

*2022 Spring*

# 计算机视觉

袁丁

北京航空航天大学

图像中心

[dyuan@buaa.edu.cn](mailto:dyuan@buaa.edu.cn)



# 第五章

## 运动恢复结构 (Structure from Motion)

# 本章主要知识点

- 什么是多视角立体视觉
- 什么是运动恢复结构
- 运动恢复结构的步骤
  - ✓ 图像特征的提取与匹配
  - ✓ 相机自标定
  - ✓ 运动参数的估计
  - ✓ 三维点云的生成
  - ✓ 集束调整

# 立体视觉

- 立体视觉:

问题核心： 给定同一目标（或场景）的一系列图像，计算得到其三维形状的表达

三维表示：深度图（Depth maps）、网格（Meshes）、点云（**Point Clouds**）、面片云（Patch Clouds）、体积模型（ Volumetric Models ）等。

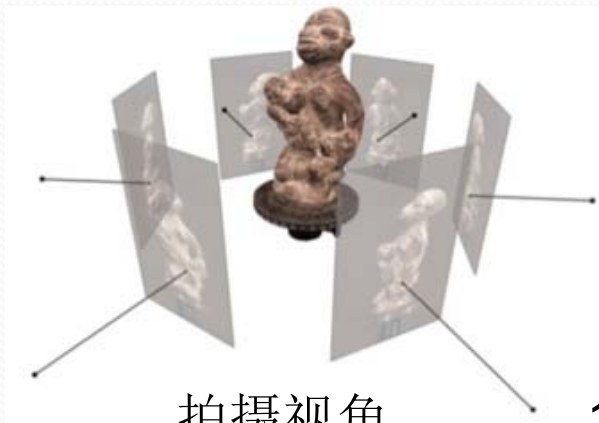




# 多视角立体视觉

- 多视角立体视觉（Multiple View Stereo, MVS）：

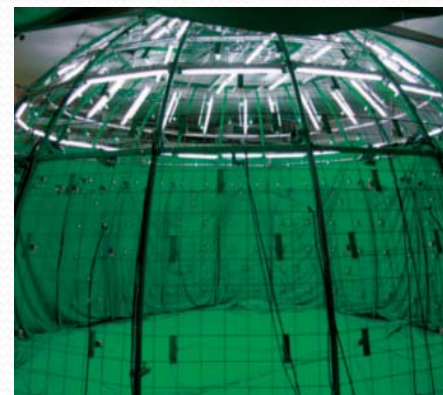
是对立体视觉的推广，能够在多个视角观察和获取景物的图像，并以此完成匹配和深度估计。



拍摄视角



120台松下GH4打造的3D照相馆

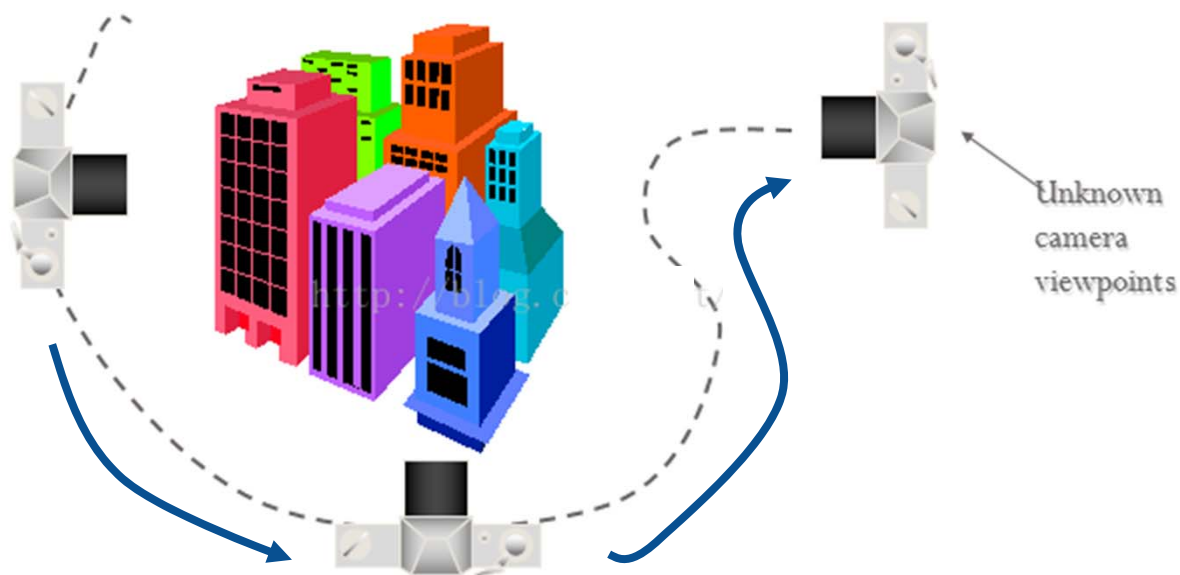


清华数据采集系统



得到的图像序列

# 什么是运动恢复结构



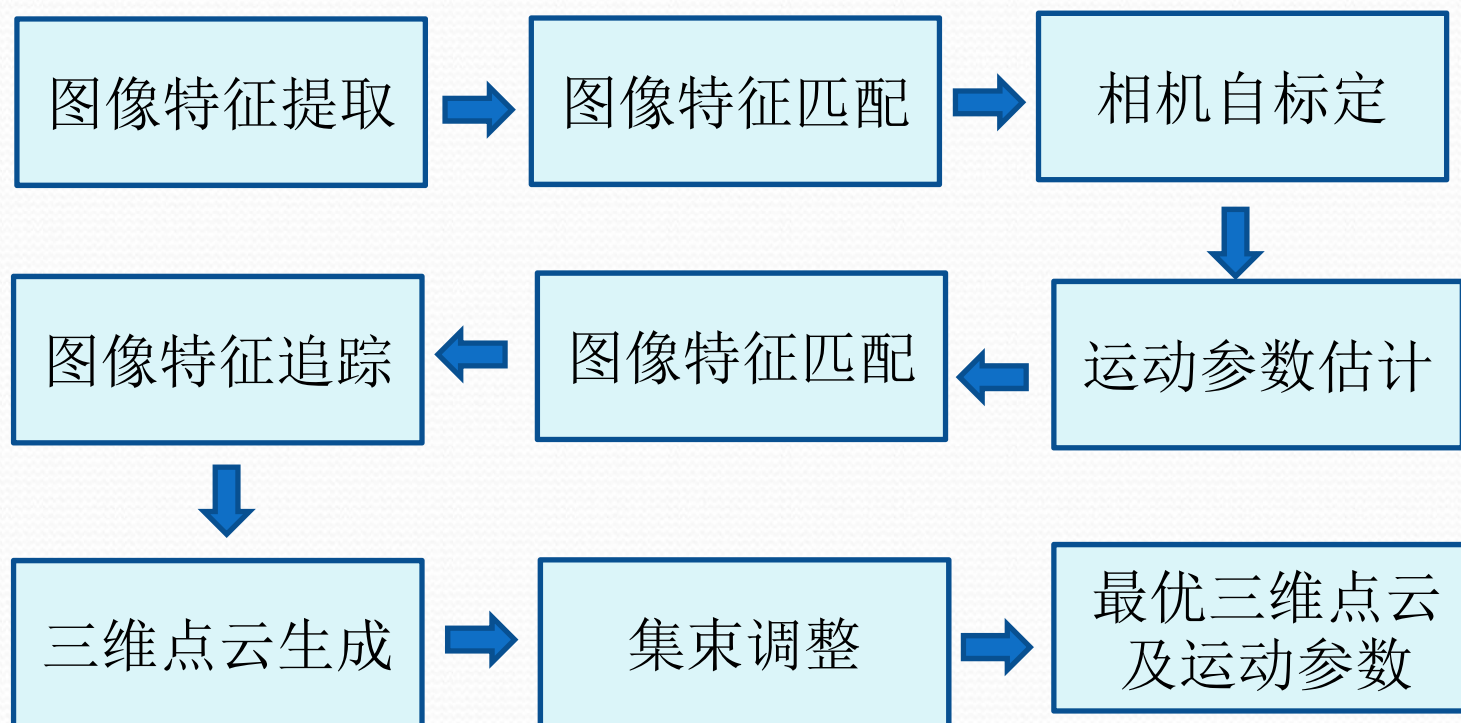
**运动恢复结构（Structure from Motion, SFM）：** 通过运动的相机拍摄图像序列，从序列图像中重建目标三维结构，并估计相机模型及其运动参数。

本质上，也是利用两幅以上的场景（或目标）图像，自动恢复场景（或目标）三维结构。

难点在于：需要对相机进行自标定（相机内部参数矩阵），以及同时获得相机的运动信息（即，不同视点之间相机坐标系变换矩阵）。

# 运动恢复结构的步骤

- 典型的运动恢复结构算法涉及到下面的若干个步骤：





# 图像特征的提取与匹配

- ① 在图像中提取关键点，提取良好的、易于正确匹配的特征  
常用特征：SIFT、SURF 等。
- ② 对于相邻两视图提取特征点后，对特征点进行匹配。  
常用匹配策略：KD-Tree 匹配搜索策略。
- ③ 通过匹配的SIFT点对，利用 8点法，计算相邻两视图间的基本矩阵  $\mathbf{F}$ 。  
匹配点对的数量越多、匹配精度越高， $\mathbf{F}$  的计算结果越精确。



# 相机自标定

- 什么是相机自标定？

摄像机自标定是指不需要标定块，仅仅通过图像点之间的匹配关系对相机进行内部参数矩阵  $K$  标定的过程。

- 优点、缺点

优点：灵活，方便

缺点：精度不太高，鲁棒性不足

- SFM中的相机内部参数矩阵的计算

可在拍摄前，固定相机焦距，利用标定板对相机内参进行标定，拍摄过程中，相机焦距不可再调节。且各视点与景物距离要相差不大。

若不利用标定板，则采用相机自标定的方法。

# 相机自标定

- 假定图像点之间的对应关系已经确定 (输入: 视图间的匹配点对)
- 一般来说, 认为在拍摄不同图像时, 摄像机的内参数没有发生变化, 即  $\mathbf{K}$  不变:

$$K = \begin{pmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

- 原理: 利用绝对二次曲线在图像平面的投影 (IAC) 作为 相机自标定的参考标定物

绝对二次曲线: 无穷远平面上的一条二次曲线



# 相机自标定

- 通过绝对二次曲线的概念，建立关于摄像机内参数矩阵的约束方程，称之为Kruppa 方程：

$$[e]_{\times}^T \omega [e]_{\times} = \lambda F^T \omega F \quad \lambda: \text{任意不为零常数}$$

$$\omega = K K^T$$

上式中，基本矩阵  $F$  已知，则内极点  $e$  已知， $\omega$  为未知数

$\omega$  有 5 个独立未知变量（对称阵，且 up to scale）

每个 Kruppa 方程最多可以提供 2 个关于未知变量的独立约束。

至少需要 3 对图像来标定相机，且相机的内参数须保持不变。

注：由于课时限制，本课程省略推导 Kruppa 方程的步骤。  
相机自标定内容仅做概念性了解。

# 运动参数估计

- 目的:

估计连续相邻视图所对应的相机坐标系的变换关系:  $\mathbf{R}$  和  $\mathbf{T}$ 。

- 系统输入: 基本矩阵  $\mathbf{F}$  与 内参矩阵  $\mathbf{K}$

由:  $\mathbf{F} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1}$

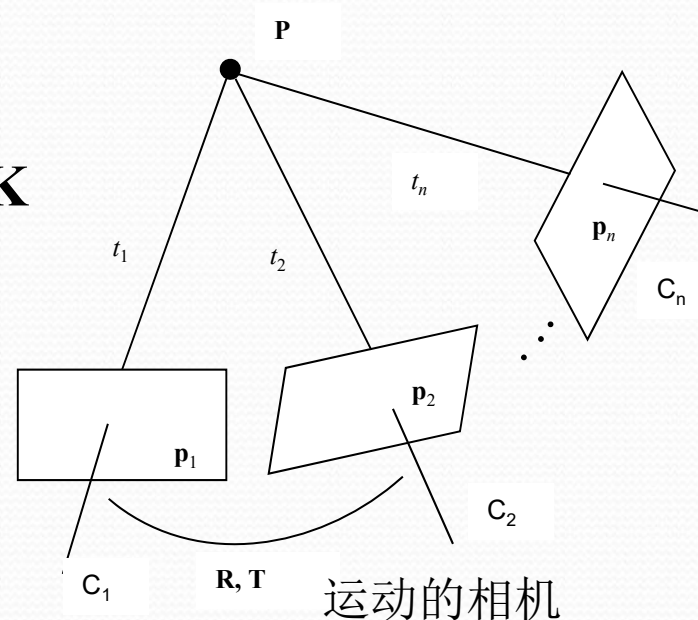


$$\mathbf{E} = \mathbf{K}^T \mathbf{F} \mathbf{K}$$

引入中间变量:  $\mathbf{E}$

$$\mathbf{E} = [\mathbf{T}]_{\times} \mathbf{R}$$

将  $\mathbf{E}$  分解得到运动参数 ---  $\mathbf{R}$ ,  $\mathbf{T}$





# R、T 的线性求解方法

- 如何由  $\mathbf{E} \rightarrow \mathbf{R}, \mathbf{T}$  ?

**定理:** 设本质矩阵  $\mathbf{E} = \mathbf{S}\mathbf{R}$  的 **SVD** (奇异值) 分解是:

$$\mathbf{E} = \mathbf{U} \operatorname{diag}(1, 1, 0) \mathbf{V}^T$$

则存在两种可能的分解 (不考虑本质矩阵的符号):

$$\mathbf{S} = \mathbf{U}\mathbf{Z}\mathbf{U}^T, \quad \mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^T \text{ 或者 } \mathbf{R} = \mathbf{U}\mathbf{W}^T\mathbf{V}^T$$

其中:

$$\mathbf{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# R、T 的线性求解方法

$$\text{由 } \mathbf{E} = \mathbf{S}\mathbf{R} \rightarrow \mathbf{S} = [\mathbf{T}]_x \rightarrow \mathbf{S}\mathbf{T} = 0$$

由于缺少景物空间尺度信息，平移分量 $\mathbf{T}$ 的模无法仅由图像点的匹配点对求出。

问题：求解  $\min(\|\mathbf{S}\mathbf{T}\|)$ ，s.t.  $\|\mathbf{T}\| = 1$

以上问题等效为：求解  $\min(\|\mathbf{U}\mathbf{Z}\mathbf{U}^T\mathbf{T}\|)$ ，s.t.  $\|\mathbf{T}\| = 1$

由于： $\mathbf{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ ，当  $\mathbf{U}^T\mathbf{T} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$  时，得到上述表达式最小值。

因此： $\mathbf{T} = \mathbf{U}(0,0,1)^T = \mathbf{u}_3$

由于 $\mathbf{E}$ 的符号不能确定，因此  $\mathbf{T} = -\mathbf{u}_3$  也成立。



# R、T 的线性求解方法

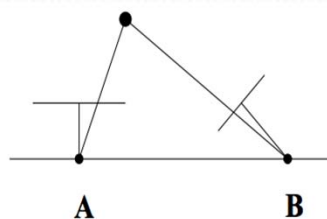
**推论：** 设本质矩阵  $\mathbf{E} = \mathbf{U} \text{diag}(1, 1, 0) \mathbf{V}^T$ ，以及相机第 1 个视点投影矩阵  $\mathbf{P} = [\mathbf{I} | \mathbf{0}]$ ，对于摄像机相机第 2 个视点的投影矩阵  $\mathbf{P}'$  有 4 种可能

$$\mathbf{P}_2 = [\mathbf{UWV}^T | +\mathbf{u}_3]$$

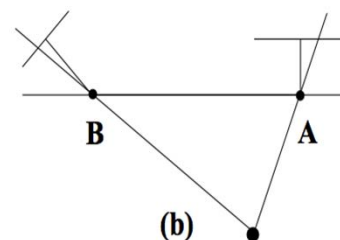
$$\mathbf{P}_2 = [\mathbf{UWV}^T | -\mathbf{u}_3]$$

$$\mathbf{P}_2 = [\mathbf{UW}^T \mathbf{V}^T | +\mathbf{u}_3]$$

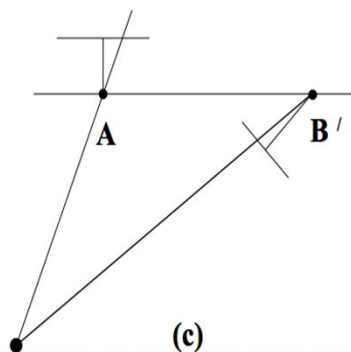
$$\mathbf{P}_2 = [\mathbf{UW}^T \mathbf{V}^T | -\mathbf{u}_3]$$



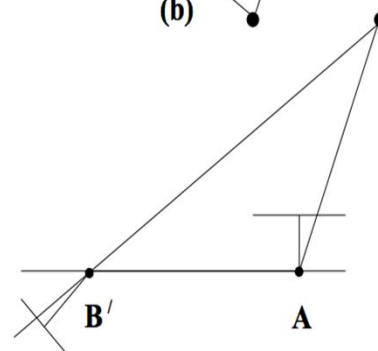
(a)



(b)



(c)



(d)

利用投影矩阵  $\mathbf{P}$  与  $\mathbf{P}'$  和图像一对匹配点对算出空间点，若计算出的空间点在两个摄像机之前，则对应的  $\mathbf{P}'$  为所求的投影矩阵。

# R、T 的线性求解方法

- 假设第1幅图像的相机坐标系与世界坐标系重合，第2幅图像的相机坐标系与世界坐标系之间的转换关系由相机外参数矩阵  $[\mathbf{R}|\mathbf{t}]$  给出，则世界坐标系与第2幅图像的相机坐标系之间的转换关系式可重写为：

$$\begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \end{bmatrix} + \mathbf{t} \quad \Rightarrow \quad \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \end{bmatrix} = \mathbf{R}^{-1} \left( \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} - \mathbf{t} \right) = \mathbf{R}^T \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} - \mathbf{R}^T \mathbf{t}$$

- 第2幅图像的相机光心  $\mathbf{C}$  在世界坐标系下的坐标为：

$$\mathbf{C} = \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \end{bmatrix} = \mathbf{R}^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \mathbf{R}^T \mathbf{t} = -\mathbf{R}^T \mathbf{t}$$

- 第2幅图像（归一化图像）的主点  $\mathbf{c}$  在世界坐标系下的坐标为：

$$\mathbf{c} = \mathbf{R}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \mathbf{R}^T \mathbf{t}$$



# R、T 的线性求解方法

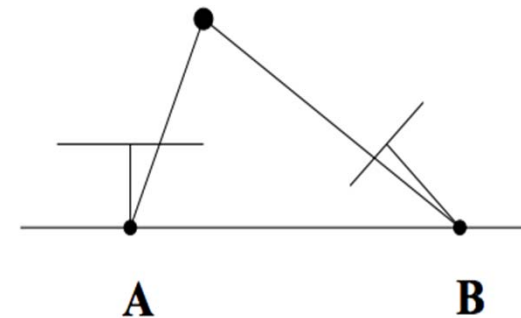
- 第 2 幅图像的相机的光轴方向为:

$$\mathbf{c} - \mathbf{C} = \left( \mathbf{R}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \mathbf{R}^T \mathbf{t} \right) - (-\mathbf{R}^T \mathbf{t}) = \mathbf{R}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{R}(3,:) ^T$$

- 选取一对匹配点对，计算其对应的空间中三维点  $\mathbf{X}$ ，在第1个视点中，需要满足  $\mathbf{X}$  与第 1 个相机的光轴夹角小于  $90^\circ$ ：  $\mathbf{X} \cdot [0 \ 0 \ 1]^T > 0$
- 从第 2 幅图像的相机中心  $\mathbf{C}$  到点  $\mathbf{X}$  的方向向量为： $(\mathbf{X}-\mathbf{C})$ 。为了确定  $\mathbf{X}$  点是否在相机的光轴正方向，只需要检测向量  $(\mathbf{X}-\mathbf{C})$  与  $\mathbf{R}(3,:)^T$  的夹角是否小于  $90^\circ$ ：

即:  $(\mathbf{X}-\mathbf{C}) \cdot \mathbf{R}(3,:)^T > 0$

满足上式的  $[\mathbf{R}|\mathbf{t}]$  则是第2个视点的投影矩阵。



# 极线校正、特征匹配、特征追踪

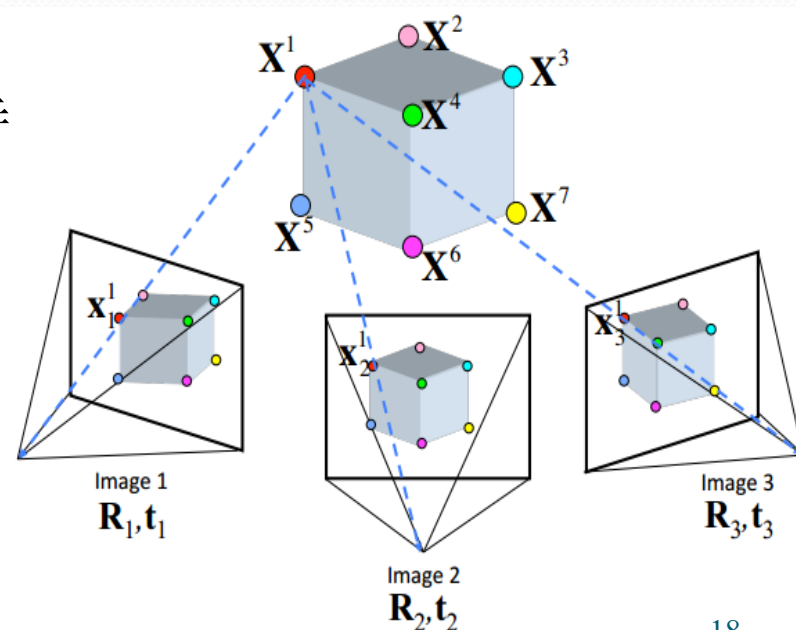
- ① 根据计算得到的相机参数矩阵，将两两相邻视图为一组，进行极线校正。
- ② 在图像中提取Harris特征，在校正后的两两相邻视图之间，进行SIFT与Harris特征匹配。

两种特征搭配使用，保证尽量可以得到更多的匹配点对。

- ③ 特征追踪：

建立匹配特征点后，进行鲁棒匹配特征点的筛选、去噪声。

SFM系统中，相邻两视点的基线长度较短，空间3维点会在相邻的几幅图像中成像。

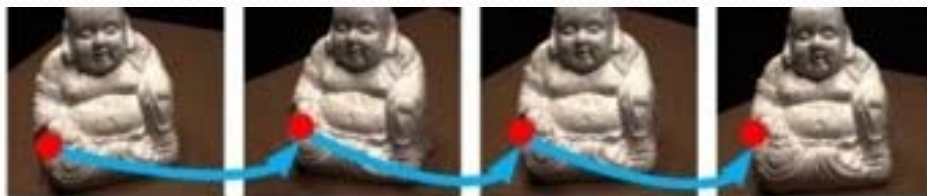




# 极线校正、特征匹配、特征追踪

## ③ 特征追踪:

- 在相邻的连续视图中考察匹配特征点对，将同一空间点的投影特征点串起来，构成特征串。
- 特征串的长度（即特征串中包含特征的数目）是反映特征串可靠性的重要指标。
- 若特征串较长，说明该特征的区分度和显著性较高，匹配鲁棒性高，由其生成的空间3D点更可靠。



特征串长度为：4



特征串长度为：3

# 三维点云生成

- 空间三维点云的计算

由视线的概念：

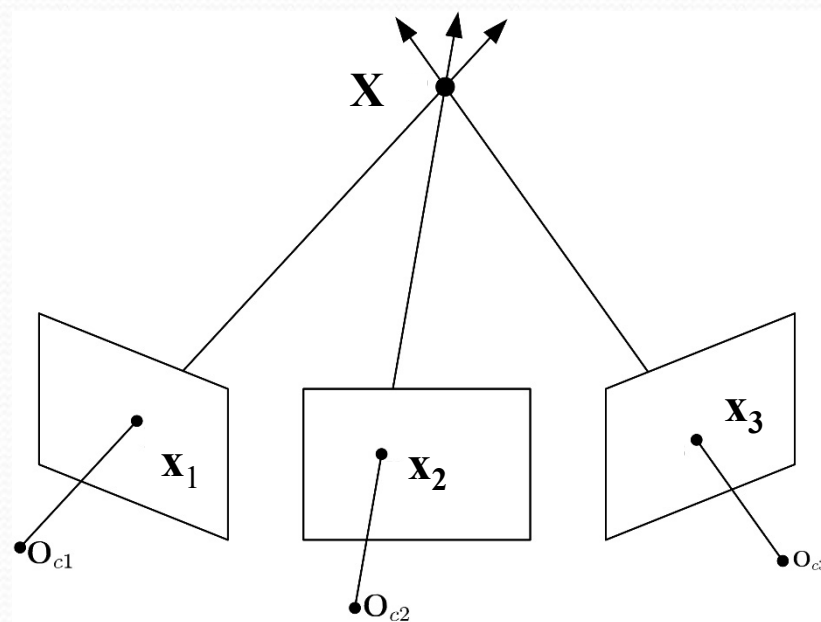
假设某一个特征串中有3个特征， $\mathbf{X}$  点由直线 $O_{c1}x_1$ 、 $O_{c2}x_2$ 、 $O_{c3}x_3$ 的交点得到。计一个特征串：

$$t = \left\{ \mathbf{x}_i = (u_i, v_i)^T \mid i = 1, 2, \dots, l \right\}$$

特征串中一个特征  $\mathbf{x}_i = (u_i, v_i)^T$ ：

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \mathbf{P}_i \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad i = 1, 2, \dots, l$$

$$\mathbf{P}_i = \begin{bmatrix} p_i^{11} & p_i^{12} & p_i^{13} & p_i^{14} \\ p_i^{21} & p_i^{22} & p_i^{23} & p_i^{24} \\ p_i^{31} & p_i^{32} & p_i^{33} & p_i^{34} \end{bmatrix}$$



空间点三维坐标计算原理



# 三维点云生成

由上式可得：

$$\begin{cases} (u_i p_i^{31} - p_i^{11})X + (u_i p_i^{32} - p_i^{12})Y + (u_i p_i^{33} - p_i^{13})Z = p_i^{14} - u_i p_i^{34} \\ (v_i p_i^{31} - p_i^{21})X + (v_i p_i^{32} - p_i^{22})Y + (v_i p_i^{33} - p_i^{23})Z = p_i^{24} - v_i p_i^{34} \end{cases}$$

通过特征串  $t = \{\mathbf{x}_i = (u_i, v_i)^T \mid i = 1, 2, \dots, l\}$ ，可构造  $2l$  个线性方程求解  $X$ 、 $Y$ 、 $Z$ 。

因此可以构造关于  $\mathbf{s} = (X, Y, Z)^T$  的超定方程组：

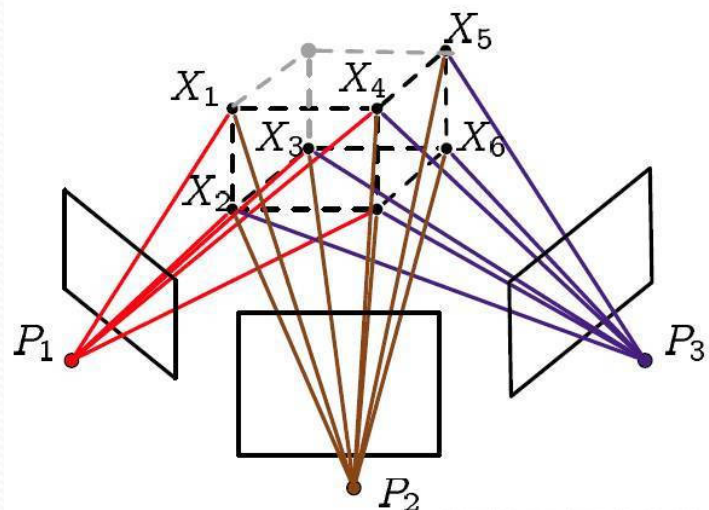
$$\mathbf{A}\mathbf{s} = \mathbf{b}$$

并利用最小二乘求最优解：

$$\mathbf{s} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

# 集束调整 (Bundle Adjustment)

- **集束调整（光束平差）**：是一项可以同时重建的三维空间点与摄像机的运动信息进行优化的技术，也是在无法避免噪声的图像中，利用特征点进行三维重建时，最常用的优化方法。
- **Bundle**：来源于bundle of light，本意指光束，即，三维空间中的点投影到像图像平面上的光束。
- **Bundle Adjustment**：本质是一个优化模型，目的是最小化重投影误差



- ✓ 第一次投影：相机在拍照的时候，三维空间点投影到图像上
- ✓ 利用图像上匹配点对，根据相机内、外参矩阵进行三角化（**triangulation**），估计三维空间点的位置，
- ✓ 最后利用计算得到的三维点的位置坐标和计算得到的相机投影矩阵进行图像上第二次模拟投影，即：**重投影**



# 集束调整 (Bundle Adjustment)

- 重投影误差数学上的描述:

$O_j$  : 第  $j$  幅图像光心位置

$\mathbf{R}_j$  、  $\mathbf{t}_j$  : 估计得到的第  $j$  幅图像相机外部参数

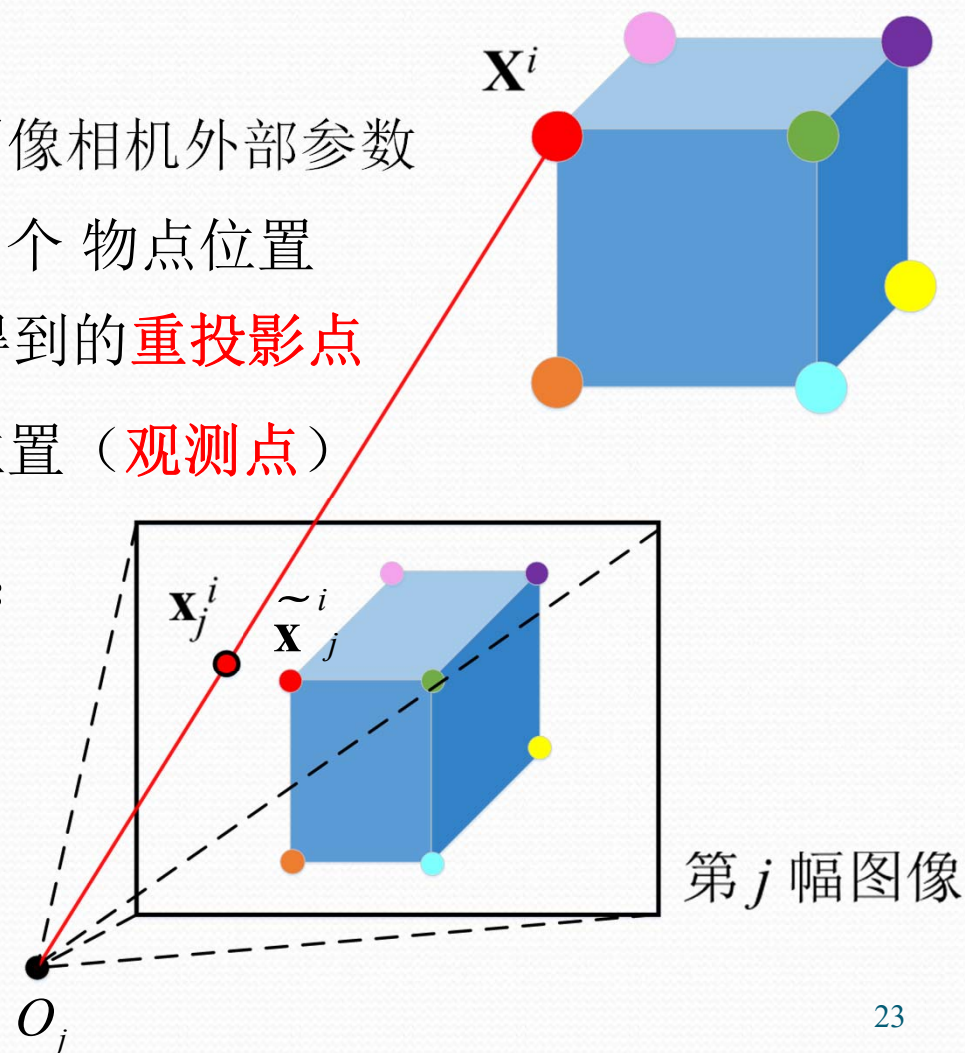
$\mathbf{X}^i$  : 估计得到的三维空间第  $i$  个物点位置

$\mathbf{x}_j^i$  : 由  $\mathbf{X}^i$  与  $\mathbf{R}_j$  、  $\mathbf{t}_j$  计算得到的重投影点

$\tilde{\mathbf{x}}_j^i$  :  $\mathbf{X}^i$  在图像中检测到的位置 (观测点)

$\mathbf{X}^i$  在  $O_j$  视点上的重投影误差:

$$\begin{aligned} E_j^i &= \left\| \tilde{\mathbf{x}}_j^i - \mathbf{x}_j^i \right\|^2 \\ &= \left\| \tilde{\mathbf{x}}_j^i - \mathbf{K} \left[ \mathbf{R}_j \mid \mathbf{t}_j \right] \mathbf{X}^i \right\|^2 \end{aligned}$$



# 集束调整 (Bundle Adjustment)

各视图的摄像机外部参数矩阵： $\mathbf{R}_j$ 、 $\mathbf{t}_j$

假设拍摄过程中相机内参矩阵  $\mathbf{K}$  保持不变

例：空间恢复的三维点在下面各视图的**重投影点**为：

	空间点 1	空间点 2	空间点 3
图像 1	$\mathbf{x}_1^1 = \mathbf{K}[\mathbf{R}_1   \mathbf{t}_1]\mathbf{X}^1$	$\mathbf{x}_1^2 = \mathbf{K}[\mathbf{R}_1   \mathbf{t}_1]\mathbf{X}^2$	
图像 2	$\mathbf{x}_2^1 = \mathbf{K}[\mathbf{R}_2   \mathbf{t}_2]\mathbf{X}^1$	$\mathbf{x}_2^2 = \mathbf{K}[\mathbf{R}_2   \mathbf{t}_2]\mathbf{X}^2$	$\mathbf{x}_2^3 = \mathbf{K}[\mathbf{R}_2   \mathbf{t}_2]\mathbf{X}^3$
图像 3	$\mathbf{x}_3^1 = \mathbf{K}[\mathbf{R}_3   \mathbf{t}_3]\mathbf{X}^1$		$\mathbf{x}_3^3 = \mathbf{K}[\mathbf{R}_3   \mathbf{t}_3]\mathbf{X}^3$

- 其中， $\mathbf{x}_j^i$  表示的是空间点  $\mathbf{X}^i$  在第  $j$  幅图像中的**重投影点**。
- 在匹配过程中提取的匹配特征点可以称为**观测点**，使用  $\tilde{\mathbf{x}}$  表示点  $\mathbf{x}$  的观测点，上表中各点的**观测点**可以表示为：

$$\begin{bmatrix} \tilde{\mathbf{x}}_1^1 & \tilde{\mathbf{x}}_1^2 & \\ \tilde{\mathbf{x}}_2^1 & \tilde{\mathbf{x}}_2^2 & \tilde{\mathbf{x}}_2^3 \\ \tilde{\mathbf{x}}_3^1 & & \tilde{\mathbf{x}}_3^3 \end{bmatrix}$$



# 集束调整 (Bundle Adjustment)

- 集束调整:

集束调整就是将所有空间3维点的重投影误差的 **和** 进行最小化

重投影误差:

$$E = \sum_j \sum_i \sigma_{ij} \left\| \tilde{\mathbf{x}}_j^i - \mathbf{K} [\mathbf{R}_j | \mathbf{t}_j] \mathbf{X}^i \right\|^2$$

其中, 当  $\mathbf{X}^i$  在  $O_j$  中有投影, 则  $\sigma_{ij}=1$ , 若  $\mathbf{X}^i$  在  $O_j$  中无投影, 则  $\sigma_{ij}=0$ 。

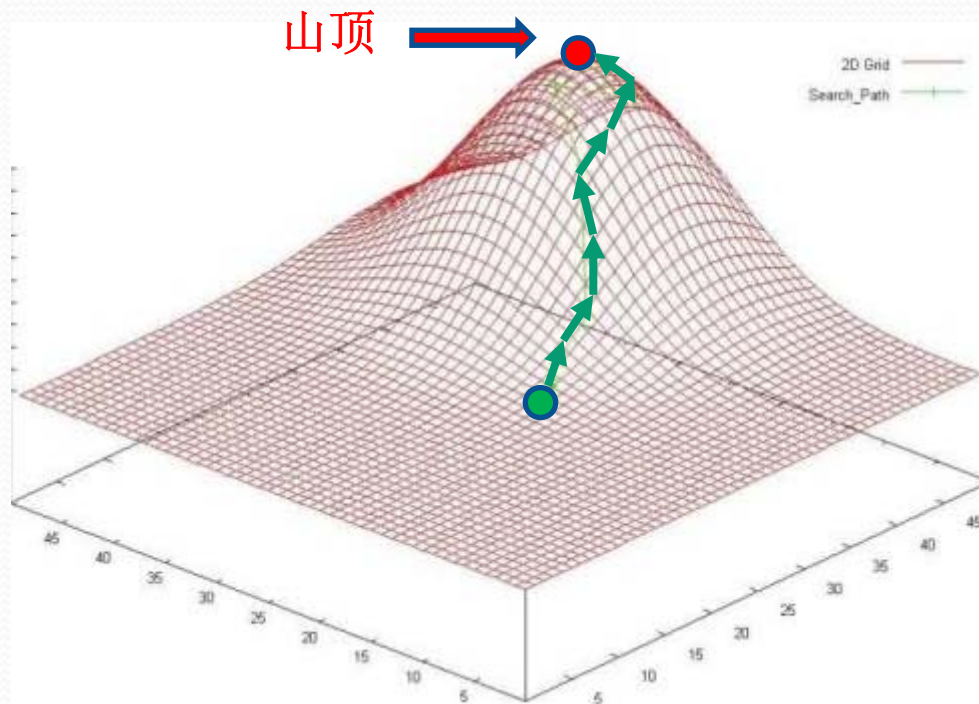
集束调整:

$$\min_{\mathbf{R}_j, \mathbf{t}_j, \mathbf{X}^i} \sum_j \sum_i \sigma_{ij} \left\| \tilde{\mathbf{x}}_j^i - \mathbf{K} [\mathbf{R}_j | \mathbf{t}_j] \mathbf{X}^i \right\|^2$$

通过求取上式的最小值, 获得最优的  $\mathbf{X}^i$ 、 $\mathbf{R}_j$ 、 $\mathbf{t}_j$ 。

# 集束调整 (Bundle Adjustment)

- 集束调整所采用的迭代优化方法：**LM (Levenberg-Marquardt)** 算法
  - ✓ 无约束的迭代优化算法
  - ✓ 通过加入阻尼因子，解决一阶Newton法无法保证每次迭代时函数保持下降的问题。常用于解决非线性最小二乘问题。



寻优过程类似爬山。

爬山算法：算法从一个随机的起点，根据函数梯度信息，判断爬坡的下一步，不断爬升直到最高点（最大值）的迭代过程。



# 集束调整 (Bundle Adjustment)

- LM (Levenberg-Marquardt) 算法的思想

对于函数  $\mathbf{Y} = f(\mathbf{X})$ , 其中,  $\mathbf{Y}$  是  $n$  维向量,  $\mathbf{X}$  是  $m$  维向量。

假设  $\mathbf{Y}$  是真值  $\bar{\mathbf{Y}}$  的测量值。由于测量误差与系统误差的出现, 并不存在变量  $\mathbf{X}$ , 使得函数  $\mathbf{Y} = f(\mathbf{X})$  严格成立。

因此, 仅能最优的  $\mathbf{X}$ , 使得关系式最佳近似满足。

以上问题即为求解最小化:

$$\min_{\mathbf{X}} \|\mathbf{Y} - f(\mathbf{X})\|$$

LM算法是一阶Newton法的一种改进。



# 集束调整 (Bundle Adjustment)

- 一阶Newton法

给定一个初始解  $\mathbf{X}^k$ , 令  $\mathbf{e}^k = \mathbf{Y} - f(\mathbf{X}^k)$ 。假定  $f$  在  $\mathbf{X}^k$  附近的值可由:

$$f(\mathbf{X}^k + \Delta^k) = f(\mathbf{X}^k) + \mathbf{J}_{\mathbf{X}^k} \Delta^k$$

一阶近似, 其中  $\mathbf{J}_{\mathbf{X}^k} = \left. \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} \right|_{\mathbf{X}=\mathbf{X}^k}$  是  $f$  的 Jacobian 矩阵在  $\mathbf{X}^k$  处的值。

寻找下一个迭代点  $\mathbf{X}^{k+1} = \mathbf{X}^k + \Delta^k$ , 使得:

$$\begin{aligned} \|\mathbf{Y} - f(\mathbf{X}^{k+1})\| &= \min_{\Delta^k} \|\mathbf{Y} - f(\mathbf{X}^k + \Delta^k)\| = \min_{\Delta^k} \|\mathbf{Y} - (f(\mathbf{X}^k) + \mathbf{J}_{\mathbf{X}^k} \Delta^k)\| \\ &= \min_{\Delta^k} \|\mathbf{Y} - f(\mathbf{X}^k) - \mathbf{J}_{\mathbf{X}^k} \Delta^k\| = \min_{\Delta^k} \|\mathbf{e}^k - \mathbf{J}_{\mathbf{X}^k} \Delta^k\| \end{aligned}$$

利用最小二乘法原理, 其最小二乘正规方程为:

$$\mathbf{J}_{\mathbf{X}^k}^T \mathbf{J}_{\mathbf{X}^k} \Delta^k = \mathbf{J}_{\mathbf{X}^k}^T \mathbf{e}^k$$

则迭代点序列为:

$$\mathbf{X}^{k+1} = \mathbf{X}^k + (\mathbf{J}_{\mathbf{X}^k}^T \mathbf{J}_{\mathbf{X}^k})^{-1} \mathbf{J}_{\mathbf{X}^k}^T \mathbf{e}^k$$

# 集束调整 (Bundle Adjustment)

- **Jacobian** 矩阵计算举例:

假设有函数  $\mathbf{Y} = f(\mathbf{X})$ :

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \begin{aligned} y_1 &= x_1 \\ y_2 &= 5x_3 \\ y_3 &= 4x_2^2 - 2x_3 \\ y_4 &= x_3 \sin x_1 \end{aligned}$$

Jacobian 矩阵为:

$$\mathbf{J} = \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} \\ \frac{\partial y_3}{\partial x_1} & \frac{\partial y_3}{\partial x_2} & \frac{\partial y_3}{\partial x_3} \\ \frac{\partial y_4}{\partial x_1} & \frac{\partial y_4}{\partial x_2} & \frac{\partial y_4}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 5 \\ 0 & 8x_2 & -2 \\ x_3 \cos x_1 & 0 & \sin x_1 \end{bmatrix}$$



# 集束调整 (Bundle Adjustment)

- **LM (Levenberg-Marquardt) 算法**

为了防止矩阵  $\mathbf{J}_{\mathbf{x}^k}^T \mathbf{J}_{\mathbf{x}^k}$  不可逆，LM算法在一阶 Newton法基础上修改了正规方程：

$$\mathbf{J}_{\mathbf{x}^k}^T \mathbf{J}_{\mathbf{x}^k} \Delta^k = \mathbf{J}_{\mathbf{x}^k}^T \mathbf{e}^k \quad \Rightarrow \quad (\mathbf{J}_{\mathbf{x}^k}^T \mathbf{J}_{\mathbf{x}^k} + \lambda \mathbf{I}) \Delta^k = \mathbf{J}_{\mathbf{x}^k}^T \mathbf{e}^k$$

每一次迭代，都是寻找一个合适的  $\lambda$ 。若解得的  $\Delta$  使得误差  $\mathbf{e}$  减小，则接受的当前  $\lambda$  的值，并在下一次迭代中，取 “ $\lambda/10$ ”。

若求解的  $\Delta$  使得误差  $\mathbf{e}$  增加，则采用 “ $10 \lambda$ ” 重新求方程，一直到求出一个  $\Delta$  使误差  $\mathbf{e}$  下降为止。



# 束调整 (Bundle Adjustment)

- LM (Levenberg-Marquardt) 算法的计算步骤

(1) 设  $k=0$ 、 $\lambda=10^{-3}$ ，选取初始点  $\mathbf{X}^0$ ，终止控制常数  $\varepsilon$ ，计算：

$$\mathbf{e}^0 = \|\mathbf{Y} - f(\mathbf{X}^0)\|$$

(2) 计算 Jacobian 矩阵  $\mathbf{J}_{\mathbf{X}^k} = \left. \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} \right|_{\mathbf{X}=\mathbf{X}^k}$ ，构造增量正规方程：

$$(\mathbf{J}_{\mathbf{X}^k}^T \mathbf{J}_{\mathbf{X}^k} + \lambda \mathbf{I}) \Delta^k = \mathbf{J}_{\mathbf{X}^k}^T \mathbf{e}^k$$

(3) 求解正规方程得到  $\Delta^k$ ：

① 若  $\|\mathbf{Y} - f(\mathbf{X}^k + \Delta^k)\| < \|\mathbf{e}^k\|$ ，则误差在减小，令  $\mathbf{X}^{k+1} = \mathbf{X}^k + \Delta^k$ 。

考察：若  $\|\Delta^k\| < \varepsilon$ ，达到目标，输出结果。否则令  $k = k+1$ ， $\lambda^{k+1} = \lambda^k / 10$

转至步骤(2)迭代。

② 若  $\|\mathbf{Y} - f(\mathbf{X}^k + \Delta^k)\| \geq \|\mathbf{e}^k\|$ ，误差增加了。则令  $\lambda = 10\lambda$ ，重新求解  $\Delta^k$   
返回 ①。

# 集束调整 (Bundle Adjustment)

- 基于LM算法的集束调整

- ✓ 测量值  $\mathbf{Y} = (\tilde{u}, \tilde{v})^T$  : 图像上检测到的特征点

- ✓ 函数  $(u, v)^T = f(\mathbf{R}, \mathbf{T}, \mathbf{X})$ : 空间估计三维点  $\mathbf{X}$  根据估计的  $\mathbf{R}$ 、 $\mathbf{T}$  重投影到图像平面的投影点。

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \cong \mathbf{K}[\mathbf{R}|\mathbf{T}]\mathbf{X} = \mathbf{P}\mathbf{X} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{bmatrix} \mathbf{X}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{P}_1 \mathbf{X}}{\mathbf{P}_3 \mathbf{X}} \\ \frac{\mathbf{P}_2 \mathbf{X}}{\mathbf{P}_3 \mathbf{X}} \end{bmatrix} \Rightarrow f(\mathbf{R}, \mathbf{T}, \mathbf{X})$$



# 集束调整 (Bundle Adjustment)

- 基于LM算法的集束调整

- ✓ 重投影误差:  $\|\mathbf{e}\| = \|\mathbf{Y} - f(\mathbf{X})\| = \left\| \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} - \begin{bmatrix} u \\ v \end{bmatrix} \right\|$

- ✓ 函数  $f$  中的变量包括: 空间估计三维点  $\mathbf{X}$ 、外部参数  $\mathbf{R}$ 、 $\mathbf{T}$

- ✓ 求解最小化问题为:

$$\min_{\mathbf{R}, \mathbf{T}, \mathbf{X}} \left\| \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} - f(\mathbf{R}, \mathbf{T}, \mathbf{X}) \right\|$$

- ✓ Jocabian矩阵的数学表达，即推导  $f(\mathbf{R}, \mathbf{T}, \mathbf{X})$  的 Jocabian 矩阵



# 集束调整 (Bundle Adjustment)

- 集束调整中 **Jacobian** 矩阵 的数学表达

✓ 一个视角  $(\mathbf{R}, \mathbf{T})$  下, 对于空间一个物点  $\mathbf{X}$ ,  $f(\mathbf{R}, \mathbf{T}, \mathbf{X})$  中未知数个数:  $\mathbf{R}$ ---3个 (三个角度值:  $\theta$ 、 $\varphi$ 、 $\psi$ )

$\mathbf{T}$  ---3个 (平移向量值:  $t_1$ 、 $t_2$ 、 $t_3$ )

$\mathbf{X}$ ---3个 (坐标值:  $X$ 、 $Y$ 、 $Z$ )

Jacobian 矩阵为  $2 \times 9$  的矩阵:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial u}{\partial \mathbf{R}_{(1 \times 3)}} & \frac{\partial u}{\partial \mathbf{T}_{(1 \times 3)}} & \frac{\partial u}{\partial \mathbf{X}_{(1 \times 3)}} \\ \frac{\partial v}{\partial \mathbf{R}_{(1 \times 3)}} & \frac{\partial v}{\partial \mathbf{T}_{(1 \times 3)}} & \frac{\partial v}{\partial \mathbf{X}_{(1 \times 3)}} \end{bmatrix}$$

# 集束调整 (Bundle Adjustment)

- 集束调整中 **Jacobian** 矩阵 的数学表达

✓ 两个视角，对于空间一个物点  $\mathbf{X}$ ， $f(\mathbf{R}, \mathbf{T}, \mathbf{X})$  中未知数个数15个  
Jacobian 矩阵为  $4 \times 15$  的矩阵：

其中，视点1：  $\mathbf{R}_1$ 、  $\mathbf{T}_1$ 、 图像重投影点  $(u_1, v_1)^T$

视点2：  $\mathbf{R}_2$ 、  $\mathbf{T}_2$ 、 图像重投影点  $(u_2, v_2)^T$

$$\mathbf{J} = \begin{bmatrix}
 \begin{matrix} \frac{\partial u_1}{\partial \mathbf{R}_1(1 \times 3)} & \frac{\partial u_1}{\partial \mathbf{T}_1(1 \times 3)} \\ \frac{\partial v_1}{\partial \mathbf{R}_1(1 \times 3)} & \frac{\partial v_1}{\partial \mathbf{T}_1(1 \times 3)} \end{matrix} & \begin{matrix} \mathbf{0}_{(1 \times 3)} & \mathbf{0}_{(1 \times 3)} \\ \mathbf{0}_{(1 \times 3)} & \mathbf{0}_{(1 \times 3)} \end{matrix} & \begin{matrix} \frac{\partial u_1}{\partial \mathbf{X}_{(1 \times 3)}} \\ \frac{\partial v_1}{\partial \mathbf{X}_{(1 \times 3)}} \end{matrix} \\
 \begin{matrix} \mathbf{0}_{(1 \times 3)} & \mathbf{0}_{(1 \times 3)} \\ \mathbf{0}_{(1 \times 3)} & \mathbf{0}_{(1 \times 3)} \end{matrix} & \begin{matrix} \frac{\partial u_2}{\partial \mathbf{R}_2(1 \times 3)} & \frac{\partial u_2}{\partial \mathbf{T}_2(1 \times 3)} \\ \frac{\partial v_2}{\partial \mathbf{R}_2(1 \times 3)} & \frac{\partial v_2}{\partial \mathbf{T}_2(1 \times 3)} \end{matrix} & \begin{matrix} \frac{\partial u_2}{\partial \mathbf{X}_{(1 \times 3)}} \\ \frac{\partial v_2}{\partial \mathbf{X}_{(1 \times 3)}} \end{matrix}
 \end{bmatrix}$$

$\mathbf{A}_j^i$   $2 \times 6$  → 
 $\frac{\partial u_1}{\partial \mathbf{R}_1(1 \times 3)}$   $\frac{\partial u_1}{\partial \mathbf{T}_1(1 \times 3)}$   
 $\frac{\partial v_1}{\partial \mathbf{R}_1(1 \times 3)}$   $\frac{\partial v_1}{\partial \mathbf{T}_1(1 \times 3)}$ 

 $\frac{\partial u_1}{\partial \mathbf{X}_{(1 \times 3)}}$   
 $\frac{\partial v_1}{\partial \mathbf{X}_{(1 \times 3)}}$ 
 ←  $\mathbf{B}_j^i$   $2 \times 3$

$j$ : 第  $j$  视点  
 $i$ : 第  $i$  个物点



# 集束调整 (Bundle Adjustment)

- 集束调整中 Jocabian 矩阵 的数学表达

✓ 3个视角，对于空间2个物点  $\mathbf{X}$ ， $f(\mathbf{R}, \mathbf{T}, \mathbf{X})$  中未知数个数：

$$6 \times 3 + 3 \times 2 = 24 \text{ 个}$$

Jocabian 矩阵为  $12 \times 24$  的矩阵：

$$\mathbf{J} = \begin{bmatrix} \mathbf{A}_1^1 & \mathbf{0} & \mathbf{0} & \mathbf{B}_1^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2^1 & \mathbf{0} & \mathbf{B}_1^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3^1 & \mathbf{B}_1^1 & \mathbf{0} \\ \mathbf{A}_1^1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_1^2 \\ \mathbf{0} & \mathbf{A}_2^1 & \mathbf{0} & \mathbf{0} & \mathbf{B}_1^2 \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3^1 & \mathbf{0} & \mathbf{B}_1^2 \end{bmatrix}$$

✓  $m$ 个视角，对于空间  $n$  个三维点，Jocabian 矩阵 的维度为：

$$2mn \times (6m + 3n)$$



# 束调整 (Bundle Adjustment)

- Jocabian 矩阵的特点:

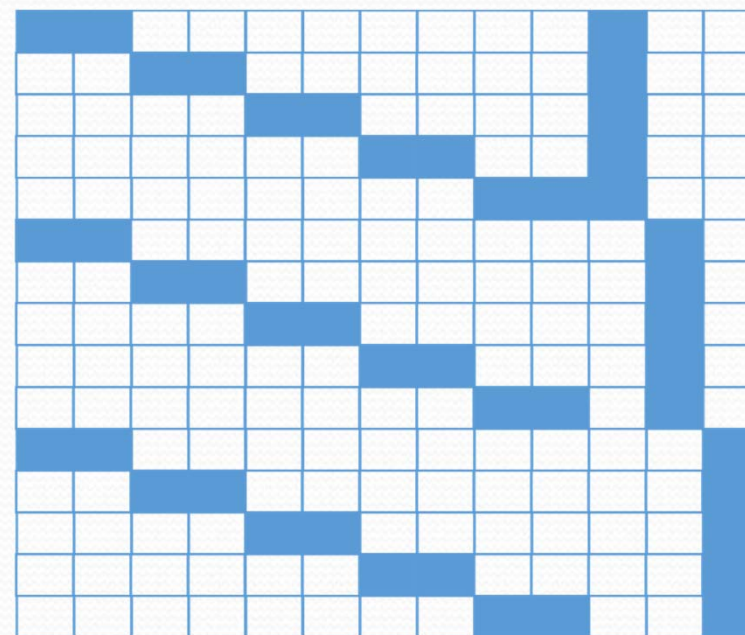
(1) 维度十分庞大, 运算不易:

$$2mn \times (6m+3n)$$

(2) 稀疏矩阵

随着空间三维点的增加,

Jocabian矩阵将变得十分稀疏



Jocabian矩阵: ■ 非“0”子阵

□ “0”子阵

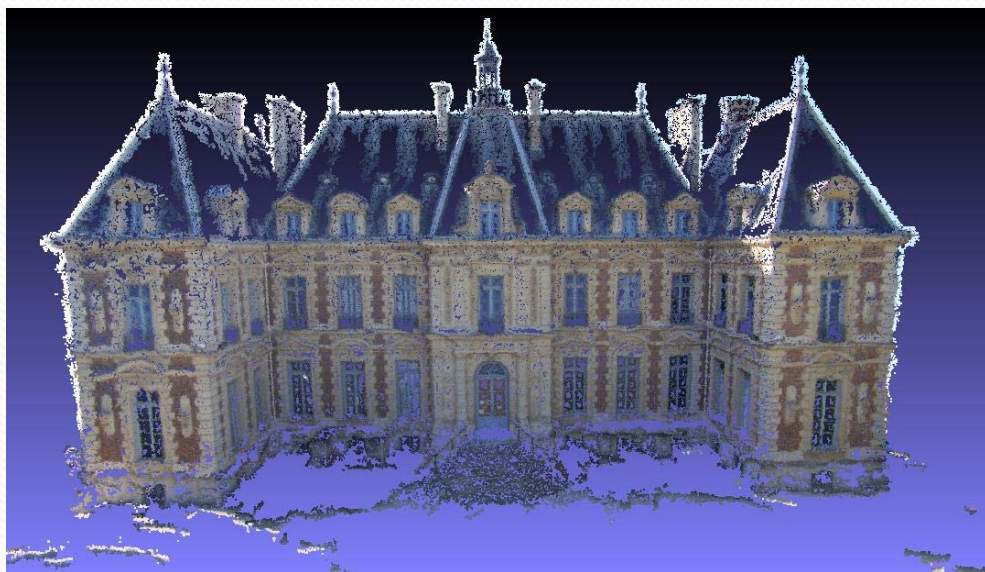
实际求解时, 可利用 Jocabian 矩阵的稀疏性, 及各种数学技巧和约束条件, 进行改进, 尽量降低运算量, 实现LM算法的迭代。

# 集束调整 (Bundle Adjustment)

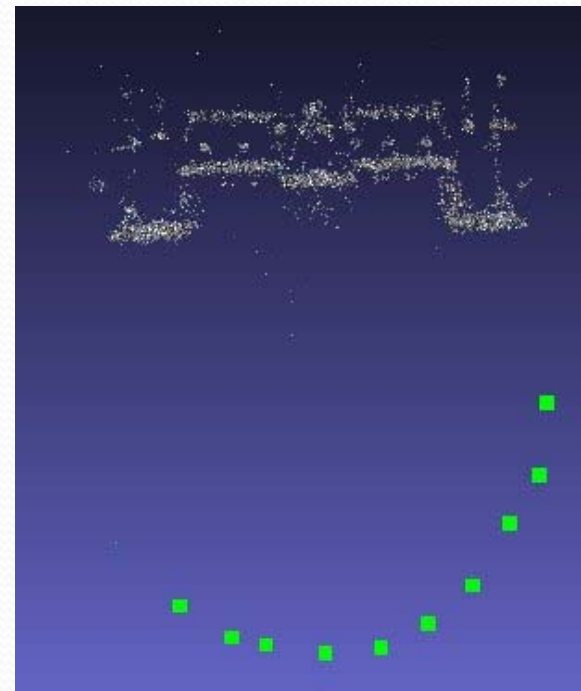
- 实验结果



输入10个视角的图像



集束调整后，重建模型任意一视角的效果



10个视角与稀疏三维点云的集束调整