Numerically Stable Iterated Filters for Bearing-Only SLAM

Presented in part at URAI 2007, accepted to ICRA2008

Hyungpil Moon*,
Stephen Tully, George Kantor, Howie Choset

*Post-doc fellow, The Robotics Institute
(now at ME, SungKyunKwan University, Suwon, Korea)

Simultaneous Localization and Mapping
SLAM

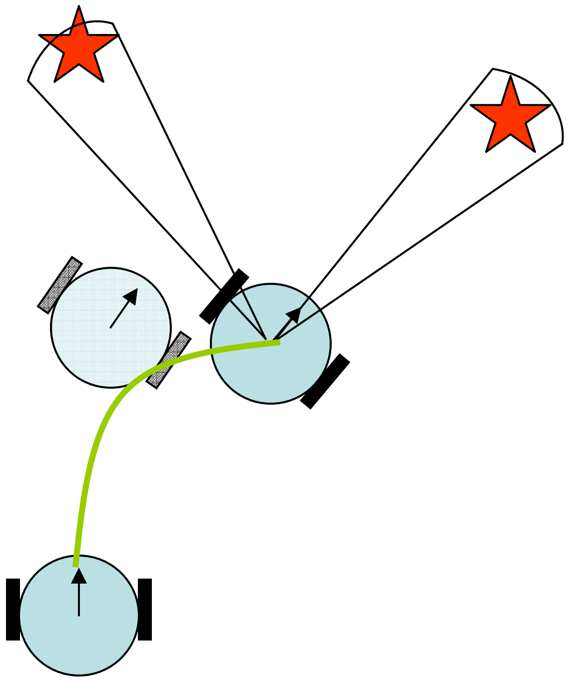
- Localization
  - For a known map
  - find the robot locations

- Mapping
  - For known robot path
  - find the map

- SLAM
  - Map and robot position are not known
  - Incrementally build a navigation map while simultaneously use this map to update its location

Conventional EKF SLAM

- Extended Kalman Filter
  - Motion uncertainty, measurement noise – Gaussian model
  - State estimation – robot pose and landmark locations

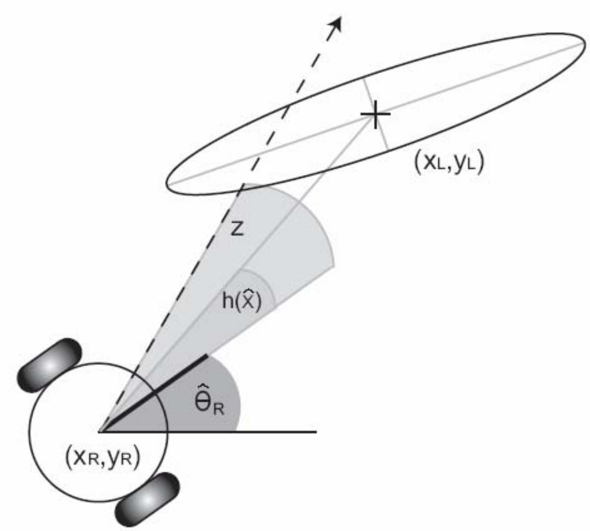


Predict using motion model :
$dx/dt = f(x, u)$
Measurement update :
$x_{t+1} = x_t + K(z-h)$
$P_{t+1} = (I-KH)P_t$

Bearing-Only SLAM

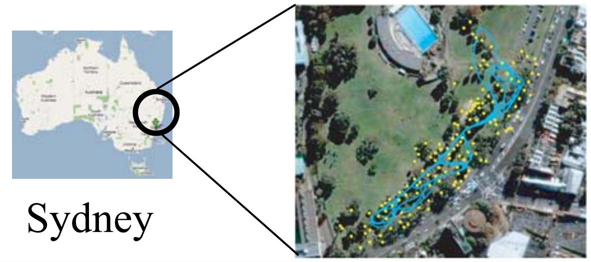
- Range-Bearing SLAM
  - State variable $[x_R \; y_R \; \theta_R \; x_L \; y_L \; \cdots]^T$
  - Measurements (r, ϕ)
  - Landmark location $\begin{cases} x_L = x_R + r\cos(\phi + \theta_R) \\ y_L = y_R + r\sin(\phi + \theta_R) \end{cases}$
  - Conventional method

    EKF : prediction using odometer + measurement update using range and bearing information
- Bearing-Only SLAM
  - Cannot determine the landmark location with only one bearing measurement
  - Requires bearing measurements at multiple difference poses

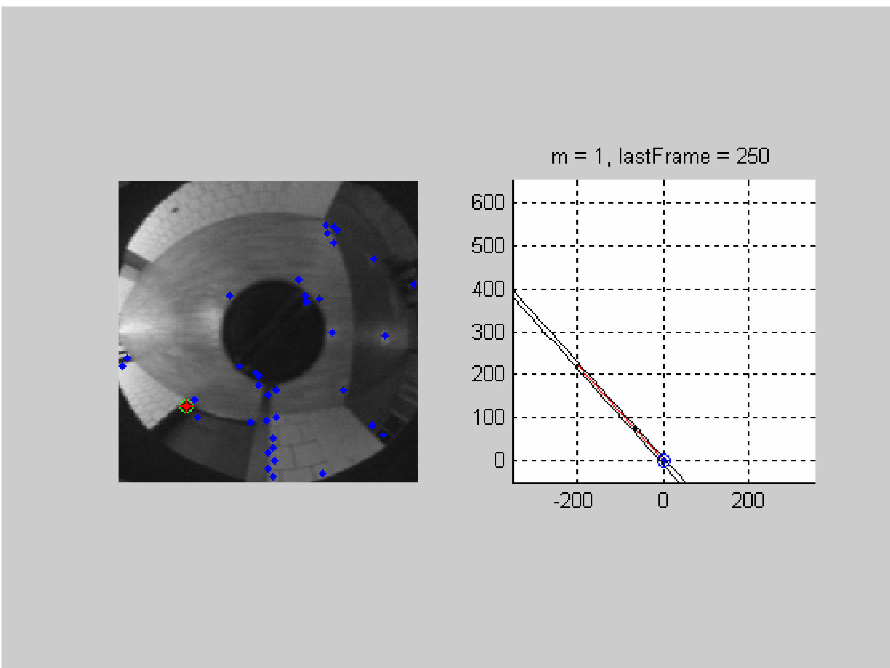


Failure of EKF with Improper Initialization

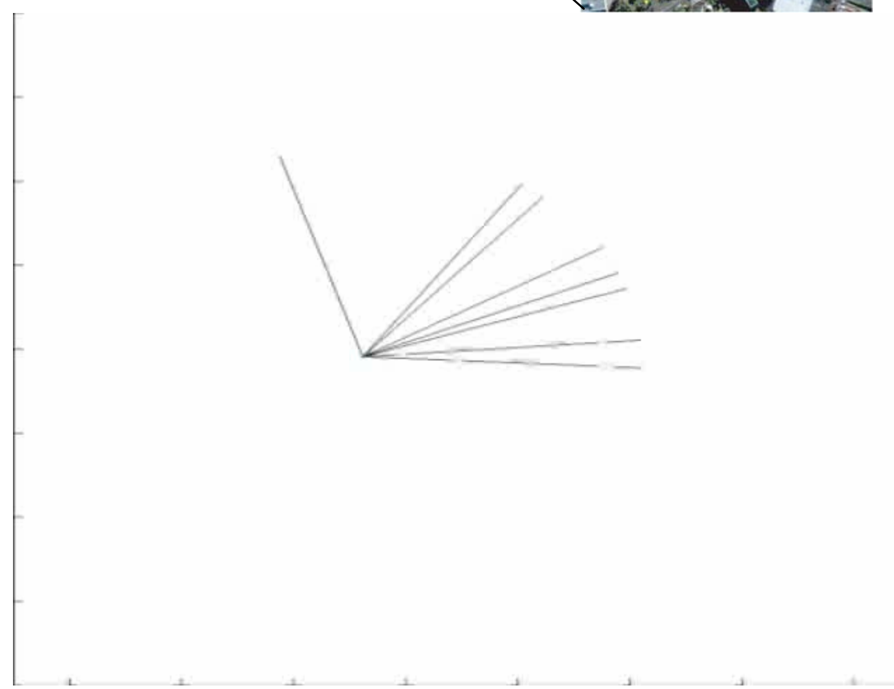
- Naïve approach : guess range information and implement it with large covariance
- Catastrophic !!!



Sydney



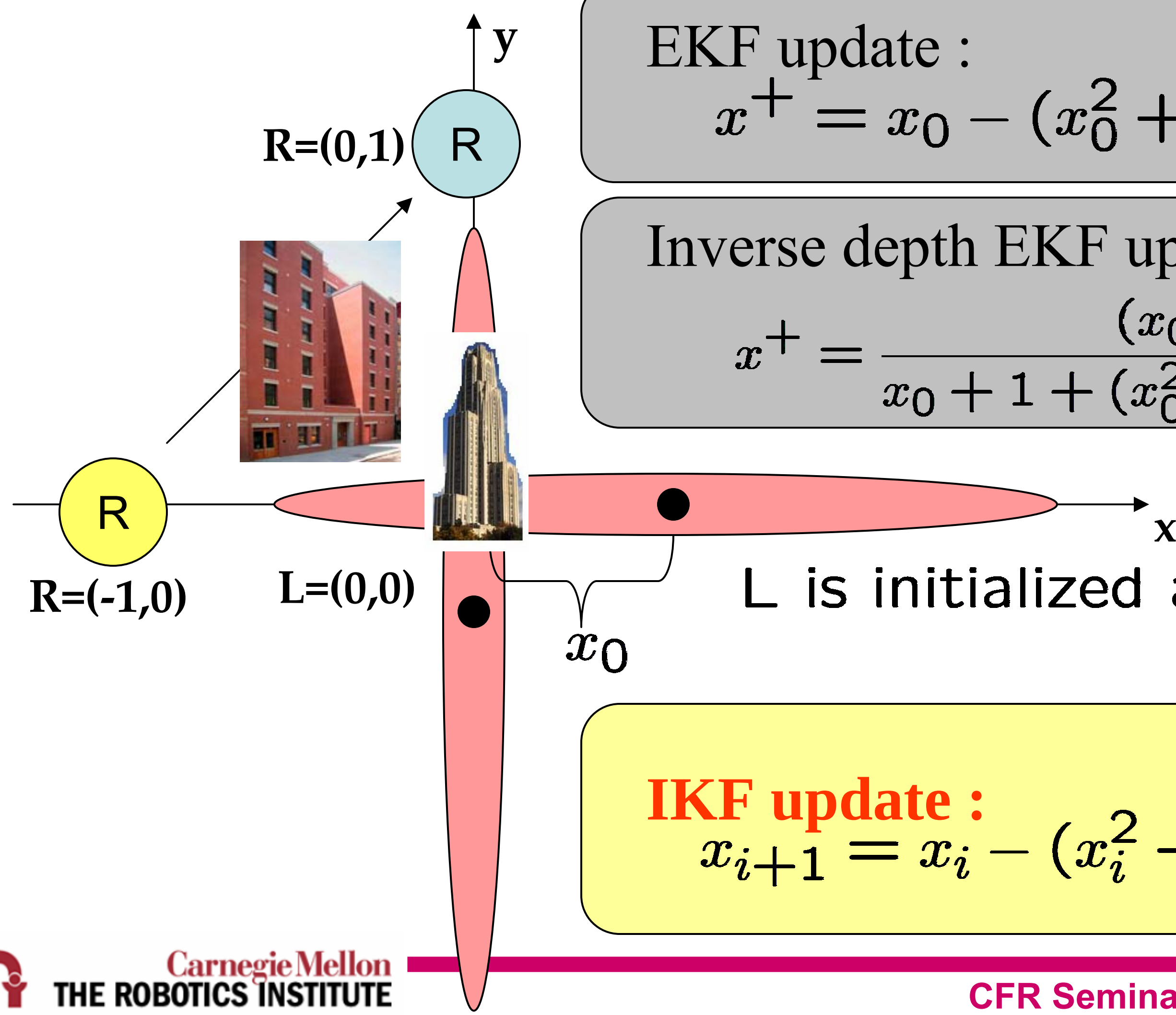
Indoor experiment



Outdoor experiment : Victoria park data set

Bearing-Only SLAM : Why fails?

No process disturbance, perfect measurement



EKF update :

$$x^+ = x_0 - (x_0^2 + 1)\arctan(x_0) \neq 0$$

Inverse depth EKF update :

$$x^+ = \frac{(x_0+1)^2}{x_0 + 1 + (x_0^2+1)\arctan(x_0)} - 1$$

L is initialized at $x_0 \neq 0$

IKF update :

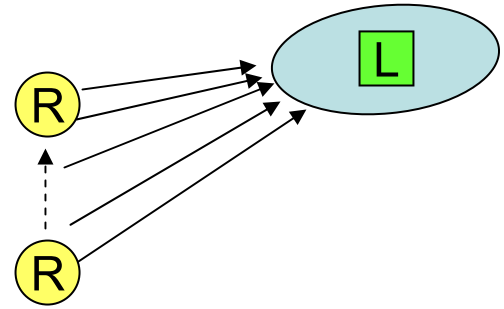
$$x_{i+1} = x_i - (x_i^2 + 1)\arctan(x_i) \rightarrow 0$$

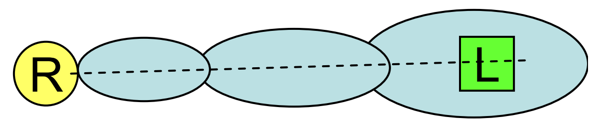
Contents

- Related work
  - Delayed methods, Sampling based methods, Inverse depth representation
- Review of IKF
  - Solving ML with Gaussian Newton method
- Bearing-Only SLAM
  - Landmark Initialization
  - Variable step-size IKF
- Square Root IKF
  - Stability issue
- Evaluation

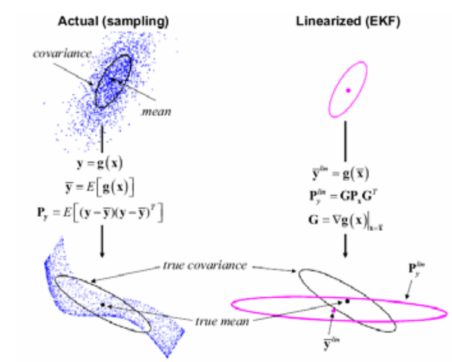
Related Work

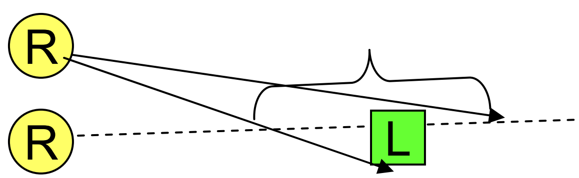
- Delayed initialization of the landmark location
  - A batch update [Deans 2000, Bailey 2003]
  - Bivariate ellipse representations [Costa 2004]
- Gaussian Sum Filter (GSF)
  - Sequential Probability Ratio Test [Kwok2005]
  - Federated Information Sharing [Sola 2005]
- Sampling methods
  - Particle Filters
    - Using Pseudo-range [Kwok2006]
    - Vision based, delayed method [Davison2003]
    - FastSLAM particle filter [Eade2006]
  - Unscented Kalman Filter [Chekhlov2006]
- Inverse depth
  - Monocular vision
  [Montiel 2006, 2007]



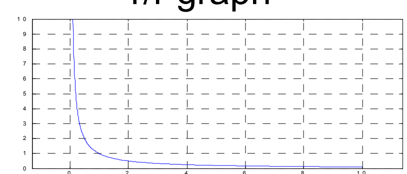


UKF



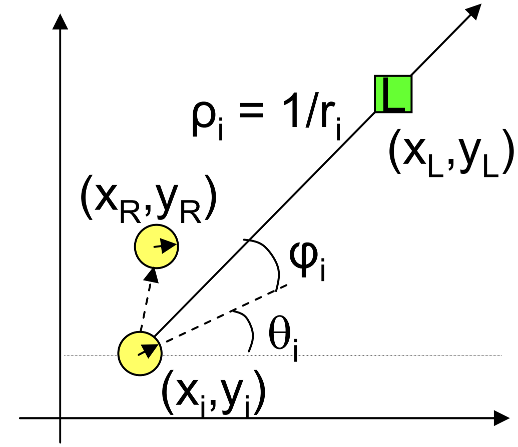


1/r graph



Inverse Depth Parameterization

- State $[x_R \ y_R \ \theta_R \ x_i \ y_i \ \rho_i \ \phi_i + \theta_i]^T$
  - The first point observation
  - Polar coordinate



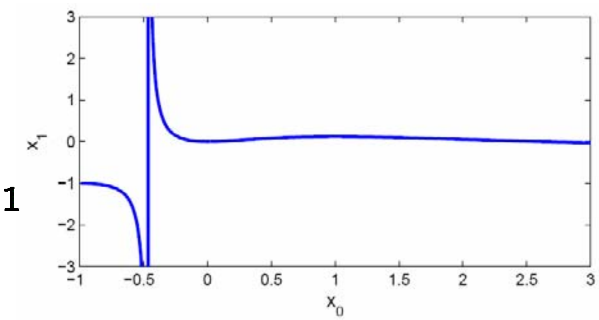
- Measurement model

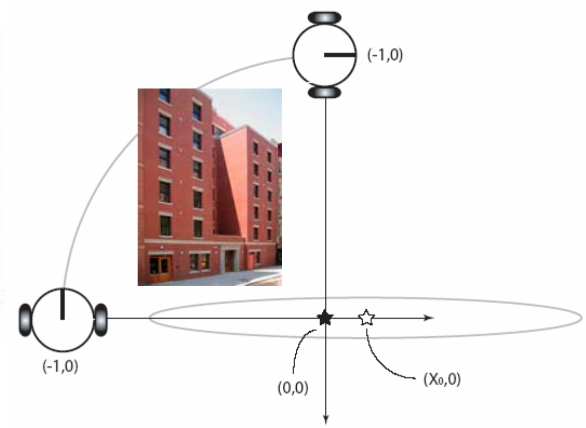
$$\tan\phi = \frac{y_L - y_R}{x_L - x_R} - \theta_R = \frac{y_i + \frac{1}{\rho_i}\sin(\phi_i + \theta_i) - y_R}{x_i + \frac{1}{\rho_i}\cos(\phi_i + \theta_i) - x_R} - \theta_R$$

- Initialization

$\rho \approx 0.5$

$$x_1 = \frac{(x_0 + 1)^2}{x_0 + 1 + (x_0^2 + 1)\arctan(x_0)} - 1$$





Initialization is still an issue !!!

EKF: Solving ML with Gauss-Newton Method

- Want to find the maximum likelihood state estimate for given measurement and predicted state estimate

$$\hat{x}^{+} = \arg\max_{x} \text{prob}(x | z, R, \hat{x}^{-}, P^{-})$$

Gaussian model ⇩

$$\hat{x}^{+} = \arg\max (L(\xi))$$

$$\text{where } L(\xi) = \frac{1}{\sqrt{(2\pi)^{m+n}|Q|}} \exp\left(-\frac{1}{2} e(\xi)^T Q^{-1} e(\xi)\right) \qquad e(x) = \begin{bmatrix} z - h(x) \\ \hat{x} - x \end{bmatrix}$$

$$Z = \begin{bmatrix} z \\ \hat{x} \end{bmatrix} \sim N(g(x), Q) \qquad g(x) = \begin{bmatrix} h(x) \\ x \end{bmatrix} \qquad Q = \begin{bmatrix} R & 0 \\ 0 & P \end{bmatrix}$$

→ Minimization problem

ML Gauss-Newton method: continues

$$\hat{x}^+ = \arg\max\,(L(\xi))$$

$$\text{where } L(\xi) = \frac{1}{\sqrt{(2\pi)^{m+n}|Q|}} \exp\left(-\tfrac{1}{2} e(\xi)^T Q^{-1} e(\xi)\right) \qquad e(x) = \begin{bmatrix} z - h(x) \\ \hat{x} - x \end{bmatrix}$$

$$Z = \begin{bmatrix} z \\ \hat{x} \end{bmatrix} \sim N(g(x), Q) \qquad g(x) = \begin{bmatrix} h(x) \\ x \end{bmatrix} \qquad Q = \begin{bmatrix} R & 0 \\ 0 & P \end{bmatrix}$$

→ Taking log : Minimization problem

$$\hat{x}^+ = \arg\min\,(l(\xi))$$

$$\text{where } l(\xi) = \tfrac{1}{2} e(\xi)^T Q^{-1} e(\xi)$$

- Solve minimization using Gauss-Newton method

$$x_{i+1} = x_i - \left(\nabla^2 l(x_i)\right)^{-1} \nabla l(x_i)$$

IKF as Gauss-Newton Method

- Gauss-Newton method

$$x_{i+1} = x_i - \left(\nabla^2 l(x_i)\right)^{-1} \nabla l(x_i)$$

$$l(\xi) = \frac{1}{2}\|r(\xi)\|^2 \qquad r(\xi) = \sqrt{Q}^{-1} e(\xi) \qquad \nabla l(\xi) = \left(\frac{dr}{d\xi}\right)^T r(\xi)$$

- Use approximated Hessian

$$\nabla^2 l(\xi) \approx \left(\frac{dr}{d\xi}\right)^T \left(\frac{dr}{d\xi}\right)$$

- Recall $Z = \begin{bmatrix} z \\ \hat{x} \end{bmatrix} \sim N(g(x), Q)$

$$e(x) = \begin{bmatrix} z - h(x) \\ \hat{x} - x \end{bmatrix} \quad g(x) = \begin{bmatrix} h(x) \\ x \end{bmatrix} \quad Q = \begin{bmatrix} R & 0 \\ 0 & P \end{bmatrix}$$

- Iterated Kalman Filter

$$\begin{aligned} x_{i+1} &= x_i + \left(g'(x_i)^T Q^{-1} g'(x_i)\right)^{-1} g'(x_i) Q^{-1}(Z - g(x_i)) \\ &= \hat{x} + K_i\left(z - h(x_i) - H_i(\hat{x} - x_i)\right) \end{aligned}$$

- Iterate just once → conventional Kalman Filter !!!

Bearing-Only SLAM using IKF

- State

$$x_k = [x_R,\ y_R,\ \theta_R,\ x_{L_1},\ y_{L_1},\ \cdots x_{L_N},\ y_{L_N}]^T$$

- Process model

$$f(x_k, u_k) = x_k + \begin{bmatrix} I_{3\times 3} \\ 0_{2N\times 3} \end{bmatrix} \begin{bmatrix} v_k \cos\theta_k \delta t \\ v_k \sin\theta_k \delta t \\ \omega_k \delta t \end{bmatrix}$$

$$F_k = \frac{\partial f}{\partial x_k}, \quad W_k = \frac{\partial f}{\partial u_k}$$

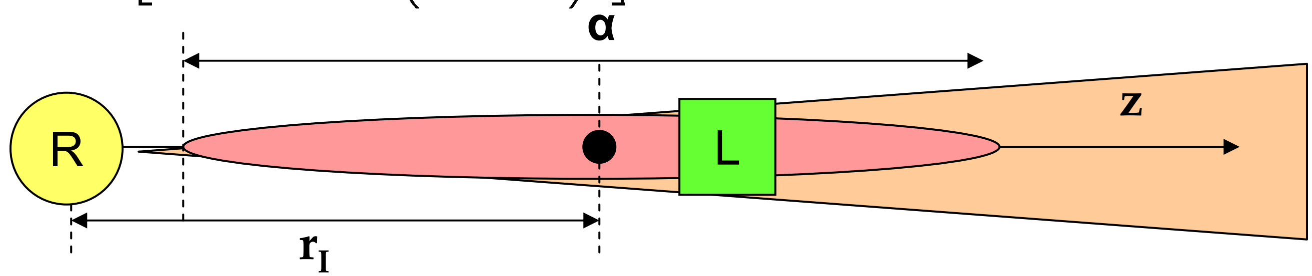
- Measurement model

$$h_i(x_k) = \arctan\left(\frac{y_R - y_{L_i}}{x_R - x_{L_i}}\right), \qquad H = \nabla h$$

Landmark Initialization and Update

- Approximate the uniform distribution by Gaussian with large mean and covariance
- Initialization is NOT delayed

$$\hat{x}_k = \begin{bmatrix} \cdots \\ x_R + r_I \cos\left(\hat{\theta}_R + z_I\right) \\ x_R + r_I \sin\left(\hat{\theta}_R + z_I\right) \end{bmatrix}, \quad P_k = \left[\begin{array}{c|cc} \cdots & \multicolumn{2}{c}{0} \\ \hline 0 & \alpha & 0 \\ & 0 & \alpha \end{array}\right]$$

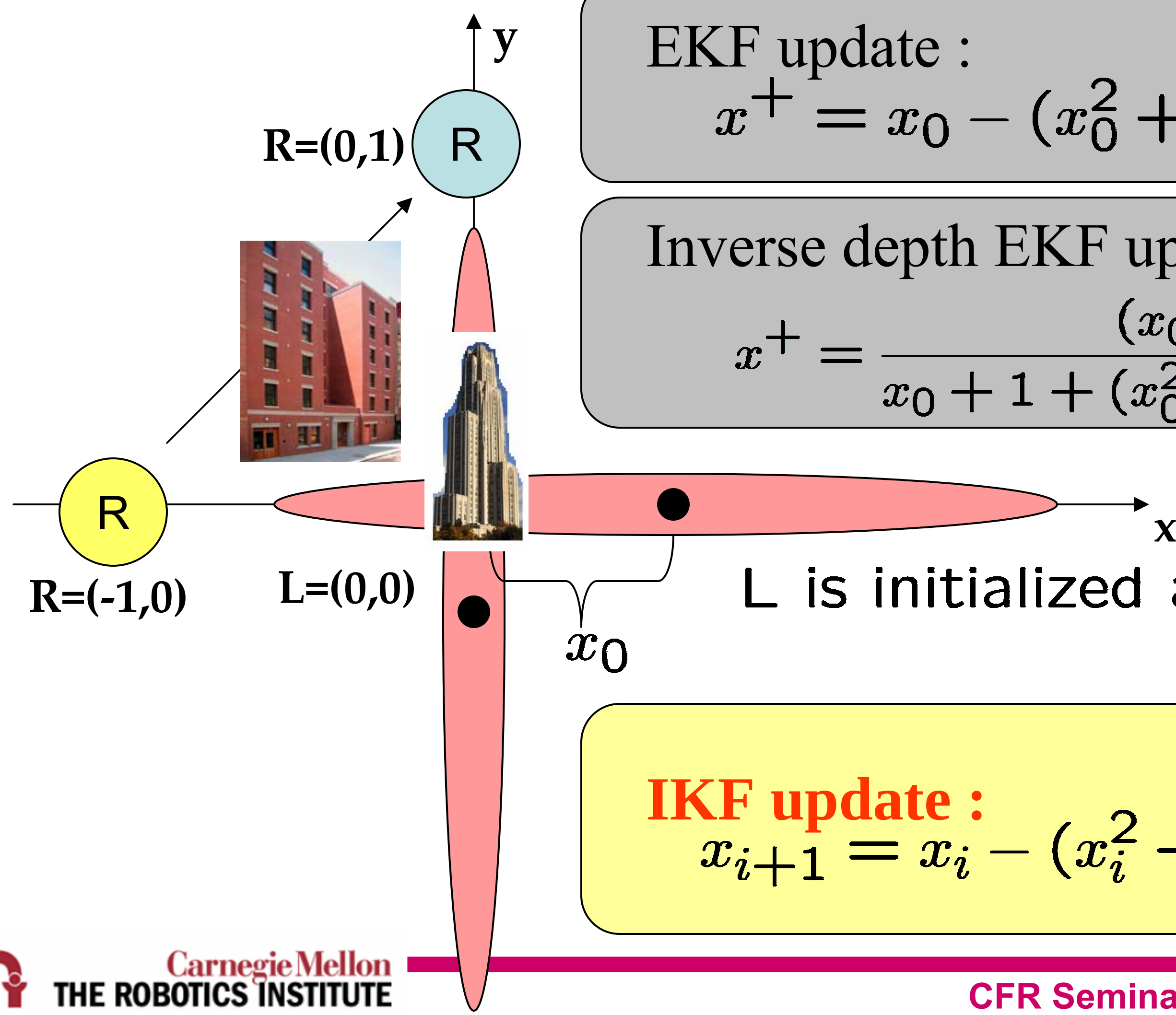


- Update equations

$$\begin{aligned} x_0 &= \hat{x}_{k+1}^-, \ P_0 = P_{k+1}^- \\ K_i &= P_0 H_i^T (H_i P_0 H_i^T + R)^{-1} \\ x_{i+1} &= x_i + \gamma_i (H_i^T R^{-1} H_i + P_0^{-1})^{-1} \\ &\quad \cdot (H_i^T R^{-1}(z - h(x_i)) + P_0^{-1}(x_0 - x_i)) \\ &= x_0 + K_i (z_{k+1} - h(x_i) - H_i(x_0 - x_i)) \end{aligned}$$

Revisit Analytic Examples

No process disturbance, perfect measurement



EKF update :

$$x^+ = x_0 - (x_0^2 + 1)\arctan(x_0) \neq 0$$

Inverse depth EKF update :

$$x^+ = \frac{(x_0 + 1)^2}{x_0 + 1 + (x_0^2 + 1)\arctan(x_0)} - 1$$

L is initialized at $x_0 \neq 0$

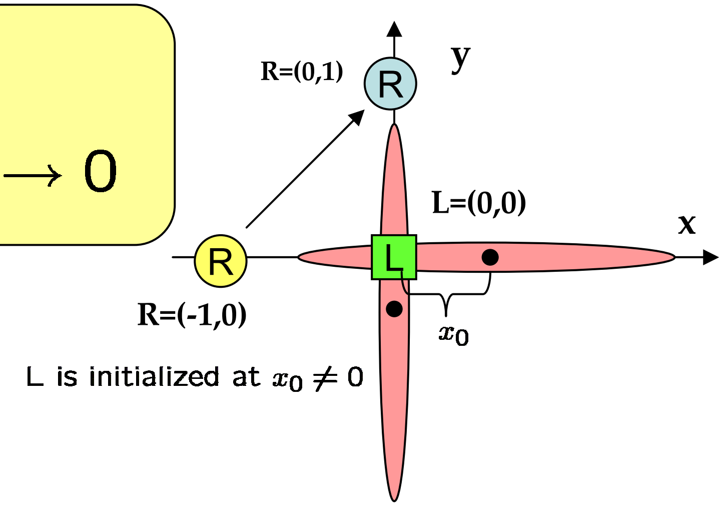
IKF update :

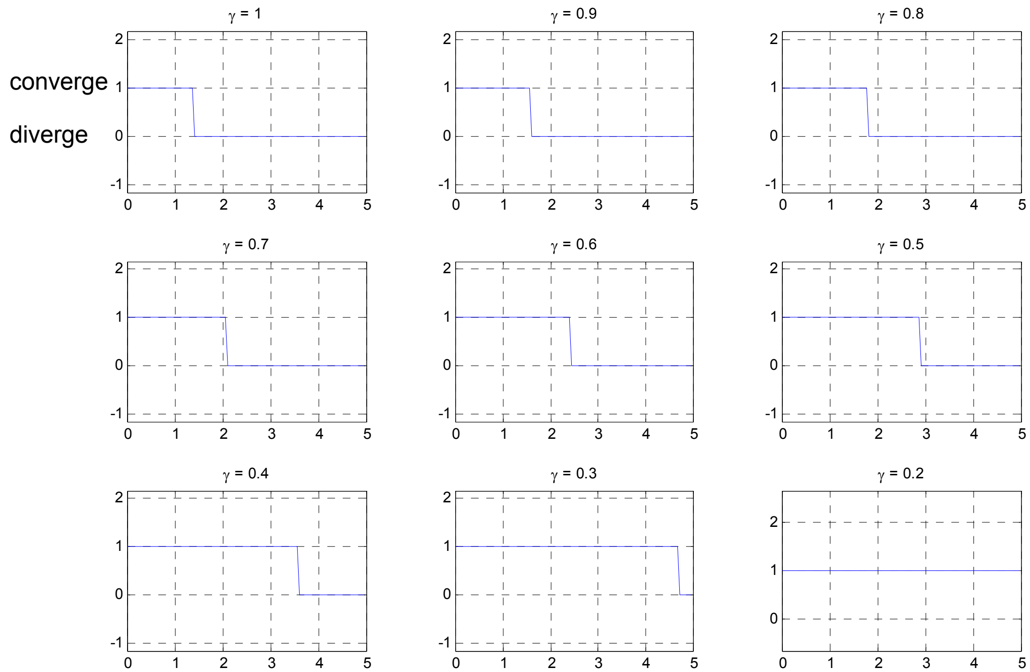
$$x_{i+1} = x_i - (x_i^2 + 1)\arctan(x_i) \to 0$$

Variable Step-Size Gauss Newton

IKF update :

$$x_{i+1} = x_i - \gamma_i (x_i^2 + 1) \arctan(x_i) \rightarrow 0$$



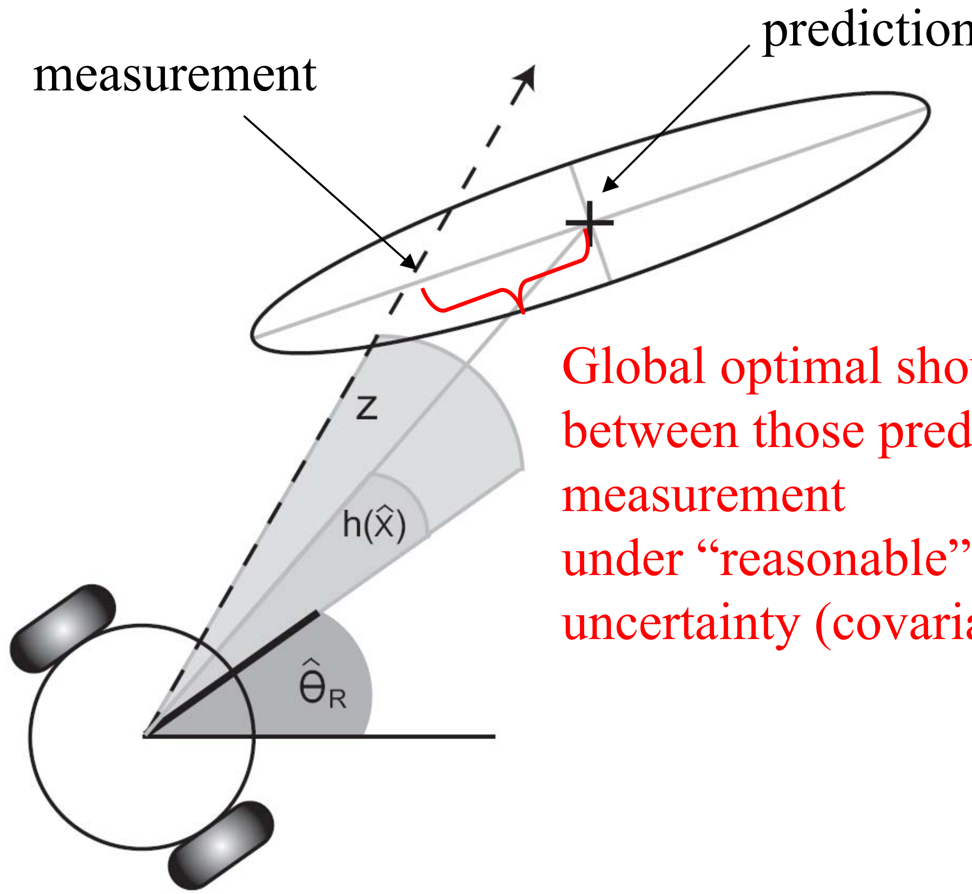


- Region of attraction is enlarged by changing step-size (back-stepping)

Story so far …

- The goal of the Kalman update is to maximize the posterior
- To maximize the posterior, we must reduce the cost (Mahalanobis distance) at each update step
- Gauss-Newton direction is good, but taking a full step along the direction may be bad
- Any optimization method will do the job as long as it reduces the cost
- Inverse depth representation with conventional EKF could go wrong
- Global minimum is not YET proved, but believable…..

Intuition Behind the Belief



Global optimal should be here
between those prediction and
measurement
under "reasonable"
uncertainty (covariance)

Mathematical Intuition according to Steve

- To maximize the posterior : $P(X|z, R, \hat{X}, P)$
- Equivalently, reduce the cost (Mahalanobis distance) at each update step

$$\hat{X}^+ = \arg\min_X \left([\hat{X} - X]^T P^{-1} [\hat{X} - X] + [h(X) - z]^T R^{-1} [h(X) - z] \right)$$

$$\phi = atan\left(\frac{y_L - y_R}{x_L - x_R}\right)$$

$$r = \sqrt{(x_L - x_R)^2 + (y_L - y_R)^2}$$

$$\hat{X}^+ = \arg\min_X \left(\begin{bmatrix} x_R \\ y_R \\ \theta_R \\ x_R + r\cos\phi \\ y_R + r\sin\phi \\ \theta_R \end{bmatrix} - \begin{bmatrix} \hat{X} \\ \phi - z \end{bmatrix} \right)^T \begin{bmatrix} P^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} \left(\begin{bmatrix} x_R \\ y_R \\ \theta_R \\ x_R + r\cos\phi \\ y_R + r\sin\phi \\ \theta_R \end{bmatrix} - \begin{bmatrix} \hat{X} \\ \phi - z \end{bmatrix} \right)$$

Mathematics continues : Assume known φ

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & \cos\phi \\ 0 & 1 & 0 & \sin\phi \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad b = \begin{bmatrix} \hat{x}_R \\ \hat{y}_R \\ \hat{\theta}_R \\ \hat{x}_L \\ \hat{y}_L \\ \phi - z \end{bmatrix}$$

$$\begin{aligned} \phi &= atan\left(\frac{y_L - y_R}{x_L - x_R}\right) \\ r &= \sqrt{(x_L - x_R)^2 + (y_L - y_R)^2} \\ y &= [x_R \; y_R \; \theta_R \; r]^T \end{aligned}$$

$$C = \begin{bmatrix} P^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix}$$

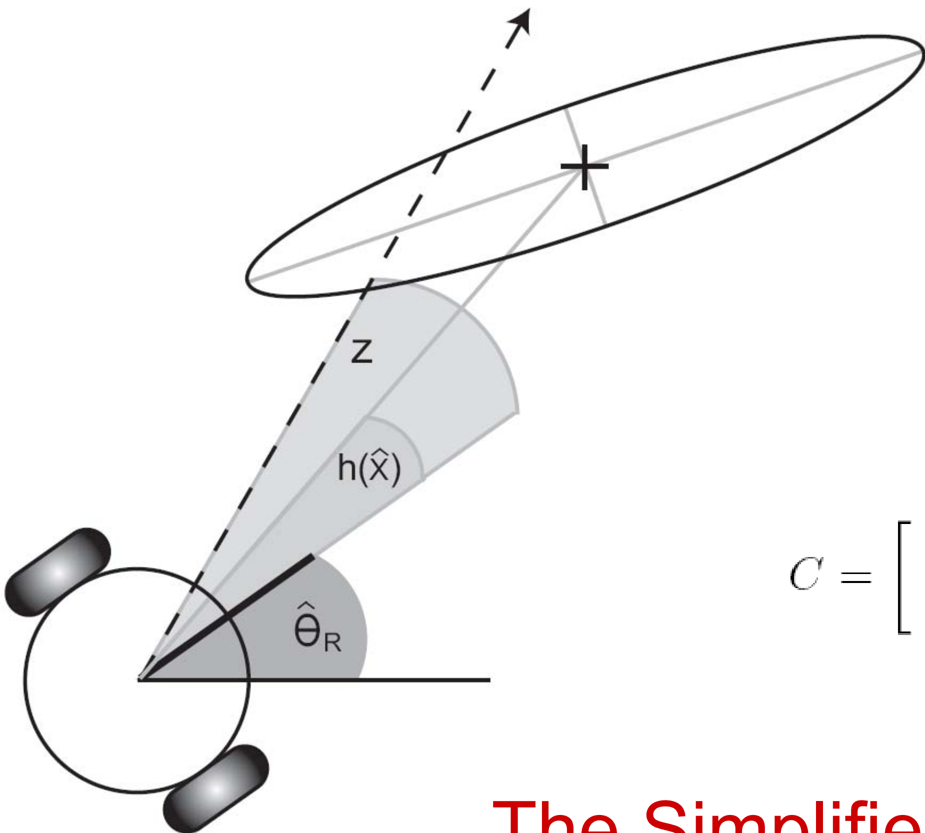
$$\hat{X}^+ = \arg\min_X \left(\begin{bmatrix} x_R \\ y_R \\ \theta_R \\ x_R + r\cos\phi \\ y_R + r\sin\phi \\ \theta_R \end{bmatrix} - \begin{bmatrix} \hat{X} \\ \phi - z \end{bmatrix} \right)^T \begin{bmatrix} P^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} \left(\begin{bmatrix} x_R \\ y_R \\ \theta_R \\ x_R + r\cos\phi \\ y_R + r\sin\phi \\ \theta_R \end{bmatrix} - \begin{bmatrix} \hat{X} \\ \phi - z \end{bmatrix} \right)$$

$$[\phi^+, y^+] = \arg\min_{\phi, y} (Ay - b)^T C (Ay - b)$$

$$y^+ = (A^T C A)^{-1} A^T C b$$

If φ is known

Mathematics continues : Plug y^+ back in



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

$$C = \begin{bmatrix} P^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} \qquad b = \begin{bmatrix} \hat{x}_R \\ \hat{y}_R \\ \hat{\theta}_R \\ \hat{x}_L \\ \hat{y}_L \\ \phi - z \end{bmatrix}$$

The Simplified Cost Function

$$C(\phi) = b^T(C - CA(A^TCA)^{-1}A^TC)b$$

Global Optimization for State Update

$$C(\phi) = b^T(C - CA(A^TCA)^{-1}A^TC)b$$

- Start with the bearing to the current landmark estimate
- Run a numerical optimization method to search for a minimum of the cost function

$$\phi_0 = atan\left(\frac{\hat{y}_L - \hat{y}_R}{\hat{x}_L - \hat{x}_R}\right)$$

NEUMERICAL OPTIMIZATION

Then we still use the conventional covariance update equation

$$P^+ = P - PH^T(R + HPH^T)^{-1}HP$$

Numerical Stability: Double-Loop Structure

- Sometimes covariance matrix becomes non-positive
- Numerical instability of the conventional KF update
- Require Inner loop stability
- Gauss-Newton method requires :

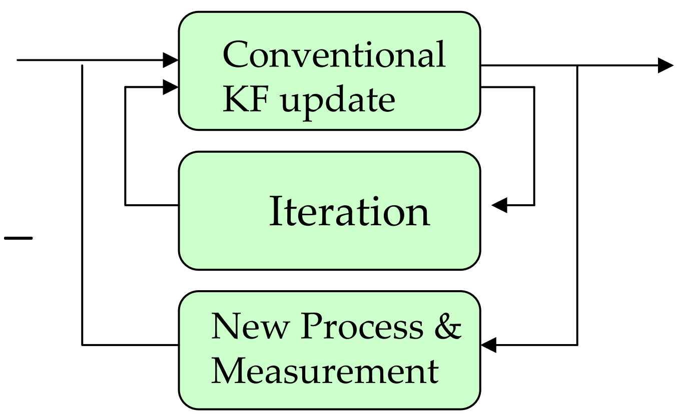
  Positive definite Hessian matrix

  → Positive definite Covariance

$$x_{i+1} = x_i - \left(\nabla^2 l(x_i)\right)^{-1} \nabla l(x_i)$$

$$P^+ = \underbrace{P^-}_{\text{Positive definite}} - \underbrace{P^- H^T (H P^- H^T + R)^{-1} H P^-}_{\text{Positive definite}}$$

Positive definite – Positive definite



Square Root IKF

- The goal is to maintain a square root form while updating the covariance

$$
\begin{aligned}
P_{k+1}^{-} &= F_k P_k^{+} F_k^T + W_k R W_k^T = F_k (P_{Vk} P_{Dk}^2 P_{Vk}^T) F_k^T + W_k R W_k^T \\
&= \begin{bmatrix} P_{Dk} P_{Vk}^T F_k^T \\ \sqrt{R}^T W_k^T \end{bmatrix}^T \begin{bmatrix} P_{Dk} P_{Vk}^T F_k^T \\ \sqrt{R}^T W_k^T \end{bmatrix}
\end{aligned}
$$

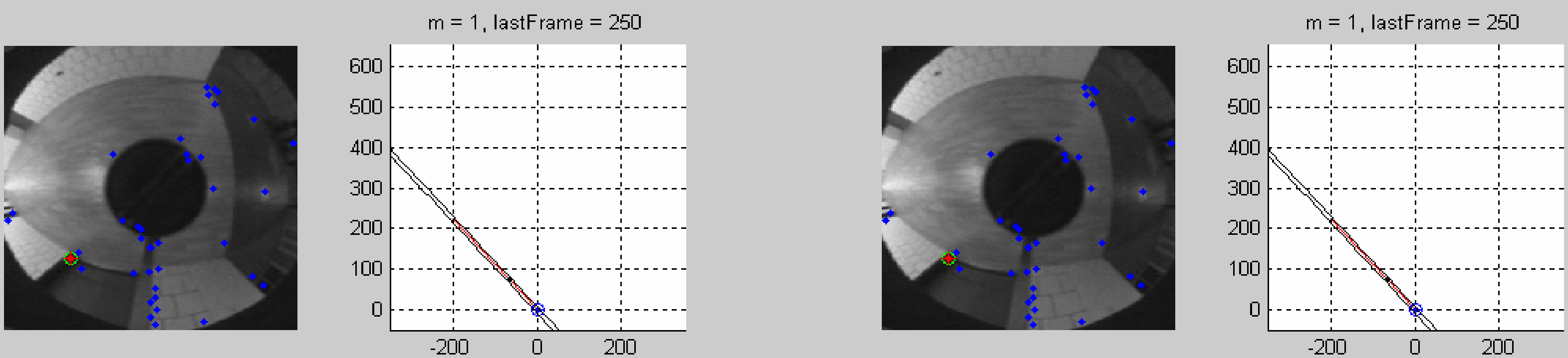
$$
\mathsf{SVD}\left(\sqrt{P_{k+1}^{-}}\right) = \mathsf{SVD}\left(\begin{bmatrix} P_{Dk} P_{Vk}^T F_k^T \\ \sqrt{R}^T W_k^T \end{bmatrix}\right) = P_U \begin{bmatrix} P_D \\ \hline 0 \end{bmatrix} P_V^T
$$

$$
\begin{aligned}
P_{k+1}^{+} &= ((P_{k+1}^{-})^{-1} + H_N^T R^{-1} H_N)^{-1} = (P_V P_D{}^{-2} P_V{}^T + H_N^T R^{-1} H_N)^{-1} \\
&= P_V \left(P_D{}^{-2} + P_V^T H_N^T R^{-1} H_N P_V\right)^{-1} P_V^T \\
&= P_V \left(T^T T\right)^{-1} P_V^T \\
&= (P_V P_V^*)\, P_D^*{}^2\, (P_V P_V^*)^T = P_{Vk+1} P_{Dk+1}^2 P_{Vk+1}^T
\end{aligned}
$$

Bearing-Only IKF SLAM Results : Indoor

Conventional method - Bearing Only EKF SLAM with a single Gaussian

Our method - Bearing Only SLAM using nonlinear filtering, modified IKF



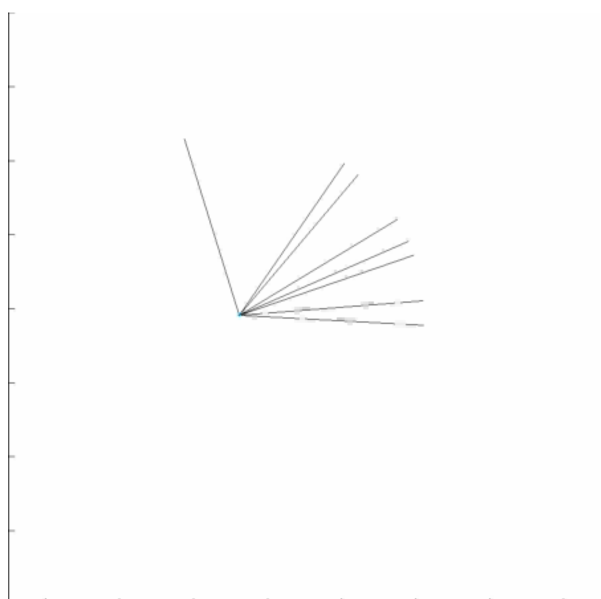
Results : Victoria Park Dataset

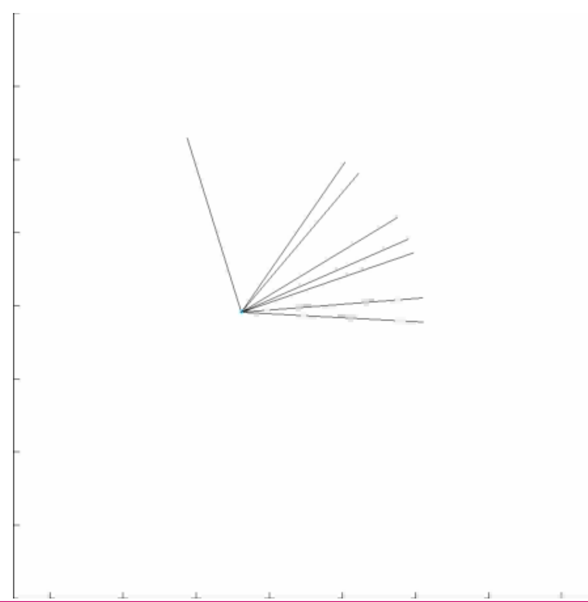
Sydney

SLAM bench mark
One of the Largest map available

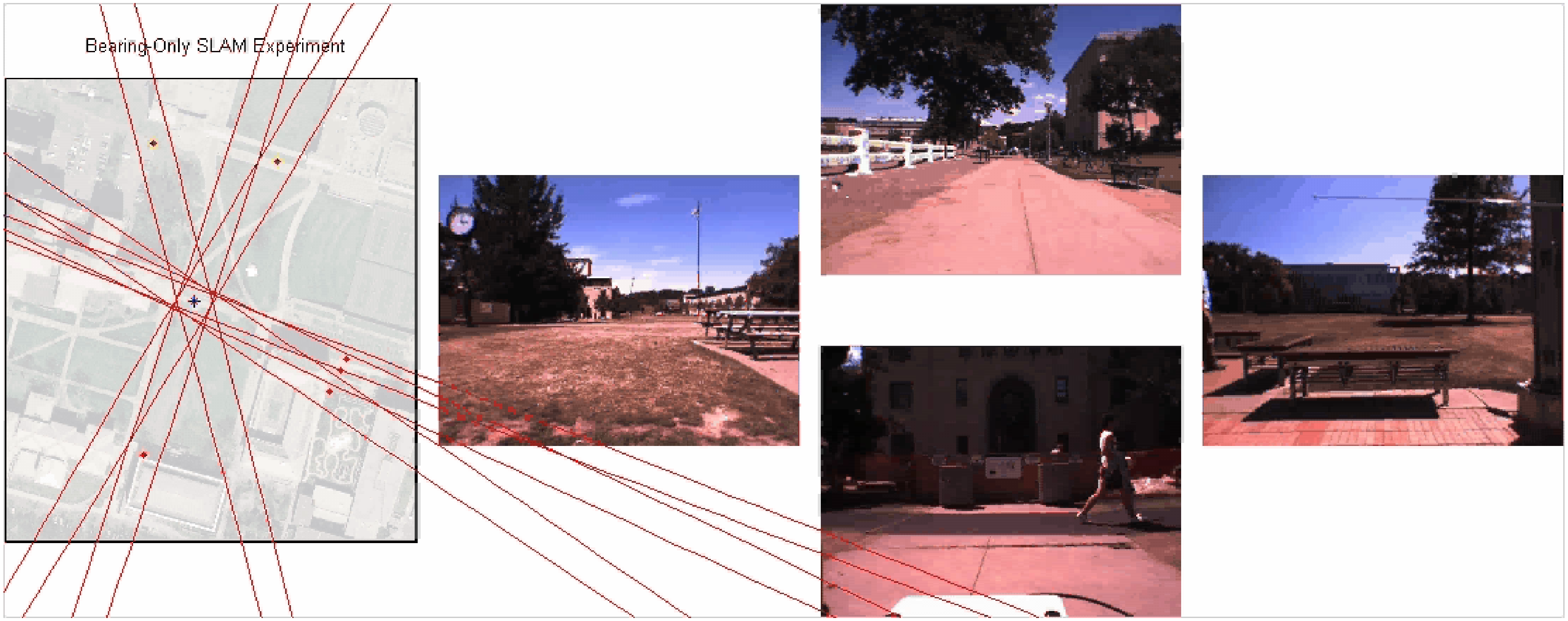
EKF fails

IKF





Results : Visual SLAM with a LAGR

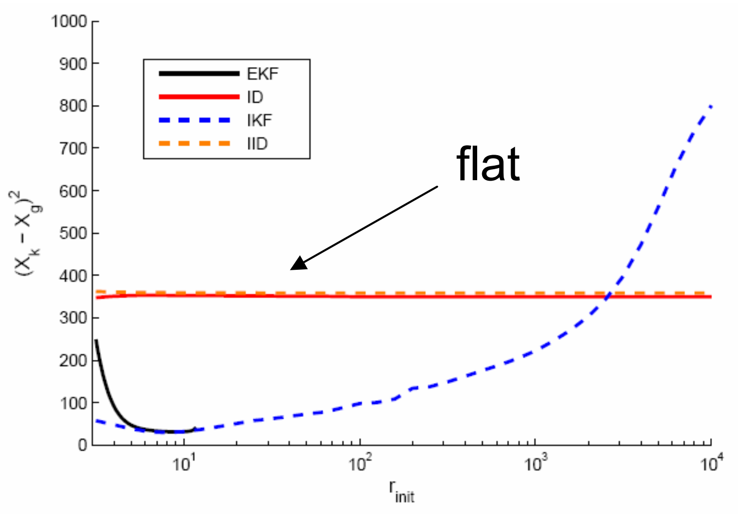


Results : SLAM with a LAGR

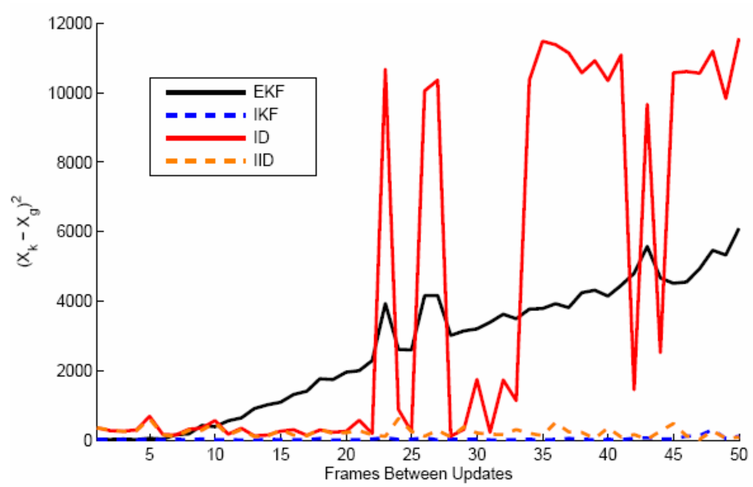
Overlapped with a satellite image

Comparisons

Map accuracy vs initialization distance



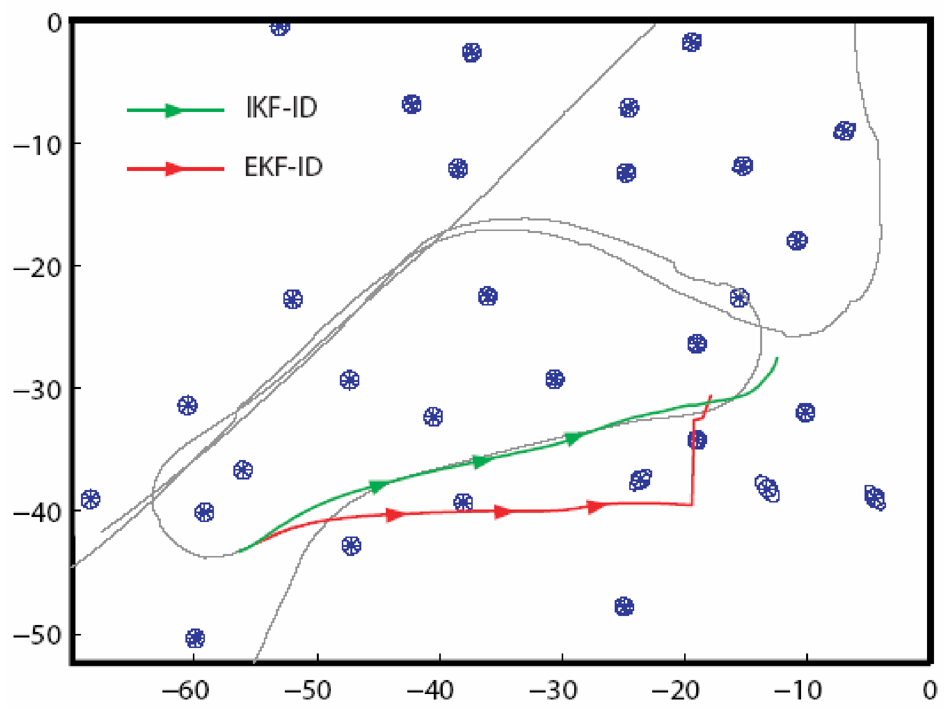
Map accuracy vs measurement frequency



High update rate ← → Low update rate

Inverse Depth with Iterations

- "Blinding" the robot for 46 consecutive update steps divergence of Inverse Depth with EKF
- Negative range
- Iterations !!!



Take-home Messages

- The goal of the Kalman update is to maximize the posterior
- To maximize the posterior, we must reduce the cost (Mahalanobis distance) at each update step
- EKF SLAM takes only one-step Gauss-Newton optimization
- Gauss-Newton direction is good, but taking a full step along the direction could be disastrous
- Any optimization method (Broyden family, Levenberg-Marquardt, etc.) will do the job as long as it reduces the cost
- Inverse depth representation with conventional EKF can run into problems
- Putting the estimated landmark behind the robot (ex. Inverse depth EKF, sigma-point UKF) is really bad
- Numerical stability could be an issue, so use square root filters

Thank you all !!!

- 7+1/2 years in US has been quite a journey
- Life in Pittsburgh was great
- Thank Howie for everything
- Thank George, Steve, Brian, …
- Thank all the members of Bio-robotics Lab
- If you come to visit Korea, please let me know
- If you want to collaborate with Korean roboticists, please let me know
- hpmoon@gmail.com
- Please find me at http://mech.skku.ac.kr/