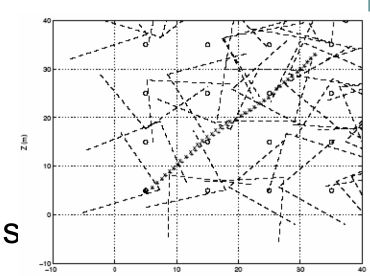
OPTIMAL SENSOR SELECTION FOR VIDEO-BASED TARGET TRACKING IN A WIRELESS SENSOR NETWORK

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

$$\begin{bmatrix} u_i S \\ v_i S \\ S \end{bmatrix} = Pr_i \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- 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: $\mathbf{x} = [x, x_v, y, y_v, z, z_v]^T$

Observation vector: $\mathbf{z_i} = [u_i, v_i]^T$
(for sensor i)

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

Bayesian Tracking Approach

Prior distribution (predict stage):

$$p(x(k) \mid Z^{k-1}) = \int p(x(k) \mid x(k-1))\, p(x(k-1) \mid Z^{k-1})\, dx(k-1)$$

$p(\mathbf{x_0}|\mathbf{z_0}) \equiv p(\mathbf{x_0})$

Posteriori distribution (update stage):

$$p(\mathbf{x}(k)|\mathbf{Z^k}) = \frac{p(\mathbf{z}(k)|\mathbf{x}(k))p(\mathbf{x}(k)|\mathbf{Z^{k-1}})}{p(\mathbf{z}(k)|\mathbf{Z^{k-1}})}$$

Normalizing factor:

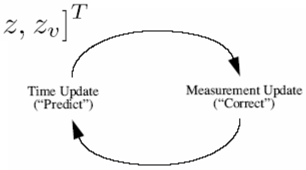
$$p(z(k) \mid Z^{k-1}) = \int \underbrace{p(z(k) \mid x(k))}_{\text{Likelihood Function}}\, p(x(k) \mid Z^{k-1})\, dx(k)$$

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: $\mathbf{z_i} = [u_i, v_i]^T$



Kalman Filter

- State update equation
  (A Linear Stochastic Difference Equation)

$$x(k) = Ax(k-1) + w(k-1)$$

- Measurement Equation

$$z(k) = Hx(k) + v(k)$$

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
  - $x_k = x_{k-1} + v_{x,k} + \alpha_{k-1}$
  - $v_k = v_{k-1} + \beta_{k-1}$

Noise

$$x(k) = Ax(k-1) + w(k-1)$$

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

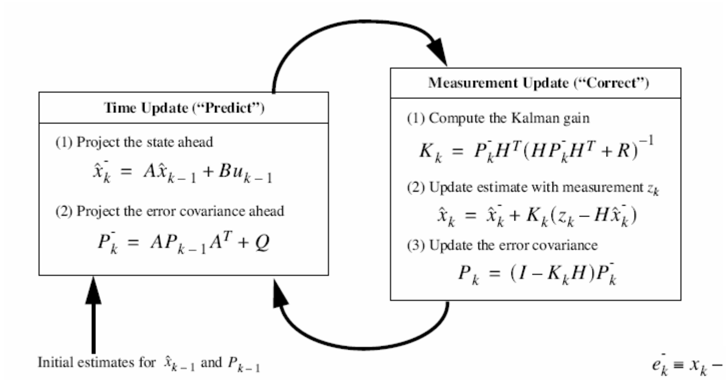
- Measurement Equation

$$z(k) = Hx(k) + v(k)$$

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

Kalman Filter - Example

Kalman Filter



$$e_k^- \equiv x_k - \hat{x}_k^-$$
$$e_k \equiv x_k - \hat{x}_k$$
$$P_k^- = E[e_k^- e_k^{-T}]$$
$$P_k = E[e_k e_k^T]$$

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

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1})$$
$$z_k = h(x_k, v_k)$$

Approximation of original model (w, v unavailable)

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0)$$
$$\tilde{z}_k = h(\tilde{x}_k, 0)$$

Linearizing through Taylor series expansion

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + W w_{k-1}$$
$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + V v_k .$$

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 3rd 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 $\bar{\mathbf{x}}$, covariance $\mathbf{P}_x$)
  - **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)

$$\mathcal{X}_0 = \bar{\mathbf{x}}$$
$$\mathcal{X}_i = \bar{\mathbf{x}} + \left(\sqrt{(L+\lambda)\mathbf{P}_\mathbf{x}}\right)_i \qquad i = 1, \ldots, L$$
$$\mathcal{X}_i = \bar{\mathbf{x}} - \left(\sqrt{(L+\lambda)\mathbf{P}_\mathbf{x}}\right)_{i-L} \qquad i = L+1, \ldots, 2L$$

$$W_0^{(m)} = \lambda/(L+\lambda)$$
$$W_0^{(c)} = \lambda/(L+\lambda) + (1 - \alpha^2 + \beta)$$
$$W_i^{(m)} = W_i^{(c)} = 1/\{2(L+\lambda)\} \qquad i = 1, \ldots, 2L$$

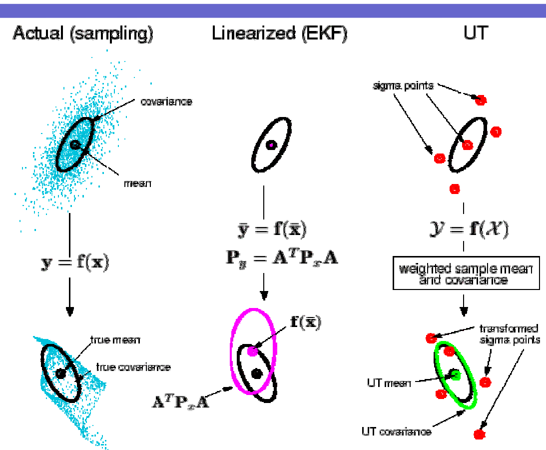
Scaling Parameter $\lambda = \alpha^2(L+\kappa) - L$

α determines the spread of the sigma points around $\bar{\mathbf{x}}$

$$\mathcal{Y}_i = g(\mathcal{X}_i) \quad i = 0, \ldots, 2L \quad ,$$

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_i$$
$$\mathbf{P}_\mathbf{y} \approx \sum_{i=0}^{2L} W_i^{(c)} \{\mathcal{Y}_i - \bar{\mathbf{y}}\}\{\mathcal{Y}_i - \bar{\mathbf{y}}\}^T$$

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

$$\mathbf{x}_k^a = [\mathbf{x}_k^T \ \mathbf{v}_\mathbf{k}^\mathbf{T} \ \mathbf{n}_\mathbf{k}^\mathbf{T}]^\mathbf{T}$$

$$P^a = \begin{bmatrix} P_x & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_n \end{bmatrix}$$

UKF Algorithm

Initialize with:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$$

$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$$

$$\hat{\mathbf{x}}_0^a = E[\mathbf{x}^a] = [\hat{\mathbf{x}}_0^T \ \mathbf{0} \ \mathbf{0}]^\mathbf{T}$$

$$\mathbf{P}_0^a = E[(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)^T] = \begin{bmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\mathbf{v} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_\mathbf{n} \end{bmatrix}$$

UKF Algorithm

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

Calculate sigma points:

$$\boldsymbol{\mathcal{X}}^a_{k-1} = \left[\hat{\mathbf{x}}^a_{k-1} \;\; \hat{\mathbf{x}}^a_{k-1} \pm \sqrt{(L+\lambda)\mathbf{P}^{\mathbf{a}}_{\mathbf{k-1}}}\right]$$

Time update:

$$\boldsymbol{\mathcal{X}}^x_{k|k-1} = \mathbf{F}[\boldsymbol{\mathcal{X}}^x_{k-1}, \boldsymbol{\mathcal{X}}^v_{k-1}]$$

$$\hat{\mathbf{x}}^-_k = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}^x_{i,k|k-1}$$

$$\mathbf{P}^-_k = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}^x_{i,k|k-1} - \hat{\mathbf{x}}^-_k][\mathcal{X}^x_{i,k|k-1} - \hat{\mathbf{x}}^-_k]^T$$

$$\boldsymbol{\mathcal{Y}}_{k|k-1} = \mathbf{H}[\boldsymbol{\mathcal{X}}^x_{k|k-1}, \boldsymbol{\mathcal{X}}^n_{k-1}]$$

$$\hat{\mathbf{y}}^-_k = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1}$$

Measurement update equations:

$$\mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}^-_k][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}^-_k]^T$$

$$\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}^-_k][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}^-_k]^T$$

$$\mathcal{K} = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}^{-1}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k}$$

$$\boxed{\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^-_k + \mathcal{K}(\mathbf{y}_k - \hat{\mathbf{y}}^-_k)}$$

$$\boxed{\mathbf{P}_k = \mathbf{P}^-_k - \mathcal{K}\mathbf{P}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} \mathcal{K}^T}$$

Mean state vector

Covariance output

$\boldsymbol{\mathcal{X}}^a = [(\boldsymbol{\mathcal{X}}^x)^T \; (\boldsymbol{\mathcal{X}}^v)^T \; (\boldsymbol{\mathcal{X}}^n)^T]^T$

λ = composite scaling parameter,
L = dimension of augmented state vector,
P_v = process noise covariance,
P_n = measurement noise covariance

Sensor Utility Measure

$$\psi_C(k+1) = -trace[\mathbf{P}_C(k+1)]$$

- $\mathbf{P}_c(k+1)$ 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
  - $U(\pi[k], k) \rightarrow \psi_{\mathcal{C}}(k)$

$$\min_{\pi_{\mathcal{N}}} \quad \sum_{k=1}^{K} -U(\pi[k], k)$$
$$s.t. \quad \sum_{k=1}^{K} \sum_{i \in N} E_i(\pi_i[k]) \leq K \cdot E_{ave}$$

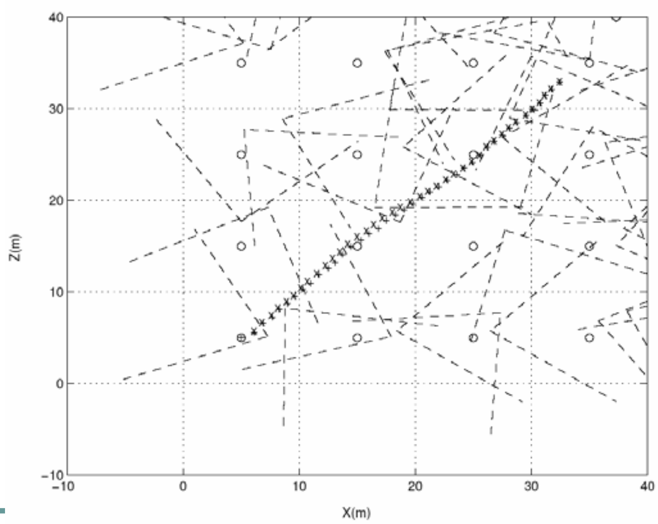
Information Utility (IU)

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

$$\min_{\pi_{\mathcal{R}}[k]} \quad -U(\pi_{\mathcal{R}}[k], k) - \sum_{j=k+1}^{k+W-1} \hat{U}(\pi_{\mathcal{R}}[j], j)$$
$$s.t. \quad \sum_{j=k-W+1}^{k} \sum_{i \in N} E_i(\pi_i[j]) \leq W \cdot E_{ave}$$
$$\sum_{i \in N} E_i(\pi_i[j]) \leq E_{pk}$$

- IU maximization problem solved using tree pruning algorithm

Simulations



Simulations

- Optimal Selection is turning on predicted suitable sensors in advance.

	Optimal Selection	Random Sensor Selection
Ave. E	4.93	2.41
rmse	0.40	0.52
% lost tracks	6.3	58.0

	Optimal Selection	Closest Sensor Selection
Ave. E	4.87	4.41
rmse	0.44	0.52
% lost tracks	10.3	11.2

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