_i$ should incorporate local usage history
- Selection of IU maximization algorithm
- Processing Node energy consumption NOT considered