Week 9

2024年11月1日 16:55

P29 1. Euclidean coordinate
2. normalized coordinate
3. pixel coordinate

内外参矩阵

Week 10

2024年11月1日 18:51

1. Structure from Motion (SfM)

Structure from Motion 是一种利用多张2D图像来恢复3D场景结构的技术。它的主要目标是从不同视角的图像序列中重建出场景的3D结构。关键步骤包括：

- **特征匹配**：在多张图像中识别并匹配同一物体点的特征（例如SIFT、ORB等）。
- **相机姿态估计**：估算每张图像的相机位置和姿态。
- **三角化**：通过多视角几何，将匹配的2D点反投影到3D空间中，恢复出对应的3D点。
- **优化**：通过 **捆绑调整（Bundle Adjustment）** 来优化相机参数和3D点的位置，提升精度。

SfM 可以在没有先验信息的情况下，通过多张图像逐步估计相机的姿态和3D点的坐标，因此适合于从完全未知的图像序列中构建场景的三维模型。最终输出通常包括场景的稀疏3D点云和相机的外参（位置和朝向）。

SVD

Singular Value Decomposition (SVD) – Any matrix can be decomposed into three matrices such that

$$\underset{m\times n}{M} = \underset{m\times m}{U}\ \underset{m\times n}{\Sigma}\ \underset{n\times n}{V} \qquad \Sigma=diag\{\sigma_1,\ldots,\sigma_m\}$$

– $\sigma_1,\ldots,\sigma_m$ are "singular values" and are similar to eigenvalues. If M is not full rank, some singular values will be 0.

– For a non-full-rank M, the column of V corresponding to a 0 singular value is in the null space of M.

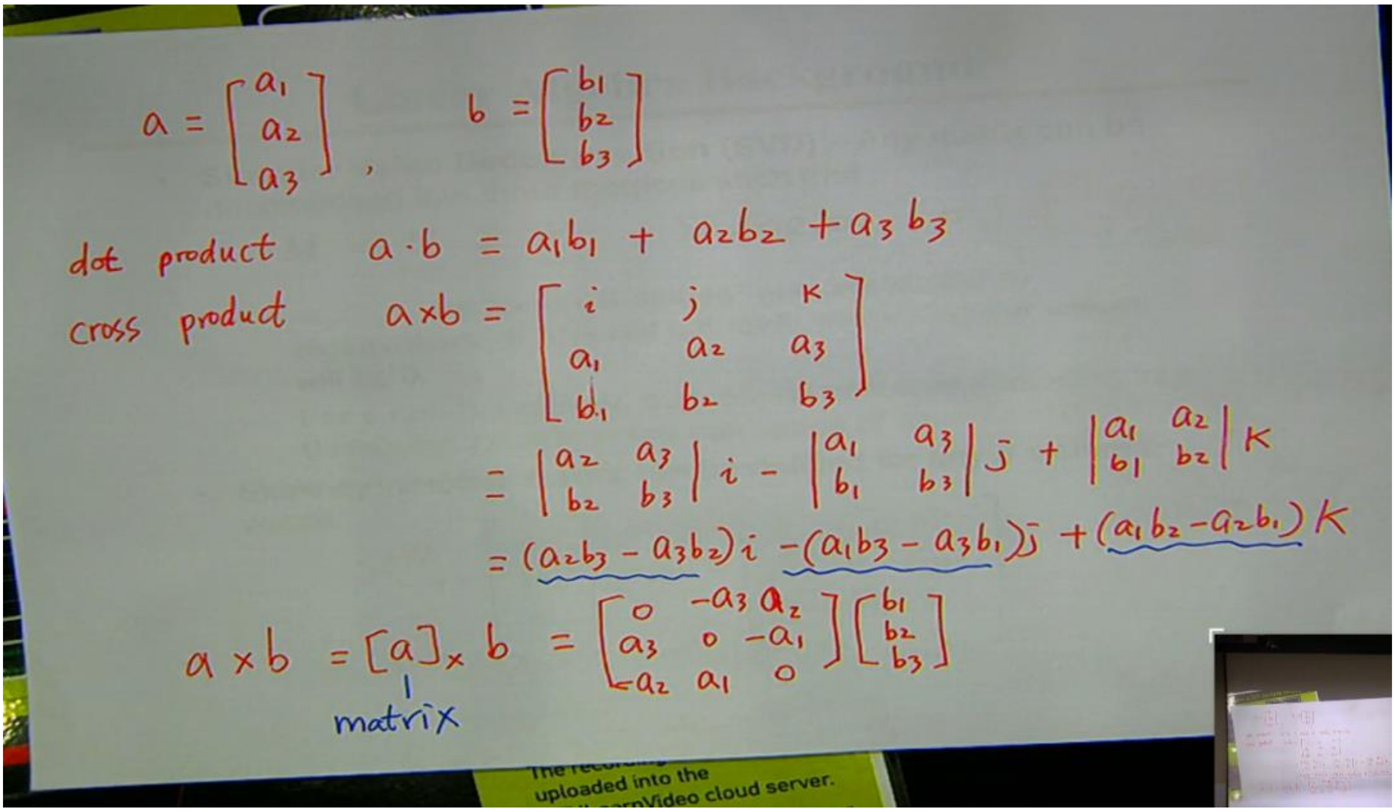
Skew symmetric matrix

we want to simplify the cross mul

Skew-symmetric matrix can be defined for any 3 element vector

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \Rightarrow \overset{\text{def}}{[a]_\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

– $[a]_x$ has the property that $[a]_x b = a \times b$



SFM1:
Essential matrix and eight-point algorithm

对于图中任何一个识别出来的点p能够从归一化坐标变换为像素坐标

$$p=\begin{bmatrix}u\\v\\1\end{bmatrix}=Am=\begin{bmatrix}f\sigma_x & -f\sigma_x\tan\alpha & u_0\\0 & f\sigma_y\sec\alpha & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}m_x\\m_y\\1\end{bmatrix}$$

当然也可以反着求

$$m=A^{-1}p$$

因为信息量不足，无法求世界坐标

相机在两个位置观测同一组点{p}，都满足世界坐标的变换关系

$$\overline{m}_j^* = \begin{bmatrix} x_j^* & y_j^* & z_j^* \end{bmatrix}^T, \quad \overline{m}_j = \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^T, \quad \forall j \in \{1 \ldots N\}$$

The coordinates of each point in the camera frames are related by

$$\bar{m}_j = R\bar{m}_j^* + x, \ \forall j \in \{1 \ldots N\}$$

重要约束是不存在4点共面

$E = [x]_\times R$ is the Essential Matrix a

$$m_j^T E m_j^* = 0$$

holds for all points in the images, and is known as the Essential Constraint or Epipolar Constraint.

The null space of M will give a vector e that satisfies the equation

$$Me = 0$$

无穷多解

一般M里有8个点，可以求出无穷多e也就是E，我们取归一化的值（然后可以分离出R和x，不考）

现在解决relative structure
也就是估计相对距离lambda

week 10.2

2024年11月1日 22:54

Homography Matrix
SFM系列方法之二，老师更喜欢这个方法，
单应性矩阵
重要约束，4点共面不共线

两个相机位置其中之一的坐标原点到特征点平面的距离d和各点之间的关系有

all $\overline{m}_j{}^*$

$$d^* = n^{*T}\overline{m}_j^*$$

其中n是垂直于平面的单位向量

- Combine equations

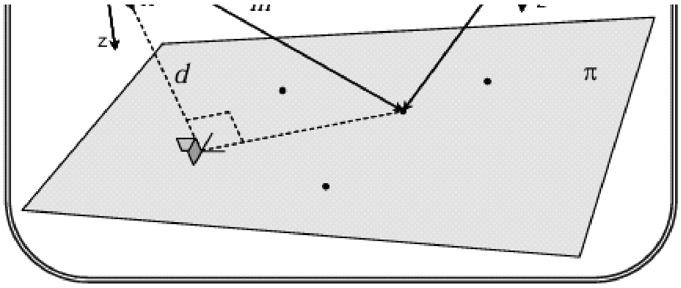
$$\overline{m}_j(t) = R(t)\overline{m}_j^* + x(t)$$

$$d^* = n^{*T}\overline{m}_j^*$$

对上面公式，刻意的在x的系数1上变为d*/d*
这样能提取mj*出来，里面保留了R x d n的信息，
并且两个zj放到一起了，就有了深度比例

to get

Loss of scale information



$$m_j = \frac{z_j^*}{z_j}\left(R + \frac{x}{d^*} n^{*T}\right) m_j^*$$

$$= \alpha_j H m_j^*$$

- $\boxed{H = R + \frac{x}{d^*} n^{*T}}$ is the **Euclidean Homography Matrix** that contains relative pose information and some structure information
- $\alpha_j = \frac{z_j^*}{z_j}$ is a ratio of depths (called depth ratio)

这里是老师说的重点

构造了一个M以及变换H为h，满足

- "Unwrap" $H_n \in \mathbb{R}^{3x3}$ to a vector $h \in \mathbb{R}^8$

$$H_n = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \longrightarrow h = [h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8]^T$$

- Define a 2x8 matrix M_j for the j^{th} feature point

$$M_j = \begin{bmatrix} m_x^*, m_y^*, 1, 0, 0, 0, -m_x^* m_x, -m_y^* m_x \\ 0, 0, 0, m_x^*, m_y^*, 1, -m_x^* m_y, -m_y^* m_y \end{bmatrix}$$ such that $M_j h = m_j$

- Concatenate all N vectors m_j and matrices M_j to form matrix $M \in \mathbb{R}^{2N\text{x}8}$

$$Mh = m$$

Week 11

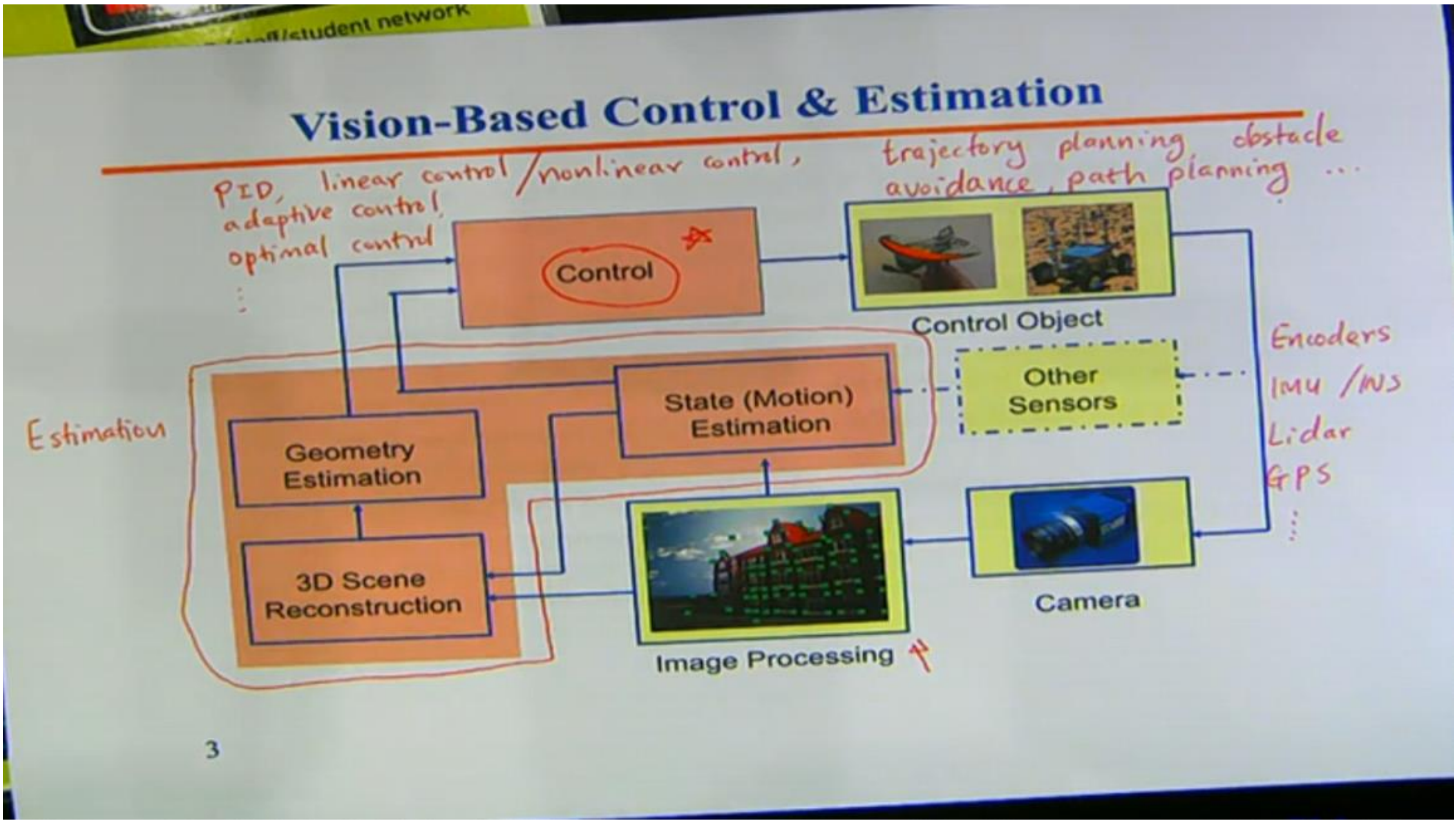
2024年11月7日 2:51

Kalman Filter

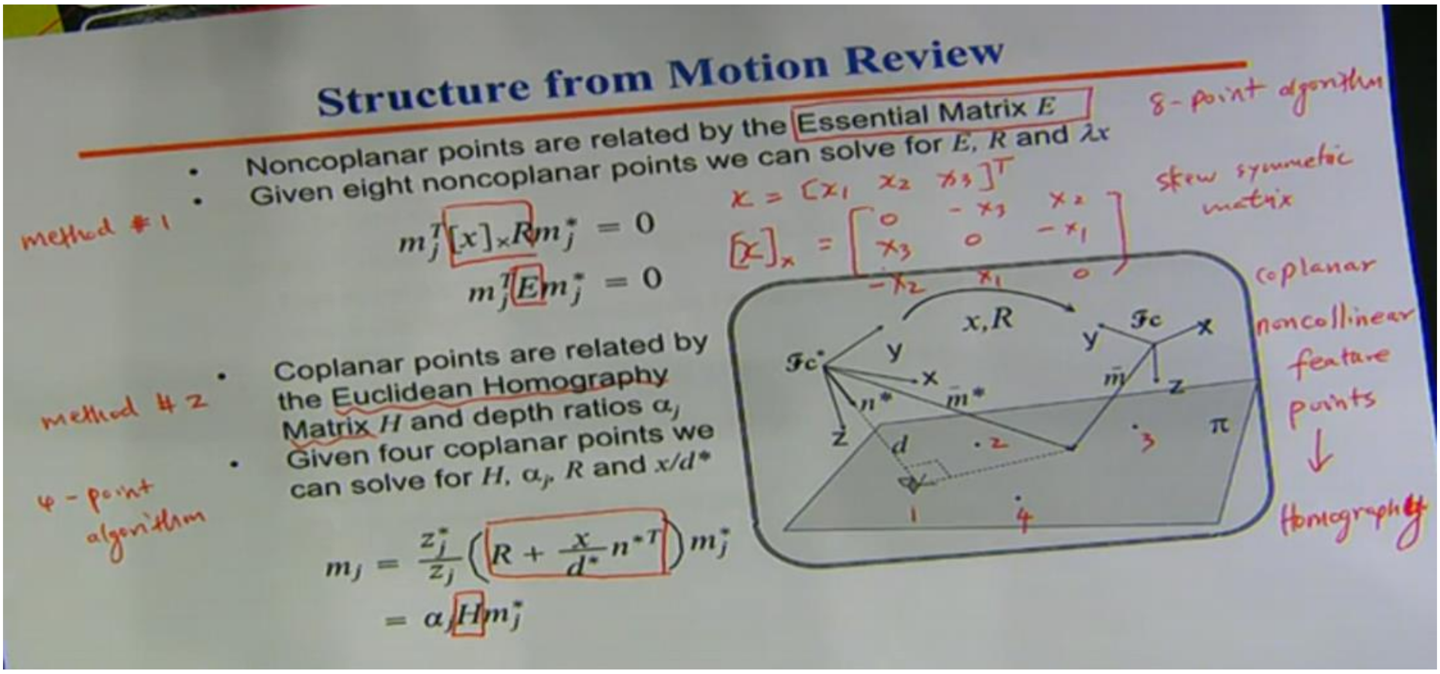
Week 12

2024年11月7日 4:42

Vision-Based Control & Estimation



视觉伺服



PBVS:

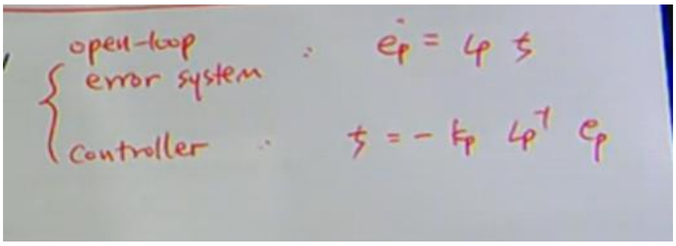
姿态误差有

$$e_p(t) = [x(t)^T, u(t)^T\theta(t)]^T \qquad x(t) \in \mathbb{R}^3, \quad u(t)\theta(t) \in \mathbb{R}^3$$

根据开环误差，姿态时间导数有

$$\dot{e}_p = \underbrace{L_p}_{\text{Jacobian}}\xi \qquad \xi(t) = [v(t)^T, \omega^T(t)]^T \in \mathbb{R}^6$$

控制器有



由此闭环误差导数

- Closed loop error dynamics given by

$$\begin{aligned} \dot{e}_p &= L_p\xi \\ &= L_p(-k_p L_p^{-1} ep) \\ &= -k_p e_p \end{aligned}$$

以上可以被证明是稳定的系统

PBVS:
优点： camera has good 3D trajectory
缺点： feature points may leave field of view

IBVS:

Image Based Visual Servoing

- Rewrite previous equation in matrix form

$$\dot{m}_x = -\frac{1}{Z}v_x + \frac{m_x}{Z}v_z + m_x m_y \omega_x - (1 + m_x^2)\omega_y + m_y \omega_z$$
$$\dot{m}_y = -\frac{1}{Z}v_y + \frac{m_y}{Z}v_z + (1 + m_y^2)\omega_x - m_x m_y \omega_y - m_x \omega_z$$

$$\Downarrow$$

$$\begin{bmatrix} \dot{m}_x \\ \dot{m}_y \end{bmatrix} = \begin{bmatrix} \frac{1}{Z} & 0 & \frac{m_x}{Z} & m_x m_y & -(1+m_x^2) & m_y \\ 0 & \frac{1}{Z} & \frac{m_y}{Z} & (1+m_y^2) & -m_x m_y & m_x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Needs depth estimate

$$\dot{m}_j = L_{ij}\xi$$

- L_{ij} is the Image Jacobian or Interaction Matrix for point j

我们想要控制的是feature point像素坐标的速度值

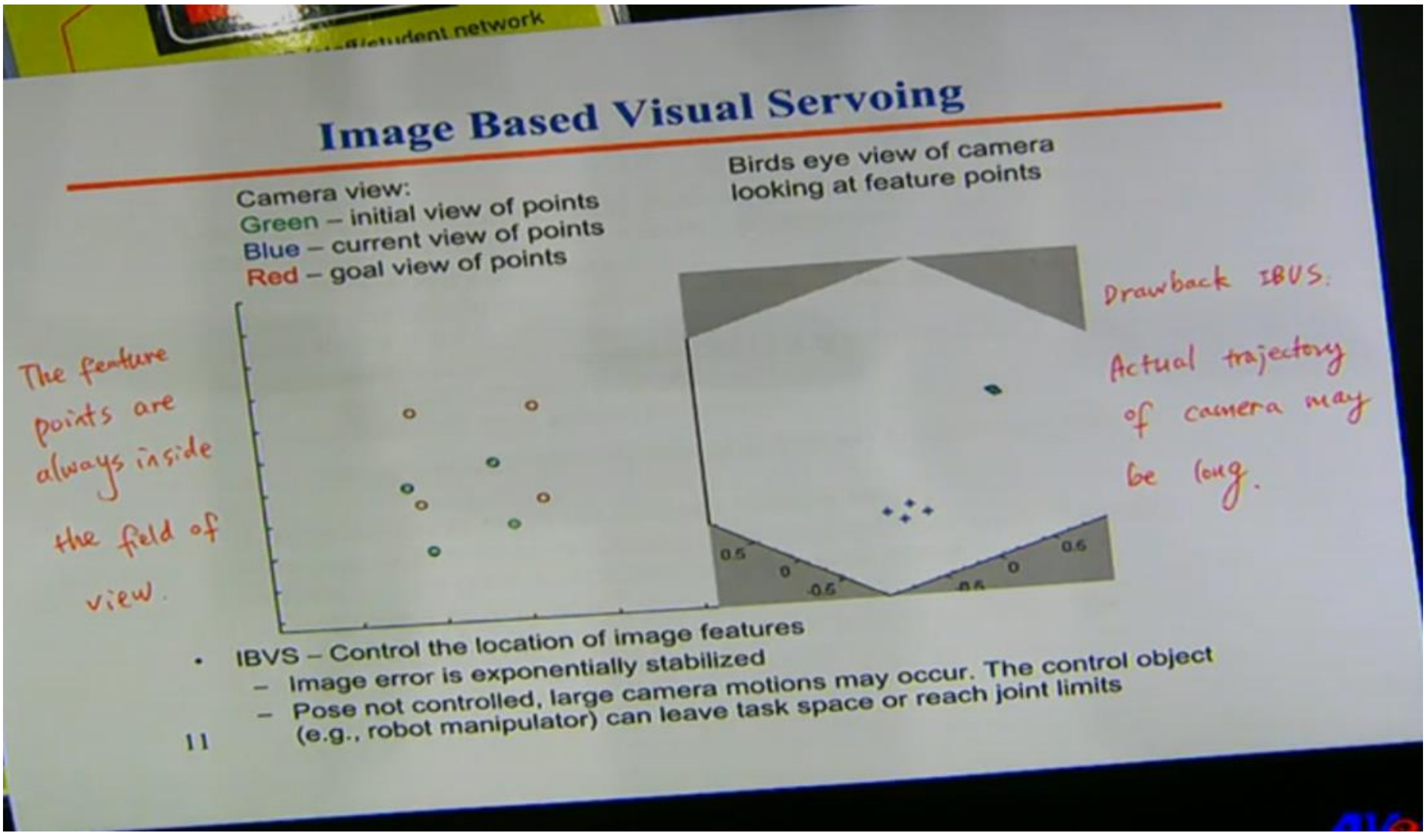


Image Based Visual Servoing
Camera view:
Green – initial view of points
Blue – current view of points
Red – goal view of points
Birds eye view of camera
looking at feature points
0.5
0
-0.5
0.5
0
-0.5
The feature points are always inside the field of view.
Drawback IBVS:
Actual trajectory of camera may be long.
• IBVS – Control the location of image features
– Image error is exponentially stabilized
– Pose not controlled, large camera motions may occur. The control object (e.g., robot manipulator) can leave task space or reach joint limits
11