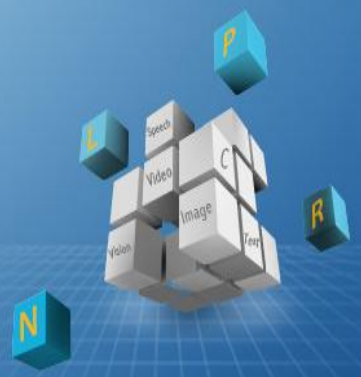
视觉跟踪

中国科学院自动化研究所
模式识别国家重点实验室
董秋雷



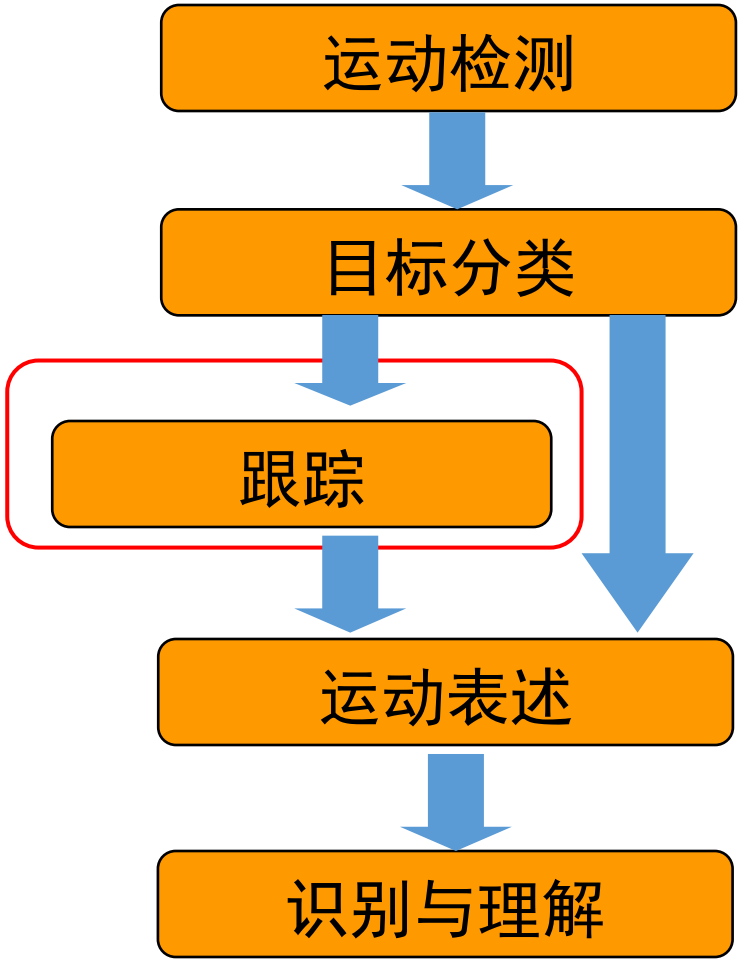
1 背景内容

2 目标跟踪

3 视觉定位

4 小节

运动分析的一般流程

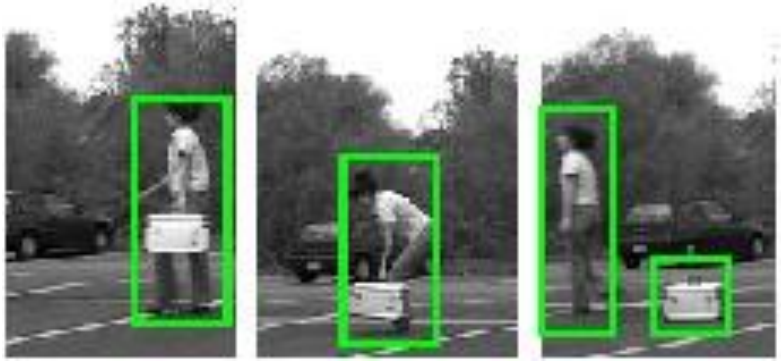


什么是跟踪(Tracking)？

- Tracking
  - 目标跟踪：在图像序列中持续地估计出感兴趣的运动目标所在区域（位置），形成运动目标的运动轨迹；有时还需要估计出运动目标的某些运动参数（比如速度、加速度等）。
  - 相机跟踪（摄像机定位）：通过图像序列，持续地计算出相机的位置、姿态，如SLAM（Simultaneous Localization And Mapping，同步定位与地图创建）。

目标跟踪问题分类

- 场景中运动目标的数目： 单运动目标 vs. 多运动目标
  - 在多目标跟踪过程中，必须考虑到多个目标在场景中会互相遮挡(Occlusion)，合并(Merge)，分离(Split)等情况。
  - 多目标跟踪中的数据关联问题(Data Association)。



目标跟踪问题分类

- 摄像机的数目： 单摄像机 vs. 多摄像机
  - 多摄像机有望解决因相互遮挡导致的运动目标丢失问题，但多摄像机的信息融合是一个关键性问题。
- 摄像机是否运动： 摄像机静止 vs. 摄像机运动
  - 摄像机的运动形式，一种是摄像机支架固定，摄像机可以偏转(Pan)，俯仰(Tilt)以及缩放(Zoom)；另一种是摄像机装在移动载体上，如车辆、飞机。
  - 摄像机的运动增加了运动目标检测的难度。

目标跟踪问题分类

- 场景中运动目标的类型： 刚体 vs. 非刚体
  - 交通车辆—刚体； 人—非刚体。
- 传感器的种类： 可见光图像 vs. 红外图像
  - 白天使用可见光图像；晚上使用红外图像。

1 背景内容

2 目标跟踪

3 视觉定位

4 小节

目标跟踪

① 运动目标的表示方法

② 传统目标跟踪方法

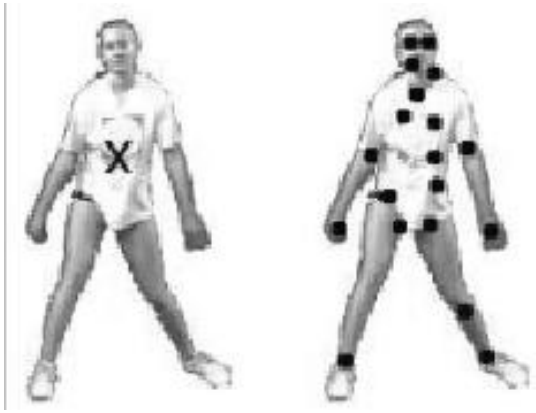
③ 基于DNN的跟踪方法

运动目标的表示方法

- 基于点的跟踪
- 基于区域的跟踪
- 基于轮廓的跟踪
- 基于模型的跟踪

基于点的跟踪

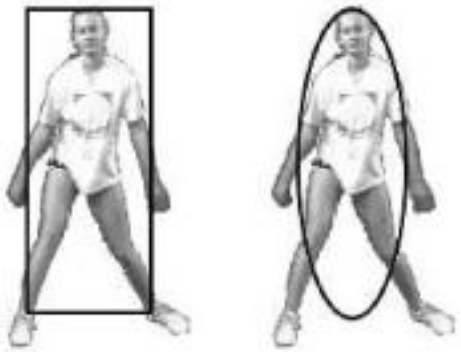
- 质心或一组特征点集



- 运动轮廓的角点

基于区域的跟踪

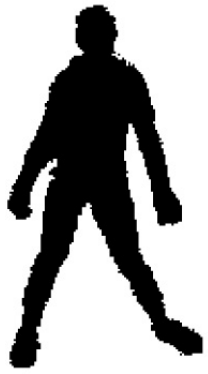
- 将运动目标用比较简单的几何形状表示，比如矩形或椭圆等



- 适合于表示简单的刚体或非刚体运动目标。
- 相较于后面要介绍的活动轮廓等表示方法精度较差。

基于轮廓的跟踪

Contour

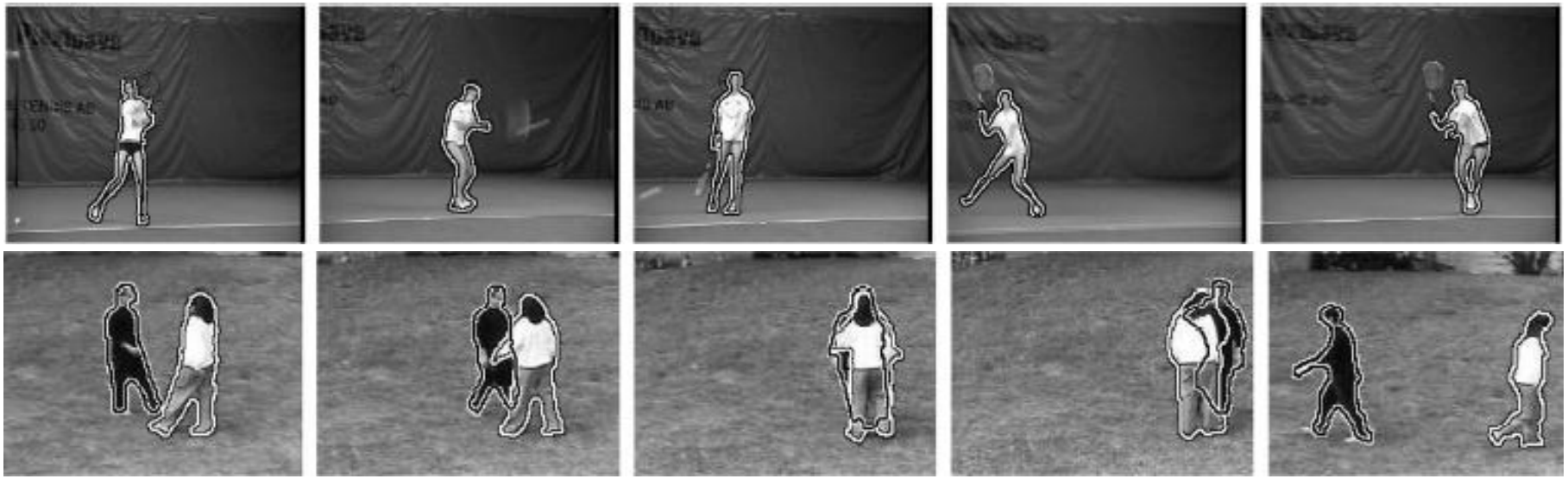


Silhouette

- Contour表示运动目标的外部轮廓
- Silhouette表示运动目标外部轮廓内的区域
- 适用于表示复杂的非刚体运动目标

基于轮廓的跟踪

- 主动轮廓 Active Contour
- 利用封闭的曲线轮廓来表示运动目标，并且该轮廓能够自动连续地更新



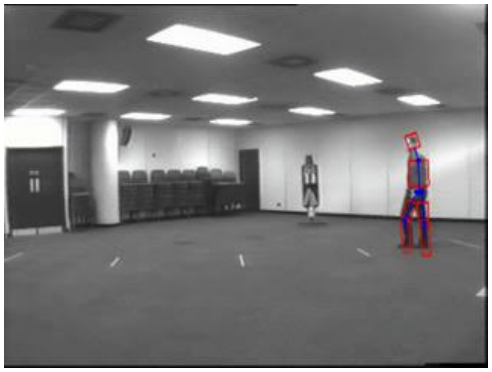
* YILMAZ, A., LI, X., AND SHAH, M. 2004. Contour based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans. Patt. Analy. Mach. Intell. 26, 11, 1531–1536.

基于模型的跟踪

- 二维形状模型

Skeletal Articulated Model

- 立体模型 Volumetric Model



运动目标的表示方法

- 基于点的跟踪
- 基于区域的跟踪
- 基于轮廓的跟踪
- 基于模型的跟踪

采用上述的哪种方法来表示运动目标和不同的应用场合、运动目标的运动特性、以及对跟踪算法的精度要求等密切相关。

目标跟踪

① 运动目标的表示方法

② 传统目标跟踪方法

③ 基于DNN的跟踪方法

目标跟踪的两种处理思路

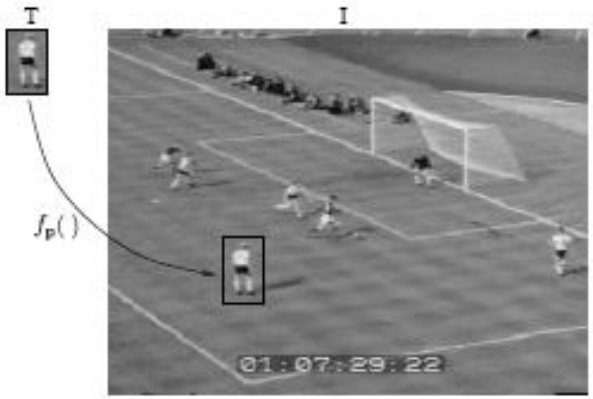
- 自底向上(Bottom-up)的处理方法
  - 数据驱动(Date-driven)的方法，不依赖于先验知识
- 自顶向下(Top-down)的处理方法
  - 模型驱动(Model-driven)的方法，依赖于所构建的模型或先验知识

目标跟踪的两种处理思路

- 自底向上(Bottom-up)的处理方法
  - → 模板匹配 (Template Match)
  - → 均值漂移 (Mean Shift) 课后练习
- 自顶向下(Top-down)的处理方法
  - → 卡尔曼滤波器 (Kalman Filter)
  - → 粒子滤波器 (Particle Filter) 课后练习

Comanniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(5): 564~577

模板匹配法（Template Matching）



- 在前一帧图像中目标位置（或模板T位置）为：(x, y)
- 在当前帧搜寻位置

$$(x', y') = (x + dx, y + dy)$$

使得

$$\arg\max_{dx,dy} \mathbf{cov}(T(x, y), I(d + dx, y + dy))$$

- 概念上相对比较简单
- 进行穷尽的搜索计算量非常大

进一步参考：SCHWEITZER, H., BELL, J. W., AND WU, F. 2002. Very fast template matching. In European Conference on Computer Vision (ECCV). 358–372.

基于卡尔曼滤波器的跟踪方法

- R. E. Kalman (1930 - 2016)
- Born 1930 in Hungary
- Studied at MIT / Columbia
- Developed filter in 1960/61

❑ Kalman filter：旨在利用线性系统状态方程，基于观测数据对系统状态进行最优估计。

❑基于卡尔曼滤波器的跟踪：通过建立状态空间模型，把跟踪问题表示为动态系统的状态估计问题。

动态系统

- 动态系统由状态转移方程和观测方程组成。
- 状态转移方程：

$$x_k = f(x_{k-1}, w_{k-1})$$

f：在很多跟踪问题中是非线性的

x_k, x_{k-1}：当前时刻与前一时刻的状态

w_{k-1}：系统噪声

动态系统

- 观测转移方程：

$$y_k = h(x_k, v_k)$$

h：在很多跟踪问题中是非线性的

y_k：测量值

x_k：当前时刻的状态

v_k：测量噪声

卡尔曼滤波器(Kalman Filter)

基本假设：

- 后验概率分布$p(x_{k-1}|y_{1:k-1})$为高斯分布
- 动态系统是线性的

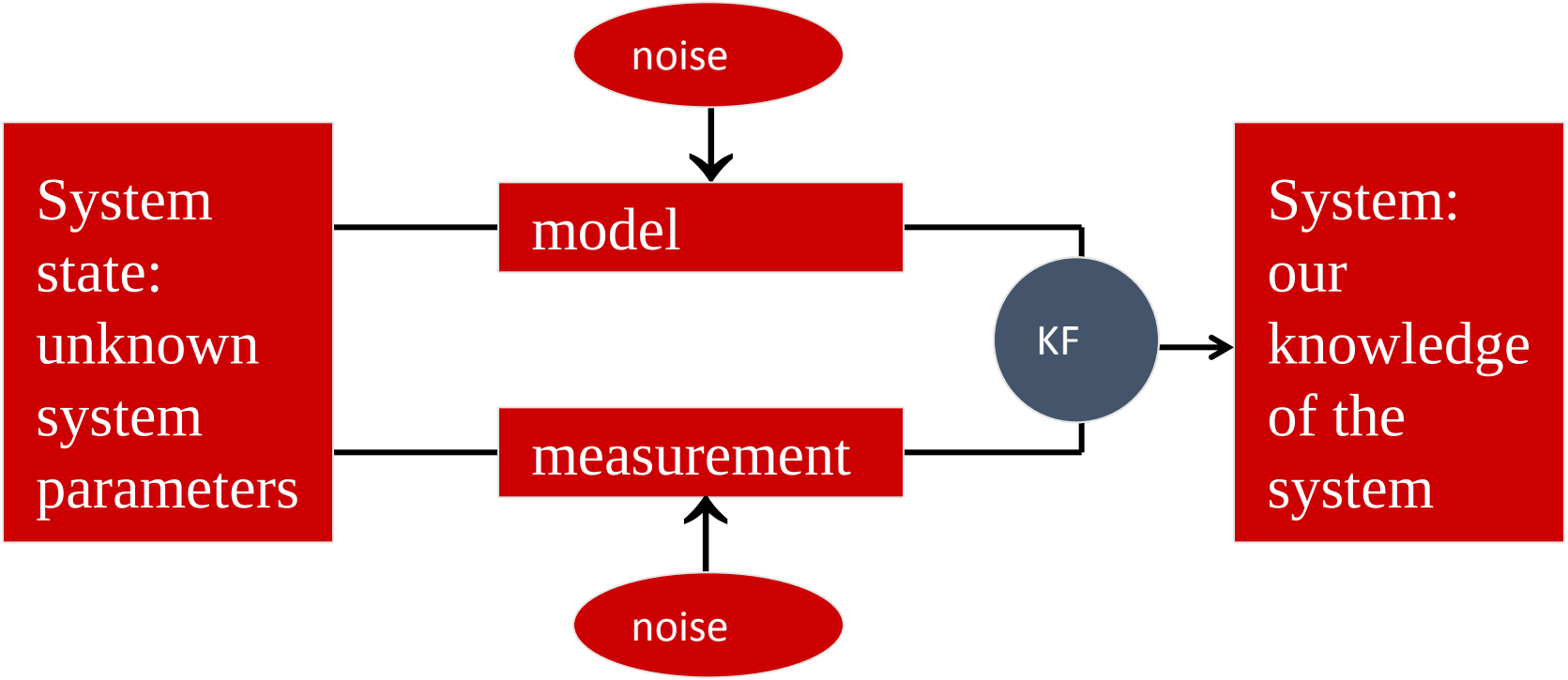
$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

$$y_k = Hx_k + v_k$$

- 系统噪声和测量噪声时高斯分布的，协方差矩阵分别为Q和R。

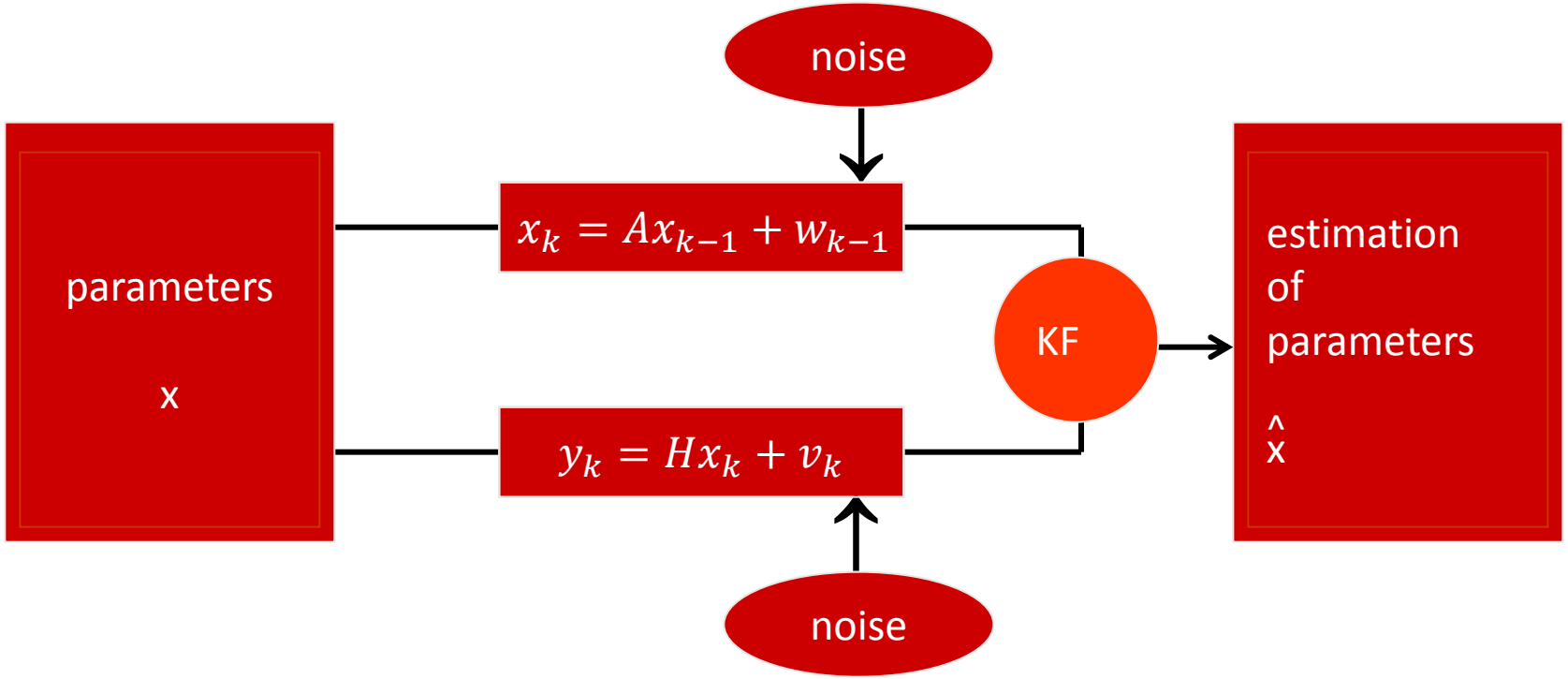
Kalman filter - KF

When and where?



Kalman filter - KF

using vectors and matrices



Noise

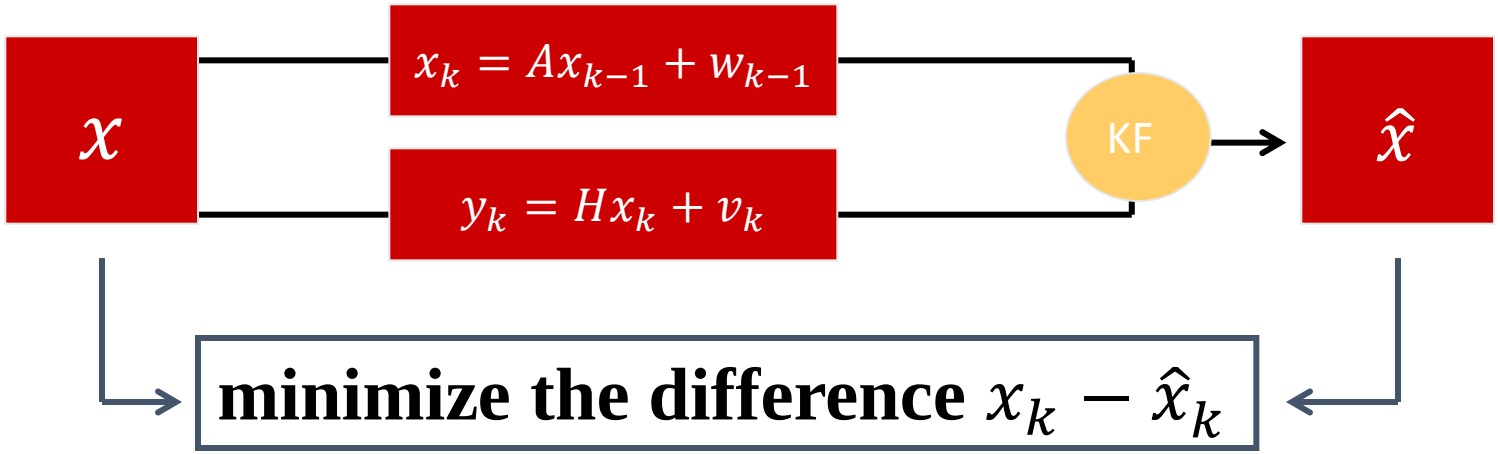
Noise: e $\quad \text{Gaussian} \Rightarrow E(e^2) = \sigma^2$

Noise covariance matrix

$$P = E(ee^T) = \begin{pmatrix} E(e_1e_1) & E(e_1e_2) & \cdots \\ E(e_2e_1) & E(e_2e_2) & \\ \vdots & & \ddots \end{pmatrix}$$

- System noise: $x_k = Ax_{k-1} + w_{k-1} \Rightarrow Q = E(ww^T)$
- Measurement noise: $y_k = Hv_k + v_k \Rightarrow R = E(vv^T)$

KF algorithm



- Prediction: $\hat{x}'_k = A\hat{x}_{k-1}$
- Correction: $\hat{x}_k = \hat{x}'_k + K(y_k - H\hat{x}'_k)$

预测—测量—更新

Kalman gain

KF algorithm

- $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$ ； $y_k = Hx_k + v_k$
- Prediction: $\hat{x}'_k = A\hat{x}_{k-1} + Bu_{k-1}$； Correction: $\hat{x}_k = \hat{x}'_k + K(y_k - H\hat{x}'_k)$
- minimize the difference $x_k - \hat{x}_k$

$$e = x - \hat{x}\ ;\ P = E(ee^T) = \begin{pmatrix} E(e_1e_1) & E(e_1e_2) & \cdots \\ E(e_2e_1) & E(e_2e_2) & \\ \vdots & & \ddots \end{pmatrix}$$

推导过程：

$$P_k = E\left[e_k e_k^T\right] = E\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\right]$$

$$P_k = E\left[[(I - K_k H)(x_k - \hat{x}'_k) - K_k v_k]\right.$$
$$\left.[(I - K_k H)(x_k - \hat{x}'_k) - K_k v_k]^T\right]$$

KF algorithm

估计值和真实值间误差的协方差矩阵

$$P_k = E\left[e_k e_k^T\right] = E\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\right]$$

$$P_k = E\left[[(I - K_k H)(x_k - \hat{x}'_k) - K_k v_k] [(I - K_k H)(x_k - \hat{x}'_k) - K_k v_k]^T\right]$$

系统状态变量和测量噪声之间是相互独立的

$$P_k = (I - K_k H) E\left[(x_k - \hat{x}'_k)(x_k - \hat{x}'_k)^T\right](I - K_k H) + K_k E\left[v_k v_k^T\right] K_k^T$$

预测值和真实值之间误差的协方差矩阵 $P'_k = E\left[e'_k e'^T_k\right] = E\left[(x_k - \hat{x}'_k)(x_k - \hat{x}'_k)^T\right]$

$$P_k = (I - K_k H) P'_k (I - K_k H)^T + K_k R K_k^T$$

$$P_k = P'_k - K_k H P'_k - P'_k H^T K_k^T + K_k \left(H P'_k H^T + R\right) K_k^T$$

KF algorithm

$$P_k = P'_k - K_k H P'_k - P'_k H^T K_k^T + K_k \left(H P'_k H^T + R\right) K_k^T$$

最小化 $T[P_k] = T[P'_k] - 2T[K_k H P'_k] + T\left[K_k \left(H P'_k H^T + R\right) K_k^T\right]$

$$\frac{dT[P_k]}{dK_k} = -2(H P'_k)^T + 2K_k \left(H P'_k H^T + R\right)$$

求得增益： $K_k = P'_k H^T \left(H P'_k H^T + R\right)^{-1}$

更新P_k

$$\begin{aligned} P_k &= P'_k - P'_k H^T \left(H P'_k H^T + R\right)^{-1} H P'_k \\ &= P'_k - K_k H P'_k \\ &= (I - K_k H) P'_k \end{aligned}$$

KF algorithm

更新P'_k

$$P'_{k+1} = E\left[e'_{k+1}e'^{T}_{k+1}\right]$$

$$= E\left[\left(x_{k+1}-\hat{x}'_{k+1}\right)\left(x_{k+1}-\hat{x}'_{k+1}\right)^T\right]$$

$$= E\left[\left(A\left(x_k-\hat{x}_k\right)+\omega_k\right)\left(A\left(x_k-\hat{x}_k\right)+\omega_{kk}\right)^T\right]$$

系统状态变量和系统噪声之间是相互独立的

$$= E\left[\left(Ae_k\right)\left(Ae_k\right)^T\right]+E\left[\omega_k\omega_k^T\right]$$

$$= AP_kA^T+Q$$

卡尔曼滤波器—时间更新和状态更新

- 时间更新

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

- 状态更新

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - H\hat{x}_k^-)$$

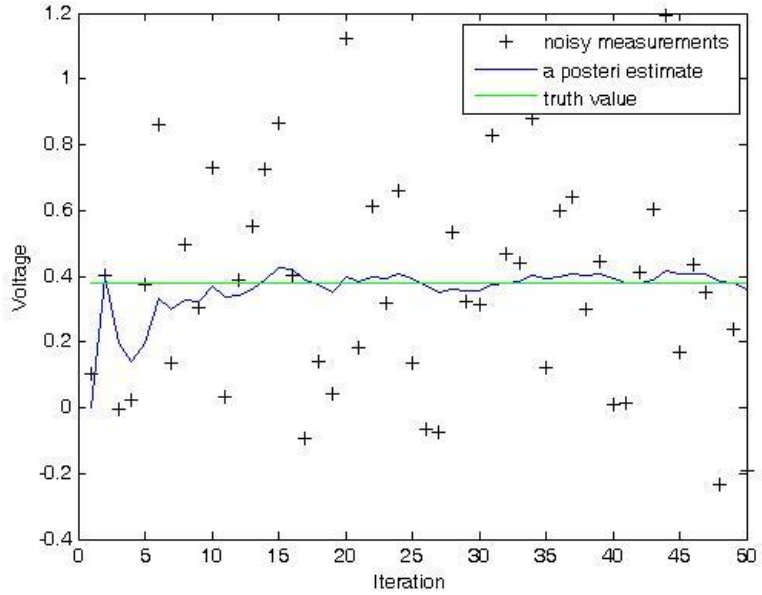
$$P_k = (I - K_k H)P_k^-$$

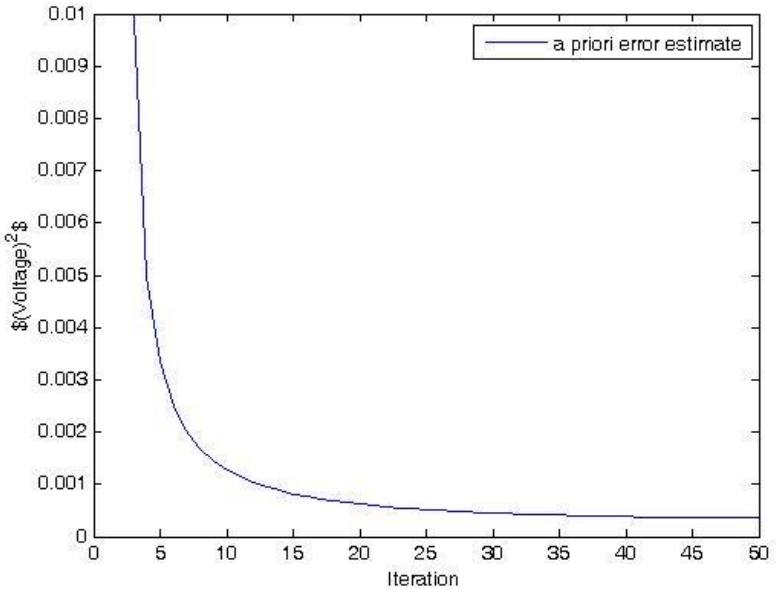
实例1

- 测量电压：假设我们可以测量这个常数的幅值，但观测幅值中掺入了幅值均方根为0.1伏的白噪声。
- 方程描述：
- $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} = x_{k-1} + w_{k-1}$
- $y_k = Hx_k + v_k = x_k + v_k$

- 过程的状态不随时间变化，A = 1；没有控制输入，u = 0；包含噪声的观测值是状态变量的直接体现，H = 1 。

实例1


noisy measurements
a posteri estimate
truth value
Voltage
Iteration
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
5
10
15
20
25
30
35
40
45
50


a priori error estimate
$(Voltage)^2$
Iteration
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
0
5
10
15
20
25
30
35
40
45
50

实例2

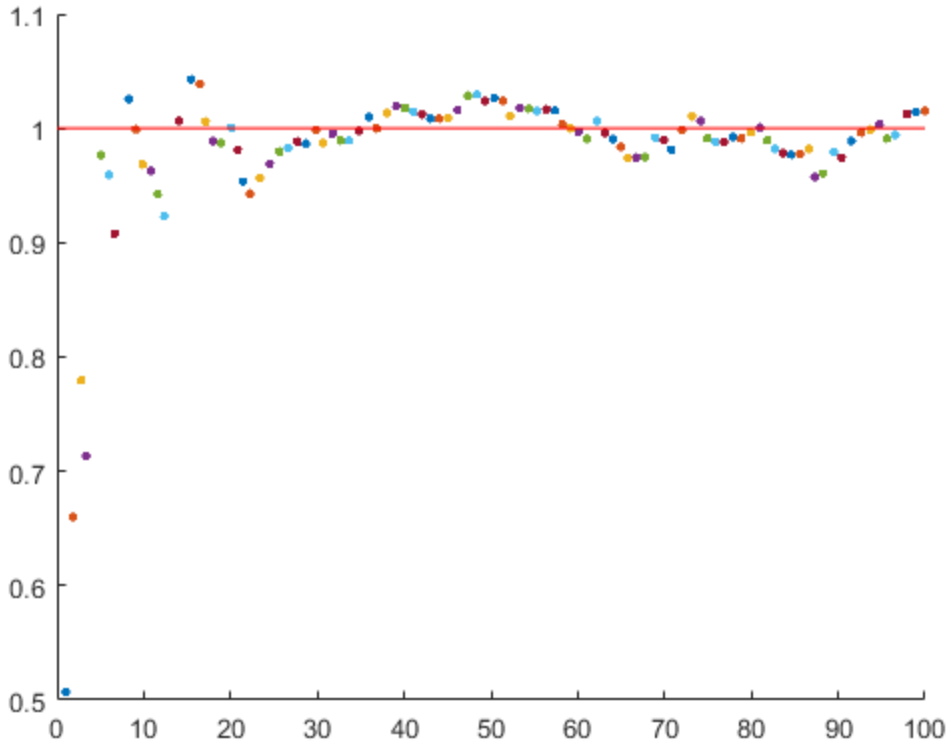
位置估计：假设一个模拟质点进行匀速（速度为1）的直线运动，可以观测到质点每一时刻的位置（掺如了幅值均方根为1的白噪声）。

- 动态系统是线性的

$$[x_k, \dot{x}_k]^T = A[x_{k-1}, \dot{x}_{k-1}]^T + w_{k-1}$$

$$y_k = H[x_k, \dot{x}_k] + v_k$$

实例2



实例3

The estimated position from the Kalman filter (red) is compared against the actual ground truth position (green).

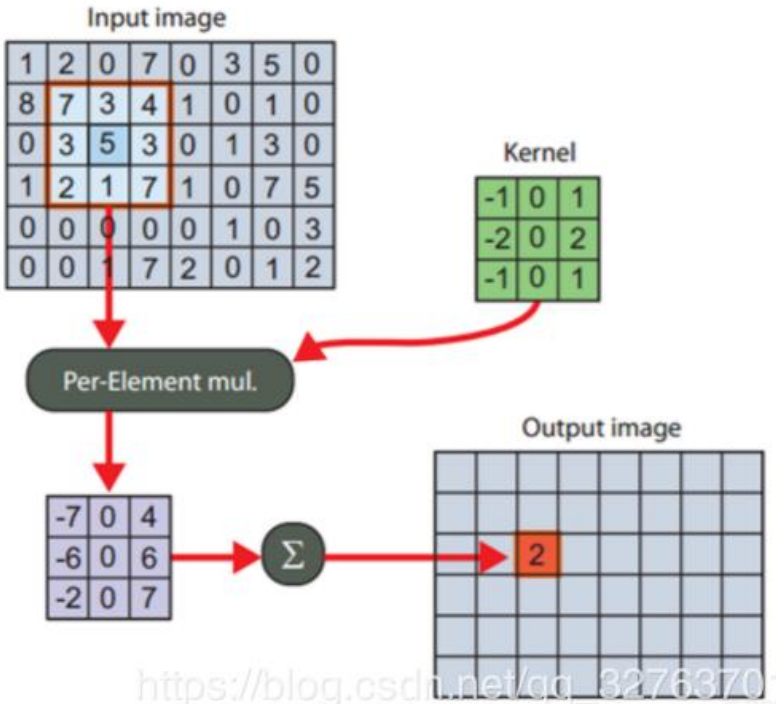
卡尔曼滤波器的扩展

- Extended Kalman Filter (EKF)
- Unscented Kalman Filter (UKF)

- 同样基于高斯分布的假设；
- 状态转移方程和测量方程为非线性函数；
- 沿用Kalman Filter的框架；
- 将非线性函数局部线性化。

相关滤波与跟踪（MOSSE）

- 相关（Correlation）
- $g = f \otimes h$ —》 $g(i,j) = \sum f(i+k, j+l) \cdot h(k,l)$
- 其中f是输入信号， h是相关核/滤波器， g是空域里的响应图



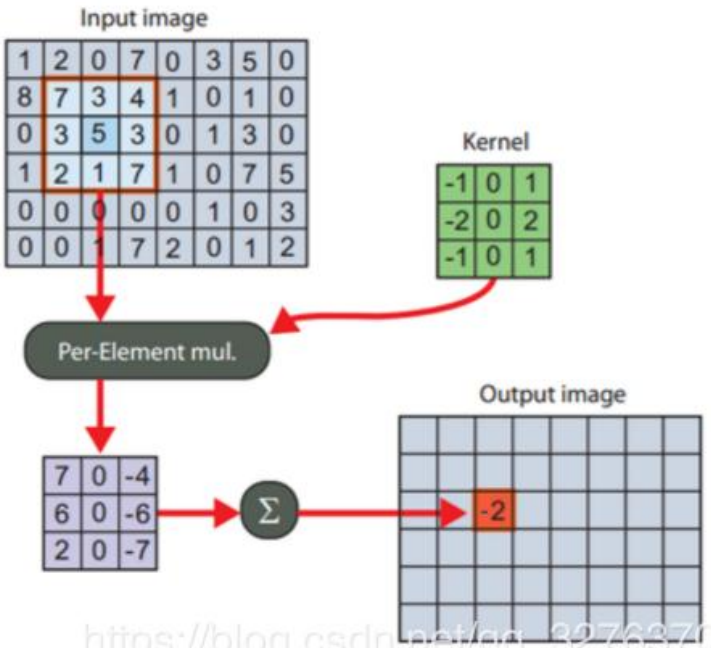
基本原理：在视频帧中利用h找到响应值最高的位置，即实现跟踪。

不足：慢！

David S. Bolme, J. Ross Beveridge, Bruce A. Draper, Yui Man Lui, Visual objecting tracking using adaptive correlation filters, CVPR, 2010

相关滤波与跟踪

- 卷积
- $g = f * h$
- $g(i,j) = \sum f(i-k, j-l) \cdot h(k,l)$
- 其中f是输入信号， h是相关核/滤波器， g是空域里的响应图



$$g = f \otimes h = f * h^*$$

$$FFT(g) = FFT(f) \cdot FFT(h^*)$$

MOSSE

- 第一帧给定bounding box；
- 通过对groundtruth中的bounding box进行随机仿射变换产生8个样本进行训练，获得H*

$$\min_{H^*} \sum_i |F_i \odot H^* - G_i|^2$$

$$H^* = \frac{\sum_i G_i \odot F_i^*}{\sum_i F_i \odot F_i^*}$$

MOSSE

- 滤波器参数更新

$$
\begin{aligned}
H_i^* &= \frac{A_i}{B_i} \\
A_i &= \eta G_i \odot F_i^* + (1-\eta) A_{i-1} \\
B_i &= \eta F_i \odot F_i^* + (1-\eta) B_{i-1}
\end{aligned}
$$

MOSSE

Algorithm	Frame Rate	CPU
FragTrack[1]	realtime	Unknown
GBDL[19]	realtime	3.4 Ghz Pent. 4
IVT [17]	7.5fps	2.8Ghz CPU
MILTrack[2]	25 fps	Core 2 Quad
MOSSE Filters	669fps	2.4Ghz Core 2 Duo

不足：

1. 特征不够稳定（输入为原始灰度像素）。
2. 较难处理目标尺度变化情况 。
3. 鲁棒性相对较弱。

相关滤波与跟踪

1. **KCF：** João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, High-Speed Tracking with Kernelized Correlation Filters, PAMI, 2015, 37(3):583-596.
2. **C-COT：** Danelljan M , Robinson A , Khan F S , et al. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking, ECCV, 2016.
3. 等等

目标跟踪

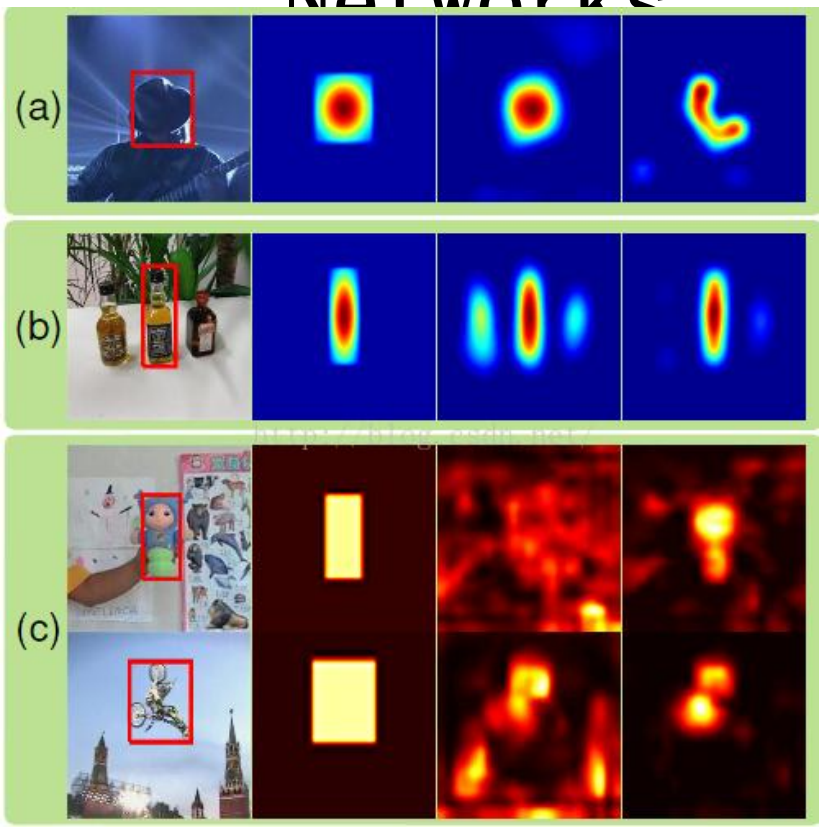
① 运动目标的表示方法

② 传统目标跟踪方法

③ 基于DNN的跟踪方法

- 策略1：DNN特征 + 相关滤波
- 策略2：直接使用DNN进行目标跟踪

Visual Tracking with Fully Convolutional Networks



- Wang L, Ouyang W, Wang X, et al. Visual Tracking with Fully Convolutional Networks[C]// IEEE International Conference on Computer Vision. IEEE, 2016:3119-3127.

Visual Tracking with Fully Convolutional Networks

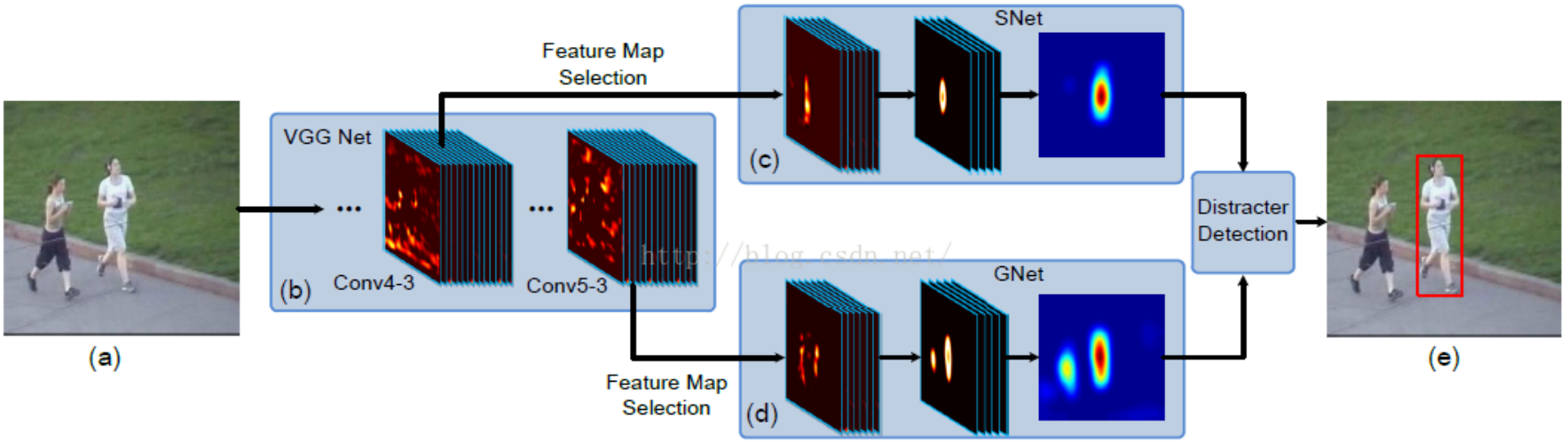


Figure 5. Pipeline of our algorithm. (a) Input ROI region. (b) VGG network. (c) SNet. (d) GNet. (e) Tracking results.

1 背景内容

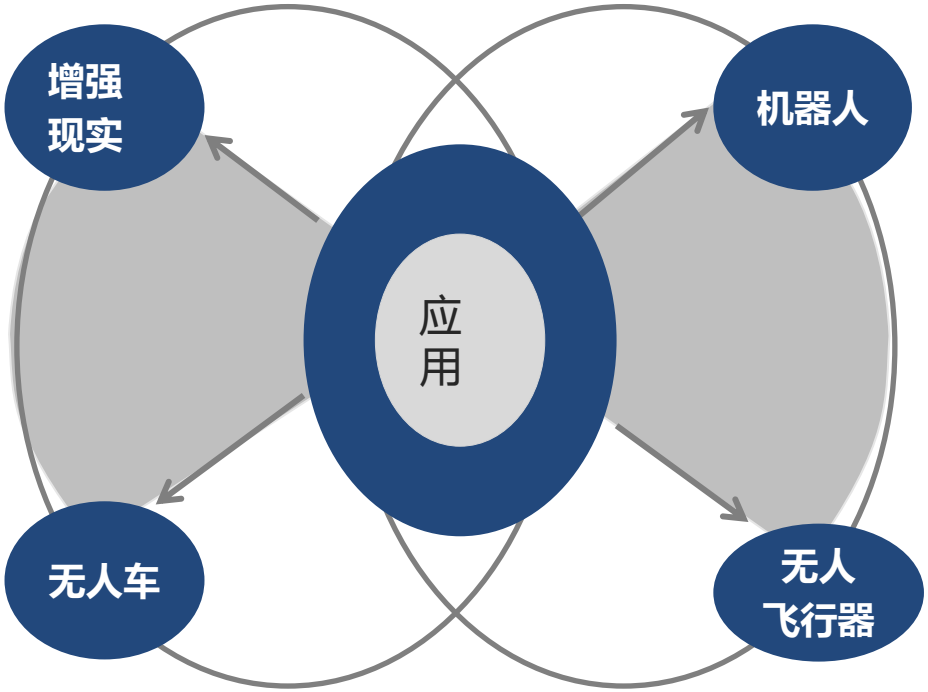
2 目标跟踪

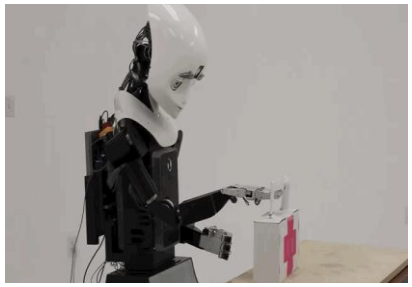
3 视觉定位

4 小节

视觉定位

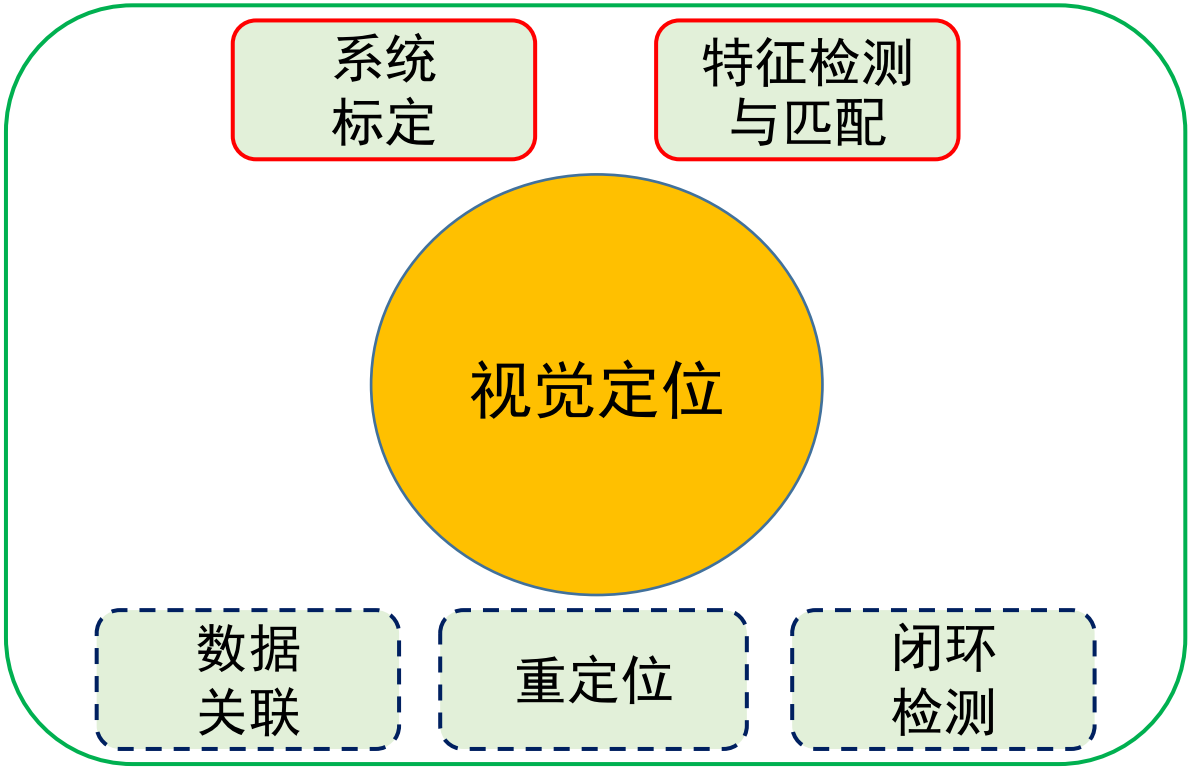
- 适用于室内、外定位；
- 基于图像的定位可以集成到手机等移动终端，方便廉价；
- 基于位置的服务几乎无处不在，如智能机器人、虚拟现实、增强现实等等；





视觉定位系统

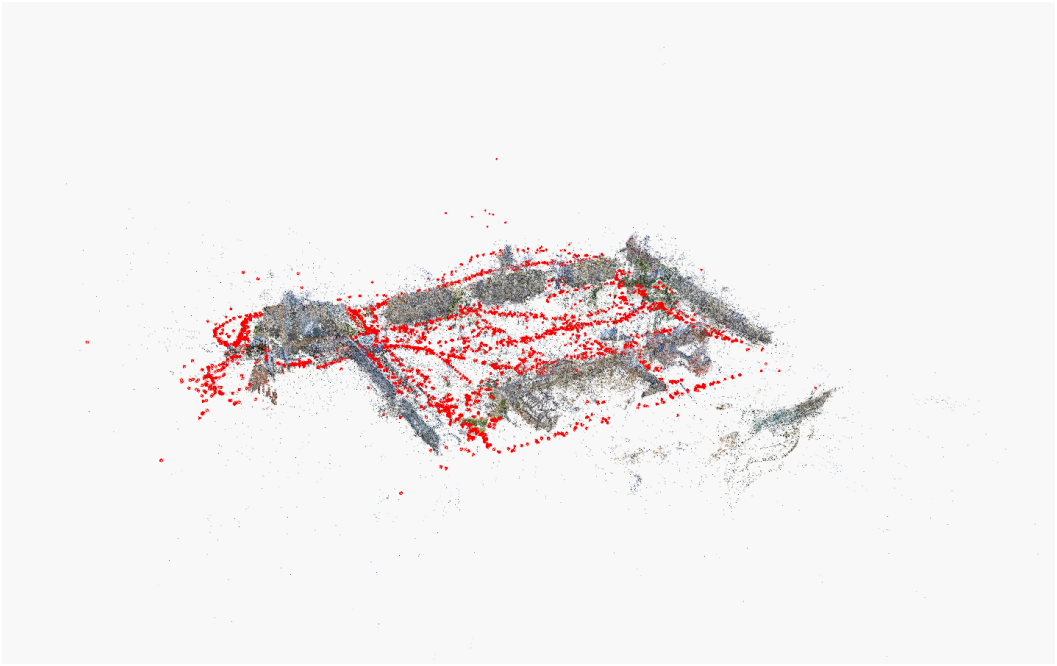
系统功能：从图像计算相机与场景的相对位置、姿态



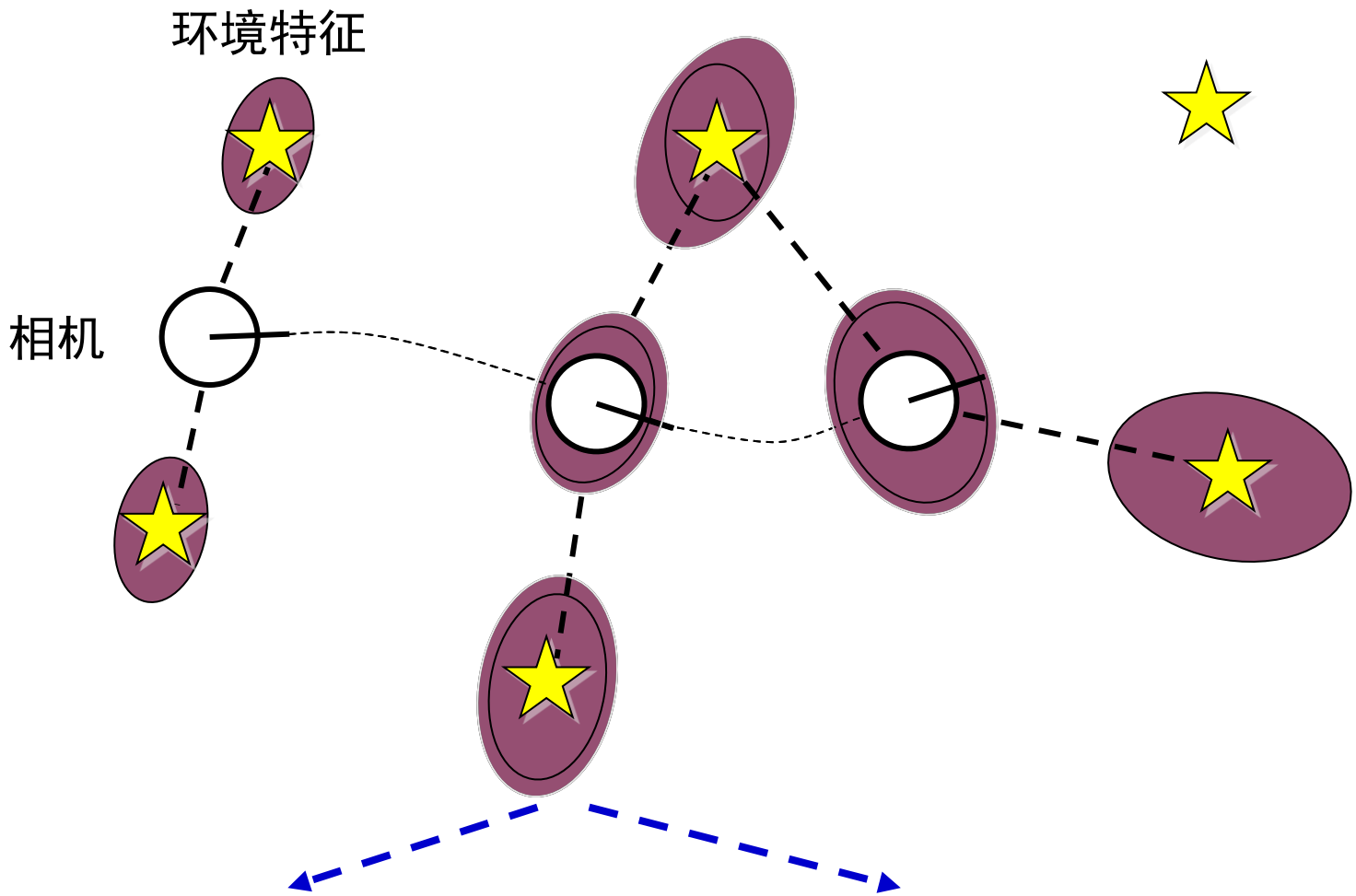
离线的视觉定位

难点：

- 图像质量参差不齐
- 计算精度与复杂度



在线的视觉定位

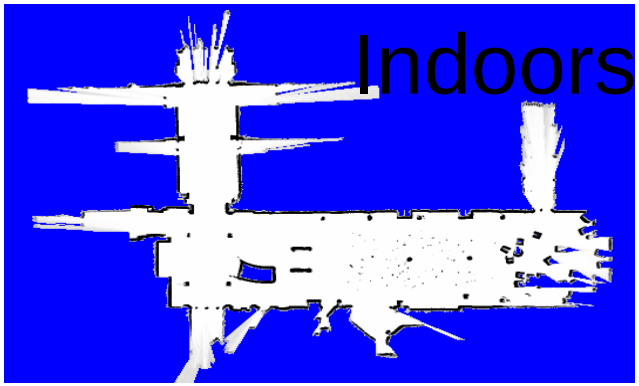


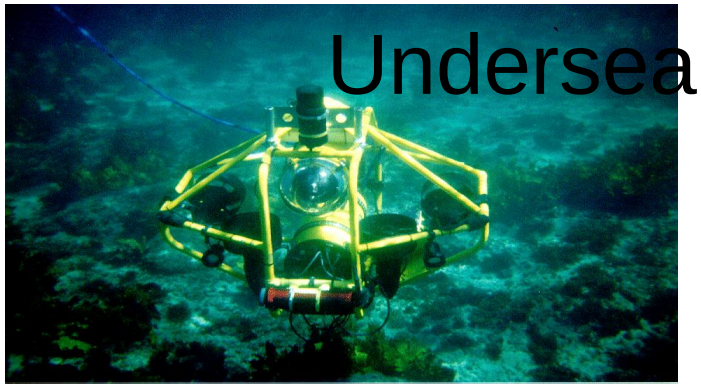
环境已知：单纯定位

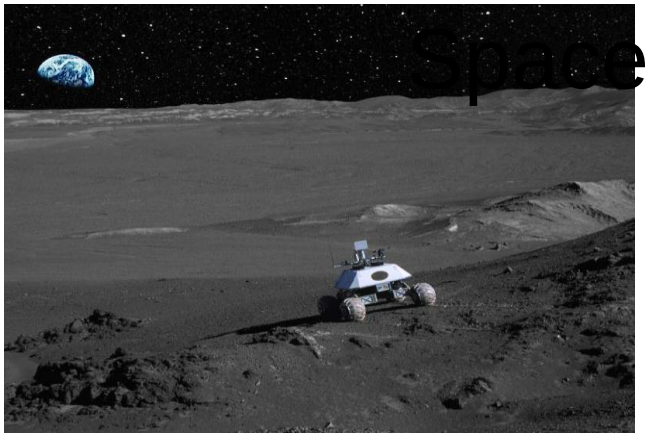
环境未知：SLAM（同步定位与地图创建）

SLAM

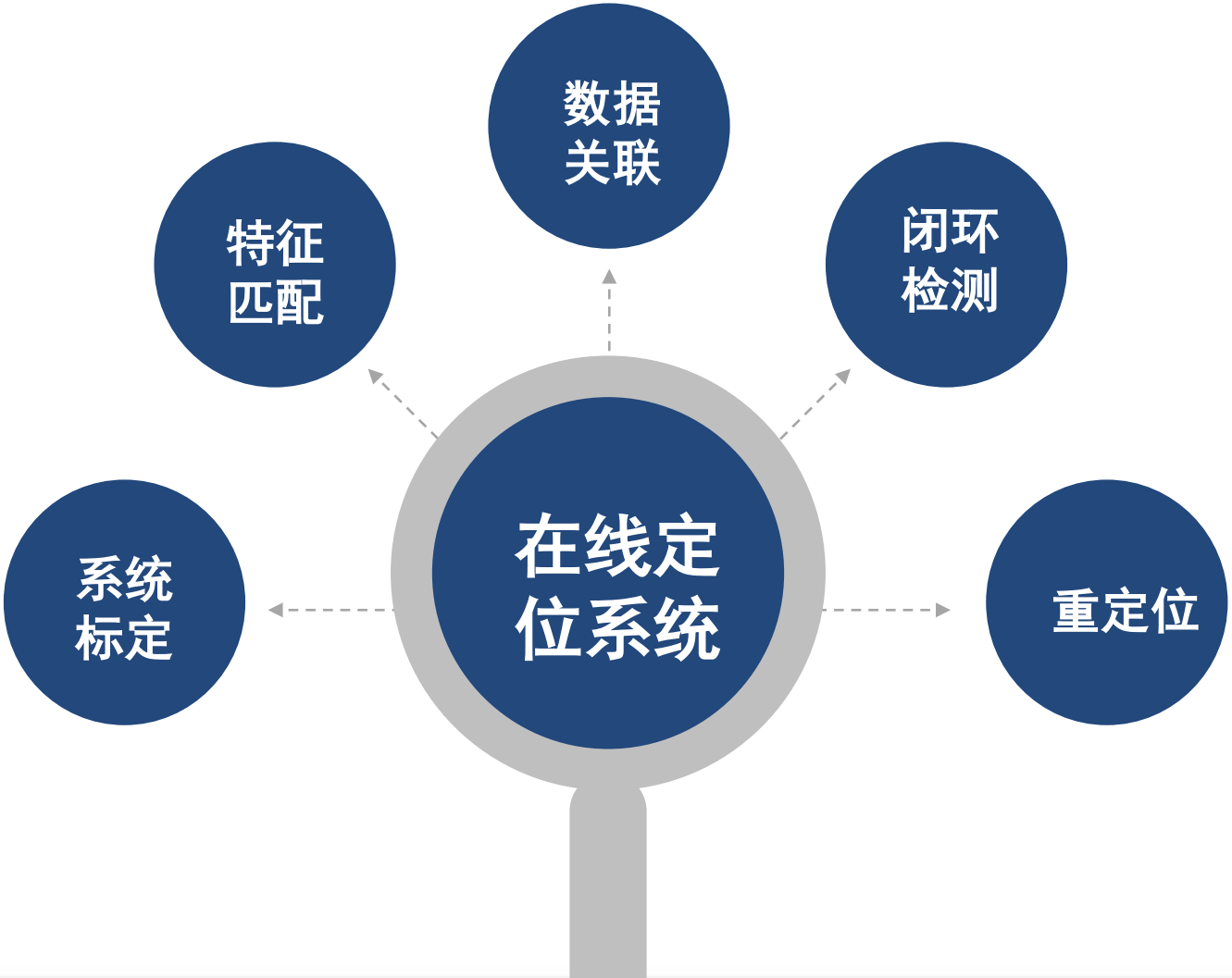
- 同步定位与地图构建（Simultaneous Localization and Mapping,SLAM），指移动物体在自身位置不确定的情况下，利用自身的传感器，在未知的环境中创建一个与环境相一致的地图，并同时确定自身在地图中的位置。
-



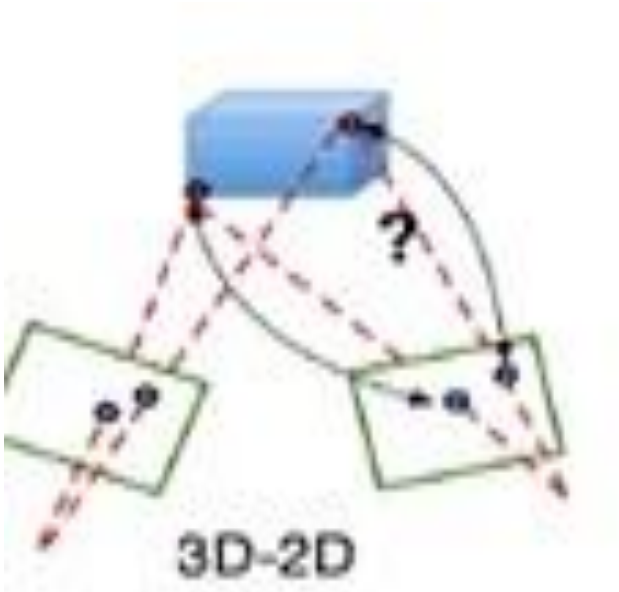


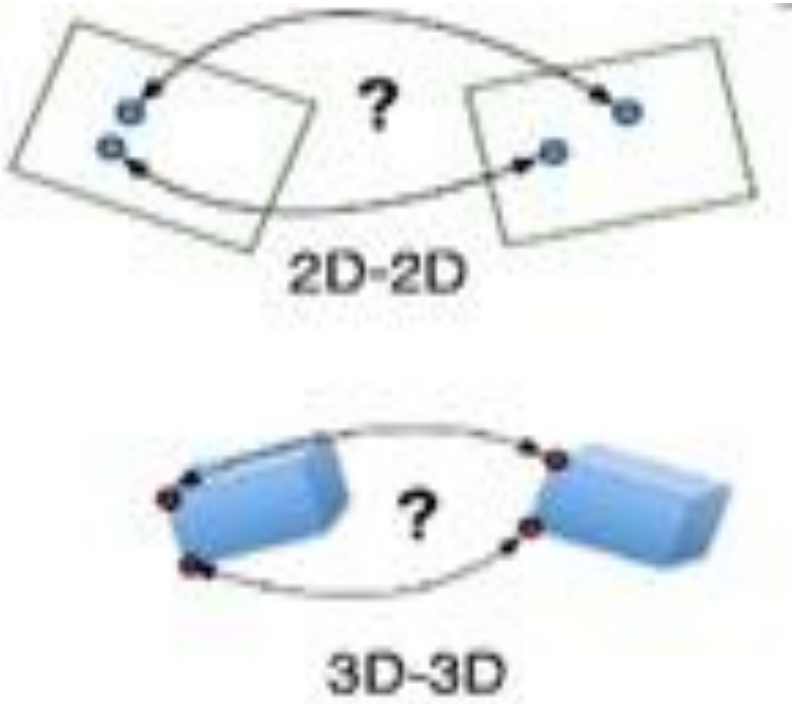


在线的视觉定位

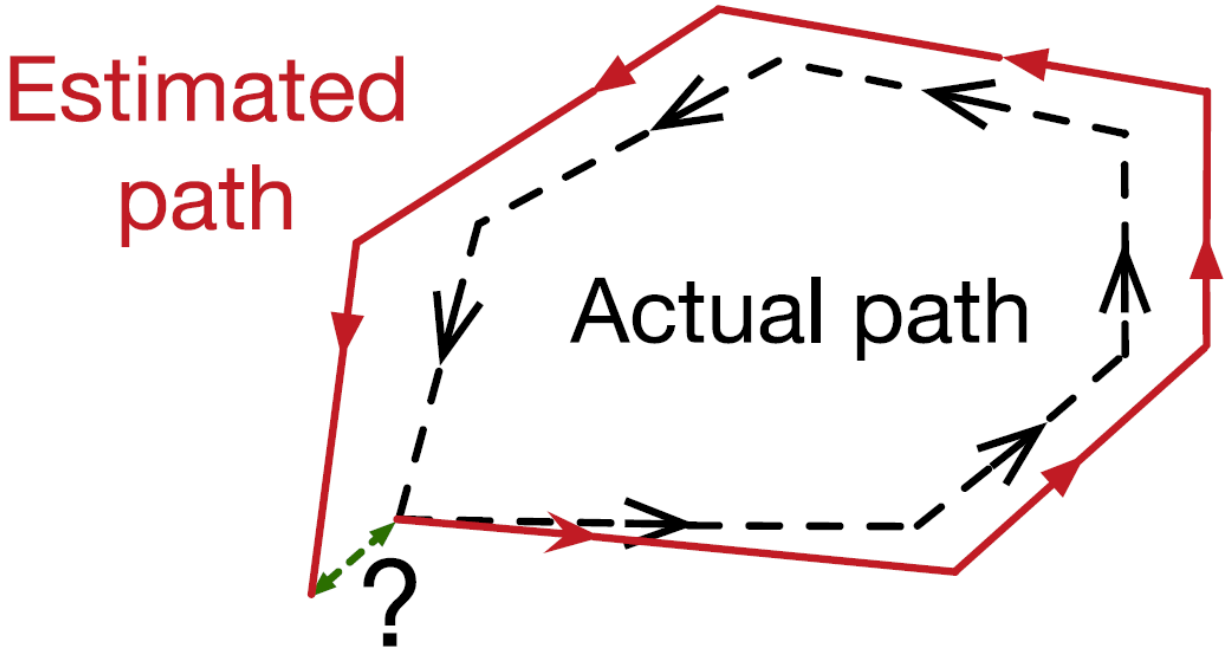


数据关联

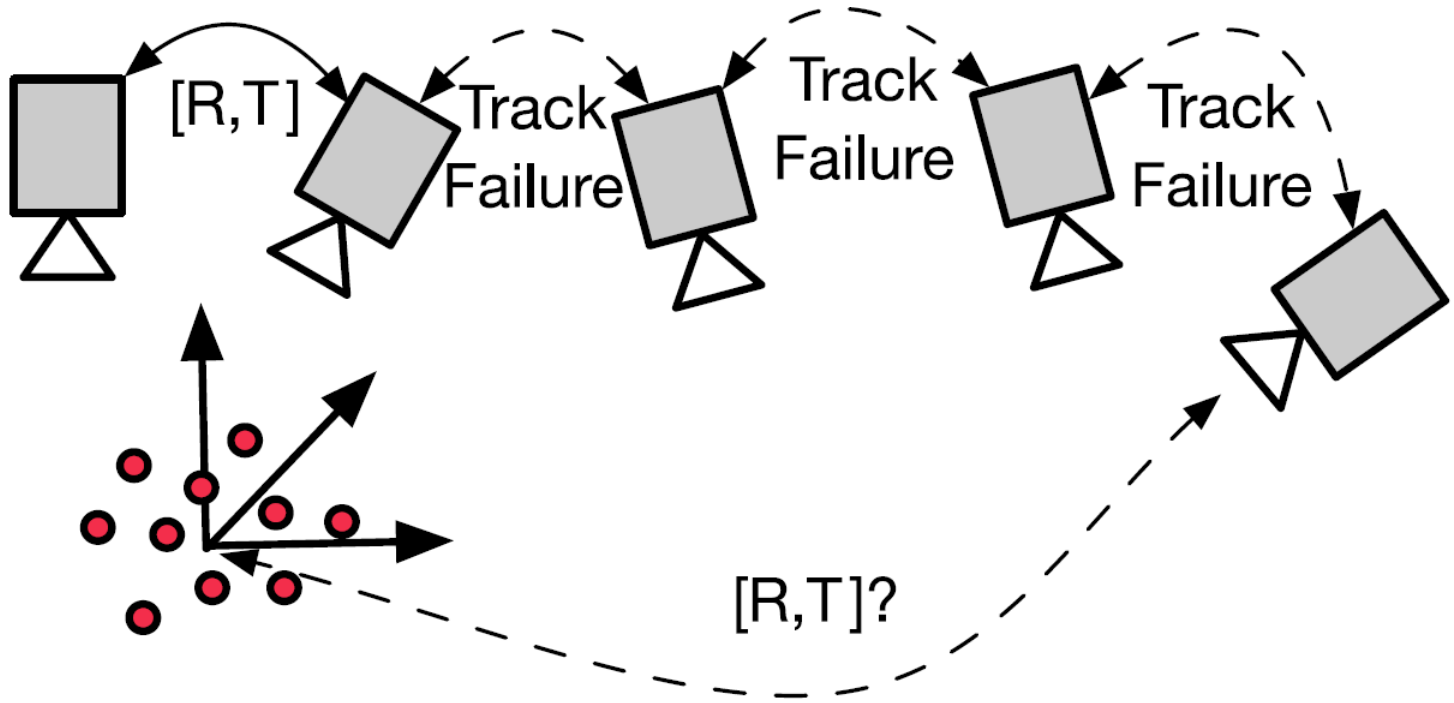




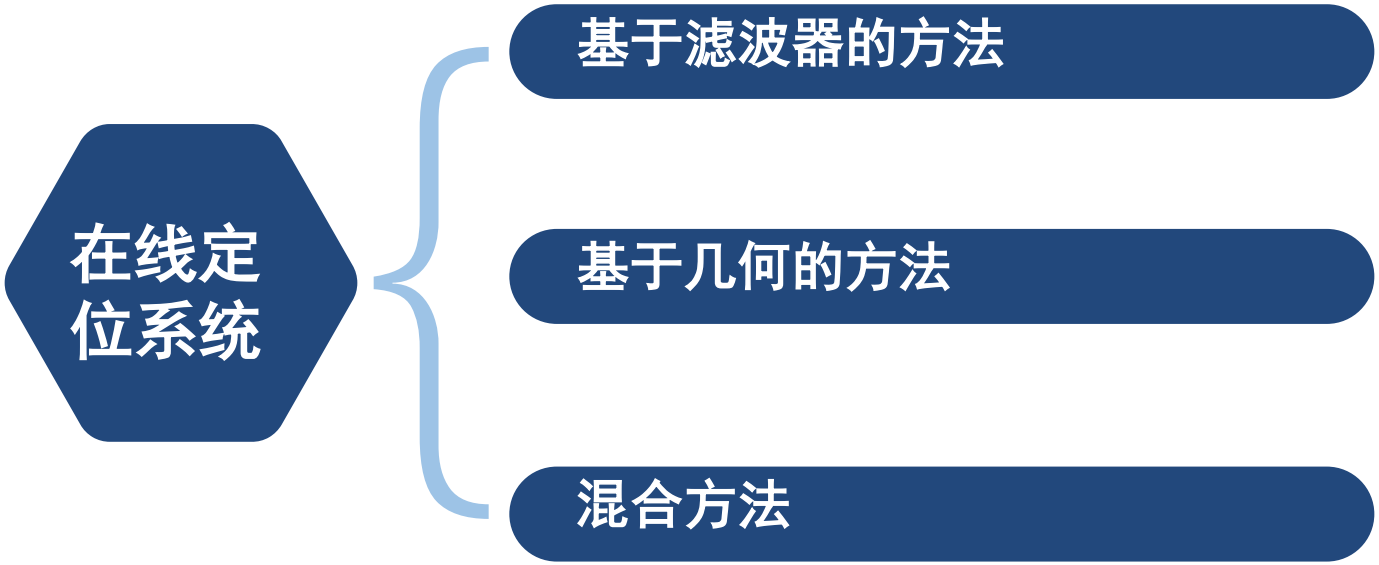
闭环检测



重定位



在线视觉定位分类



难点：

- 实时性与计算精度
- 定位失败后的视觉重定位

基于滤波器的实时定位方法

核心思想：将相机位置、姿态和地图特征等未知信息作为滤波器的状态量，利用相机的观测特征不断地估计相机相机位置、姿态和地图特征。
常用滤波器：卡尔曼滤波器、粒子滤波器等。

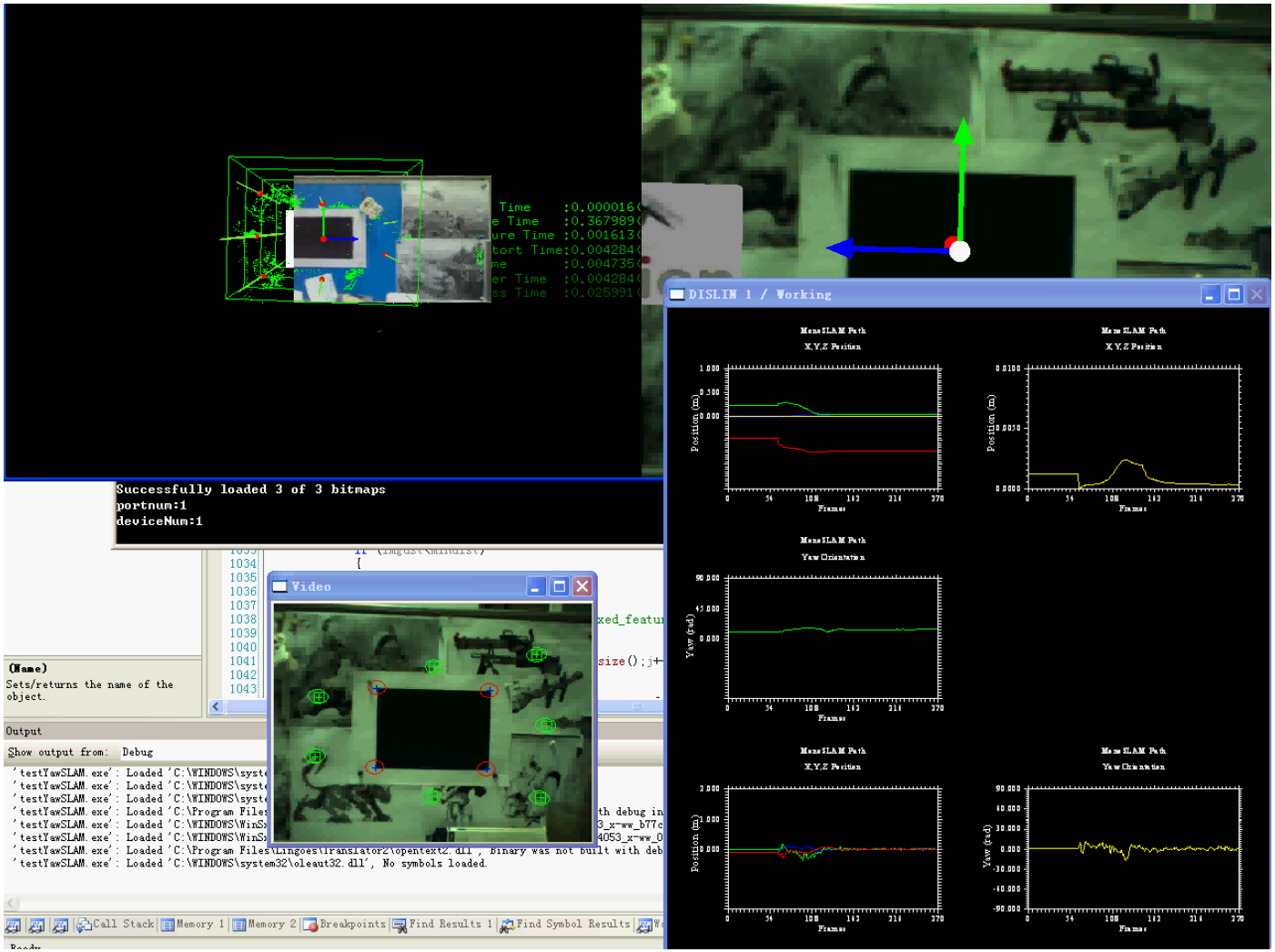
卡尔曼滤波器状态方程

$$x_{c,t} = \begin{pmatrix} r_t \\ \gamma_t \\ v_t \\ \omega_{\gamma,t} \end{pmatrix} = \begin{pmatrix} r_{t-1} + (v_{t-1} + n_v)\Delta t \\ \gamma_{t-1} + (\omega_{\gamma,t-1} + n_{\omega_\gamma})\Delta t \\ v_{t-1} + n_v \\ \omega_{\gamma,t-1} + n_{\omega_\gamma} \end{pmatrix}$$

卡尔曼滤波器观测方程

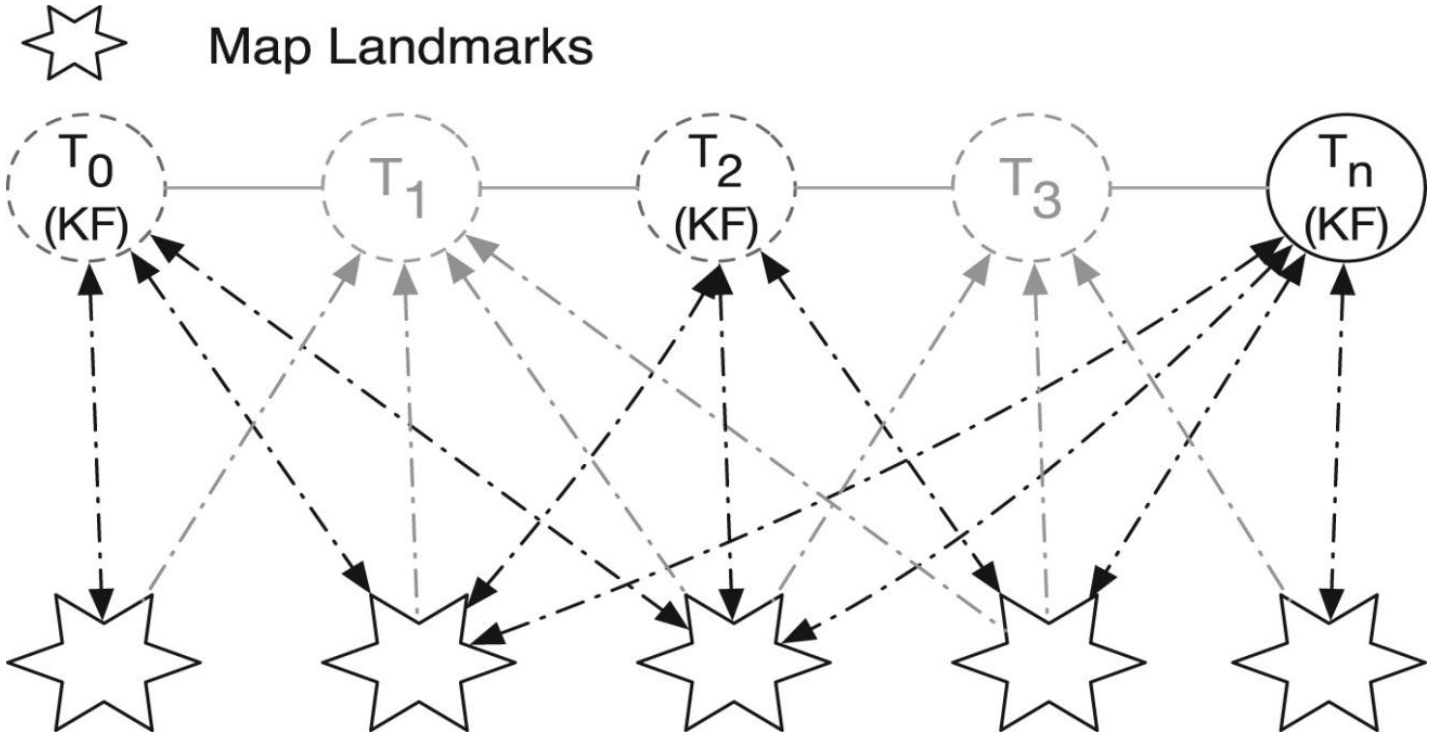
$$z_{i,t} = K(R_{w-c,t} f_i + T) + n_m \quad i = 1, 2, ..., M$$

基于滤波器的实时定位方法



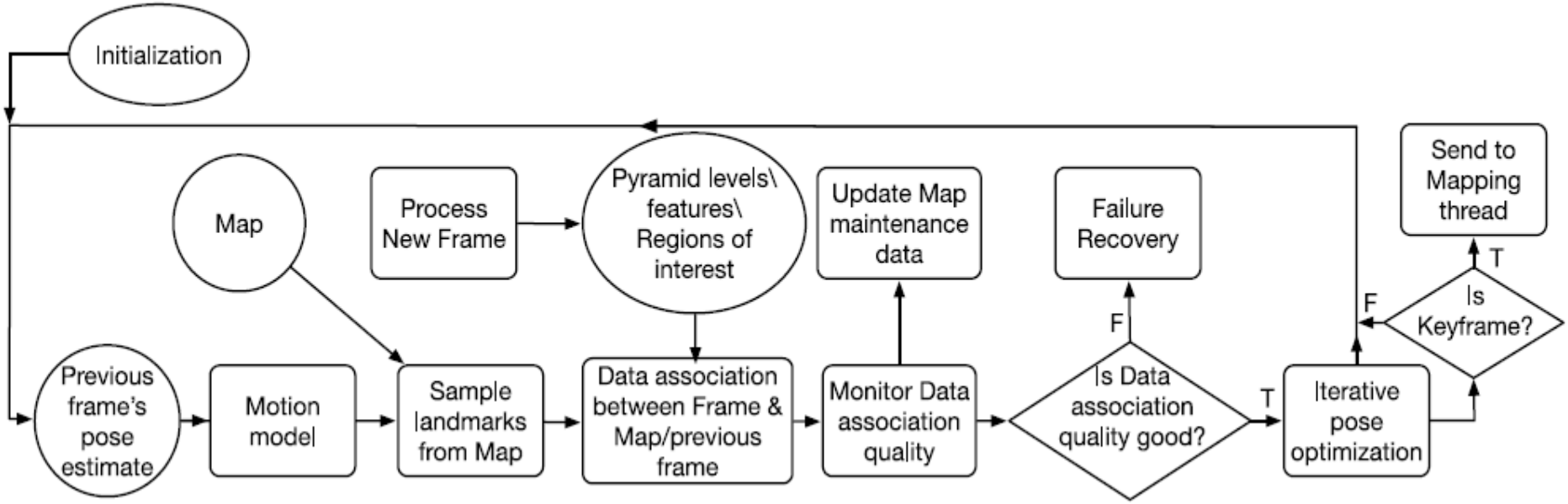
基于几何的实时定位方法

基于关键帧的实时定位方法



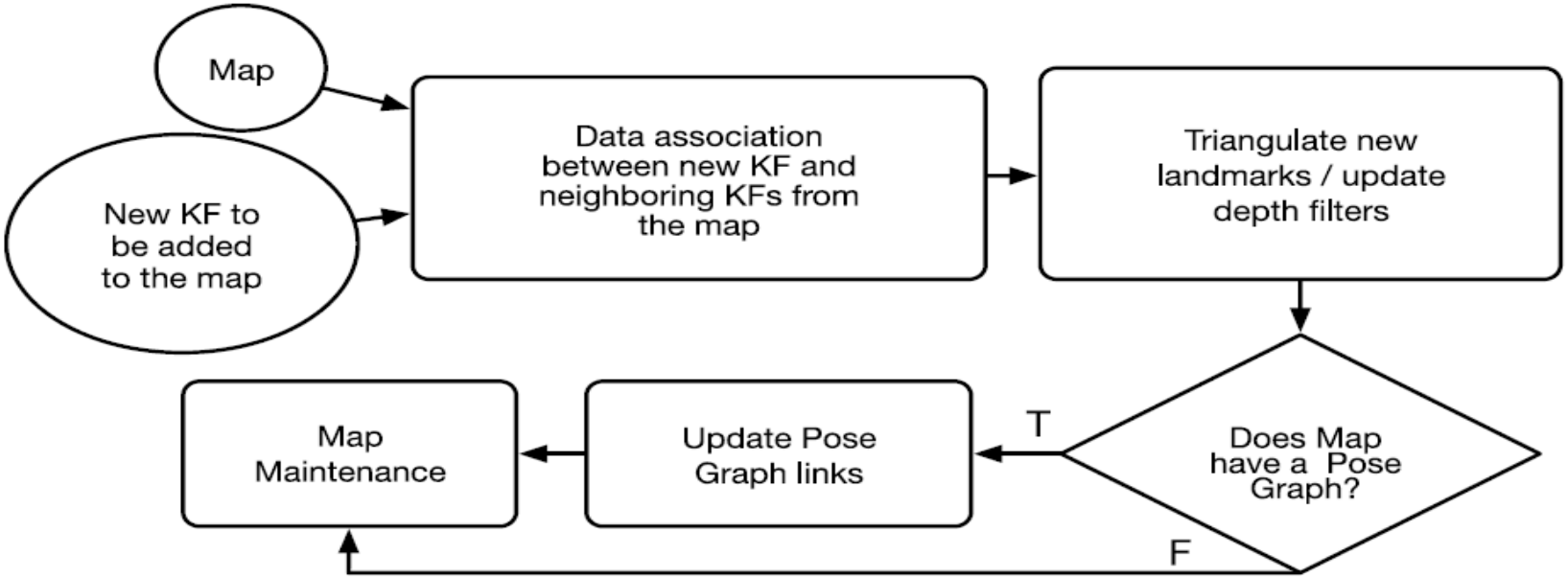
基于关键帧的实时定位方法

Tracking Thread



基于关键帧的实时定位方法

Mapping Thread



基于几何的实时定位方法

双（多）线程处理：

- 地图线程：利用几何重建方法构建环境地图
- 定位（跟踪）线程：利用图像信息和地图信息，实时计算相机的位置姿态

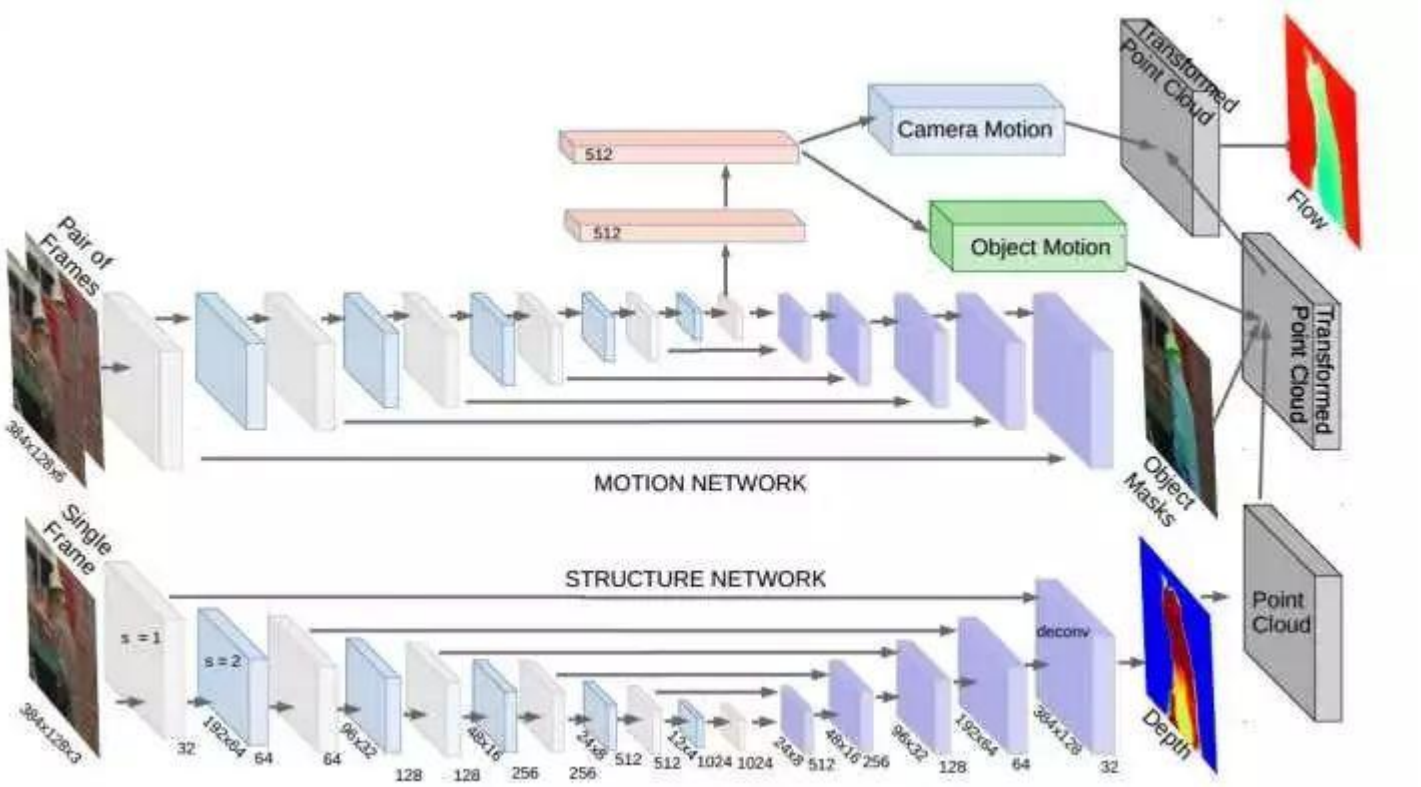
数据关联

闭环检测

重定位

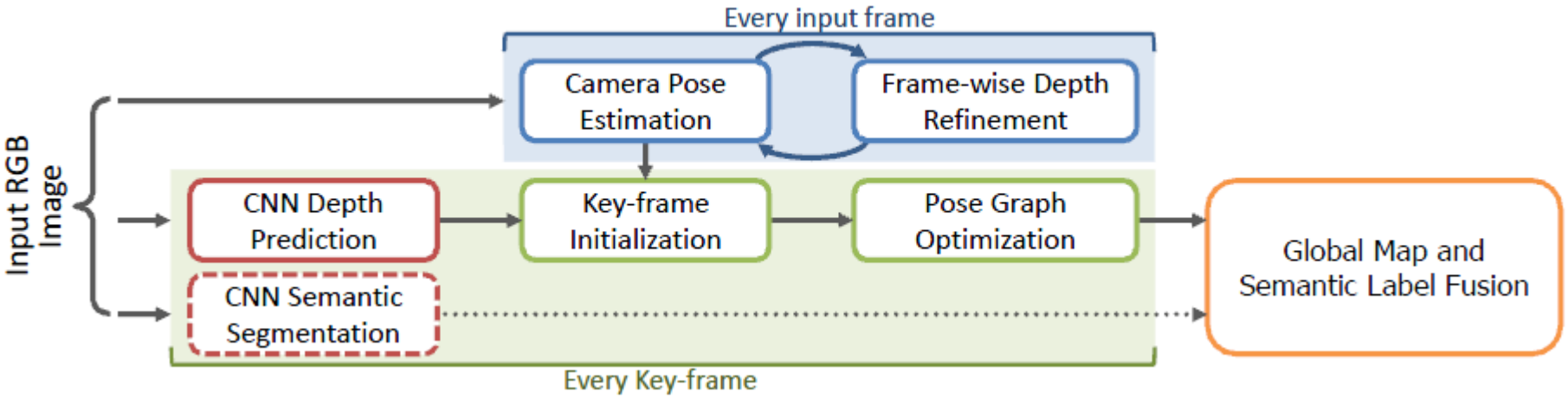
SfM-Net

- Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, Katerina Fragkiadaki， “SfM-Net: Learning of Structure and Motion from Video”, Learning of Structure and Motion from Video （https://arxiv.org/abs/1704.07804）



CNN-SLAM

- Keisuke Tateno, Federico Tombari, Iro Laina, Nassir Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction", in CVPR 2017



1 背景内容

2 目标跟踪

3 视觉定位

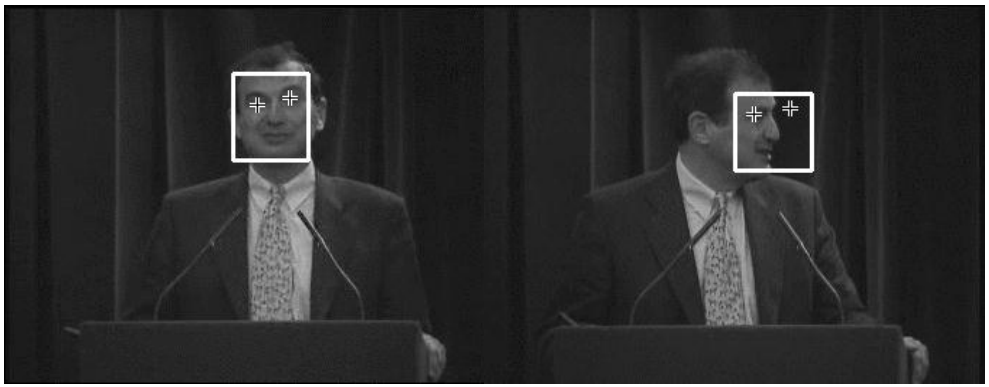
4 小节

小节

- 目标跟踪
  - 模板匹配法
  - 基于Kalman滤波器的跟踪方法
  - 基于相关滤波的跟踪方法
  - 基于CNN的跟踪方法
- 视觉定位
  - 基于Kalman滤波器的定位方法
  - 基于关键帧的定位方法

小节

- 视觉跟踪所面临的主要难点：鲁棒性、准确性、快速性。
- 鲁棒性：跟踪算法能够在各种环境条件下实现对运动目标（摄像机）持续稳定的跟踪。
- 准确性
- 快速性：在保证所要求的跟踪精度的前提下，实现实时地跟踪。



课后练习作业

- 针对43页实例3，试编程实现基于Kalman滤波器的目标跟踪。
- 试实现C-COT跟踪算法。

参考文献

1. Comanniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(5): 564~577.
2. João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, High-Speed Tracking with Kernelized Correlation Filters, PAMI, 2015, 37(3):583-596.
3. Danelljan M , Robinson A , Khan F S , et al. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking, ECCV, 2016.

谢谢！