“Filtro de Kalman e Rastreamento em Seqüências de Vídeo - I"

Jorge de Jesus Gomes Leandro

Aluno de mestrado - DCC - IME - USP

Orientador: Prof. Roberto Marcondes César Junior

Co-Orientador: Prof. Luciano da Fontoura Costa – S.Carlos

http://www.vision.ime.usp.br/~createvision/

jleandro@vision.ime.usp.br

Sumário – parte I

- Introdução
  - Motivação
  - Intuição
- Estimativas Ótimas e o princípio da ortogonalidade
- Quem foi Kalman?
- Características do Filtro de Kalman
- Rastreamento:
  - Estimadores Recursivos
  - Perspectiva probabilística
- Formulação do problema e derivação simplificada
- Panorama do método
- Alguma bibliografia para derivação completa e aplicações
- O algoritmo
- Vantagens e Desvantagens
- Mais informações

Introdução

Motivação

Fins militares

- Wiener (Filtro de Wiener, 1949)
  - posicionar o canhão a fim de que o projetil atinja a aeronave com menor erro possível (II Guerra).
  - para séries temporais estacionárias
- Kalman (Filtro de Kalman, 1960)
  - extensão do filtro de Wiener para processos não-estacionários
  - prever a trajetória de mísseis balísticos cujas trajetórias, nas fases de lançamento e chegada, tem comportamento não-estacionário

Introdução

Motivação

Rastreamento em Seqüências de Vídeo

- Aplicações:
  - Segurança de patrimônio, comércio, militar, etc...
- Etapas principais:
  - Detecção de Regiões de Movimento (Proc. Imagens):
    - ✓Subtração de Fundo
      - ❖Média, Mediana, Média Móvel, Média Móvel com Seletividade, EingenBackgrounds, Mistura de Gaussianas, etc...
  - Rastreamento (estabelecer relação temporal entre objetos):
    - ✓Kalman, Kalman estendido, Filtro de Partículas (ConDensAtion), etc...

Introdução

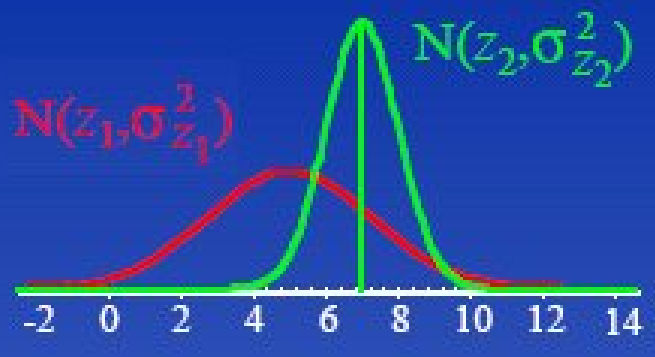
Intuição

Dois instrumentos (observadores) fornecem conjuntos de medidas Z_1 e Z_2.

$z_1, \sigma^2_{z_1}$ $z_2, \sigma^2_{z_2}$

$\hat{x}_1 = z_1$ $\hat{x}_2 = \ldots ?$

$\hat{\sigma}^2_1 = \sigma^2_{z_1}$ $\hat{\sigma}^2_2 = \ldots ?$



Introdução

Intuição

Combinar estimativas:

$$\hat{x}_2 = \left[\frac{\sigma_{z_2}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}\right] z_1 + \left[\frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}\right] z_2$$

$$= \hat{x}_1 + K_2\left[z_2 - \hat{x}_1\right]$$

Onde:

$$K_2 = \left[\frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}\right]$$

Combinar variâncias:

$$\frac{1}{\sigma_2^2} = \frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2}$$

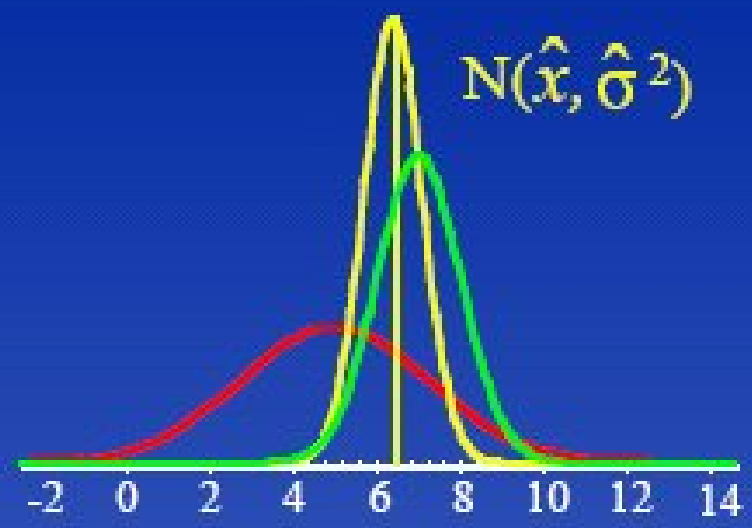
Introdução

Intuição

Combinação das Densidades Estimadas:

$$\hat{x} = \hat{x}_2$$

$$\hat{\sigma}^2 = \sigma_2^2$$



Introdução

Estimativas Ótimas e o princípio da ortogonalidade

Suponha que tenhamos o processo estocástico observável (mensurável):

$$y(k) = x(k) + v(k) \qquad (1)$$

Onde:

$x(k) \equiv$ processo desejado e desconhecido

$v(k) \equiv$ componente ruído aditivo

Seja

$\hat{x}(k)$ uma estimativa a posteriori de x(k) dados y(k), k=1,2,....

Introdução

Estimativas Ótimas: alguns conceitos

- Para obter uma estimativa ótima para $\hat{x}(k)$ precisamos de uma função custo que penalize erros para a estimativa.
- Esta função custo deve ser:
  - ✓Não negativa
  - ✓Não decrescente no erro, o qual é estimado como:

$$e(k) = x(k) - \hat{x}(k) \qquad (2)$$

- As duas restrições são satisfeitas pelo desvio quadrático médio (energia do processo erro), definido como:

$$J(k) = E\left[\left(x(k) - \hat{x}(k)\right)^2\right] = E\left[\left|e(k)\right|^2\right] \qquad (3)$$

Introdução

Estimativas Ótimas: alguns conceitos

- Seja:

$$\hat{x}(k) = \sum_{i=0}^{\infty} \omega^{*}(i) y(k-i) \qquad (4)$$

Onde:

$$\omega(i) = a(i) + jb(i)$$

Com $a(i), b(i) \in R$ e $j = \sqrt{-1}$

Introdução

Estimativas Ótimas: alguns conceitos

- Então minimizamos a função custo J(k), tomando:

$$\nabla_i J(k) = 0, i = 0,1,2,... \qquad (5)$$

Onde ∇_i é o operador gradiente complexo:

$$\nabla_i = \frac{\partial}{\partial a(i)} + j\frac{\partial}{\partial b(i)}$$

Assim, de (2), (3) e (5) vem:

$$\nabla_i J(k) = E\left\{\frac{\partial e(k)}{\partial a(i)}e^*(k) + j\frac{\partial e(k)}{\partial b(i)}e^*(k) + e(k)\frac{\partial e^*(k)}{\partial a(i)} + je(k)\frac{\partial e^*(k)}{\partial b(i)}\right\} = 0$$

(6)

Introdução

Estimativas Ótimas: alguns conceitos

Onde:

$$\frac{\partial e(k)}{\partial a(i)} = \frac{\partial}{\partial a(i)} \sum_{i=0}^{\infty} \omega^{*}(i) y(k-i) = -y(k-i)$$

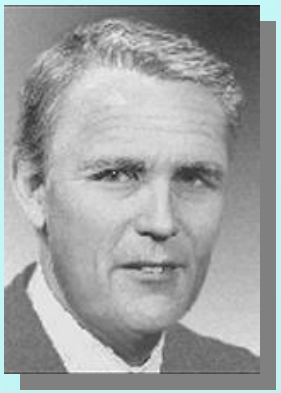
E analogamente:

$$\frac{\partial e^{*}(k)}{\partial a(i)} = -y^{*}(k-i) \qquad \frac{\partial e(k)}{\partial b(i)} = iy(k-i) \qquad \frac{\partial e^{*}(k)}{\partial b(i)} = -iy^{*}(k-i)$$

Portanto: $$\nabla_i J(k) = -2E\left\{y(k-i)e^{*}(k)\right\} = 0 \qquad i = 1,2,...$$

Logo: $$\nabla_i J(k) = E\left\{\hat{x}(k)e^{*}(k)\right\} = 0 \qquad (7)$$

Princípio da Ortogonalidade: o processo medido é ortogonal à estimativa de erro. A estimativa desejada também é ortogonal à estimativa de erro.



⌘ Quem foi Kalman?

- Rudolf Emil Kalman ...
  - ... ✵19/05/1930, Budapest - Hungary
  - ... Recebeu do MIT seu BS (1953) e MS (1954) em engenharia elétrica
  - ... Recebeu seu DSci (1955-1957) da Columbia University
  - ...Publicou um artigo seminal sobre filtragem e previsão em 1960
  - ... Prêmio *Kyoto* (1985)

⌘ Características do Filtro de Kalman – Teoria

- O **Filtro de Kalman** (KF) é ...
  - ... um **Estimador** : estima o estado de um sistema baseado no conhecimento de suas entradas e saídas.
  - ... baseado em **mínimos quadrados** : minimiza a inconsistência entre todas as informações à disposição.
  - ... baseado em **modelo** e **linear** : modelo de sistema (equação **linear** de estado) e modelo de observação (equação **linear** de medição)
  - ... **estocástico**: a confiança a respeito das informações é expressa em termos de distribuições de probabilidade
  - ... **recursivo** : se a informação é disponibilizada incrementalmente (on-line), o KF é formulado recursivamente
  - ... **ponderado** : ao minimizar a soma de mínimos quadrados, atribui pesos às informações, conforme a certeza acerca daquelas.

⌘ Rastreamento – Estimadores Recursivos

- Problema: rastrear objetos em seqüências temporais.
  - **Modelo**: sistema dinâmico estocástico, descrito por vetores de estado x e um modelo de sistema (equações)
  - **Vetor x de estado:** pode conter **variáveis dinâmicas** do modelo de sistema que descrevam a evolução temporal de x. Ex: coordenadas (x,y), orientação. Pode conter também um modelo de forma: Bezier, B-splines, etc.
  - **Hipótese**: o sistema é um **processo estocástico de Markov**: o estado x, é função somente do estado anterior x_{t-1} e sujeito à incerteza representada pelo vetor aleatório w_{t-1} (média zero).

$$x_t = f_{t-1}(x_{t-1}) + W_{t-1}$$

  - **Observações/Medições**: Um conjunto de medições z_t pode ser efetuado em cada instante t.
  - Hipótese: z_t é uma função linear do estado desconhecido x_t e está sujeito a um ruído de observação (incertezas na medição).

$$y_t = g_t(x_t) + V_t$$

  - **Hipótese:** Os vetores aleatórios e independentes W_t e V_t representam ruído branco, gaussiano, com média zero e matrizes de covariância q_t e r_t.

⌘ Rastreamento – Estimadores Recursivos

Perspectiva probabilística: Inferência Bayesiana

- **Objetivo:** estimar recursivamente a probabilidade condicional de ocorrer um estado x_t, dado um conjunto de observações $z_{1:t,}$ isto é: $p(x_t \mid z_{1:t})$
- **$p(x_t \mid z_{1:t})$** é chamada **Densidade de Filtro, Densidade de Estado, Densidade a Posteriori.**

  Conceitualmente, poderia ser obtida em duas etapas:

  - **1- Previsão:**
    - **Hipótese:** São conhecidos o estado **x_{t-1} e** a observação **z_{t-1}** no instante t-1, isto é a **Posteriori $p(x_{t-1}|z_{1:t-1})$**
    - **Hipótese:** O sistema é um **processo de Markov,** ou seja, x_t só depende de x_{t-1}, mas não de $x_{1:t-2}$,
    - Obter a **Previsão propagando a Posteriori** estimada no instante anterior **$p(x_{t-1}|z_{1:t-1})$** via **Densidade de Transição $p(x_t|x_{t-1})$,** mediante a equação de ***Chapman-Kolmogorov:***

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1}, z_{1:t-1})\, p(x_{t-1} \mid z_{1:t-1})\, dx_{t-1} = \int p(x_t \mid x_{t-1})\, p(x_{t-1} \mid z_{1:t-1})\, dx_{t-1}$$

⌘ Rastreamento – Estimadores Recursivos

Perspectiva probabilística: Inferência Bayesiana

1-PREVISÃO - Demonstração:

Supondo conhecida a **Posteriori** $\mathbf{p(x_{t-1}|z_{1:t-1})}$, estamos interessados em prever $\mathbf{p(x_t|z_{1:t-1})}$.

A probabilidade marginal de x_t, dadas as evidências $z_{1:t-1}$ é :

$$p(x_t \mid z_{1:t-1}) = \int p(x_t, x_{t-1} \mid z_{1:t-1}) dx_{t-1}$$

Aplicando a regra de Bayes na relação anterior, temos:

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1}, z_{1:t-1})\, p(x_{t-1} \mid z_{1:t-1}) dx_{t-1}$$

Assumindo que o processo estocástico em questão é Markoviano, o estado x_t independe das observações anteriores $z_{1:t-1}$ e a expressão anterior se reescreve:

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1})\, p(x_{t-1} \mid z_{1:t-1}) dx_{t-1}$$

Também conhecida como esquação de **Chapman-Kolmogorov**

⌘ Rastreamento – Estimadores Recursivos

- **Objetivo:** estimar recursivamente a probabilidade condicional de ocorrer um estado x_t, dado um conjunto de observações $z_{1:t,}$ isto é: $p(x_t \mid z_{1:t})$
  - **2- Atualização / Correção / Assimilação:**
    - Assimilar a **nova observação z_t** no instante t, **atualizando/corrigindo** a probabilidade a Priori, usando a **Regra de Bayes:**

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})}$$

Onde:

$$p(z_t \mid z_{1:t-1}) = \int p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}) dx_t$$

➢ **As equações que descrevem conceitualmente as etapas (1) e (2) do rastreamento constituem a solução bayesiana ótima, mas não podem ser resolvidas analiticamente por envolverem integrais multidimensionais**

⌘ Rastreamento – Estimadores Recursivos

Perspectiva probabilística: Inferência Bayesiana

🖂 2-Atualização - Demonstração:

No instante t uma medida z_t é obtida. Podemos atualizar a probabilidade a priori pela regra de Bayes:

$$p(x_t \mid z_{1:t}) = \frac{p(z_{1:t} \mid x_t)\, p(x_t)}{p(z_{1:t})}$$

Da independência entre as evidências (medidas) obtidas segue que:

$$p(z_{1:t} \mid x_t) = p(z_t \mid x_t).p(z_{1:t-1} \mid x_t)$$

Logo, temos:

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t)\, p(z_{1:t-1} \mid x_t)\, p(x_t)}{p(z_{1:t})}$$

Mas da regra de Bayes, sabemos que:

$$p(z_{1:t-1} \mid x_t)\, p(x_t) = p(x_t \mid z_{1:t-1}).p(z_{1:t-1})$$

⌘ Rastreamento – Estimadores Recursivos

Perspectiva probabilística: Inferência Bayesiana

2-Atualização - Demonstração:

Assim:

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1})\, p(z_{1:t-1})}{p(z_{1:t})}$$

Note que:

$$\frac{p(z_{1:t})}{p(z_{1:t-1})} = \frac{p(z_t, z_{1:t-1})}{p(z_{1:t-1})} = p(z_t \mid z_{1:t-1})$$

Portanto:

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})}$$

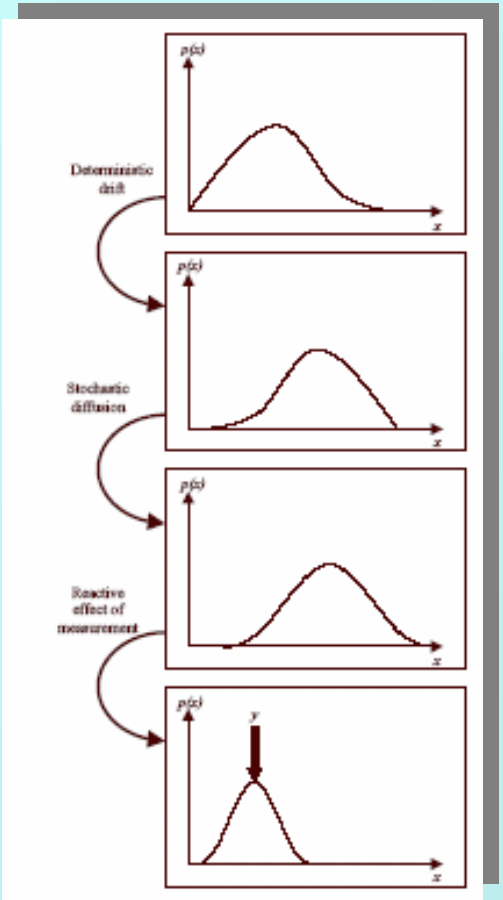
Onde a constante de normalização é:

$$p(z_t \mid z_{1:t-1}) = \int p(z_t \mid x_t).p(x_t \mid z_{1:t-1}) dx_t$$

⌘ Rastreamento – Estimadores Recursivos

Perspectiva probabilística: Inferência Bayesiana

- Filtro de Kalman visto como Propagação de Densidade de Probabilidade



⌘ Rastreamento – Estimadores Recursivos

Formulação do Problema e Derivação simplificada

- Problema: **estimar** a densidade a **posteriori** $p(x_t|z_{1:t})$
  - **Filtros Bayesianos Recursivos:** da Teoria de Controle, aproximação recorrente da densidade a posteriori
  - **Filtro de Kalman:** é o estimador recursivo em que a densidade a **posteriori** $p(x_t|z_{1:t})$ é modelada por uma distribuição unimodal gaussiana, a ser propagada no tempo. Sua média é o estado mais provável e variância a incerteza.
  - **Hipóteses**
    - Não há variações bruscas de movimentos em quadros subseqüentes
    - O modelo de sistema (equações) pode ser escrito matricialmente como

$$x_t = A_{t-1}x_{t-1} + w_{t-1} \qquad (1)$$

      - A_{t-1}: transição de estado do instante t-1 para t
      - W_{t-1}: incerteza associada aos erros de previsão do sistema (gaussiana com média zero e covariância q_{t-1})
    - A observável z_t é uma transformação linear do vetor de estado desconhecido x_t:

$$y_t = H_t x_t + v_t \qquad (2)$$

      - H_t: matriz de sensibilidade de medições
      - V_t: incerteza associada aos ruídos no medição (gaussiana com média zero e covariância r_{t-1})

⌘ Rastreamento – Estimadores Recursivos

Formulação do Problema e Derivação simplificada

- **Previsão:**
  - Suponha que uma medida em um sistema linear dinâmico descrita pelas eqs (1) e (2) foi realizada no instante t.
  - Usar a informação na nova medida em t para atualizar a estimativa x_t.
  - A estimativa a posteriori pode ser expressa como uma combinação linear da estimativa a priori e da nova medida:

$$\hat{x}_t = K_t^{(1)} \hat{x}_t^- + K_t y_t \qquad (3)$$

  - E o vetor de estado erro é definido como:

$$e_t = x_t - \hat{x}_t \qquad (4)$$

  - Aplicando o princípio da ortogonalidade, podemos escrever:

$$E[e_t \hat{y}_i^T] = 0 \qquad i = 1,2,...,t-1 \qquad (5)$$

⌘ Rastreamento – Estimadores Recursivos

Formulação do Problema e Derivação simplificada

- **Previsão:**
  - Usando as equações (2), (3), (4) e (5), aplicando o princípio da ortogonalidade sempre que se fizer necessário e fazendo simplificações, obtém-se:

$$\hat{x}_t = \hat{x}_t^- + K_t(y_t - H_t \hat{x}_t^-) = (1 - K_t H_t)\hat{x}_t^- + K_t y_t \qquad (6)$$

  - A matriz K_t é chamada **Ganho de Kalman**.
  - Para encontrar o **Ganho de Kalman**, partimos do **princípio da ortogonalidade,** que também vale para o desvio na medida:

$$E[e_t \tilde{y}_i^T] = 0, i = 1,2,...,t-1 \qquad (7)$$

  - O desvio na medida também é chamado de inovação no processo

$$\tilde{y}_t = y_t - \hat{y}_t = H_t x_t + v_t - H_t \hat{x}_t^- = v_t + H_t e_t^- \qquad (8)$$

⌘ Rastreamento – Estimadores Recursivos

Formulação do Problema e Derivação simplificada

Previsão:

- Usando as equações (2) e (6) podemos reescrever o desvio:

$$e_t = (I - K_t H_t) e_t^- - K_t v_t \qquad (10)$$

- Substituindo (8) e (10) em (7) temos:

$$E[\{(I - K_t H_t) e_t^- - K_t v_t\}(H_t e_t^- + v_t)] = 0, i = 1,2,...,t-1 \qquad (11)$$

- Considerando a independência entre o ruído de medida e o desvio e definindo a matriz de covariância a priori:

$$P_t^- = E[e_t e_t^{-T}] \qquad (12)$$

- A (11) se torna:

$$(I - K_t H_t) P_t^- H_t^T - K_t R_t = 0 \qquad (13)$$

- Que resolvida para o ganho de Kalman K_t, resulta

$$K_t = P_t^- H_t^T \left[H_t P_t^- H_t^T + R_t\right]^{-1} \qquad (14)$$

- Onde R_t é a matriz de covariância de v_t

⌘ Rastreamento – Estimadores Recursivos

Formulação do Problema e Derivação simplificada

- **Atualização:**
  - A eq. (14) para o Ganho de Kalman é definida em termos da matriz de covariância a priori. Consideremos uma propagação de covariância, refletindo o efeito do tempo sobre as matrizes de covariância de estimação de erro. Esta propagação se dá em 2 etapas:
    - Calcular a matriz de covariância a postgriori P_k dada a matriz de covariância a priori P_k^- , definida como:

$$P_t = E[e_t e_t^T] \quad (15)$$

    - Usando (10) , (15) e (14), vem:

$$P_t = (I - K_t H_t) P_t^- \quad (16)$$

  - Usando a equação (1) e uma análoga para a estimativa a priori, podemos estimar o erro a priori como:

$$e_t^- = x_t - \hat{x}_t^- = A_{t,t-1} e_{t-1} + w_{t-1} \quad (17)$$

  - Usando a definição (12) e notando a independência entre o ruído w_t e estimativa:

$$P_t^- = A_{t,t-1} P_{t-1} A_{t,t-1}^T + Q_{t-1} \quad (18)$$

  - Onde R_t é a matriz de covariância de v_t

⌘ Rastreamento – Estimadores Recursivos

Formulação do Problema e Derivação simplificada

- **Inicialização:**
  - Na ausência de medições no instante t=0, a estimativa é:

$$x_0 = E[x_0]$$

  - E a matriz de covariância a posteriori inicial:

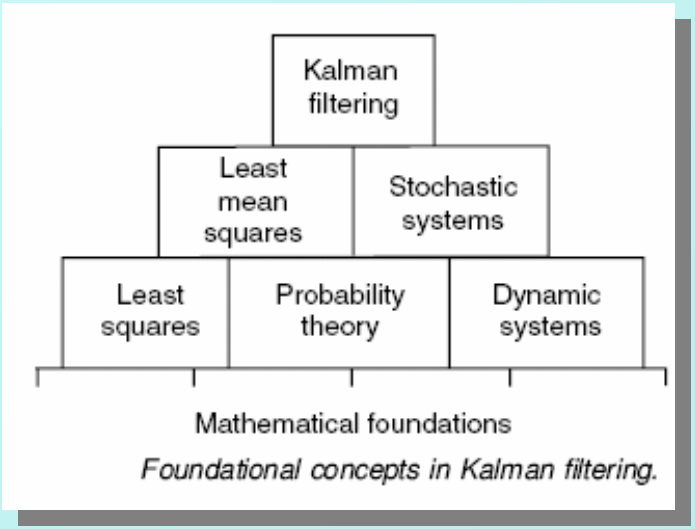
$$P_t = E\left[(x_0 - E[x_0])(x_0 - E[x_0])^T\right]$$

⌘ Rastreamento (Filtro de Kalman – Teoria)

Panorama do Método:

- **Modelos:** forma e movimento (sistema)
- **Etapas por instante t**
  - Previsão: onde o objeto deveria estar
  - Observação/Medição: observar prováveis destinos do objeto
  - Correção/Atualização/Assimilação: atualizar o modelo e estado do objeto combinando as informações de previsão e medição.

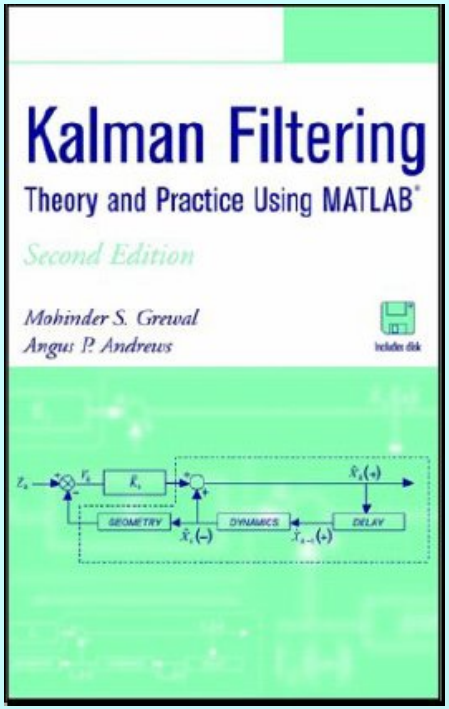
Fundamentos Teóricos:

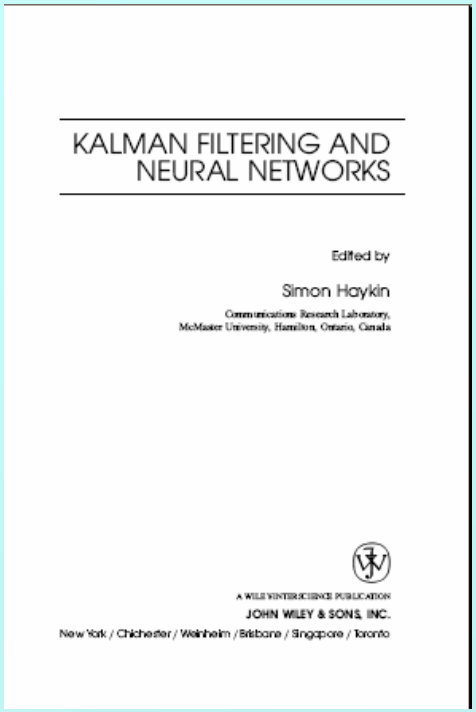


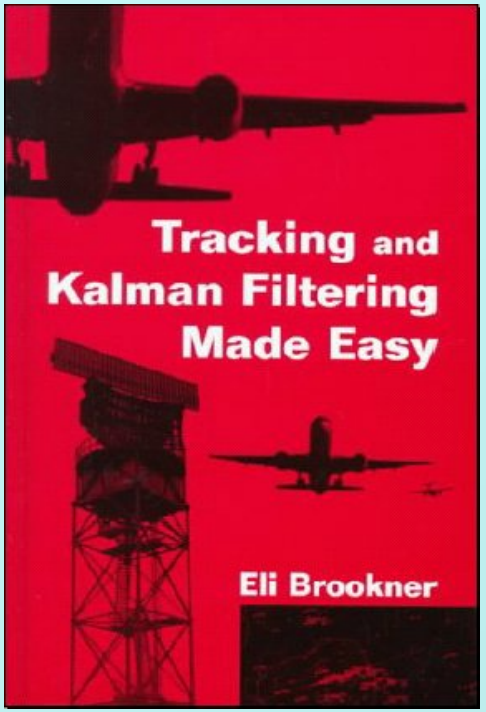
Foundational concepts in Kalman filtering.

⌘ Rastreamento – Filtro de Kalman

- **Alguma Bibliografia para Derivação completa e aplicações:**

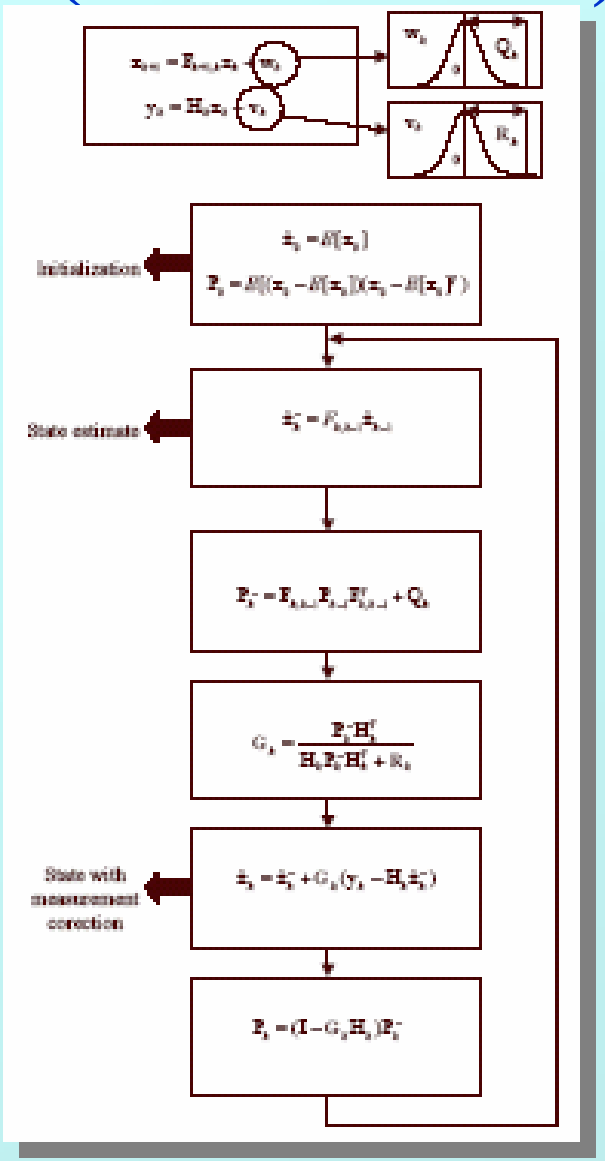






⌘ **Rastreamento** (Filtro de Kalman) - algoritmo

- O Algoritmo



⌘ **Rastreamento** (Filtro de Kalman)

- Vantagens
  - Computacionalmente eficiente (recursivo)
- Desvantagens
  - Falha quando objeto desaparece repentinamente, reaparecendo posteriormente (oclusão)

Mais informaçoes:

- http://www.vision.ime.usp.br/~creativision/
- http://www.cs.unc.edu/~welch/kalman/

“Filtro de Kalman e Rastreamento em Seqüências de Vídeo - II"

Jorge de Jesus Gomes Leandro

Aluno de mestrado - DCC - IME - USP

Orientador: Prof. Roberto Marcondes César Junior

Co-Orientador: Prof. Luciano da Fontoura Costa – S.Carlos

http://www.vision.ime.usp.br/~createvision/

jleandro@vision.ime.usp.br

Sumário – parte II

- Introdução
  - Revisão – Parte I
    - Hipóteses
    - Algoritmo – Recursão
    - Equações
- Aplicações
  - Exemplo – Rastreamento de Veículos
- Limitações
- Implementações
  - OpenCV
  - Matlab
- Bibliografia

Introdução

- **Rastreamento – Estimadores Recursivos**

Hipóteses

- Problema: **estimar** a densidade a **posteriori** $p(x_t|z_{1:t})$
  - **Filtros Bayesianos Recursivos:** da Teoria de Controle, aproximação recorrente da densidade a posteriori
  - **Filtro de Kalman:** é o estimador recursivo em que a densidade a **posteriori** $p(x_t|z_{1:t})$ é modelada por uma distribuição unimodal gaussiana, a ser propagada no tempo. Sua média é o estado mais provável e variância a incerteza.
  - **Hipóteses**
    - Não há variações bruscas de movimentos em quadros subseqüentes
    - O modelo de sistema (equações) pode ser escrito matricialmente como

$$x_t = A_{t-1}x_{t-1} + w_{t-1} \qquad (1)$$

      - A_{t-1}: transição de estado do instante t-1 para t
      - W_{t-1}: incerteza associada aos erros de previsão do sistema (gaussiana com média zero e covariância q_{t-1})
    - A observável z_t é uma transformação linear do vetor de estado desconhecido x_t:

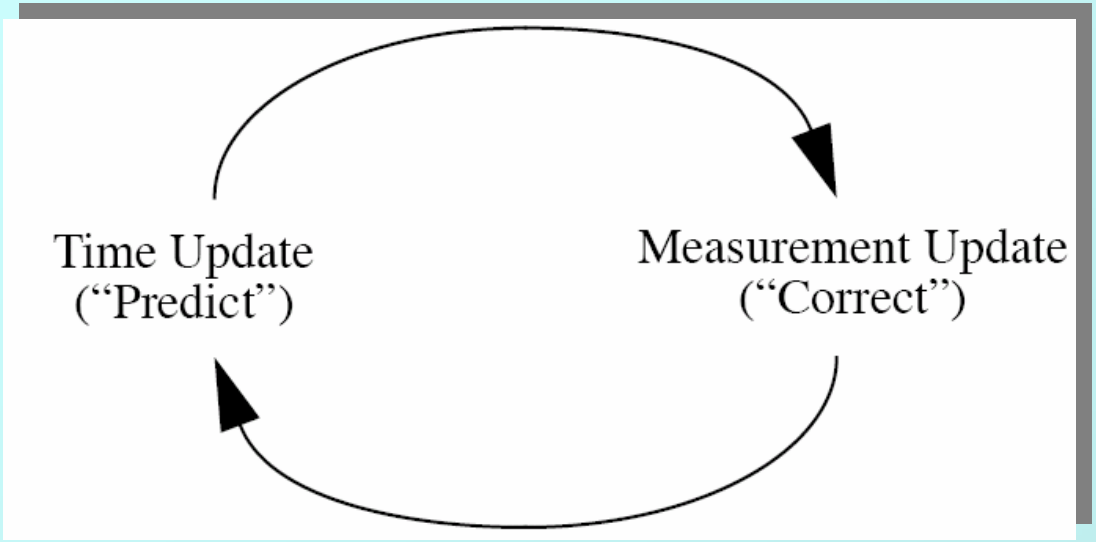
$$y_t = H_t x_t + v_t \qquad (2)$$

      - H_t: matriz de sensibilidade de medições
      - V_t: incerteza associada aos ruídos no medição (gaussiana com média zero e covariância r_{t-1})

Introdução

- Filtro de Kalman - Algoritmo

  - O Ciclo



A atualização temporal projeta o estado corrente para o instante posterior. A atualização por observação ajusta a estimativa projetada por uma medida real naquele instante de tempo.

Introdução

⌘ Filtro de Kalman - Algoritmo

➢Recursão

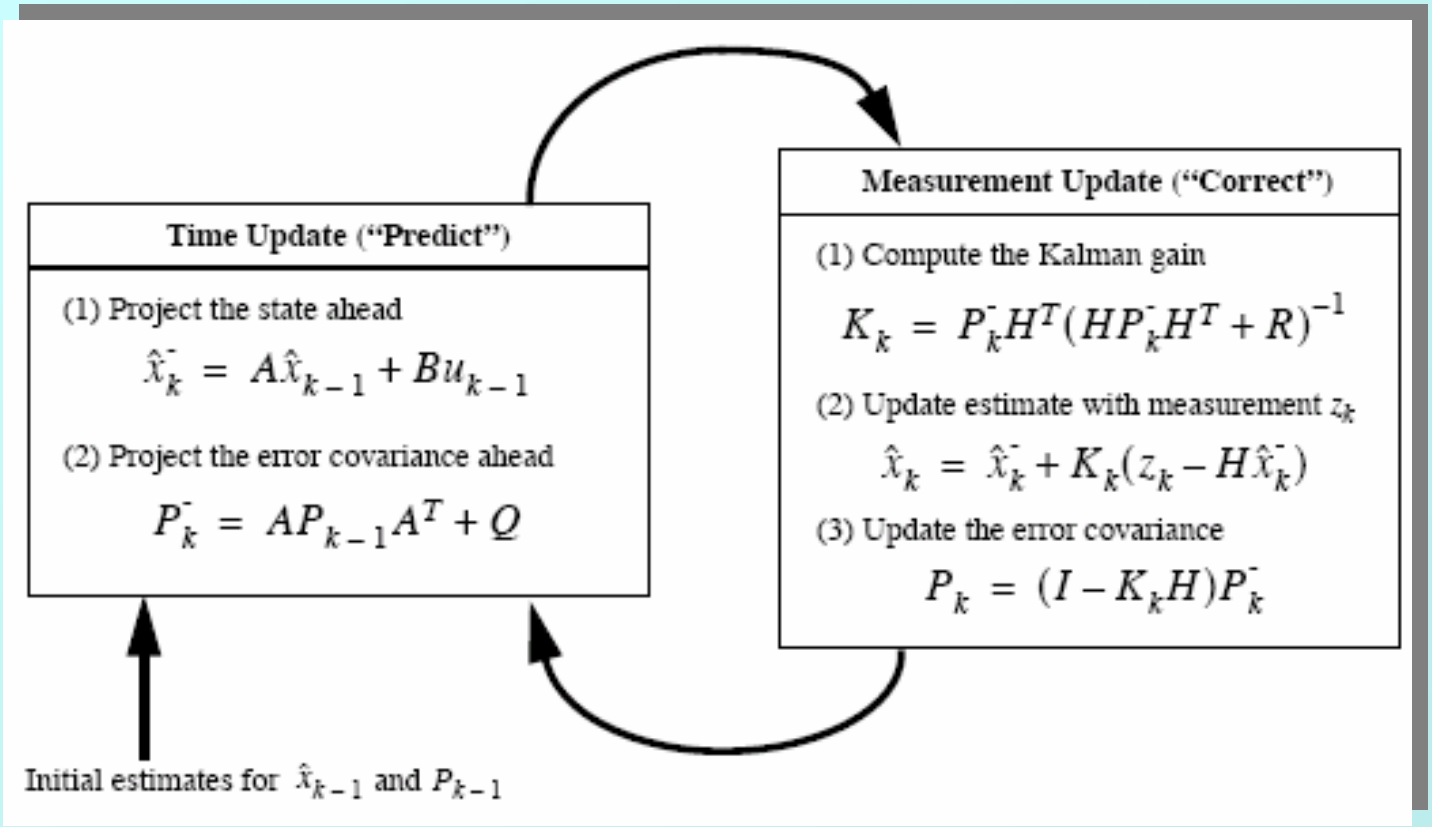


Imagem completa da operação do Filtro de Kalman

Introdução

- Filtro de Kalman - Equações

$x'_t = A_{t-1}\, x_{t-1}$ previsão do estado

$p'_t = A_{t-1}\, P_{t-1}\, A^T_{t-1} + q_{t-1}$ previsão da covariância do estado

$K_t = p'_t H_t^T (H_t p_t' t H_t^T + r_t)^{-1}$ ganho de Kalman

$x_t = A_{t-1}\, x_{t-1} + K_t(z_t - H_t A_{t-1}\, x_{t-1})$ estado final

$p_t = \mathrm{cov}(x_t) = (I - K_t)\, p'_t\ (I - K_t)^T + K_t r_t K_t^T$ covariância do estado final

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- Problema: rastrear veículos numa estrada
  - Hipótese: o veículo se move a uma velocidade quase constante.
  - Hipótese: sabemos medir a posição de uma característica do veículo que queremos rastrear em cada frame.

Aplicações

- Exemplo : Rastreamento de Veículos

  - Problema: rastrear veículos numa estrada
    - Assumimos que o veículo se move com velocidade constante
      - Nosso estado é composto pela posição do veículo e sua velocidade
      - Em cada instante de tempo a velocidade incrementa a posição

Aplicações

- Equações de Atualização de Estado

$$x_t = x_{t-1} + \dot{x}_{t-1}$$

$$y_t = y_{t-1} + \dot{y}_{t-1}$$

$$\dot{x}_t = \dot{x}_{t-1}$$

$$\dot{y}_t = \dot{y}_{t-1}$$

$$s = \left[x, y, \dot{x}, \dot{y} \right]^T$$

Matricialmente

$$\begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{bmatrix}$$

$$s_t = A s_{t-1}$$

Aplicações

- Exemplo : Rastreamento de Veículos

  - Problema: rastrear veículos numa estrada

    - Em cada instante de tempo, podemos detectar características numa imagem

      - As quais compõem nossas medidas z_t

      - Podemos medir diretamente a posição do veículo, mas não sua velocidade

Aplicações

- Exemplo : Rastreamento de Veículos

  - Equações de Medição

$$z_t = [x, y]^T$$

Matricialmente

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \bullet \\ x_{t-1} \\ \bullet \\ y_{t-1} \end{bmatrix}$$

$$z_t = As_t$$

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- Estimativa Inicial
  - ✓Atribuimos um valor grosseiro para x e y como informação de qual característica estamos rastreando
  - ✓Não sabemos nada sobre $\dot{x}$ e $\dot{y}$
  - ✓Então usamos

$$s_0 = [100,170,0,0]^T$$

- Também precisamos fornecer a incerteza
  - ✓Nossa estimativa de posição é boa dentro de alguns pixels
  - ✓Nossa estimativa de movimento não é boa, mas esperamos que o deslocamento seja pequeno
  - ✓Representamos esta informação como uma matriz de covariâncias

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- ➢ Matrizes de Covariância
  - ✓ O que é uma matriz de covariância mesmo?
    - ✓ Informa a relação entre conjuntos de variáveis

$$\mathrm{var}(x) = E[(x - \bar{x})^2]$$

    - ✓ Covariância entre duas variáveis x e y :

$$\mathrm{cov}(x, y) = E[(x - \bar{x})(y - \bar{y})]$$

- ➢ Dado um vetor de variáveis

$$x = [x_1, x_2, ..., x_k]$$

  - ✓ A covariância C é uma matriz k x k
  - ✓ O elemento i,j de C é : $C_{i,j} = \mathrm{cov}(x, y)$

Aplicações

- Exemplo : Rastreamento de Veículos

  - Cont.

    - ✓Um elemento da diagonal $C_{i,i}$ fornece a variância na variável x_i
    - ✓C é simétrica

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- Covariância no Ruído
  - ✓Os termos de ruído w e v precisam ser estimados
    - ❖Eles têm média zero, e covariância Q e R respectivamente
    - ❖Precisamos estimar estas matrizes
    - ❖Q e R informam quão certos estamos do nosso modelo
  - ✓Para estimar Q
    - ❖Nossa estimativa inicial será dentro de poucos pixels, por exemplo $\sigma = 3$
    - ❖A velocidade é um pouco menos certa, mas não será muito grande, por exemplo $\sigma = 5$
    - ❖Não há nenhuma razão para pensar que os erros estão relacionados, assim os termos de covariância serão zero.

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- Covariância Inicial
  - ✓As variâncias em x e y são $3^2 = 9$
  - ✓As variâncias em $\dot{x}$ e $\dot{y}$ são $5^2 = 25$
  - ✓Já que assumimos independência, os elementos fora da diagonal principal serão nulos

$$P_0 = \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

Aplicações

- Exemplo : Rastreamento de Veículos

  - Incerteza no Modelo

    - ✓ As equações do modelo possuem termos de ruído

      - ❖ w representa o fato de nosso modelo de atualização de estado poder não ser acurado
      - ❖ v representa o fato de que as medidas serão sempre ruidosas
      - ❖ Precisamos estimar suas covariâncias

    - ✓ Em geral

      - ❖ Quase sempre os termos são independentes. Se for o caso, os elementos fora da diagonal principal serão nulos.
      - ❖ Escolher os valores da diagonal principal é mais difícil

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- Atualização da Covariância do Estado
  - ✓A equação de atualização de estado não é perfeita
    - ❖Ela assume que o movimento é constante, mas $\dot{x}$ e $\dot{y}$ poderiam variar com o tempo
    - ❖Ela assume que todo o movimento é representado por $\dot{x}$ e $\dot{y}$, mas outros fatores poderiam afetar x e y.
  - ✓Estes erros provavelmente serão pequenos
    - ❖O movimento é lento e bem suave
    - ❖Então a variância nestes termos é provavelmente um pixel ou menos, digamos $\sigma = 0.5$

$$Q = \begin{bmatrix} 0{,}25 & 0 & 0 & 0 \\ 0 & 0{,}25 & 0 & 0 \\ 0 & 0 & 0{,}25 & 0 \\ 0 & 0 & 0 & 0{,}25 \end{bmatrix}$$

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- ➢ Atualização da Covariância do Estado
  - ✓ As medidas que obtemos serão ruidosas
    - ❖ As características estão localizadas somente no pixel mais próximo
    - ❖ Devido ao ruído na imagem, *aliasing*, etc, elas podem ser deslocadas por um pixel ou dois.
  - ✓ Estes erros são um pouco mais fáceis de estimar
    - ❖ Provavelmente a característica está no lugar certo, ou um pixel deslocado
    - ❖ Então a variância nestes termos é provavelmente $\sigma^2 = 1$

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

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- ➢ Previsão de Estado
  - ✓ Podemos agora aplicar o filtro
    - ❖ Primeiramente, fazemos uma previsão do estado em t=1, baseados em nossa estimativa inicial em t=0

$$s_1^- = As_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix}$$

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- Previsão da Covariância

$$P_1^- = AP_0A^T + Q =$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0{,}25 & 0 & 0 & 0 \\ 0 & 0{,}25 & 0 & 0 \\ 0 & 0 & 0{,}25 & 0 \\ 0 & 0 & 0 & 0{,}25 \end{bmatrix} =$$

$$= \begin{bmatrix} 34{,}25 & 0 & 25 & 0 \\ 0 & 34{,}35 & 0 & 25 \\ 25 & 0 & 35{,}25 & 0 \\ 0 & 25 & 0 & 25{,}25 \end{bmatrix}$$

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- Obtendo uma medida
  - ✓A previsão de estado nos dá uma idéia de onde a característica pode estar
    - ❖Esperamos que ela esteja próxima a (100,170)
    - ❖A variância na coordenada de posição x é 34.25
    - ❖A variância na coordenada de posição y também é 34.25
  - ✓Podemos usar esta informação para restringir nossa busca pela característica
    - ❖Temos 95% de certeza que a característica cai dentro de um círculo de raio 2σ, centrado na previsão (seria uma elipse se as vars fossem diferentes)

$$\sigma = \sqrt{34.25} \approx 5.85$$

    - ❖Procuramos pela característica nesta região

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- Obtendo uma medida
  - ✓Dentro da região de interesse (neste caso, um círculo)
    - ❖Calculamos um valor que nos diga a probabilidade de cada ponto ser uma característica
    - ❖Encontramos o ponto com maior valor dentro da região
    - ❖Isto é

$$z_t = [103,163]^T$$

Procuramos por uma característica próxima de nosso valor previsto, e as covariâncias dizem quão ampla pode ser essa busca

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- ➢ O Ganho de Kalman
  - ✓ Agora combinamos a previsão e a medida
    - ❖ Calculamos a matriz do Ganho de Kalman
    - ❖ Que leva em conta a certeza relativa que temos quanto às duas informações

$$K_1 = P_1^- H^T \left(H P_1^- H^T + R \right)^{-1} =$$

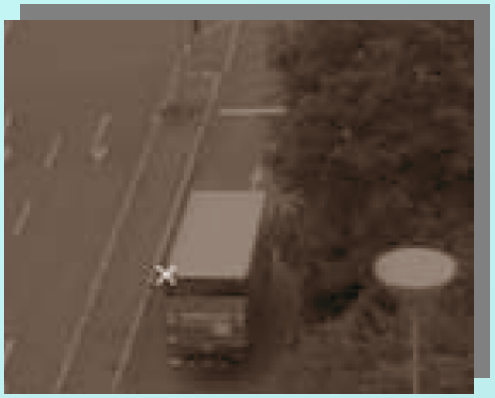
$$= \begin{bmatrix} 0.972 & 0 \\ 0 & 0.972 \\ 0.709 & 0 \\ 0 & 0.709 \end{bmatrix}$$

    - ❖ Os primeiros componentes estão bem próximos de 1, o que atribui mais confiança (maior peso) à medida

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- A Estimativa Final
  - ✓Podemos agora realizar uma estimativa final
    - ❖Combinamos a previsão e a medida
  - ✓Também calculamos a covariância desta estimativa
    - ❖Isto pode ser usado para nos informar até onde podemos confiar nesta estimativa
    - ❖Também é usada para fazer uma previsão para o próximo frame



Aplicações

- Exemplo : Rastreamento de Veículos

  - Estimativa de estado

$$s_1 = s_1^- + K_1\left(z_1 - Hs_1^-\right)$$

$$\approx \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0{,}972 & 0 \\ 0 & 0{,}972 \\ 0{,}709 & 0 \\ 0 & 0{,}709 \end{bmatrix} * \left\{ \begin{bmatrix} 103 \\ 163 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix} \right\} \approx \begin{bmatrix} 102{,}9 \\ 163{,}2 \\ 2{,}13 \\ -4{,}96 \end{bmatrix}$$

Aplicações

- Exemplo : Rastreamento de Veículos

  - Covariância de Estado

$$P_1 = P_1^- + K_1 H P_1^- =$$

$$\approx \begin{bmatrix} 34{,}25 & 0 & 25 & 0 \\ 0 & 34{,}25 & 0 & 25 \\ 25 & 0 & 25{,}25 & 0 \\ 0 & 25 & 0 & 25{,}25 \end{bmatrix} + \begin{bmatrix} 0{,}972 & 0 \\ 0 & 0{,}972 \\ 0{,}709 & 0 \\ 0 & 0{,}709 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 34{,}25 & 0 & 25 & 0 \\ 0 & 34{,}25 & 0 & 25 \\ 25 & 0 & 25{,}25 & 0 \\ 0 & 25 & 0 & 25{,}25 \end{bmatrix} \approx$$

$$\approx \begin{bmatrix} 0{,}971 & 0 & 0{,}71 & 0 \\ 0 & 0{,}971 & 0 & 0{,}71 \\ 0{,}71 & 0 & 7{,}52 & 0 \\ 0 & 0{,}71 & 0 & 7{,}52 \end{bmatrix}$$

Aplicações

⌘ Exemplo : Rastreamento de Veículos

- ➢ Iteração
  - ✓ Repetimos este cálculo para cada frame
    - ❖ Com o passar do tempo, as previsões de estado se tornam mais acuradas
    - ❖ O Ganho de Kalman leva isto em consideração e atribui maior peso às previsões
  - ✓ Para implementar o filtro de Kalman
    - ❖ Precisamos de um conjunto de rotinas para manipulação de matrizes
    - ❖ Muito trabalhoso, mas há diversas bibliotecas disponíveis
    - ❖ São necessárias somente as operações básicas: +, -, *, transposição e inversão.

Limitações

- Filtro de Kalman

  - o estado de cada objeto rastreado é modelado por uma única distribuição unimodal gaussiana. O que fazer quando a densidade de um estado for multimodal e não gaussiano, por exemplo, o objeto pode ser encontrado em dois pontos, mas não entre eles?
  - muito sensível a ruído originado pelo fundo (desorganizado/não-uniforme) e oclusões.
  - o modelo de movimento é suposto linear. O que fazer quando o movimento não for linear?

Limitações

- Filtro de Kalman
  - usar Filtro de Partículas (ConDensAtion e outros)
  - Usar Filtro Estendido de Kalman
    - ✓Aproximação linear para uma função não-linear (expansão em série de potências – MacLaurin)
    - ✓Depende da acurácia da aproximação
    - ✓Não há provas formais, mas funciona bem na prática

Implementações

- **OpenCV** (Open Source Computer Vision Library)
  - biblioteca aberta de Visão Computacional da Intel para C/C++
  - Implementa o Filtro de Kalman em quatro funções:
    1. cvCreateKalman
    2. cvReleaseKalman
    3. cvKalmanUpdateByTime ou cvKalmanPredict
    4. cvKalmanUpdateByMeasurement ou cvKalmanCorrect

Implementações

- **OpenCV** (Open Source Computer Vision Library)

1. CvKalman cvCreateKalman (int DynamParams, int MeasureParams, int controlParams=0);

- É a primeira função a ser usada quando o filtro for iniciado.
- Recebe:
  - ✓ um inteiro DynamParams para a dimensão (número de variáveis dinâmicas) do vetor de estados x_t.
  - ✓um inteiro MeasureParams para a dimensão (número de características extraídas: coordenadas, área, bordas, cor,...) do vetor de observações z_t e
  - ✓um inteiro controlParams para número de parâmetros externos de controle (alguma força, como viscosidade, por exemplo).
- Devolve:
  - ✓ um ponteiro para uma estrutura CvKalman, contendo informações sobre os estados, as observações, e seus parâmetros.

Implementações

⌘ **OpenCV** (Open Source Computer Vision Library)

2. **void cvReleaseKalman (CvKalman ** kalman)**
   - ➢ Usada quando todo o trabalho estiver concluído.
   - ➢ Recebe:
     - ✓ um ponteiro duplo para a estrutura CvKalman que deve ser destruída, liberando a memória a ela alocada.

3. **void cvKalmanUpdateByTime (CvKalman * kalman)** ou
   const CvMat* cvKalmanPredict(CvKalman* kalman, const CvMat* control=NULL);

```
#define cvKalmanUpdateByTime cvKalmanPredict
```

   - ➢ Recebe:
     - ✓ um ponteiro para a estrutura CvKalman, cujo estado queremos projetar no futuro.
   - ➢ Implementa:
     - ✓ a etapa da **Previsão** de um estado x'_t, considerando apenas o modelo dinâmico (movimento) adotado.

Implementações

- **OpenCV** (Open Source Computer Vision Library)

4. **void cvKalmanUpdateByMeasurement (CvKalman * kalman)** ou
void cvKalmanCorrect(CvKalman* kalman, const CvMat* measurement=NULL);
#define cvKalmanUpdateByMeasurement cvKalmanCorrect

- Recebe:
  - ✓ um ponteiro para a estrutura CvKalman que deve ser atualizada/corrigida, assimilando as observações/medições, já efetuadas de alguma maneira, da etapa de **Observação/Medição** em algum ponto do código anterior à chamada desta função.
- Implementa:
  - ✓ a etapa da **Atualização/Correção/Assimilação**, assimilando uma observação/medição z_t para atualizar/corrigir a previsão x'_t.

Implementações

⌘ Matlab

- Kalman Filter Toolbox for Matlab (free)
  - ➢ http://www.cs.ubc.ca/~murphyk/Software/Kalman/kalman.html
- The KalmTool Toolbox – Version 2 (free)
  - ➢ http://www.iau.dtu.dk/research/control/kalmtool.html
- M-KFTOOL 2.01 – Biblioteca para Matlab - ~U$ 500,00

Implementações

⌘ Matlab

- ReBEL: Recursive Bayesian Estimation Library (free)
  - ➢ http://choosh.ece.ogi.edu/rebel/
    - ✓ **Kalman filter**
    - ✓ **Extended Kalman filter**
    - ✓ **Sigma-Point Kalman filters (SPKF)**
      - ❖ Unscented Kalman filter (UKF)
      - ❖ Central difference Kalman filter (CDKF)
      - ❖ Square-root SPKFs
      - ❖ Gaussian mixture SPKFs
      - ❖ Iterated SPKF
    - ✓ **Particle filters**
      - ❖ Generic particle filter
      - ❖ Gaussian sum particle filter
      - ❖ Sigma-point particle filter
      - ❖ Gaussian mixture sigma-point particle filter

Bibliografia

1. D. Fox, J. Hightower, H. Kautz, L. Liao, and D. Patterson. ***Bayesian techniques for location estimation***. In Proceedings of The 2003 Workshop on Location-Aware Computing
2. Bob Fisher **– *Target Tracking with Noise and Bouncing*** - Lectures on Advanced Vision – Lecture 8 – School of Informatics – University of Edingburgh
3. Ng Teck Khim – ***CS5245 "Computer Vision and Graphics for Special Effects"*** – Match Move Lectures Notes
4. Tony Pridmore and Steve Mills – ***G5BVIS Computer Vision – Lecture 10 : Motion – The Kalman Filter / Lecture 11:Motion – More on the Kalman Filter –*** Image Processing and Interpretation – University of Nottingham
5. A W Krings - ***Fault-Tolerant Systems – Lecture 10: Markov Process*** – University of Idaho
6. Thomas B. Moeslund and Erik Granum – ***A Survey of Computer Vision-Based Human Motion Capture*** – Computer Vision and Image Understanding 81, 231-268 (2001) – Academic Press
7. **Wikipedia**, the free encyclopedia. (http://en.wikipedia.org/wiki/)
8. Intel - ***Open Source Computer Vision Library Reference Manual –*** 1999-2001- Intel Corporation
9. Greg Welch and Gary Bishop – ***An Introduction to the Kalman Filter*** - Technical Report: TR95-041 (1995)
10. Rudolph Kalman – ***A New Approach to Linear Filtering and Prediction Problems*** – Transactions of the ASME – Journal of Basic Engineering 82, D – 35-45 (1960)
11. **MathWorld** Wolfram Research – (http://mathworld.wolfram.com)
12. Curso de Visão Computacional - **SCHOOL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY** -Tony Pridmore - http://www.cs.nott.ac.uk/~tpp/