相机模型与运动恢复结构

章国锋

浙江大学CAD&CG国家重点实验室

视频场景重建的流程

运动恢复结构

深度恢复

三维重建

齐次坐标

在原有的坐标上增加一个维度：

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

新增的维度并不会增加自由度：

$$(x, y, z, w) \quad w \neq 0 \rightarrow (x/w, y/w, z/w)$$

齐次坐标

使用2D坐标完成平移：

$$x' \rightarrow \begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} u + t_u \\ v + t_v \end{bmatrix} = x + t$$

使用齐次坐标完成平移：

$$x' \rightarrow \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = Tx$$

齐次坐标

使用齐次坐标判断点是否在线上

$$l = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$$

$$x^T l = \begin{bmatrix} u & v & 1 \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = 0$$

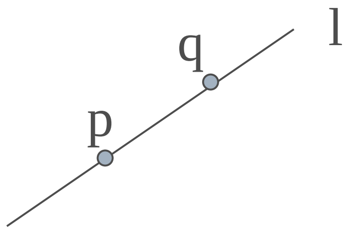
使用齐次坐标判断点是否在平面上

$$\pi = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ d \end{bmatrix}$$

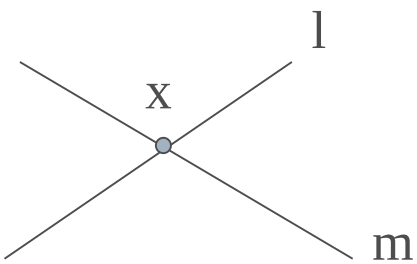
$$x^T \pi = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ d \end{bmatrix} = 0$$

齐次坐标

- 两个点定义一条直线: $\mathbf{l} = \mathbf{p} \times \mathbf{q}$



- 两条直线定义一个点: $\mathbf{x} = \mathbf{l} \times \mathbf{m}$



齐次坐标

使用齐次坐标完成平移和放缩：

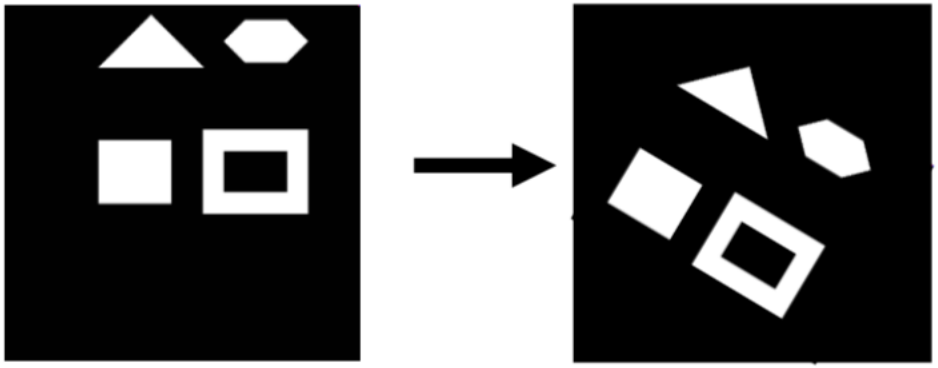
$$x' \rightarrow \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} s_u & 0 & s_u t_u \\ 0 & s_v & sv_t v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_u & 0 & 0 \\ 0 & s_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = STx$$

使用齐次坐标完成旋转和平移：

$$x' \rightarrow \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & t_u \\ sin\theta & cos\theta & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = TRx$$

等距变换

2D的等距变换具有三自由度，这种变换是保距离的

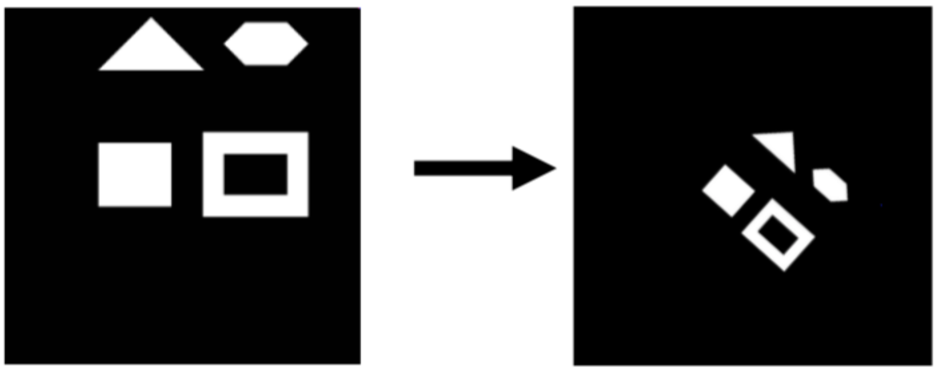


$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

R为旋转矩阵，t为平移向量

相似变换

2D的相似变换具有四自由度，这种变换是保角度的

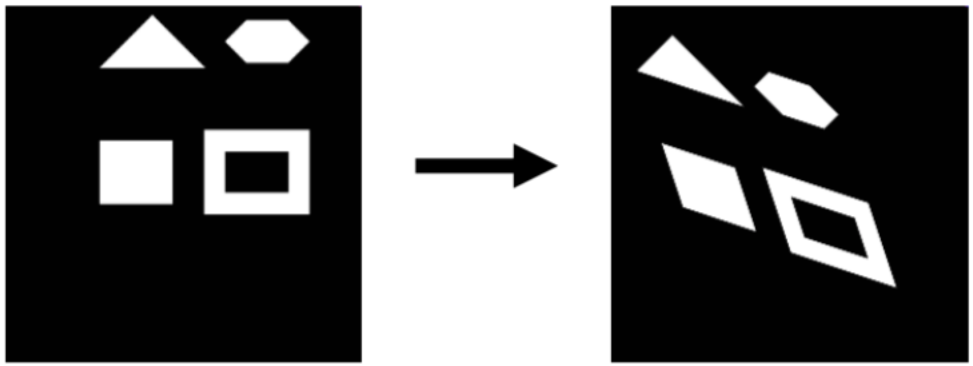


$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

s为相似变换因子

仿射变换

2D的仿射变换具有六自由度，这种变化是保平行的

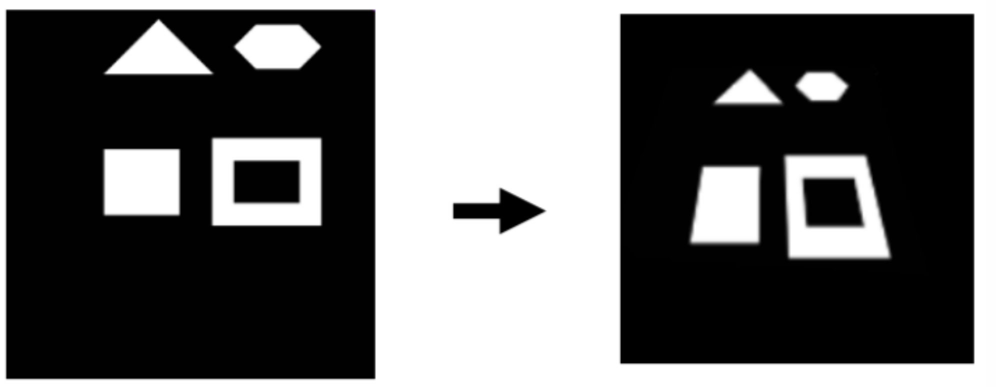


$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta)R(-\phi)SR(\phi) \qquad S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

射影变换

2D的射影变换具有八自由度，这种变化是保同线性的



$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ v & b \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$H = \begin{bmatrix} A & t \\ v & b \end{bmatrix}$$

求解射影变换

$$H = \begin{bmatrix} A & t \\ v & b \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \qquad x' = Hx$$

$$u_2 = \frac{h_1 u + h_2 v + h_3}{h_7 u + h_8 v + h_9}$$

$$v_2 = \frac{h_1 u + h_2 v + h_3}{h_4 u + h_5 v + h_6}$$

展开两个等式中得到：

$$h_1 u_1 + h_2 u_2 + h_3 - h_7 u_1 u_2 - h_8 v_1 u_2 - h_9 u_2 = 0$$

$$h_4 u_1 + h_5 u_2 + h_6 - h_7 u_1 v_2 - h_8 v_1 v_2 - h_9 v_2 = 0$$

求解射影变换

$$h_1u_1 + h_2u_2 + h_3 - h_7u_1u_2 - h_8v_1u_2 - h_9u_2 = 0$$

$$h_4u_1 + h_5u_2 + h_6 - h_7u_1v_2 - h_8v_1v_2 - h_9v_2 = 0 \qquad (1)$$

我们使用下述表示

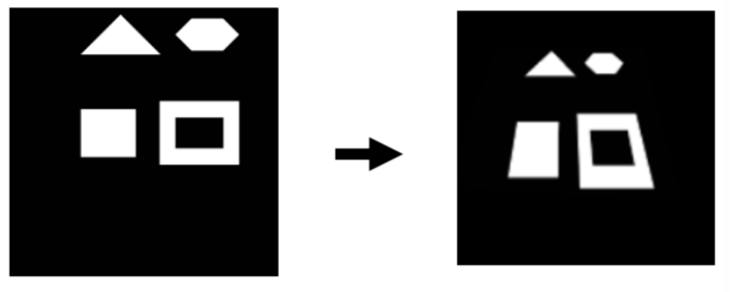
$$A_i = \begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1u_2 & -v_1u_2 & -u_2 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_1v_2 & -v_1v_2 & -v_2 \end{pmatrix}$$

$$\mathbf{h}^{\bullet} = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 \end{pmatrix}$$

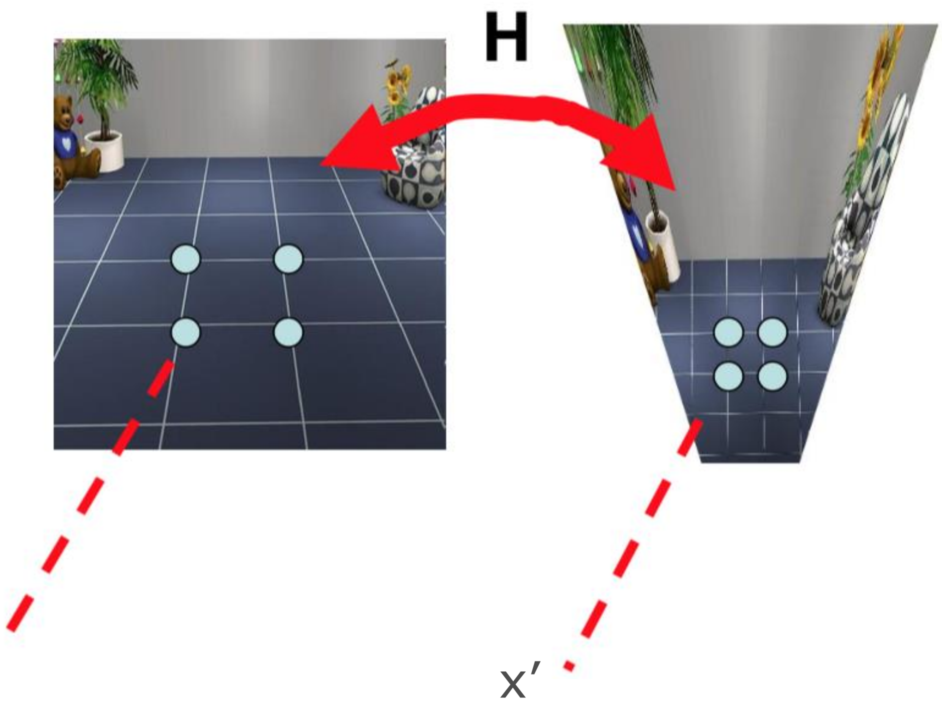
则等式(1)可以改写成如下形式：

$$A_i\mathbf{h} = 0$$

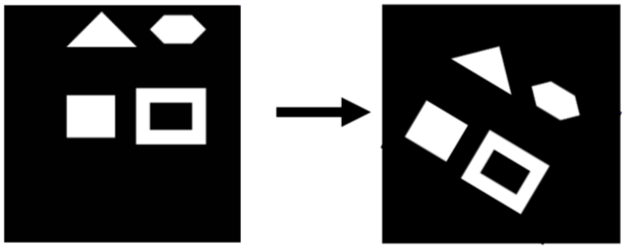
求解射影变换



$A_i \mathbf{h} = 0$



2D齐次坐标的线性变换



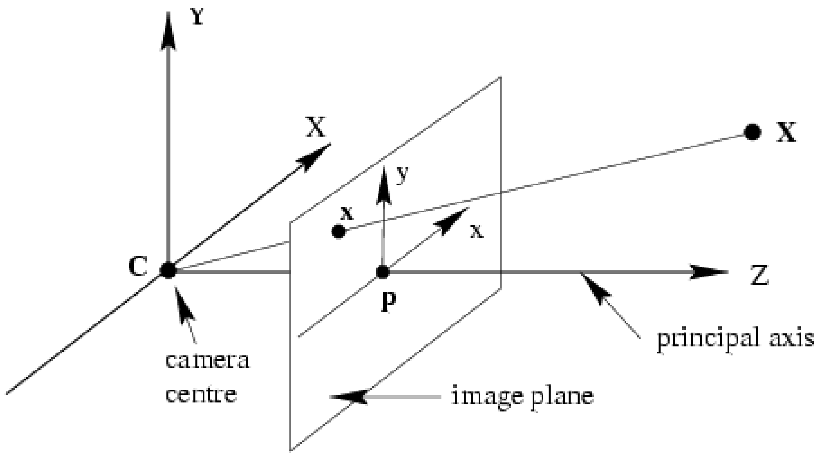
$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

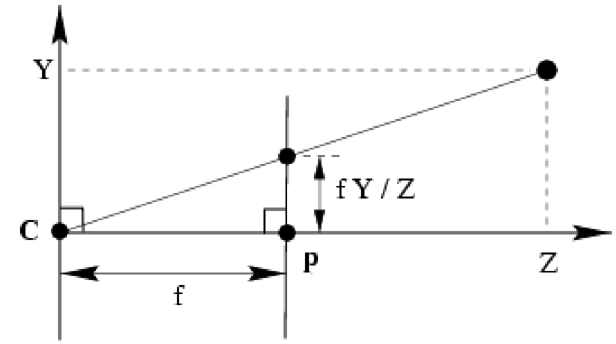
$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ v & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

针孔相机模型





投影方程：

$$x = f\frac{X}{Z}$$

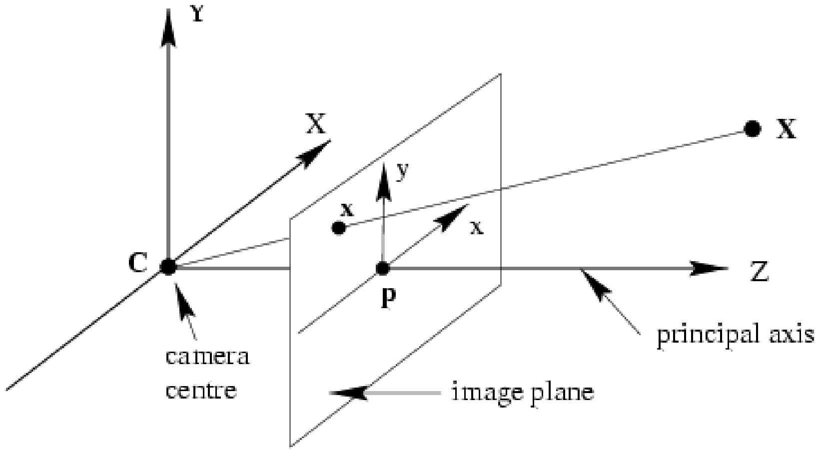
$$y = f\frac{Y}{Z}$$

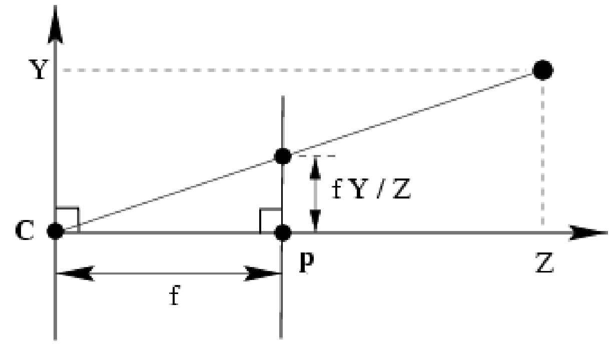
齐次坐标表示：

$$\begin{pmatrix} x \\ y \\ f \end{pmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Richard Hartley and Andrew Zisserman."Multiple View Geometry in Computer Vision". Cambridge University Press, Second Edition 2004.

求解射影变换



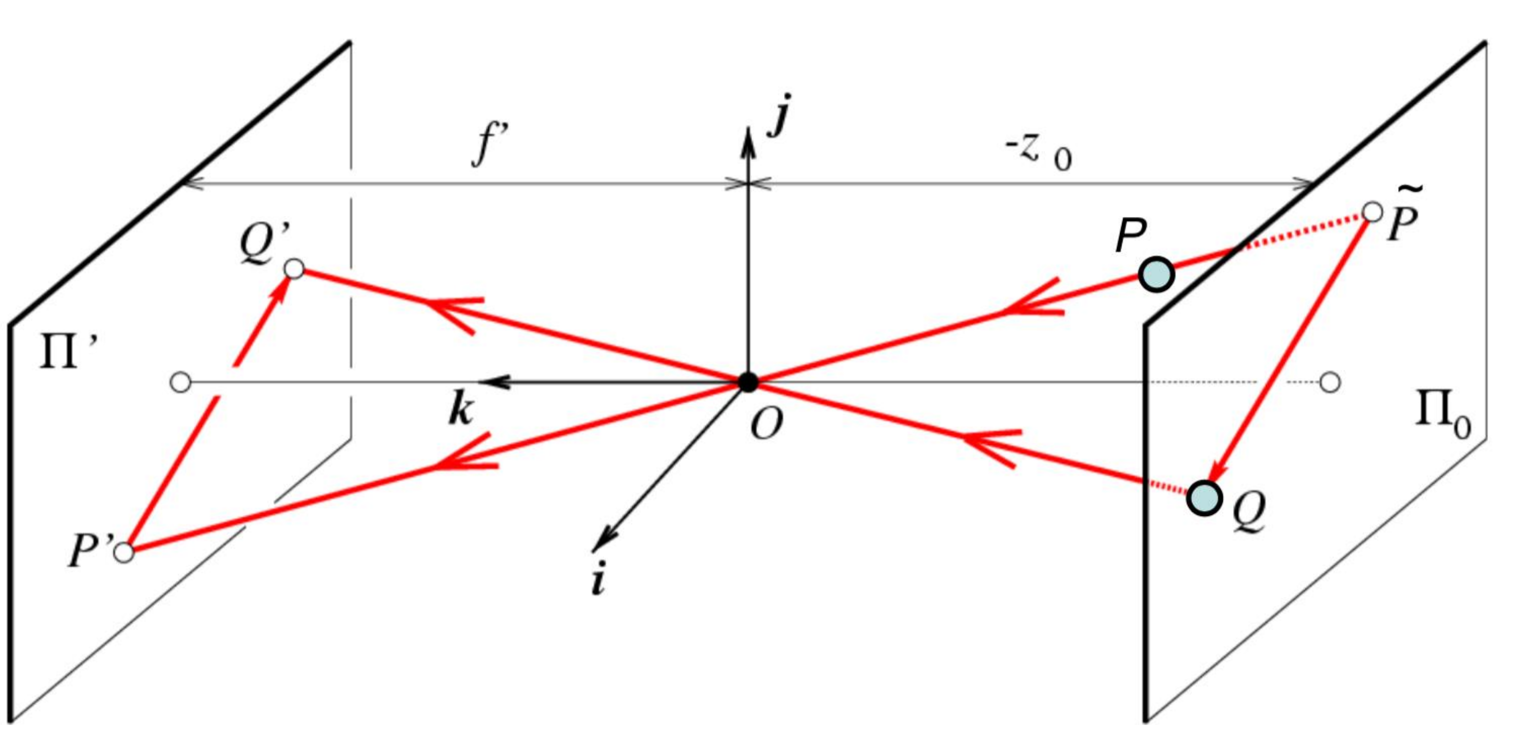


$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

K　　　[R|t]

Richard Hartley and Andrew Zisserman."Multiple View Geometry in Computer Vision". Cambridge University Press, Second Edition 2004.

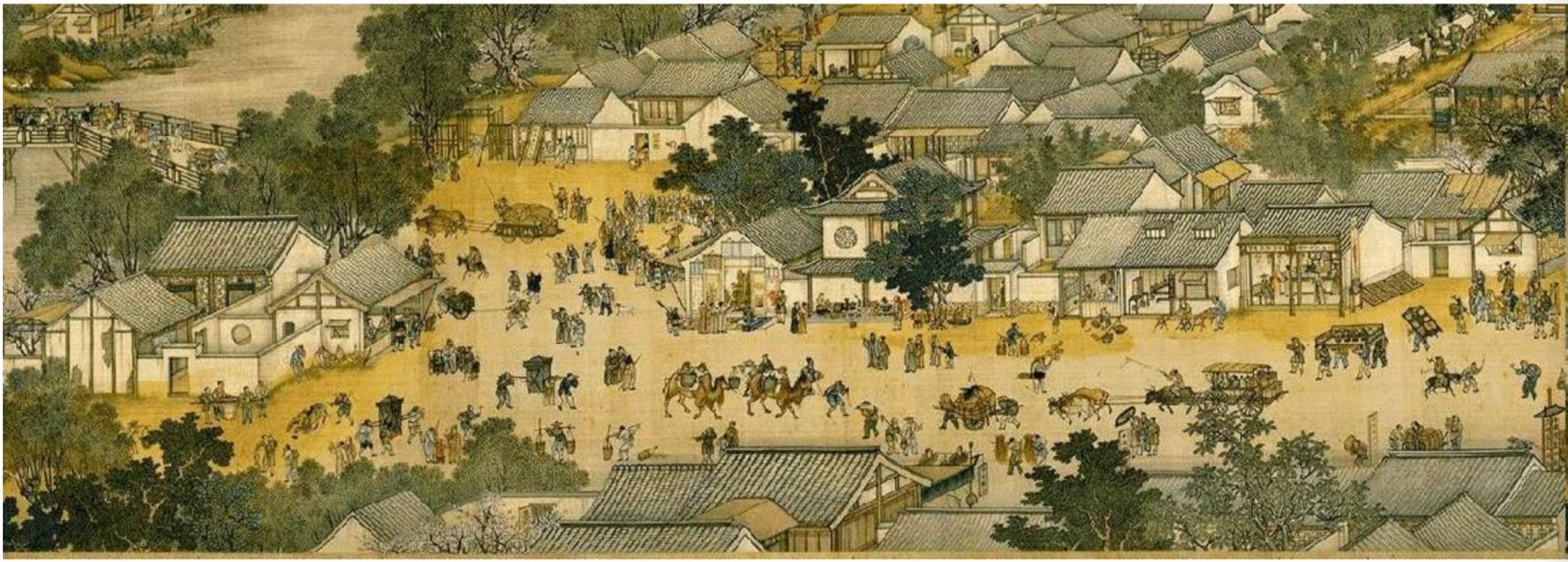
弱透视投影



$$(u,v) = (-f\frac{x}{z}, -f\frac{y}{z}) \rightarrow (u,v) = (-mx, -my)$$

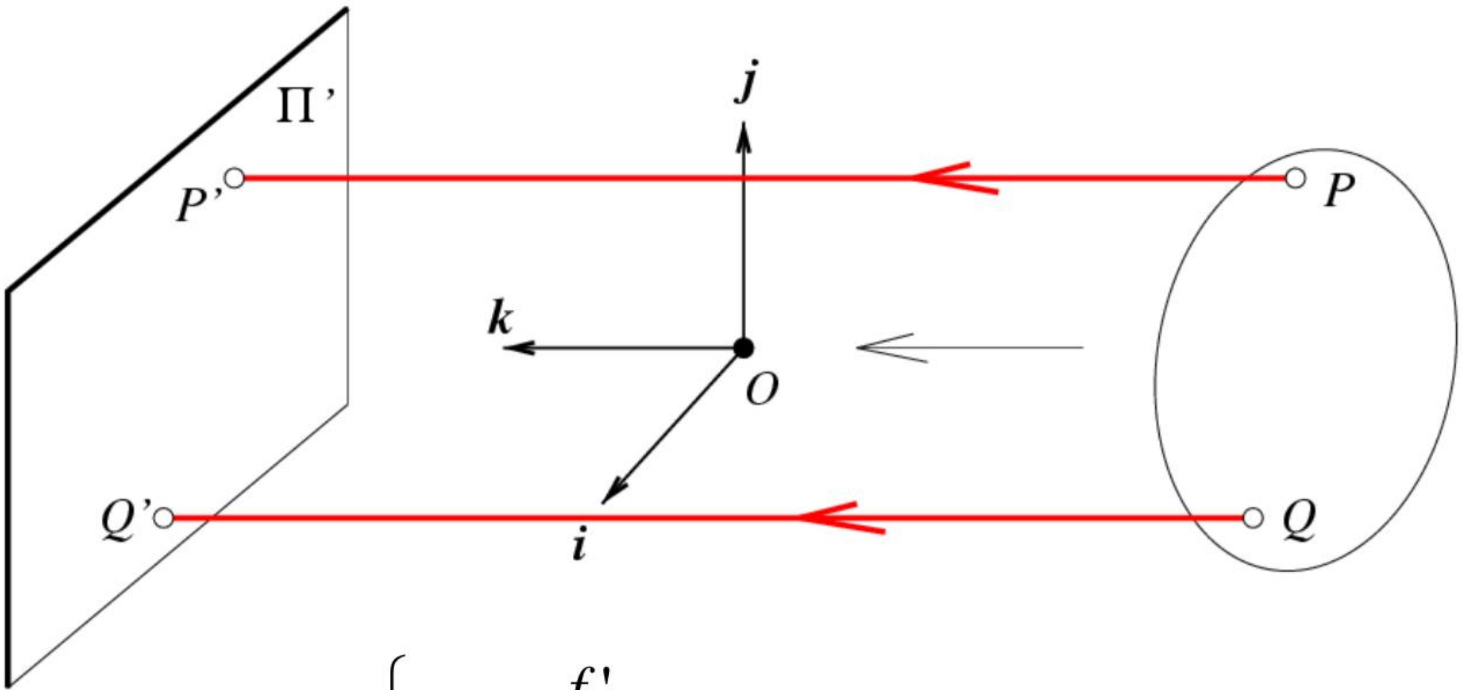
Richard Hartley and Andrew Zisserman."Multiple View Geometry in Computer Vision". Cambridge University Press, Second Edition 2004.

弱透视投影



清明上河图节选

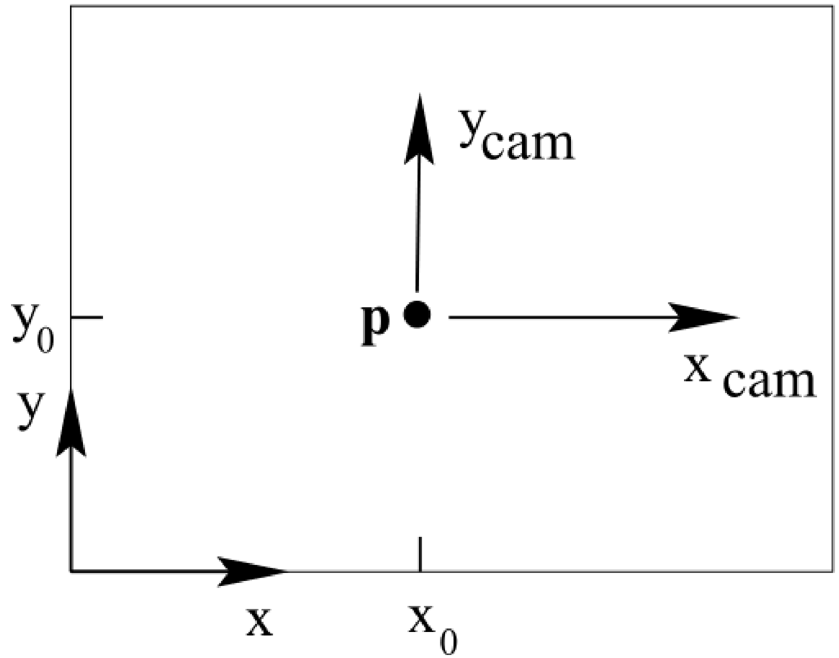
正交投影



$$
\begin{cases} x' = -\dfrac{f'}{z} x \\ y' = -\dfrac{f'}{z} y \end{cases} \rightarrow \begin{cases} x' = -x \\ y' = -y \end{cases}
$$

Richard Hartley and Andrew Zisserman."Multiple View Geometry in Computer Vision". Cambridge University Press, Second Edition 2004.

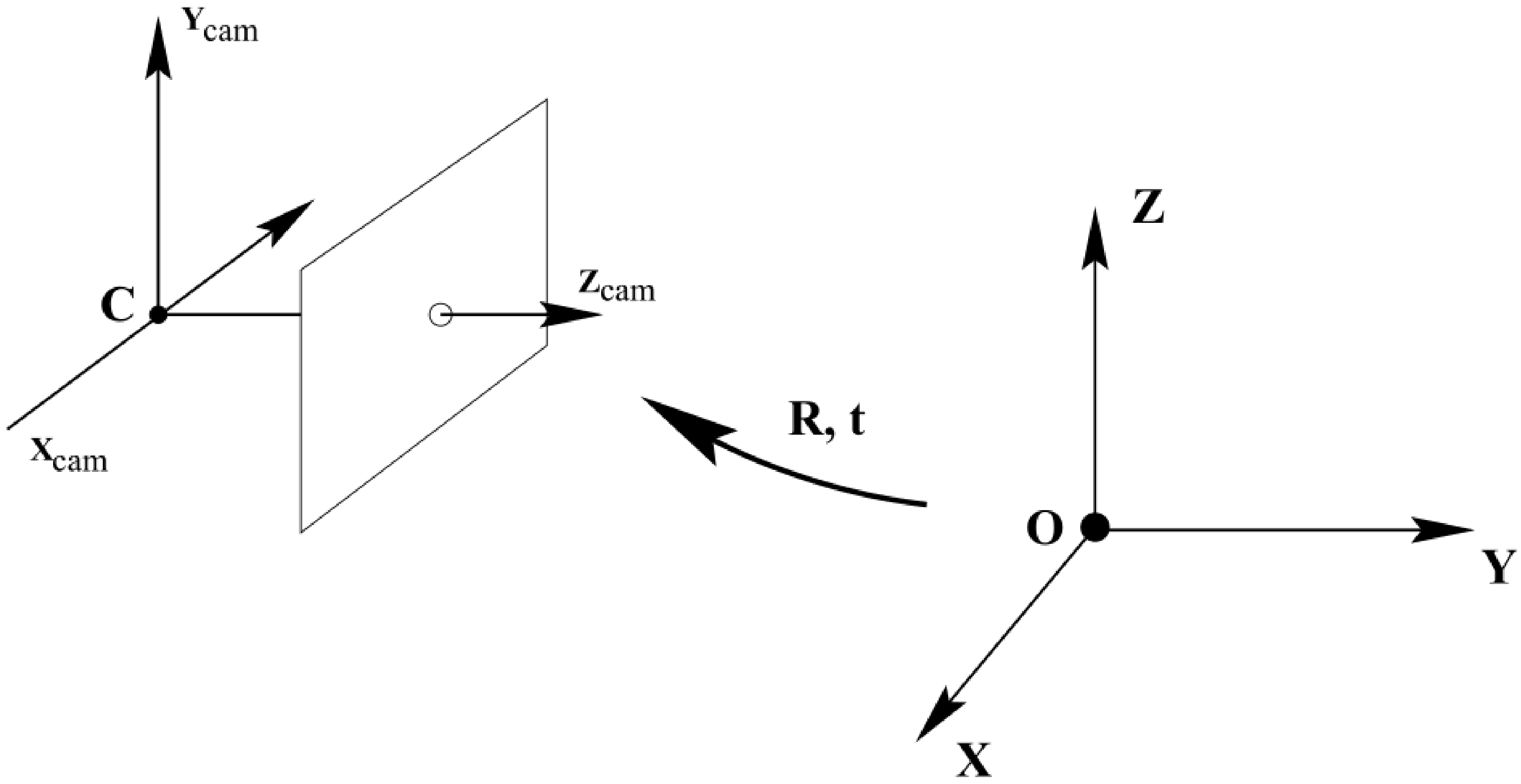
主点的偏移



$$\begin{pmatrix} fX/Z + x_0 \\ fY/Z + y_0 \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX + Zx_0 \\ fY + Zy_0 \\ Z \end{pmatrix} = \begin{bmatrix} f & & x_0 & 0 \\ & f & y_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Richard Hartley and Andrew Zisserman."Multiple View Geometry in Computer Vision". Cambridge University Press, Second Edition 2004.

相机的外部参数



Richard Hartley and Andrew Zisserman."Multiple View Geometry in Computer Vision". Cambridge University Press, Second Edition 2004.

透视相机模型

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{P} = \mathrm{K}[\mathrm{R} \mid \mathrm{t}]$$

11 DoF (5+3+3)

径向畸变

- 比如鱼眼镜头：

- 数学模型:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} R\left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}^\top & -\mathbf{R}^\top \mathbf{t} \\ \mathbf{0}_3^\top & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \right)$$

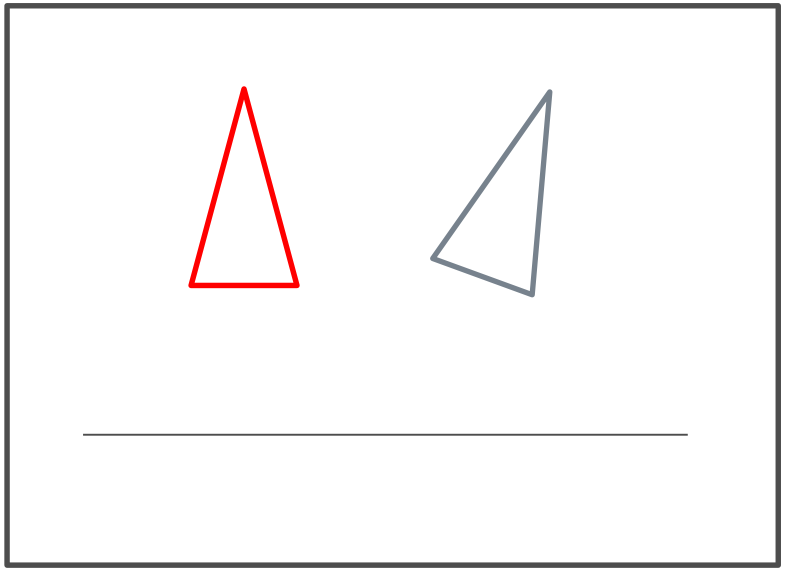
$$R\,(x,y) = (1 + K_1(x^2+y^2) + K_2(x^2+y^2)^2 + \dots)\begin{bmatrix} x \\ y \end{bmatrix}$$

(Marc Pollefeys)

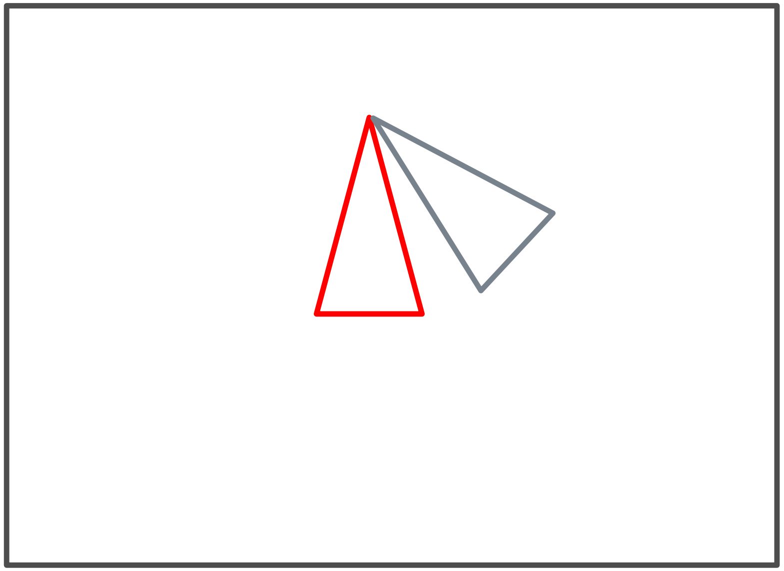
径向畸变矫正例子

(Marc Pollefeys)

单应性矩阵



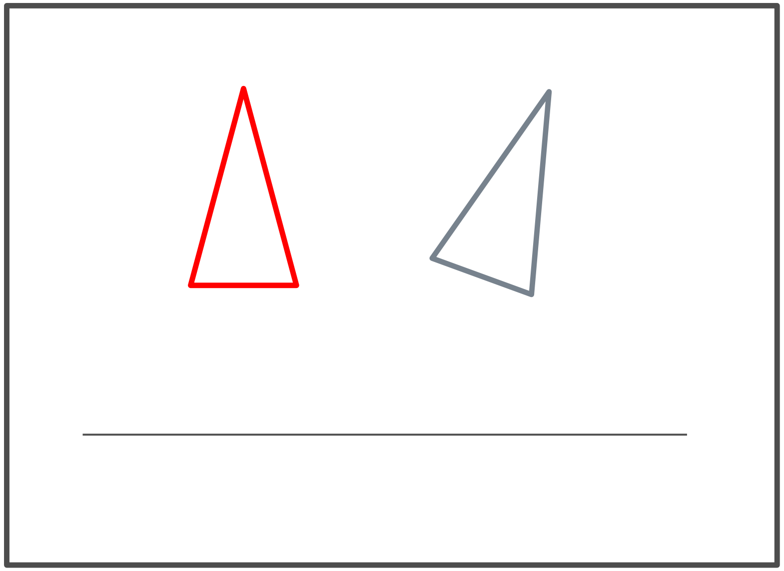
拍摄平面



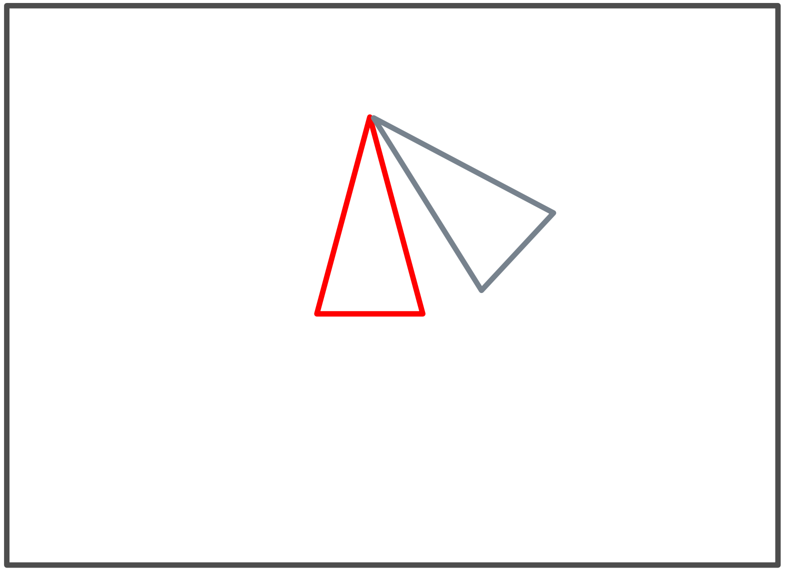
纯旋转

$$c\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$$

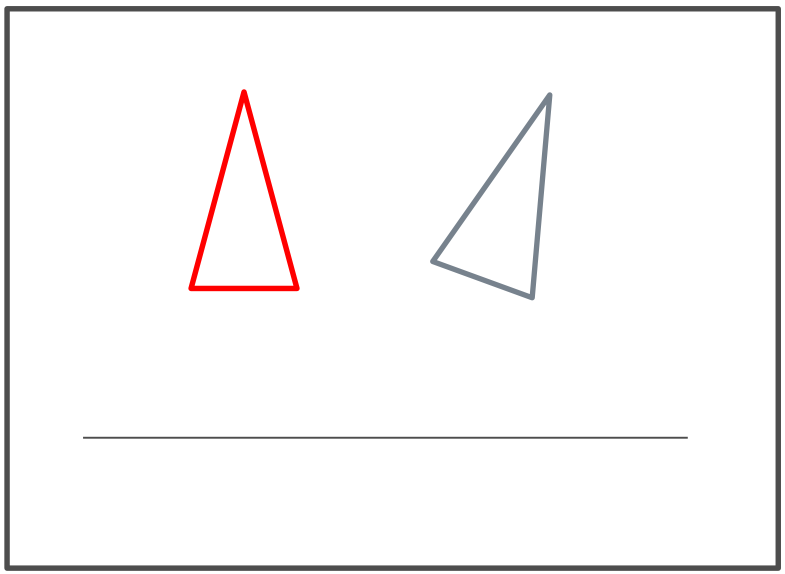
单应性矩阵



单应性矩阵



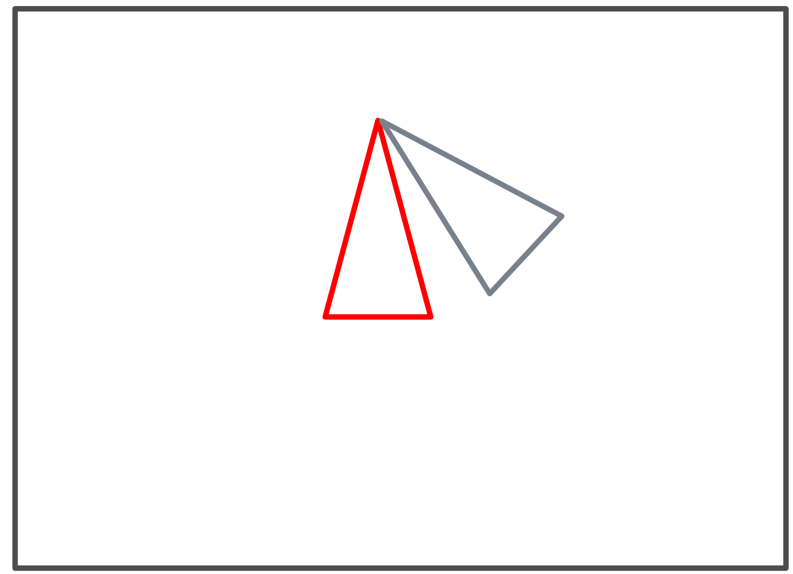
单应性矩阵



拍摄平面

$$\pi = \begin{bmatrix} \mathbf{n} \\ d \end{bmatrix}$$

$$H = K_2(R - \frac{tn}{d})K_1^{-1}$$



纯旋转

$$H = K_2 R K_1^{-1}$$

向量外积的矩阵表示

向量的外积 **v**×**x** 可以被表示为矩阵乘法形式

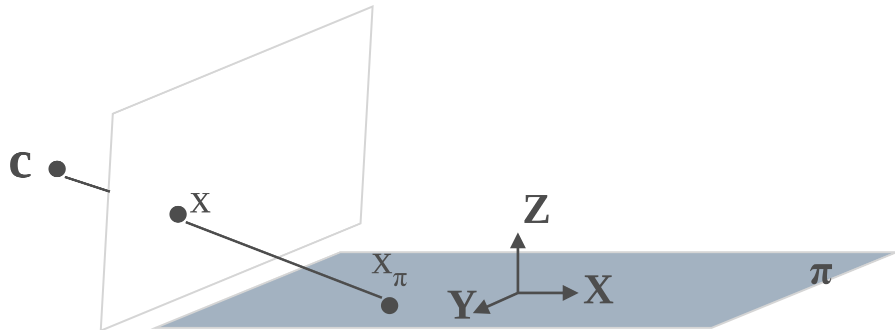
$$\mathbf{v}\times\mathbf{x}=[\mathbf{v}]_\times \mathbf{x}$$

其中

$$[v]_\times=\begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$$

- [**v**]× 是一个 3 × 3 秩为2的斜对称矩阵.
- **v** 是矩阵[**v**]×的零向量： **v** × **v** = [**v**]×**v** = **0**.

平面射影变换



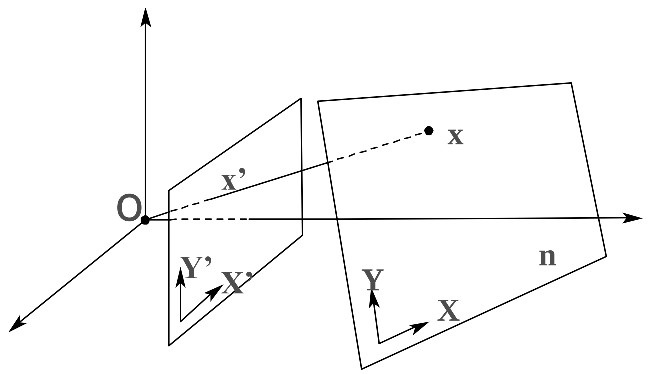
选择该平面作为世界坐标系的Z=0的平面，那么3 ×4的摄影矩阵可以被归约：

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{21} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{pmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{pmatrix}$$

其中该3×3 的矩阵代表了从一个平面到另一个平面的变换

射影变换

$$\begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$



$\mathbf{x}' = \mathrm{H}\mathbf{x}$, 其中 H 是一个 3 × 3 非奇异的齐次矩阵.

- 在使用透视相机成像时，这是一种常用的变换来构建世界和成像平面之间的关系
- 射影变换又被称为具有单应性或共线性。
- H 拥有 8 个自由度

多视图几何

- 运动恢复结构
  - 从多张图片或者视频序列中自动回复相机参数和场景三维结构

Noah Snavely, Steven M. Seitz, Richard Szeliski.
"Photo tourism: Exploring photo collections in 3D". 2016.

双视图几何

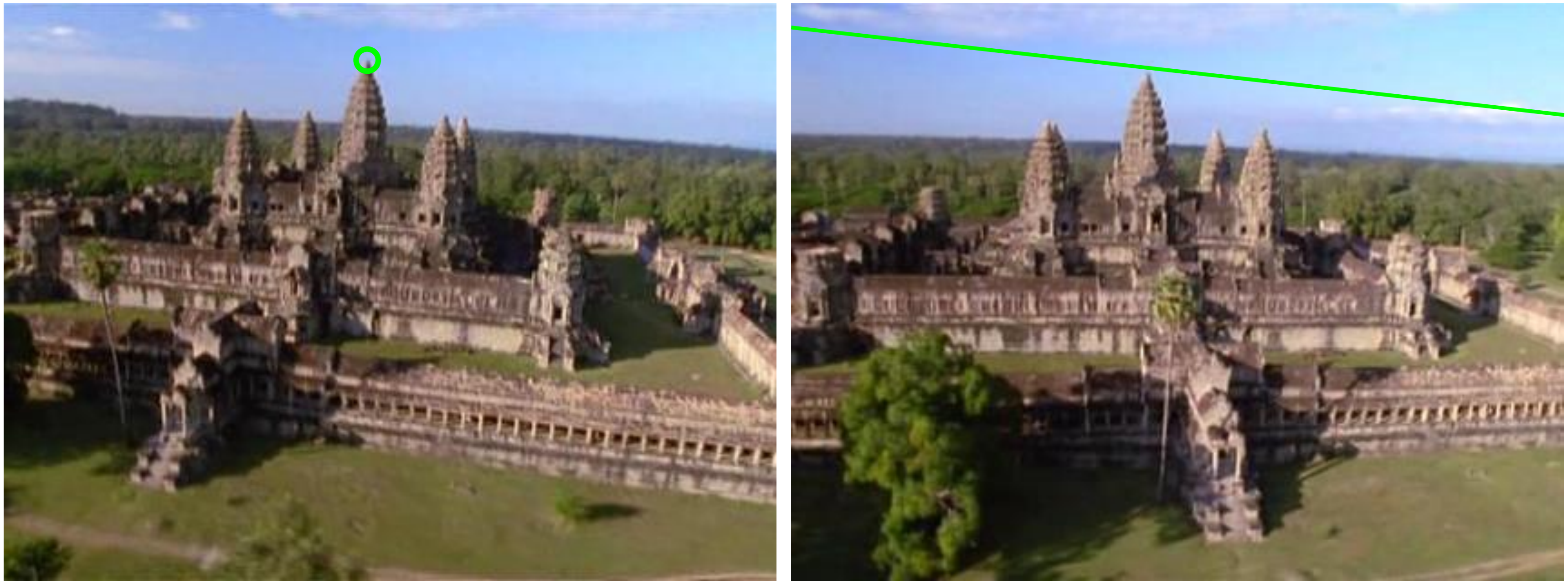
3D???


双视图几何

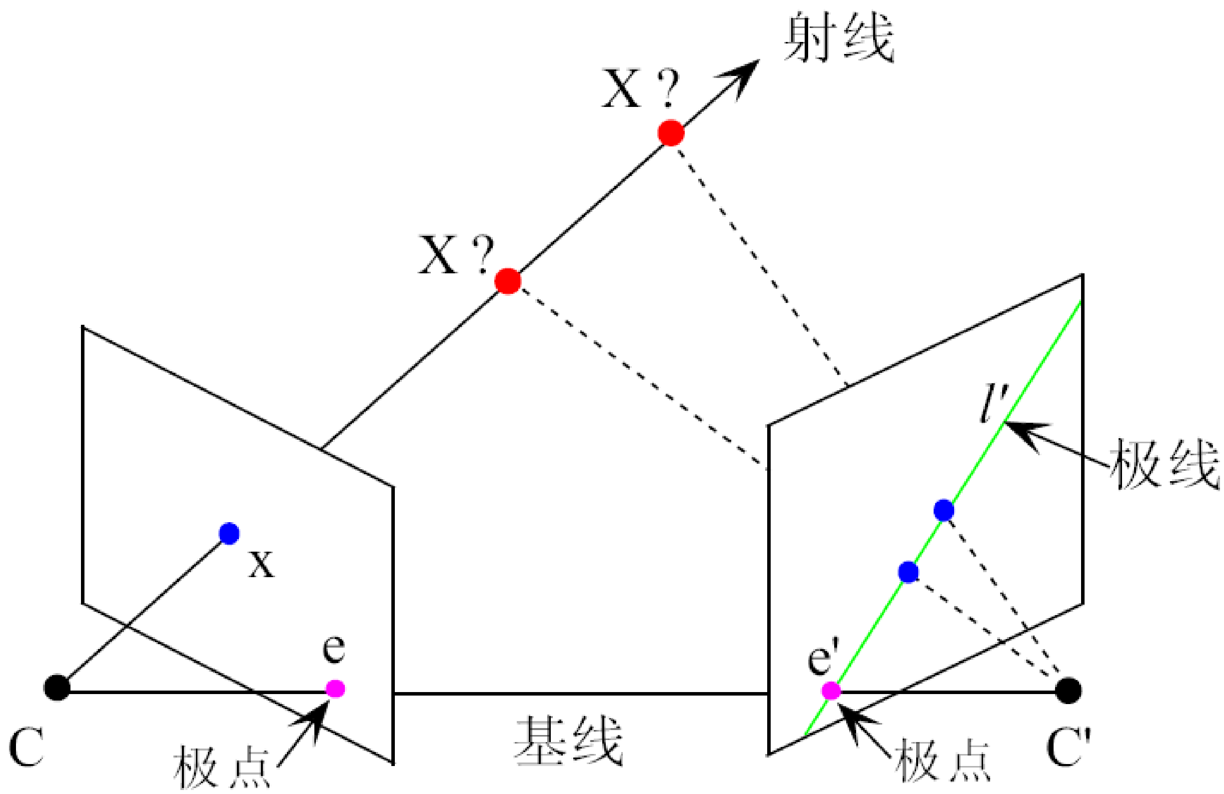
3D???


双视图几何

3D: 极线几何



极线几何



$$\hat{\mathbf{x}}'^{\top} F \hat{\mathbf{x}} = 0$$

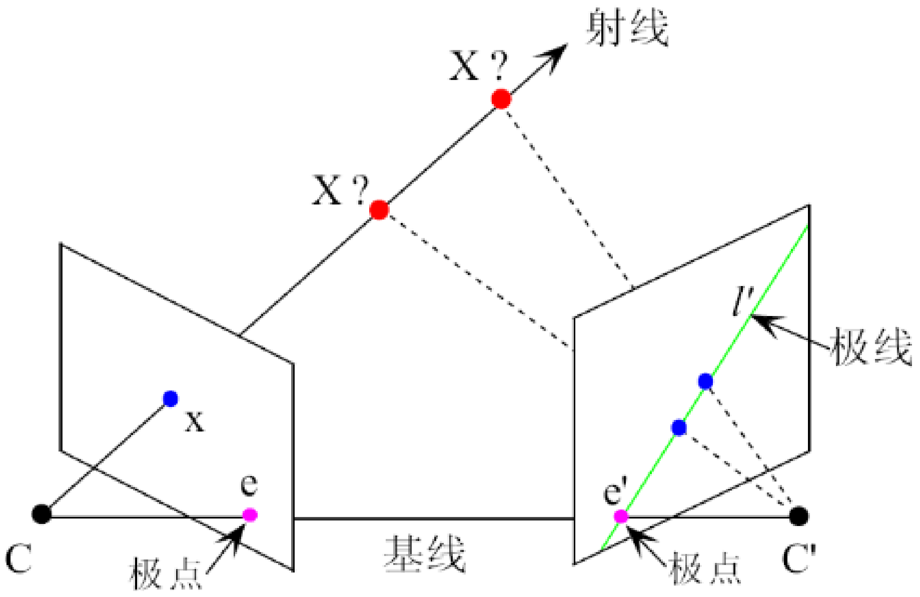
基础矩阵

- 只跟两个视图的相对相机姿态和内参有关

$$F = K_2^{-T}[t]_\times RK_1^{-1}$$

- F 是一个3 × 3 秩为2的矩阵
- Fe = **0**
- 7个自由度
- 最少7对匹配点就可以求解
  - 七点法
  - 八点法

OpenCV: cvFindFundamentalMat()



八点法求解基础矩阵

根据对极几何关系，基本矩阵 F 满足

$$\hat{x}'^{\top} F \hat{x} = 0$$

若设 $\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^{\top}$

那么对极几何关系又可以写作：

$$\begin{pmatrix} \hat{x}'_1\hat{x}_1 & \hat{x}'_1\hat{x}_2 & \hat{x}'_1 & \hat{x}'_2\hat{x}_1 & \hat{x}'_2\hat{x}_2 & \hat{x}'_2 & \hat{x}_1 & \hat{x}_2 & 1 \end{pmatrix} \mathbf{f} = 0$$

若存在 n 对对应点，F 应满足如下的线性系统：

$$A\mathbf{f} = \begin{pmatrix} \hat{x}'_{11}\hat{x}_{11} & \hat{x}'_{11}\hat{x}_{12} & \hat{x}'_{11} & \hat{x}'_{12}\hat{x}_{11} & \hat{x}'_{12}\hat{x}_{12} & \hat{x}'_{12} & \hat{x}_{11} & \hat{x}_{12} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{x}'_{n1}\hat{x}_{n1} & \hat{x}'_{n1}\hat{x}_{n2} & \hat{x}'_{n1} & \hat{x}'_{n2}\hat{x}_{n1} & \hat{x}'_{n2}\hat{x}_{n2} & \hat{x}'_{n2} & \hat{x}_{n1} & \hat{x}_{n2} & 1 \end{pmatrix} \mathbf{f} = 0$$

基础矩阵

- $\mathbf{f}$ 为 9 维向量，若要有解，rank(F) 至多为 8
  - 在 rank(F) = 8 时，$\mathbf{f}$ 的方向是唯一的
  - 通过至少 8 对对应点，可恰好得到使 $\mathbf{f}$ 方向唯一的 F
- $\mathbf{f}$ 为 F 的右零空间的基向量，可用 SVD(F) 求得
- 当对应点超过8对时且可能有外点时，我们一般先使用RANSAC方法来求解并筛选出内点，并求解得到最优的F^*。

基础矩阵

- 在得到初解后，我们一般还要根据所有内点对F做非线性优化，其中g为距离度量函数， x_i, x_i'为匹配点对：

$$\operatorname{argmin}_{F} \sum_{i} g(x_i, x_i')$$

- 一般使用LM算法来优化该目标函数。
- 常见的两种距离度量
  - 辛普森距离
  - 对称极线距离

基础矩阵

- 一阶几何误差(first-order geometric error)，又名辛普森距离(Sampson distance)　：
  - 令 $e = x_i'^T F x_i$，$J = \frac{\delta(x_i'^T F x_i)}{\delta x_i}$则第 i 对对应点的辛普森距离为

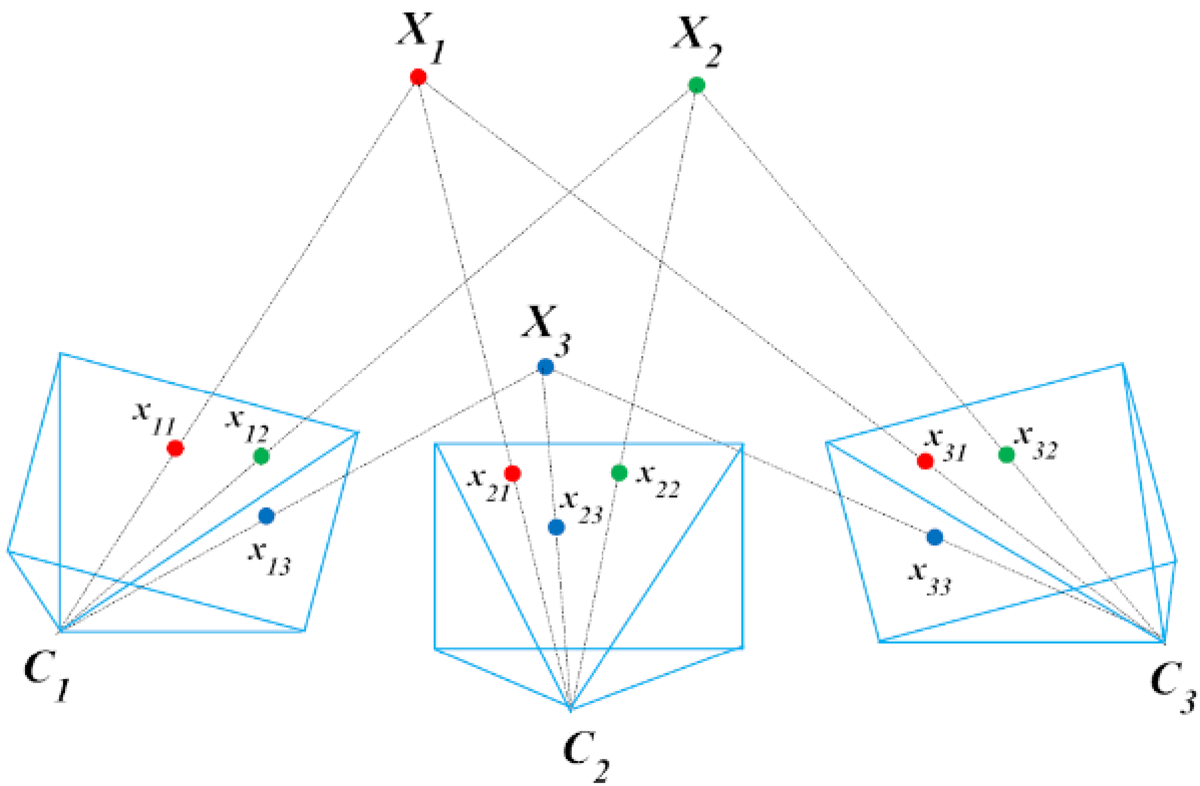
$$\frac{e^T e}{JJ^T} = \frac{(x_i'^T F x_i)^2}{JJ^T} = \frac{(x_i'^T F x_i)^2}{(Fx_i)_1^2+(Fx_i)_2^2+(F^T x_i'^T)_1^2+(F^T x_i'^T)_2^2}$$

- 对称极线距离(symmetric epipolar distance)，它形式上与辛普森距离很像，但是度量的是点到极线的距离：

$$\frac{(x_i'^T F x_i)^2}{(Fx_i)_1^2+(Fx_i)_2^2}+\frac{(x_i'^T F x_i)^2}{(F^T x_i'^T)_1^2+(F^T x_i'^T)_2^2}$$

运动恢复结构

多视图几何



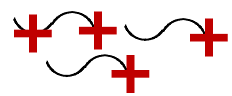
$$\mathbf{x}_{ij} = \pi(\mathbf{P}_i X_j)$$

投影函数 $\pi(x, y, z) = (x/z, y/z)$ $\mathbf{P}_i = \mathbf{K}_i[\mathbf{R}_i|\mathbf{T}_i]$

运动恢复结构

- 流程
  - 特征跟踪
    - 获得一堆特征点轨迹

$$\mathcal{X} = \{\mathbf{x}_i|_{i=1,\dots,m}\}$$



  - 运动恢复结构
    - 求解相机参数和特征点轨迹的三维位置

$$\mathbf{x}_{ij} = \pi(\mathbf{P}_i X_j) \qquad \mathbf{P}_i = \mathbf{K}_i[\mathbf{R}_i|\mathbf{T}_i]$$

$$E(\mathbf{P}_1, \dots, \mathbf{P}_m, X_1, \dots, X_n) = \sum_{i=1}^{m}\sum_{j}^{n} w_{ij}||\pi(\mathbf{P}_i X_j) - \mathbf{x}_{ij}||^2$$

图像特征

- 图像中显著、容易区分和匹配的内容
  - 点
  - 角点
  - 线: 直线, 曲线,…
  - 边: 二维边, 三维边
  - 形状：长方形, 圆, 椭圆, 球,…
  - 纹理
- 不变性
  - 视角不变(尺度, 方向,平移)
  - 光照不变
  - 物体变形
  - 部分遮挡

Harris 角点检测

- 核心思想：统计图像梯度的分布
  - 平坦区域：梯度不明显
  - 边缘区域：梯度明显，方向一致
  - 角点区域：梯度明显，方向不一致
- 方法：
  - 计算像素邻域的梯度二阶矩

$$H = \sum_{(u,v)} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

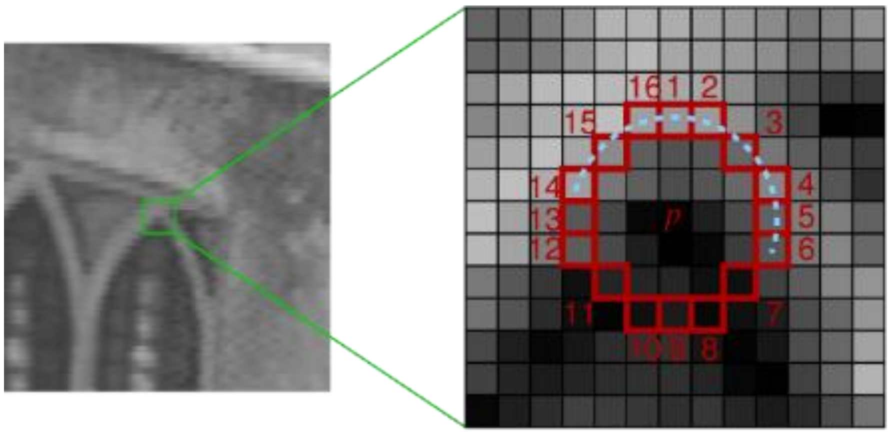
  - 计算上述矩阵的角点响应指标

$$R = \det(H) - \alpha \cdot \mathrm{trace}(H)^2$$

  - 对*R*进行阈值过滤和非极大值抑制

FAST

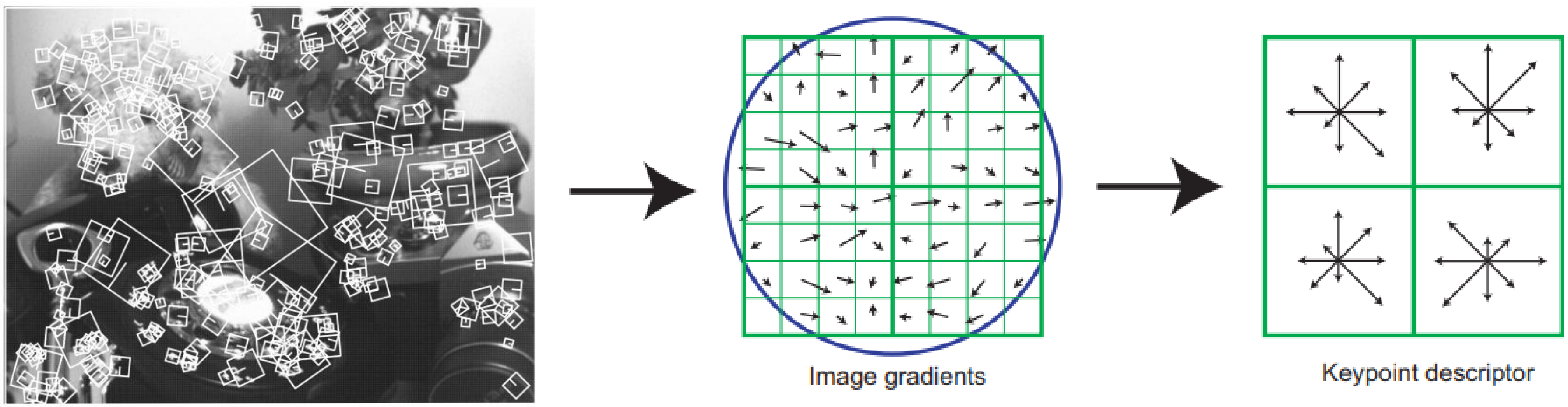
- 核心思想
  - 角点与周围邻域内足够多的像素的灰度差异较大
- 通过少量像素点的比较，加速角点提取
- 考虑中心点周围的16个像素，设中心点亮度为 p
  - 如果有连续 n 个像素亮度都大于 p+t，或者都小于 p-t (如图中的 14~16, 1 ~ 6)
  - 首先检查 1、5、9、13 四个位置，如果是角点，四个位置中应当有三个满足上面的条件
  - ……
- 速度快，但对噪音不鲁棒



Edward Rosten, Tom Drummond. Machine Learning for High-Speed Corner Detection. ECCV (1) 2006: 430-443.

SIFT

- Scale-Invariant Feature Transform
- SIFT通过在不同级别的图像DoG上寻找极值点来确定特征点的位置和对应的尺度，后续的特征提取在与其尺度最邻近的图像DoG上进行。这使它有良好的尺度不变性。



David G. Lowe.Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2): 91-110 (2004).

More Invariant Features

- SIFT之后陆续出现了各种尺度不变特征描述量提取算法
  - 如 RIFT、GLOH、SURF等
  - 保证了一定的视觉不变性，又能很好地对抗噪声
- SURF
  - 使用了Haar小波卷积替代SIFT中的高斯核
  - 用积分图像进行了加速，使得计算速度达到SIFT的3~7倍
- ORB
  - 使用FAST提取特征点
  - 使用轻量级的二进制描述子
  - 由于其极快的提取速度得到了广泛使用。

特征提取

精度高

速度快

SIFT

极佳的尺度不变性，能一定程度上适应视角变化和亮度变化

SURF

能够处理严重的图像模糊，速度要高于SIFT，但精度不如SIFT

ORB

极快的提取速度，在实时应用中常用来替代SIFT

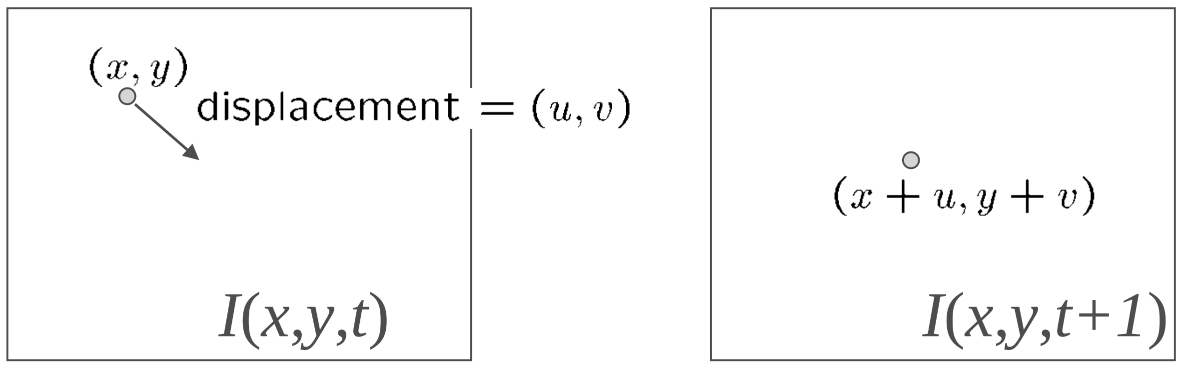
以上三种特征提取算法均在OpenCV中有实现

特征匹配

- 模板匹配
  - 直接在目标图像中寻找给定的图像块

特征匹配

在小运动假设下，可以采用 KLT 跟踪方法：



$$
\begin{aligned}
I(x,y,t) &= I(x+u, y+v, t+1) \\
&\approx I(x,y,t) + \boxed{I_x u + I_y v + I_t}
\end{aligned}
\rightarrow \nabla I \cdot \begin{pmatrix} u \\ v \end{pmatrix} + I_t = 0
$$

一个等式，两个未知量

特征匹配

进一步假设：相邻像素运动一致

$$\nabla I \cdot \begin{pmatrix} u \\ v \end{pmatrix} + I_t = 0$$

(单个像素)

$$\begin{pmatrix} I_x(p_1) & I_y(p_1) \\ \vdots & \vdots \\ I_x(p_n) & I_y(p_n) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} I_t(p_1) \\ \vdots \\ I_t(p_n) \end{pmatrix} = 0$$

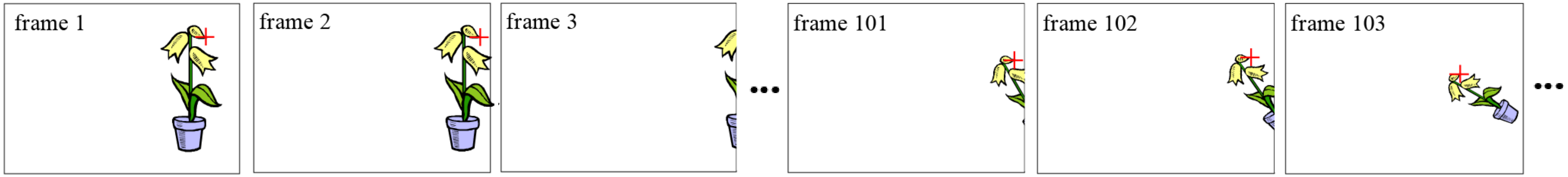
(邻域窗口)

特征匹配

- 大运动情况下的匹配
  - 通过比较特征描述量的距离进行匹配
  - SIFT = 128 维、SURF = 64 维、ORB = 256bits
    - 暴力匹配
    - 快速最近邻匹配
  - OpenCV中提供了相应的匹配算法

Non-Consecutive Feature Tracking

- How to efficiently match the common features among different subsequences?

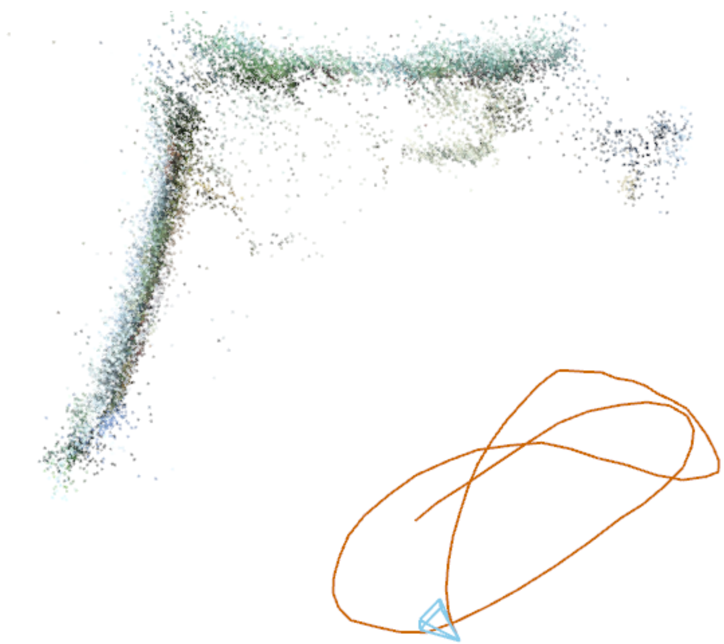


Non-Consecutive Feature Tracking

Consecutive Feature Tracking

Non-Consecutive Track Matching

Structure from Motion



Framework Overview

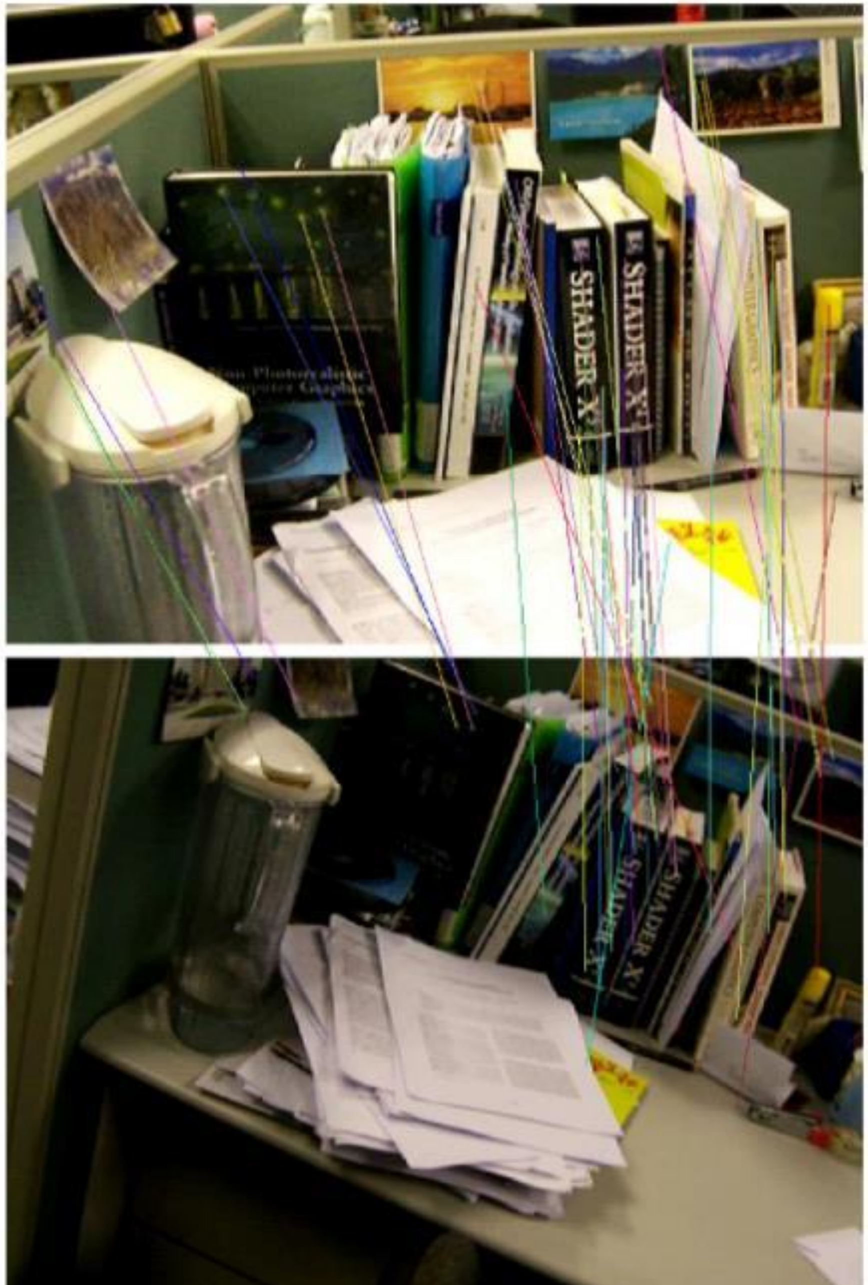
1. Detect SIFT features over the entire sequence.
2. **Consecutive point tracking**:
    2.1 Match features between consecutive frames with descriptor comparison.
    2.2 Perform the second-pass matching to extend track lifetime.
3. **Non-consecutive track matching**:
    3.1 Use hierachical k-means to cluster the constructed tracks.
    3.2 Estimate the matching matrix with the grouped tracks.
    3.3 Detect overlapping subsequences and join the matched tracks.

Two-Pass Matching

- SIFT Feature Extraction
- First-Pass Matching by Descripto
  Comparison

$$c = \frac{||\mathbf{p}(\mathcal{N}_1^{t+1}(\mathbf{x}_t)) - \mathbf{p}(\mathbf{x}_t)||}{||\mathbf{p}(\mathcal{N}_2^{t+1}(\mathbf{x}_t)) - \mathbf{p}(\mathbf{x}_t)||}$$

$c < \varepsilon$ Global distinctive

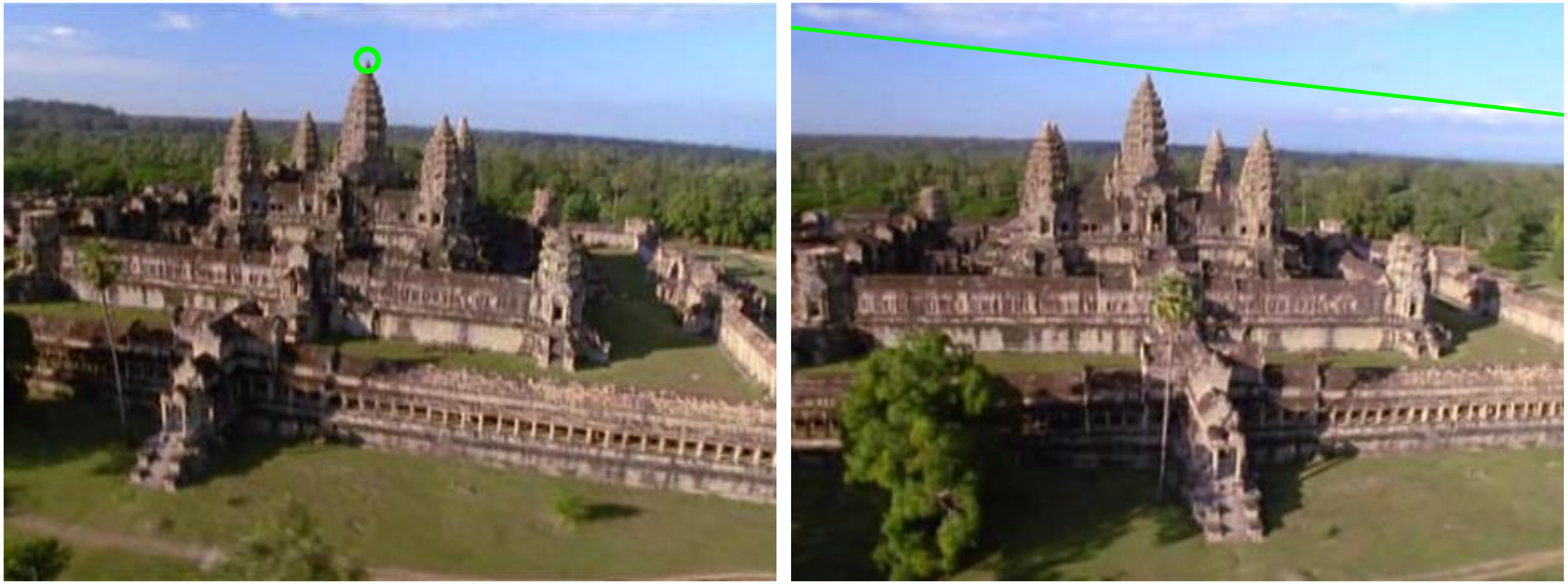

SHADER X
SHADER X

Two-View Geometry

3D???

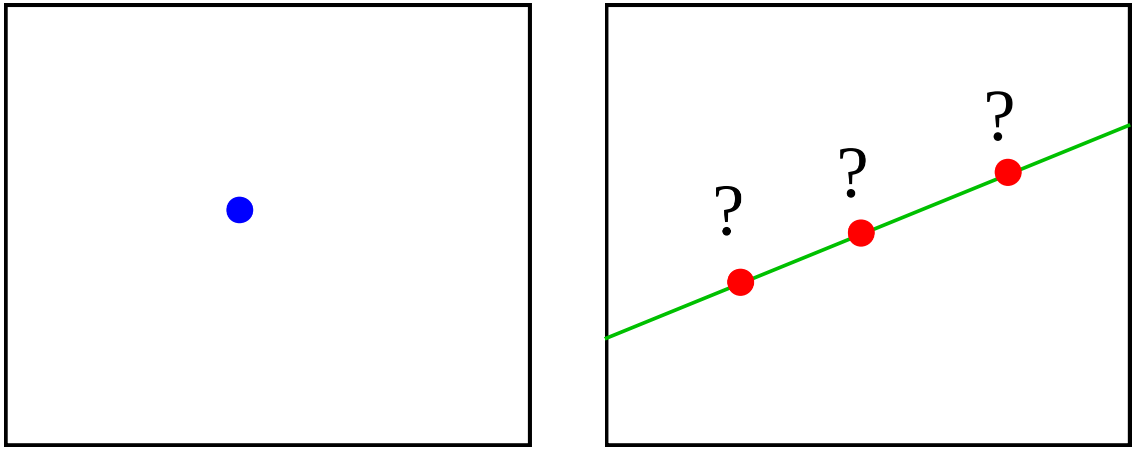
Two-View Geometry

3D: Epipolar Geometry



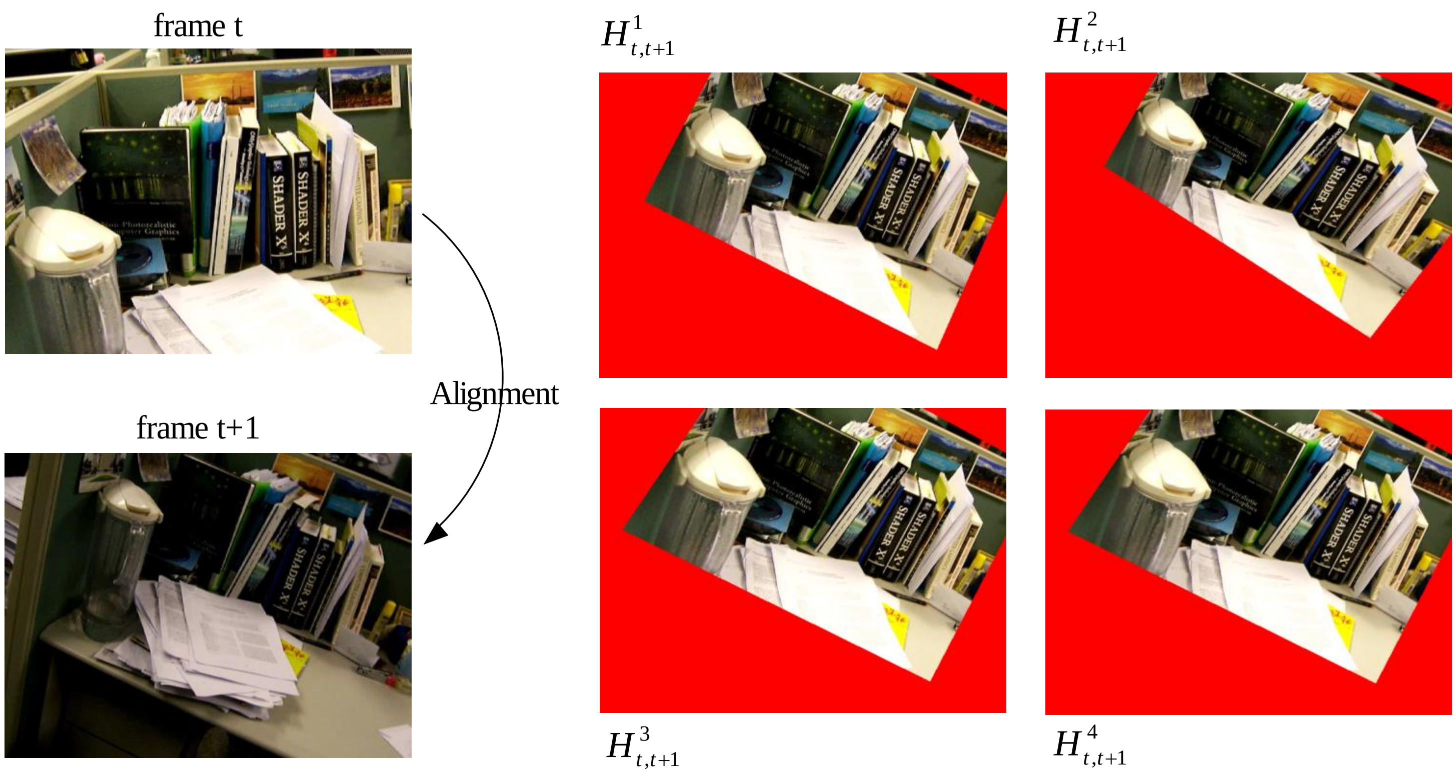
Not Enough!

- How to handle image distortion?
  - Naïve window-based matching becomes unreliable!
- How to give a good position initializaton?
  - Whole line searching is still time-consuming and ambiguous with many potential correspondences.



Second-Pass Matching by Planar Motion Segmentation

- Estimate a set of homographi $\{H^k_{t,t+1} | k = 1, \ldots, N\}$
  - Using inlier matches in first-pass matching



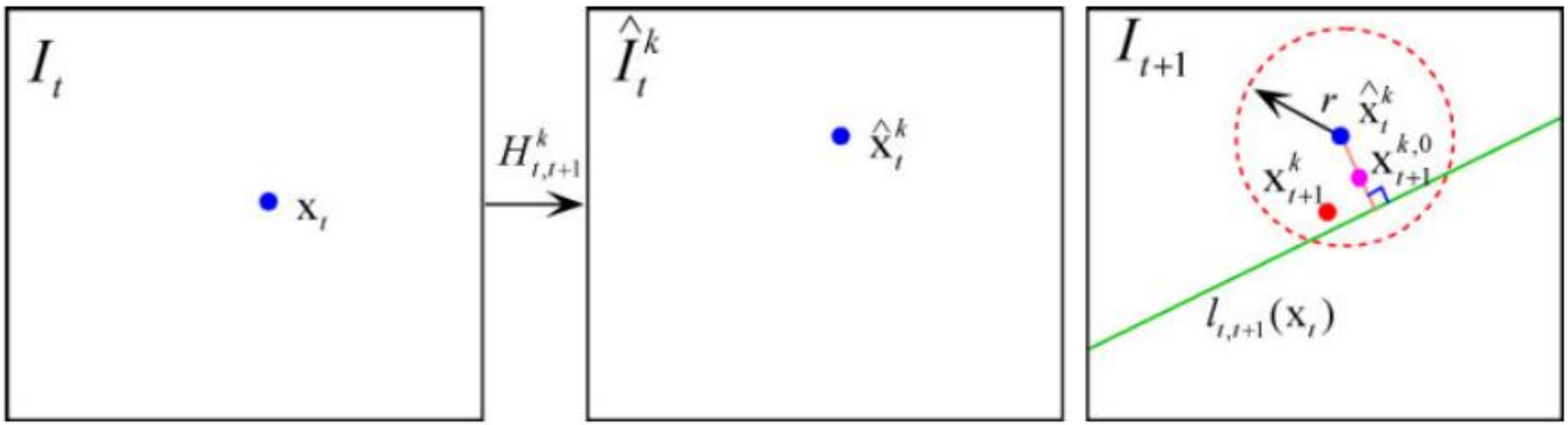
Second-Pass Matching by Planar Motion Segmentation

- Guided matching

$$S^k_{t,t+1}(\mathbf{x}^k_{t+1}) = \sum_{\mathbf{y}\in W} ||\hat{I}^k_t(\hat{\mathbf{x}}^k_t + \mathbf{y}) - I_{t+1}(\mathbf{x}^k_{t+1} + \mathbf{y})||^2 +$$

$$\lambda_e d(\mathbf{x}^k_{t+1}, l_{t,t+1}(\mathbf{x}_t))^2 + \lambda_h ||\hat{\mathbf{x}}^k_t - \mathbf{x}^k_{t+1}||^2$$

Epipolar constraint Homography constraint



Second-Pass Matching with Multi-Homographies

First-Pass Matching
(53 matches)

Direct Searching
(11 matches added)

Our Second-Pass Matching
(346 matches added)

Non-Consecutive track matching

- Fast Matching Matrix Estimation
- Detect overlapping subsequences and join the matched tracks.

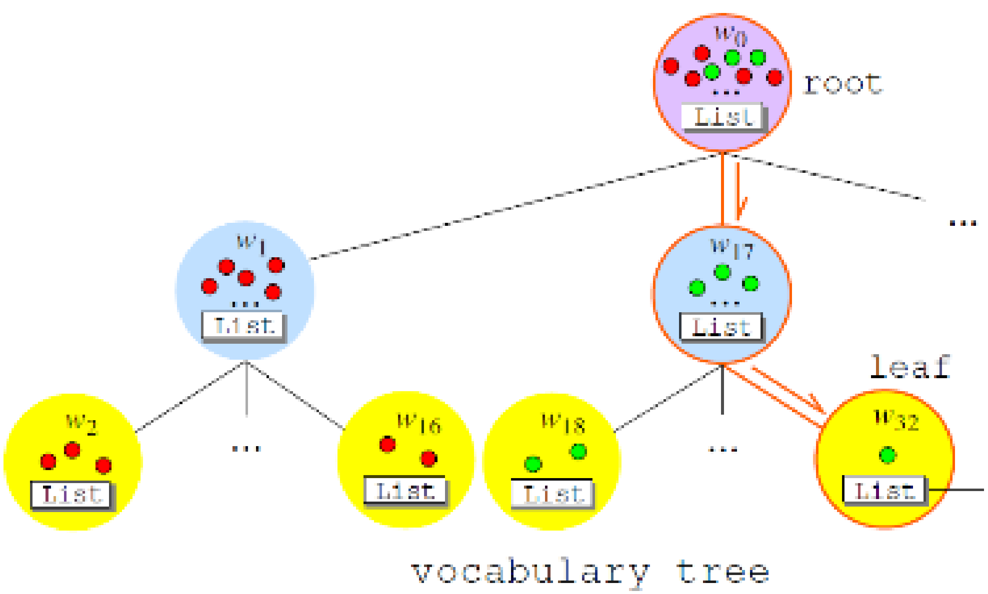
Fast Matching Matrix Estimation

- Each track has a group of description vectors

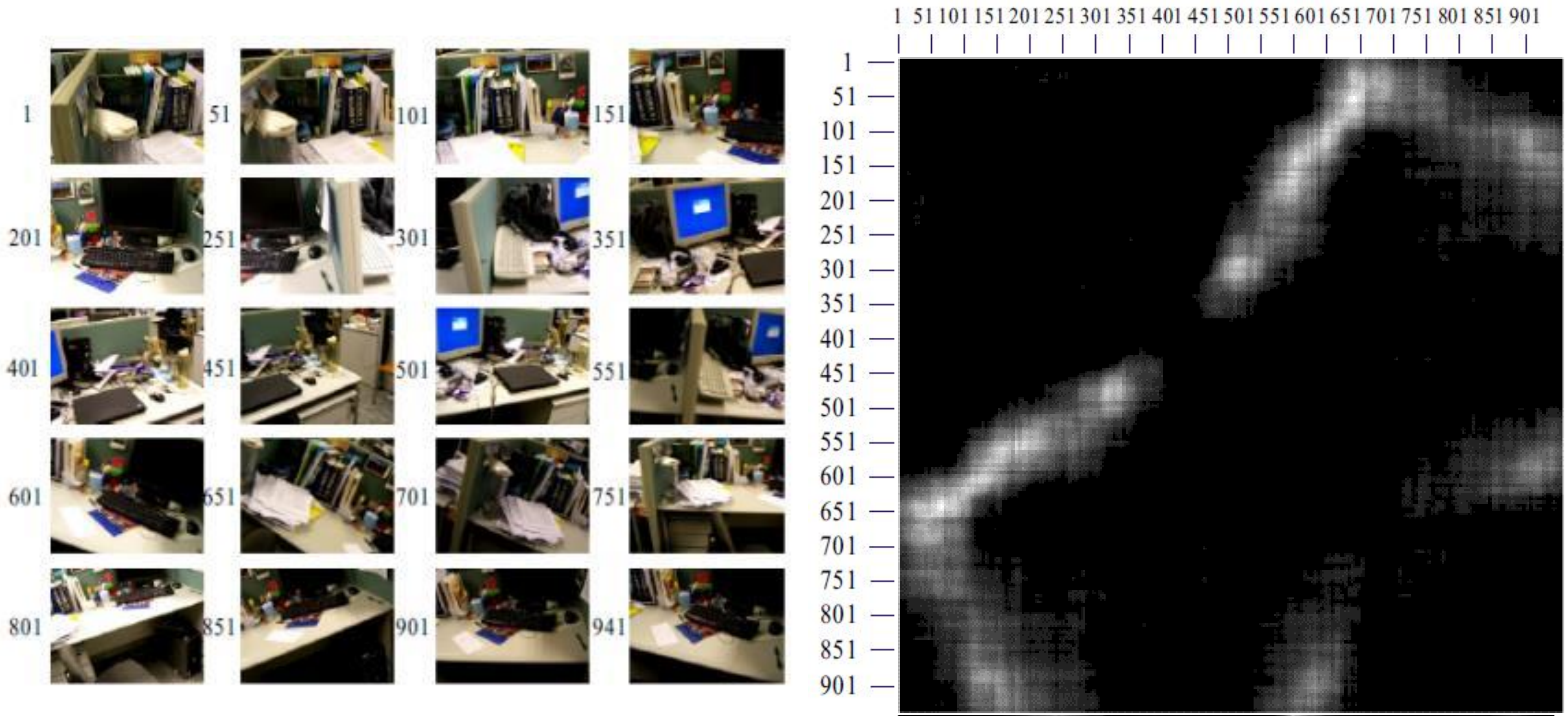
$$\mathcal{P}_{\mathcal{X}} = \{\mathbf{p}(\mathbf{x}_t) | t \in f(\mathcal{X})\}$$

- Track descriptor $\mathbf{p}(\mathcal{X})$

- Use a hierarchical K-means approach to cluster the track descriptors

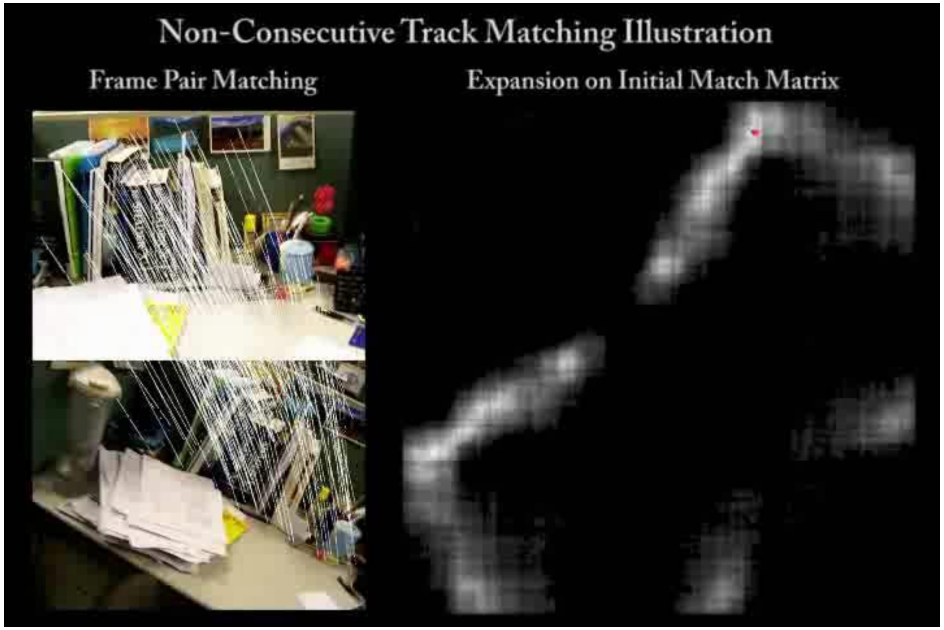


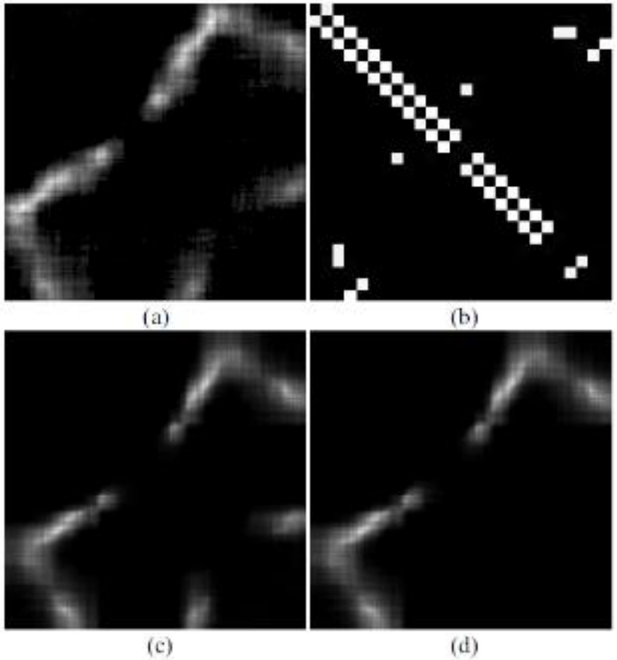
Fast Matching Matrix Estimation



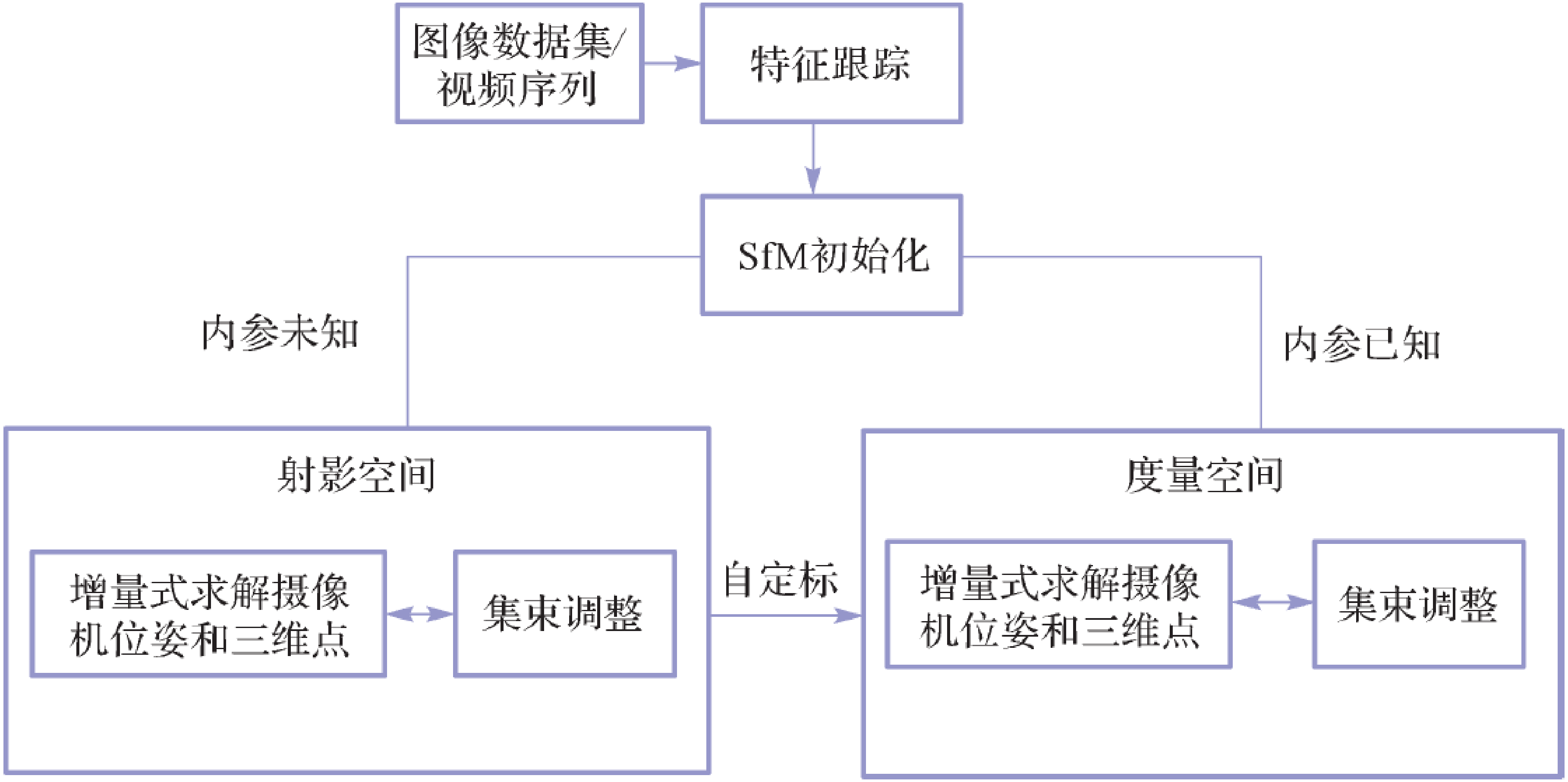
Non-Consecutive Track Matching

- Simultaneously Match Images and Refine Matching Matrix
  - Refine the matching matrix after matching the common features of the selected image pairs.
  - More reliably find the best matching images with the updated matching matrix.





常用的增量式SfM 系统框架



基于自定标的单序列增量式SfM 求解框架

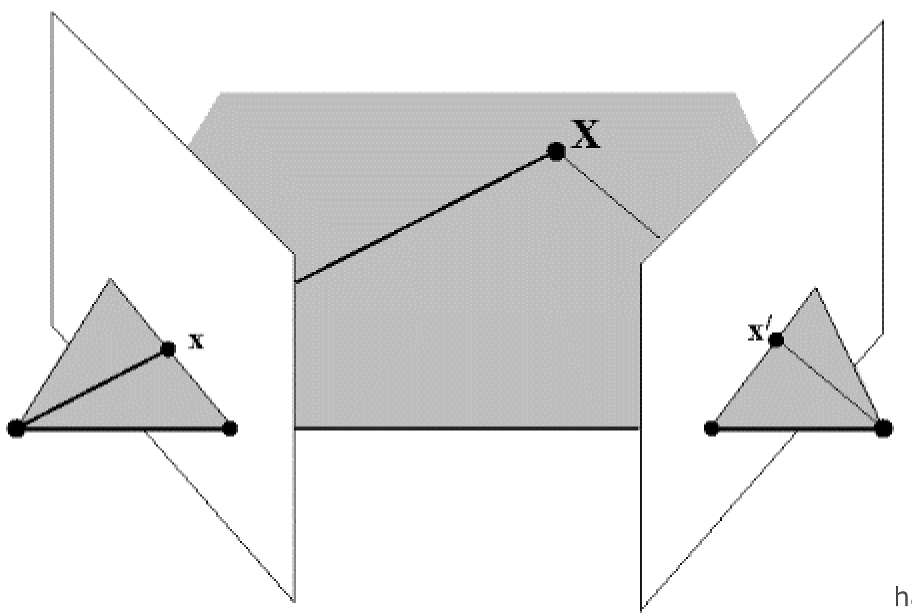
- 1.自动抽取特征点并匹配；
- 2.抽取关键帧组成关键帧序列；
- 3.初始化度量空间下的三维结构和运动：
  - 3.1 选择合适的三帧组进行射影重建的初始化；
  - 3.2 采用增量式求解，并选择合适时机进行自定标，将射影重建转换到度量重建；
- 4.对于每一个新加入求解的关键帧：
  - 4.1 初始化新求解帧的相机参数和相关的三维点；
  - 4.2 用局部集束调整算法对局部已经求解的结构和运动进行求精；
- 5.求解所有非关键帧的相机参数；
- 6.对整个序列恢复的结构和运动用集束调整进行最后优化。

三角化

- 已知**F**, 计算 **P** 和 **P'**

$$P = [I \mid 0] \;;\; P' = [[e']_{\times}F \mid e'] = [M \mid e']$$

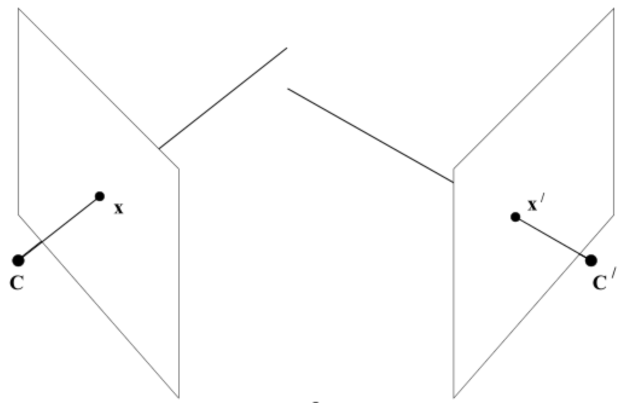
- 已知**x** 和 **x'**
- 计算**X**：　$x = \mathbf{P}X$　　$x' = \mathbf{P}'X$



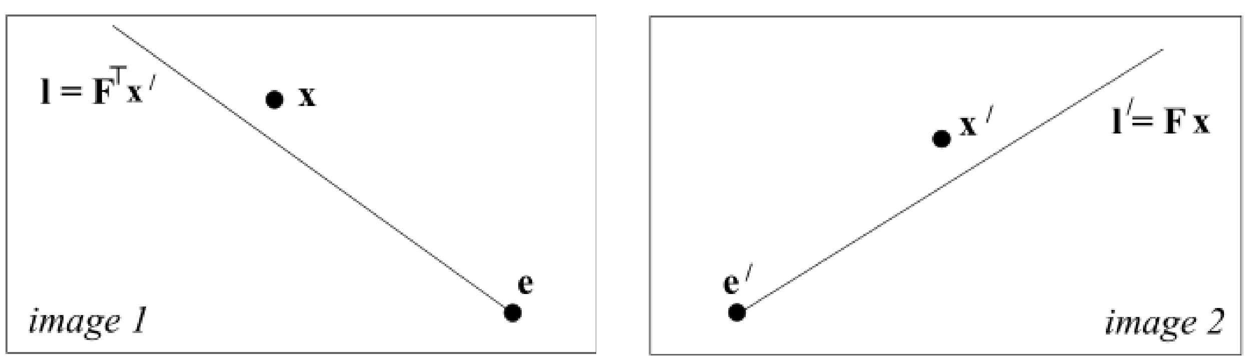
hard Hartley and Andrew Zisserman.“Multiple View Geometry in Computer Vision”. Cambridge University Press, Second Edition 2004.

有噪声情况下的三角化

- 由于存在噪声，反投到三维空间上的射线并不会严格相交



优化投影点到对应极线的距离



Richard Hartley and Andrew Zisserman."Multiple View Geometry in Computer Vision". Cambridge University Press, Second Edition 2004.

线性三角化方法

- 给定方程

  $\mathrm{x}=\mathbf{P}\mathrm{X}$

  $\mathrm{x}'=\mathbf{P}'\mathrm{X}$

  - $\mathbf{p}^{iT}$ 表示P的第i行.
  - 写成矩阵和向量相乘的形式

$$\begin{bmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{bmatrix} \mathrm{x} = 0$$

  - 直接解析求解.
  - 没有几何意义 — 不是最优.

优化几何误差

- 目标函数

$$X = \arg\min_X \sum_i ||\pi(\mathbf{P}_i X) - \mathbf{x}_i||^2$$

- 用Levenberg-Marquart算法求解

已知三维，求解相机位姿

- Compute Projection Matrix

$$\mathbf{P}_i = \arg\min_{\mathbf{P}_i} \sum_j ||\pi(\mathbf{P}_i X_j) - \mathbf{x}_{ij}||^2$$

- Decomposition for Metric Projection Matrix

$$P = K[R \mid t] = [KR \mid Kt] = [M \mid Kt]$$

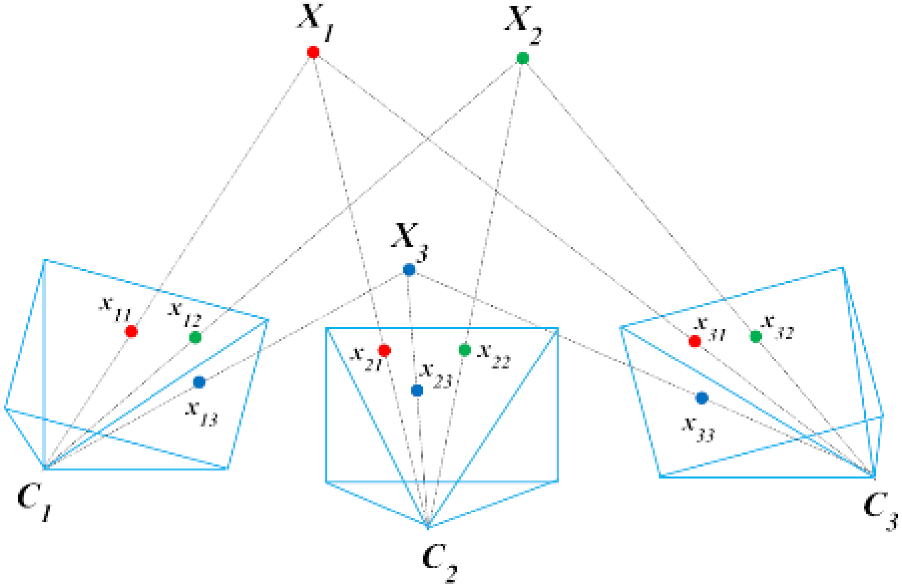
Decompose *M* into *K*, *R* by QR decomposition

$$t = K^{-1}(p_{14}, p_{24}, p_{34})^T$$

Bundle Adjustment

- Definition
  - Refining a visual reconstruction to produce jointly optimal 3D structure and viewing parameter (camera pose and/or calibration) estimates.

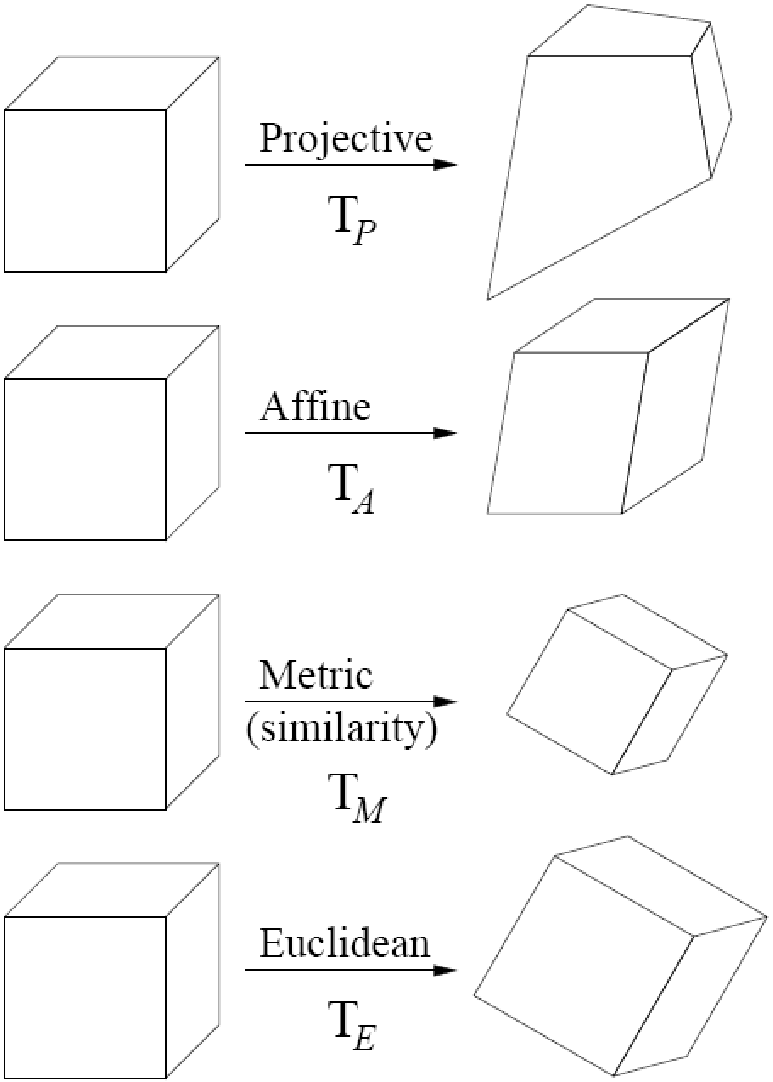
$$\arg\min_{\mathbf{P}_k, \mathrm{X}_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D\left(\mathrm{x}_{ki}, \mathbf{P}_k\left(\mathrm{X}_i\right)\right)^2$$



B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In Workshop on Vision Algorithms, pages 298–372, 1999.

Geometric Ambiguities

ambiguity	DOF	transformation	invariants
projective	15	$\mathbf{T}_P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$	cross-ratio
affine	12	$\mathbf{T}_A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	relative distances along direction parallelism *plane at infinity*
metric	7	$\mathbf{T}_M = \begin{bmatrix} \sigma r_{11} & \sigma r_{12} & \sigma r_{13} & t_x \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & t_y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	relative distances angles *absolute conic*
Euclidean	6	$\mathbf{T}_E = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	absolute distances



Projective Reconstruction $\xrightarrow{\text{Self-Calibration}}$ Metric Reconstruction

Marc Pollefeys. “Visual 3D Modeling from Images”

Self-Calibration

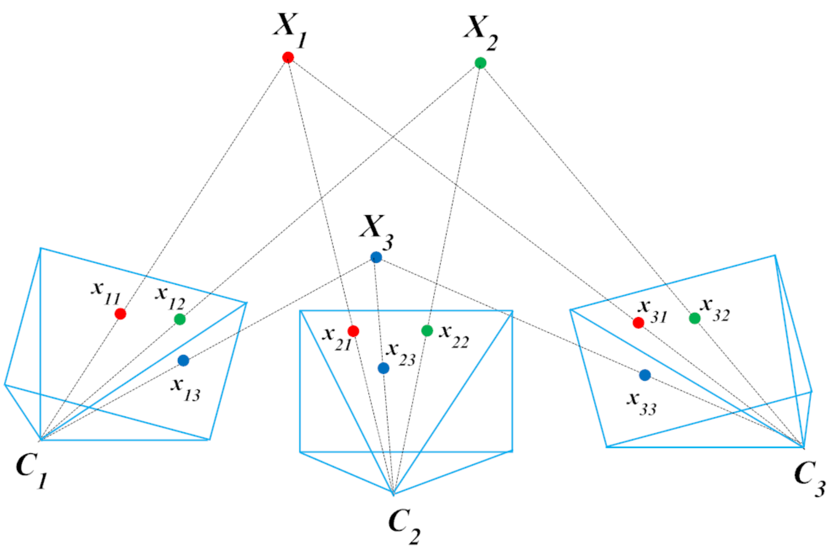
- State-of-the-Art References
  - R.I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, second ed. Cambridge Univ. Press, 2004.
  - M. Pollefeys, L.J. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, Visual Modeling with a Hand-Held Camera, Int'l J. Computer Vision, vol. 59, no. 3, pp. 207-232, 2004.
  - G. Zhang, X. Qin, W. Hua, T.-T. Wong, P.-A. Heng, and H. Bao, Robust Metric Reconstruction from Challenging Video Sequences, Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, 2007.

Bundle Adjustment

Introduction of BA

- Bundle Adjustment (BA) is to jointly optimize all cameras and points, by minimizing reprojection errors

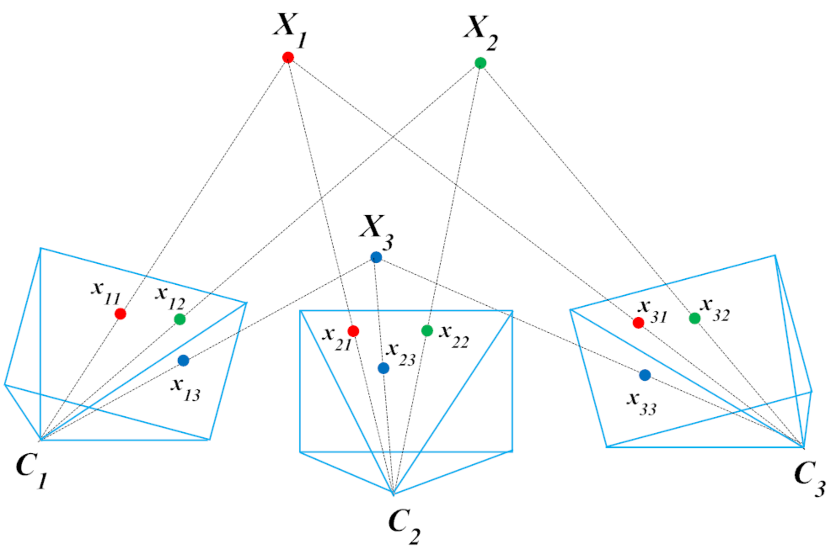
$$\underset{C_1,\ldots C_{N_c},X_1,\ldots,X_{N_p}}{\operatorname{argmin}} \sum \left\| \pi(C_i, X_j) - x_{ij} \right\|^2$$



Introduction of BA

- Bundle Adjustment (BA) is to jointly optimize all cameras and points, by minimizing reprojection errors

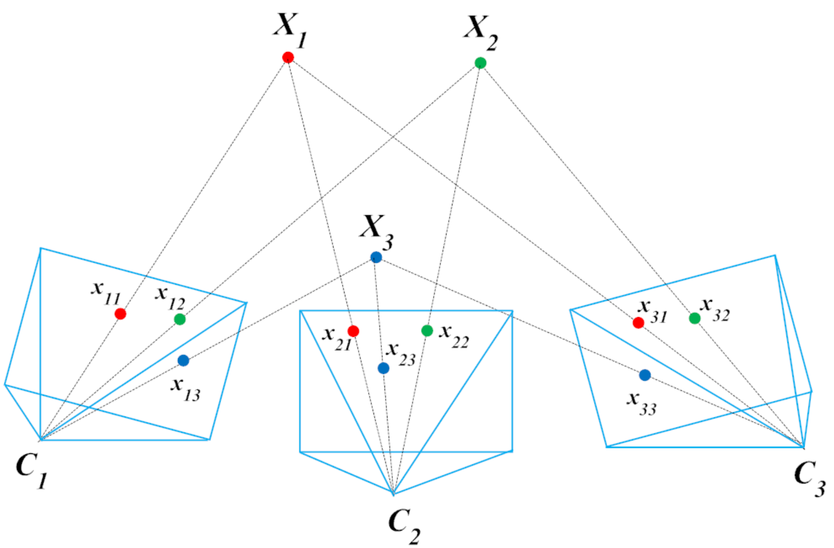
$$\underset{C_1,\ldots C_{N_c},X_1,\ldots,X_{N_p}}{\operatorname{argmin}} \sum \left\| \pi(C_i, X_j) - x_{ij} \right\|^2$$



Introduction of BA

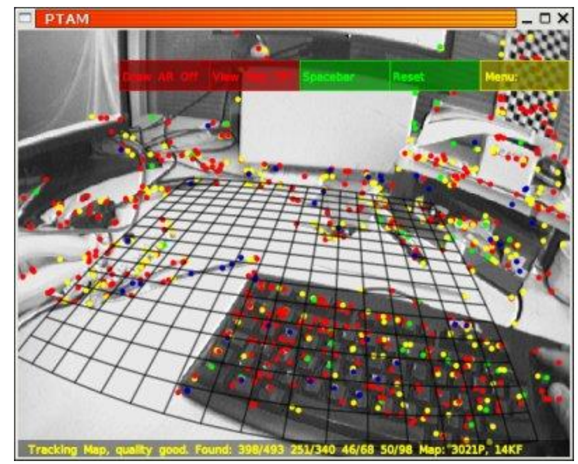
- Bundle Adjustment (BA) is to jointly optimize all cameras and points, by minimizing the reprojection errors

$$\underset{C_1,\dots C_{N_c},X_1,\dots,X_{N_p}}{\operatorname{argmin}} \sum \left\| \pi(C_i, X_j) - x_{ij} \right\|^2$$



Introduction of BA

- BA is a golden step for almost all SfM and SLAM systems



Bundler (SfM)

PTAM (SLAM)

Outline

- Theories in BA
- BA for large scale SfM
- BA for realtime SLAM

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}}\, E(x)$$

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}} E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}} E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^TA)x + 2(A^Tb)x + b^Tb$$

Nonlinear Least Sqaures

$$x^* = \arg\min_x E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^T A)x + 2(A^T b)x + b^T b$$

$$\frac{\partial E(x)}{\partial x} = 2(A^T A x + A^T b) = 0$$

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}}\, E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^TA)x + 2(A^Tb)x + b^Tb$$

$$\frac{\partial E(x)}{\partial x} = 2(A^TAx + A^Tb) = 0$$

$$A^TA\,x = -A^Tb$$

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}} E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^T A)x + 2(A^T b)x + b^T b$$

$$\frac{\partial E(x)}{\partial x} = 2(A^T Ax + A^T b) = 0$$

$$A^T A\, x = -A^T b$$

Nonlinear case

$$E(x) = \|\varepsilon(x)\|^2$$

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}} E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^TA)x + 2(A^Tb)x + b^Tb$$

$$\frac{\partial E(x)}{\partial x} = 2(A^TAx + A^Tb) = 0$$

$$A^TA\,x = -A^Tb$$

Nonlinear case

$$E(x) = \|\varepsilon(x)\|^2$$

$$x^* = \hat{x} + \delta_x$$

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}} E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^TA)x + 2(A^Tb)x + b^Tb$$

$$\frac{\partial E(x)}{\partial x} = 2(A^TAx + A^Tb) = 0$$

$$A^TA\,x = -A^Tb$$

Nonlinear case

$$E(x) = \|\varepsilon(x)\|^2$$

$$x^* = \hat{x} + \delta_x$$

$$\varepsilon(x^*) \approx \varepsilon(\hat{x}) + J\ \delta_x$$

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}}\, E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^TA)x + 2(A^Tb)x + b^Tb$$

$$\frac{\partial E(x)}{\partial x} = 2(A^TAx + A^Tb) = 0$$

$$A^TA\,x = -A^Tb$$

Nonlinear case

$$E(x) = \|\varepsilon(x)\|^2$$

$$x^* = \hat{x} + \delta_x$$

$$\varepsilon(x^*) \approx \varepsilon(\hat{x}) + \boxed{J}\,\delta_x$$

Jacobian matrix

$$J = \left.\frac{\partial \varepsilon}{\partial x}\right|_{x=\hat{x}}$$

Nonlinear Least Sqaures

$$x^* = \operatorname*{argmin}_x E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^TA)x + 2(A^Tb)x + b^Tb$$

$$\frac{\partial E(x)}{\partial x} = 2(A^TAx + A^Tb) = 0$$

$$A^TA\,x = -A^Tb$$

Nonlinear case

$$E(x) = \|\varepsilon(x)\|^2$$

$$x^* = \hat{x} + \delta_x$$

$$\varepsilon(x^*) \approx \varepsilon(\hat{x}) + \boxed{J}\,\delta_x$$

Jacobian matrix

$$J = \left.\frac{\partial \varepsilon}{\partial x}\right|_{x=\hat{x}}$$

$$E(x) \approx {\delta_x}^T(J^TJ)\delta_x + 2(J^T\varepsilon)\delta_x + \varepsilon^T\varepsilon$$

$$J^TJ\,\delta_x = -J^T\varepsilon$$

Nonlinear Least Sqaures

$$x^* = \underset{x}{\operatorname{argmin}}\, E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^T A)x + 2(A^T b)x + b^T b$$

$$\frac{\partial E(x)}{\partial x} = 2(A^T A x + A^T b) = 0$$

$$\boxed{A^T A}\, x = -A^T b$$

Hessian matrix

Nonlinear case

$$E(x) = \|\varepsilon(x)\|^2$$

$$x^* = \hat{x} + \delta_x$$

$$\varepsilon(x^*) \approx \varepsilon(\hat{x}) + \boxed{J}\, \delta_x$$

Jacobian matrix

$$J = \left.\frac{\partial \varepsilon}{\partial x}\right|_{x=\hat{x}}$$

$$E(x) \approx {\delta_x}^T (J^T J)\delta_x + 2(J^T \varepsilon)\delta_x + \varepsilon^T \varepsilon$$

$$\boxed{J^T J}\, \delta_x = -J^T \varepsilon$$

Nonlinear Least Sqaures

$$x^* = \operatorname*{argmin}_x E(x)$$

Linea case

$$E(x) = \|Ax + b\|^2$$

$$= x^T(A^TA)x + 2(A^Tb)x + b^Tb$$

$$\frac{\partial E(x)}{\partial x} = 2(A^TAx + A^Tb) = 0$$

$$\boxed{\boxed{A^TA}\,x = -A^Tb}$$

Nonlinear case

$$E(x) = \|\varepsilon(x)\|^2$$

$$x^* = \hat{x} + \delta_x$$

$$\varepsilon(x^*) \approx \varepsilon(\hat{x}) + \boxed{J}\,\delta_x$$

$$E(x) \approx \delta_x{}^T(J^TJ)\delta_x + 2(J^T\varepsilon)\delta_x + \varepsilon^T\varepsilon$$

$$\boxed{\boxed{J^TJ}\,\delta_x = -J^T\varepsilon}$$

Jacobian matrix

$$J = \left.\frac{\partial \varepsilon}{\partial x}\right|_{x=\hat{x}}$$

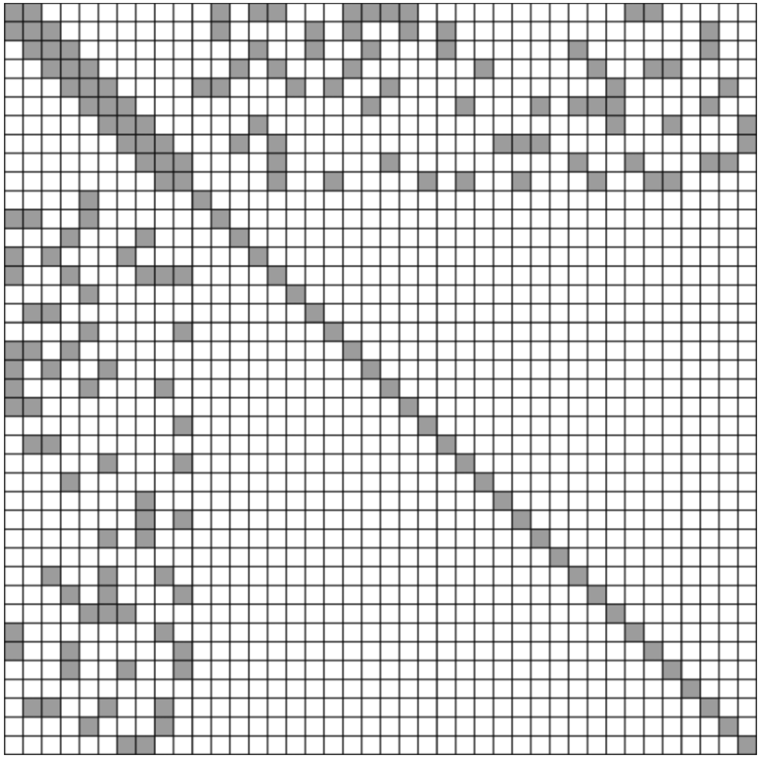
Hessian matrix

normal equation

Sparse Bundle Adjustment

$$\underset{C_1,\ldots C_{N_c},X_1,\ldots,X_{N_p}}{\operatorname{argmin}} \sum \left\| \pi(\underbrace{C_i}_{\text{1 camera}}, \underbrace{X_j}_{\text{1 point}}) - x_{ij} \right\|^2$$

Sparsity pattern of Hessian



Sparse Bundle Adjustment

- An simple example
  - 3 cameras
  - 4 points
  - all points are visible in all cameras

Sparse Bundle Adjustment

$$
\begin{array}{c}
\qquad\quad \overbrace{\hphantom{A_{11}\quad 0\quad 0}}^{\text{3 cameras}}\qquad \overbrace{\hphantom{B_{11}\quad 0\quad 0\quad 0}}^{\text{4 points}} \\
J = \begin{pmatrix}
A_{11} & 0 & 0 & B_{11} & 0 & 0 & 0 \\
A_{12} & 0 & 0 & 0 & B_{12} & 0 & 0 \\
A_{13} & 0 & 0 & 0 & 0 & B_{13} & 0 \\
A_{14} & 0 & 0 & 0 & & 0 & B_{14} \\
0 & A_{21} & 0 & B_{21} & 0 & 0 & 0 \\
0 & A_{22} & 0 & 0 & B_{22} & 0 & 0 \\
0 & A_{23} & 0 & 0 & 0 & B_{23} & 0 \\
0 & A_{24} & 0 & 0 & 0 & 0 & B_{34} \\
0 & 0 & A_{31} & B_{31} & 0 & 0 & 0 \\
0 & 0 & A_{32} & 0 & B_{32} & 0 & 0 \\
0 & 0 & A_{33} & 0 & 0 & B_{33} & 0 \\
0 & 0 & A_{34} & 0 & 0 & 0 & B_{34}
\end{pmatrix}, e = \left.\begin{pmatrix}
e_{11} \\ e_{12} \\ e_{13} \\ e_{14} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{24} \\ e_{31} \\ e_{32} \\ e_{33} \\ e_{34}
\end{pmatrix}\right\} \text{12 re-projections}
\end{array}
$$

Sparse Bundle Adjustment

$$J^T J \delta_x = -J^T \varepsilon$$

Sparse Bundle Adjustment

$$\boxed{J^T J}\delta_x = -J^T \varepsilon$$

$$J^T J = \begin{pmatrix} U & W \\ W^T & V \end{pmatrix} = \begin{pmatrix} U_1 & 0 & 0 & W_{11} & W_{12} & W_{13} & W_{14} \\ 0 & U_2 & 0 & W_{21} & W_{22} & W_{23} & W_{24} \\ 0 & 0 & U_3 & W_{31} & W_{32} & W_{33} & W_{34} \\ W_{11}^T & W_{21}^T & W_{31}^T & V_1 & 0 & 0 & 0 \\ W_{12}^T & W_{22}^T & W_{32}^T & 0 & V_2 & 0 & 0 \\ W_{13}^T & W_{21}^T & W_{33}^T & 0 & 0 & V_3 & 0 \\ W_{14}^T & W_{24}^T & W_{34}^T & 0 & 0 & 0 & V_4 \end{pmatrix}$$

$$U_i = \sum_{j=1}^{4} A_{ij}^T A_{ij}, V_j = \sum_{i=1}^{3} B_{ij}^T B_{ij}, W_{ij} = A_{ij}^T B_{ij}$$

Sparse Bundle Adjustment

$$J^T J \boxed{\delta_x} = -J^T \varepsilon$$

$$d_x = \begin{pmatrix} d_C \\ d_X \end{pmatrix} = \begin{pmatrix} d_{C_1}^T & d_{C_2}^T & d_{C_3}^T & d_{X_1}^T & d_{X_2}^T & d_{X_3}^T & d_{X_4}^T \end{pmatrix}^T$$

Sparse Bundle Adjustment

$$J^T J \delta_x = -\boxed{J^T \varepsilon}$$

$$J^T \mathbf{e} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_1 & u_2 & u_3 & v_1 & v_2 & v_3 & v_4 \end{pmatrix}^T$$

$$u_i = \sum_{j=1}^{4} A_{ij}^T \mathbf{e}_{ij}$$

$$v_j = \sum_{i=1}^{3} B_{ij}^T \mathbf{e}_{ij}$$

Sparse Bundle Adjustment

- In general, NOT all points are visible in all cameras

$$U_i = \sum_{j=1}^{4} A_{ij}^T A_{ij}, V_j = \sum_{i=1}^{3} B_{ij}^T B_{ij}, W_{ij} = A_{ij}^T B_{ij}$$

  - $A_{ij} = B_{ij} = 0$ if j-th point is not observed in i-th camera
  - More sparse structure, more speedup

Sparse Bundle Adjustment

$$J^T J \delta_x = -J^T \varepsilon$$

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = - \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{pmatrix} U - WV^{-1}W^T & 0 \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = - \begin{pmatrix} u - WV^{-1}v \\ v \end{pmatrix}$$

$$S = U - WV^{-1}W^T$$ Schur Complement

$$Sd_C = -(u - WV^{-1}v)$$ Compute cameras first (# cameras << # points)

$$Vd_X = -v - W^T d_C$$ back substitution for points

Schur Complement for Cameras

$$(U - WV^{-1}W^{T})d_{C} = -(u - WV^{-1}v)$$

Schur Complement for Cameras

$$(U - \boxed{WV^{-1}W^T})d_C = -(u - WV^{-1}v)$$

$$WV^{-1}W^T = \begin{pmatrix} S_{11} & S_{12} & S_{13} \\ S_{12}^T & S_{22} & S_{23} \\ S_{13}^T & S_{23}^T & S_{33} \end{pmatrix}$$

$$S_{i_1 i_2} = \sum_{j=1}^{4} W_{i_1 j} V_j^{-1} W_{i_2 j}^T$$

Schur Complement for Cameras

$$(U - WV^{-1}W^T)d_C = -(u - \boxed{WV^{-1}v})$$

$$WV^{-1}e_X = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix}$$

$$g_i = \sum_{j=1}^{4} W_{ij} V_j^{-1} v_j$$

Schur Complement for Cameras

- Again, in general NOT all points are visible in all cameras

$$S_{i_1 i_2} = \sum_{j=1}^{4} W_{i_1 j} V_j^{-1} W_{i_2 j}^T$$

  - $S_{i_1 i_2} = 0$ if i_1-th camera has no common points with i_2-th camera
  - More sparse structure more speedup

Back Substitution for Points

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

$$(U - WV^{-1}W^T)d_C = -(u - WV^{-1}v)$$

$$W^T d_C + V d_X = -v$$

$$d_{X_j} = -v_j - \sum_{i=1}^{3} W_{ij}^T d_{C_i}$$

- Each point can be solve independently
- Again, $W_{ij} = 0$ if j-th point is not observed in i-th camera

Probability Interpretation

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix}\begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

joint density $\mathrm{P}(\delta_C, \delta_X) = P(\delta_C)P(\delta_X|\delta_C)$

$$(U - WV^{-1}W^T)d_C = -(u - WV^{-1}v)$$

marginalize out $\mathrm{P}(\delta_X)$ to get $\mathrm{P}(\delta_C)$

$$W^T d_C + V d_X = -v$$

conditional probability $P(\delta_X|\delta_C)$

Factor Graph Interpretation

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix}\begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

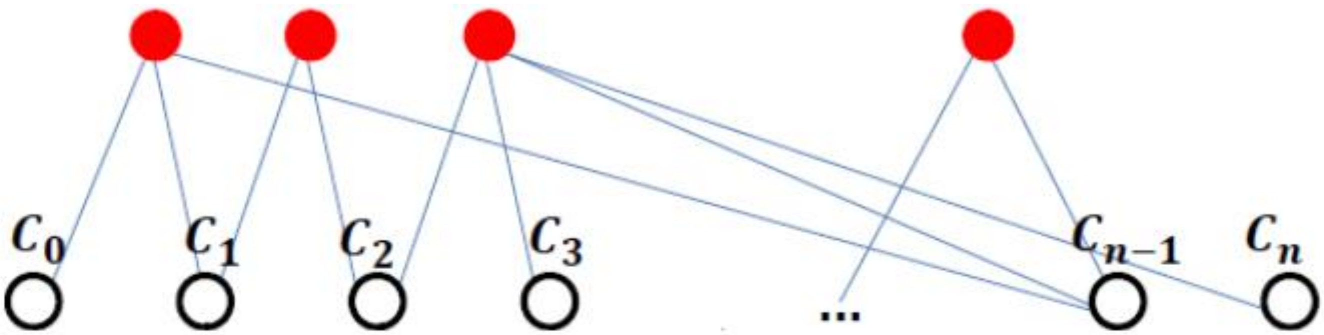
joint density $\mathrm{P}(\delta_C, \delta_X) = P(\delta_C)P(\delta_X|\delta_C)$

$$(U - WV^{-1}W^T)d_C = -(u - WV^{-1}v)$$

marginalize out $\mathrm{P}(\delta_X)$ to get $\mathrm{P}(\delta_C)$

$$W^T d_C + V d_X = -v$$

conditional probability $P(\delta_X|\delta_C)$



Factor Graph Interpretation

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix}\begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

joint density $\mathrm{P}(\delta_C, \delta_X) = P(\delta_C)P(\delta_X|\delta_C)$

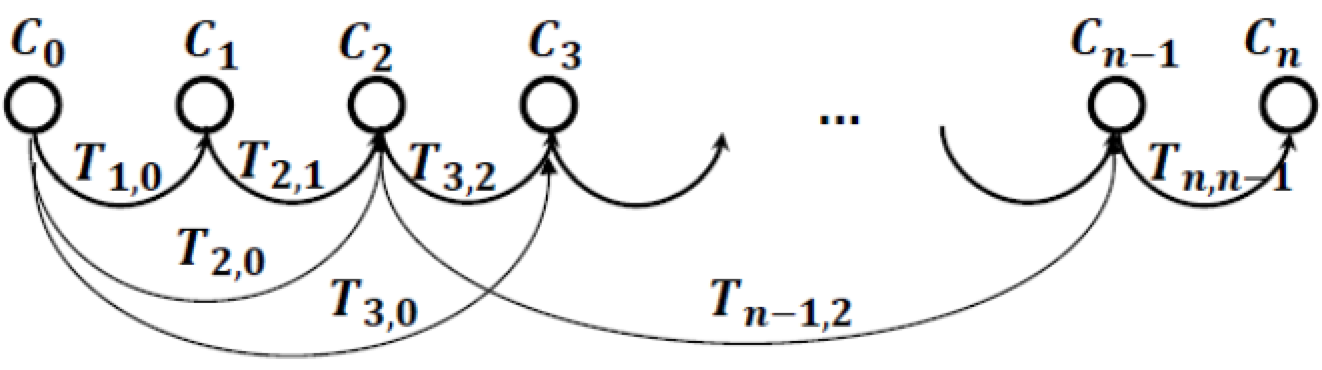
$$(U - WV^{-1}W^T)d_C = -(u - WV^{-1}v)$$

marginalize out $\mathrm{P}(\delta_X)$ to get $\mathrm{P}(\delta_C)$

$$W^T d_C + V d_X = -v$$

conditional probability $P(\delta_X|\delta_C)$

$$\underset{C_1,\ldots C_{N_C}}{\operatorname{argmin}} \sum \left\| C_i \ominus C_j \ominus T_{i,j} \right\|^2$$



Pose Graph Optimization

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = - \begin{pmatrix} u \\ v \end{pmatrix}$$

joint density $\mathrm{P}(\delta_C, \delta_X) = P(\delta_C)P(\delta_X|\delta_C)$

$$(U - WV^{-1}W^T)d_C = -(u - WV^{-1}v)$$

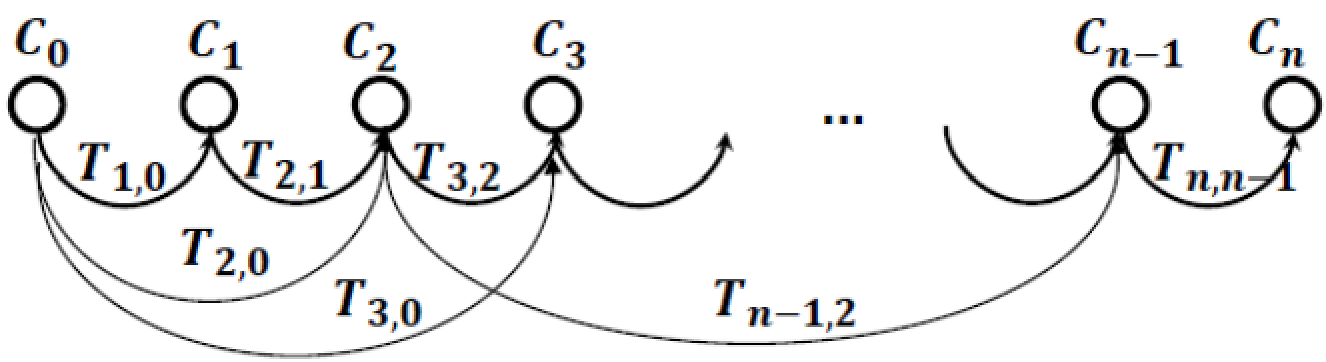
marginalize out $\mathrm{P}(\delta_X)$ to get $\mathrm{P}(\delta_C)$

$$\cancel{W^T d_C + V d_X = -v}$$

conditional probability $P(\delta_X|\delta_C)$

$$\underset{C_1,\ldots C_{N_C}}{\operatorname{argmin}} \sum \left\| C_i \ominus C_j \ominus T_{i,j} \right\|^2$$

Pose graph optimization is an approximation of BA



Steps in BA

$$\begin{pmatrix} \boxed{\begin{matrix} U & W \\ W^T & V \end{matrix}} \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} \boxed{\begin{matrix} u \\ v \end{matrix}} \end{pmatrix}$$

$$(U - WV^{-1}W^T)d_C = -(u - WV^{-1}v)$$

$$W^T d_C + V d_X = -v$$

1. Construct normal equation

$\mathbf{U} = \mathbf{0}$; $\mathbf{V} = \mathbf{0}$; $\mathbf{W} = \mathbf{0}$; $\mathbf{u} = \mathbf{0}$; $\mathbf{v} = \mathbf{0}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
  Construct linearized equation (11)
  $\mathbf{U}_{ii} += \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{C}_{ij}}$
  $\mathbf{V}_{jj} += \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
  $\mathbf{u}_i += \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{e}_{ij}$
  $\mathbf{v}_j += \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{e}_{ij}$
  $\mathbf{W}_{ij} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
end for

Steps in BA

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

$$\left(\boxed{U - WV^{-1}W^T}\right) d_C = -\left(\boxed{u - WV^{-1}v}\right)$$

$$W^T d_C + V d_X = -v$$

1. Construct normal equation
2. Construct Schur complement

$\mathbf{S} = \mathbf{U}$
for each point j and each camera pair $(i_1, i_2) \in \mathcal{V}_j \times \mathcal{V}_j$
do
  $\mathbf{S}_{i_1 i_2} -= \mathbf{W}_{i_1 j} \mathbf{V}_{jj}^{-1} \mathbf{W}_{i_2 j}^{\top}$
end for
$\mathbf{g} = \mathbf{u}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
  $\mathbf{g}_i -= \mathbf{W}_{ij} \mathbf{V}_{jj}^{-1} \mathbf{v}_j$
end for

Steps in BA

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

$$(U - WV^{-1}W^T)\boxed{d_C} = -(u - WV^{-1}v)$$

$$W^T d_C + V d_X = -v$$

1. Construct normal equation
2. Construct Schur complement
3. Solve cameras
    - Sparse Cholesky factorization
    - Preconditioned Conjugate Gradient (PCG)
        - explicitly leverages the sparseness

Steps in BA

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} d_C \\ d_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

$$(U - WV^{-1}W^T)d_C = -(u - WV^{-1}v)$$

$$W^T d_C + V\boxed{d_X} = -v$$

1. Construct normal equation
2. Construct Schur complement
3. Solve cameras
4. Solve points

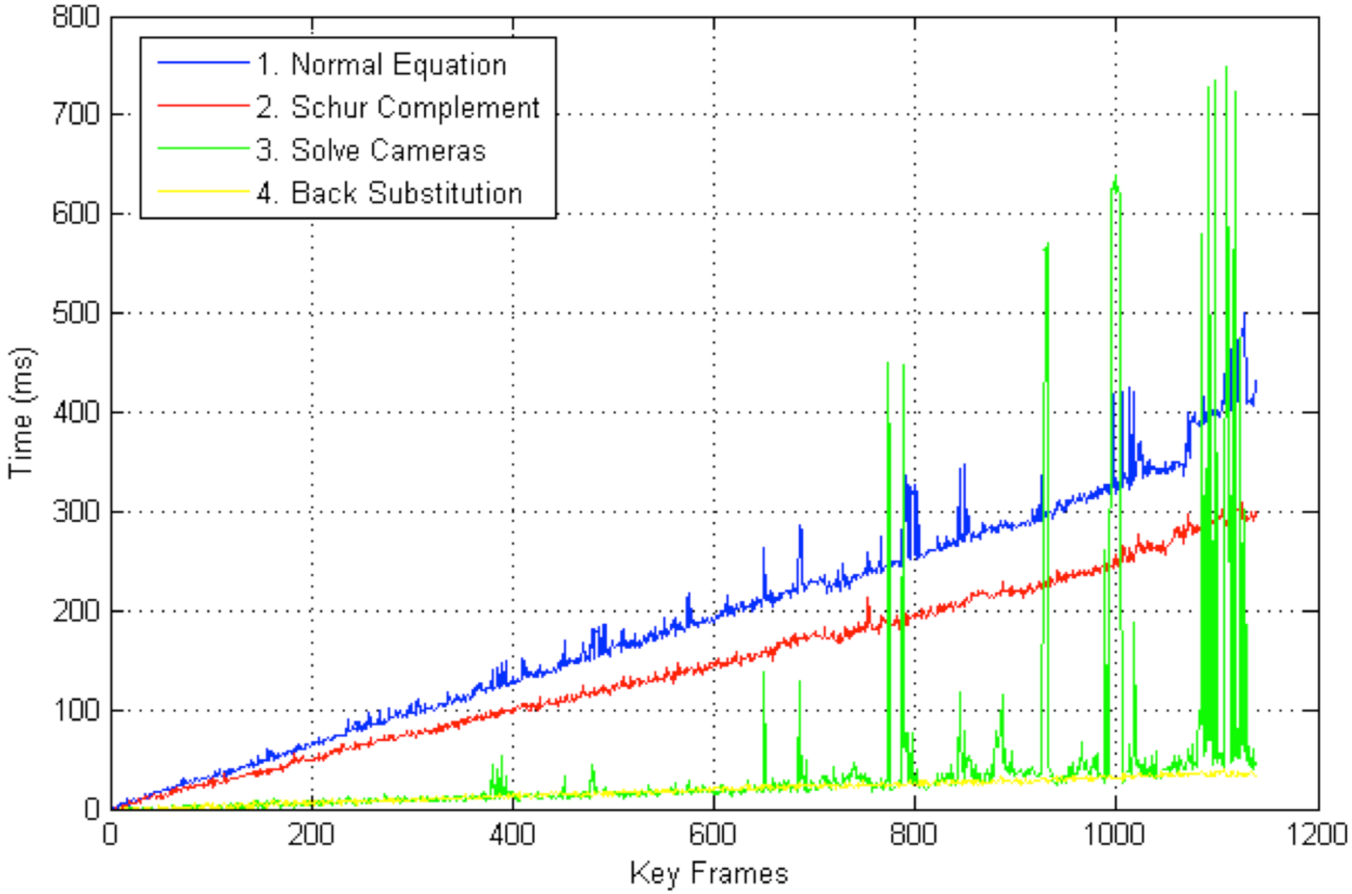
for each point j **do**

$$\delta \mathbf{x}_j = \mathbf{V}_{jj}^{-1}\left(\mathbf{v}_j - \sum_{i \in \mathcal{V}_j} \mathbf{W}_{ij}^{\top} \delta \mathbf{c}_i\right)$$

end for

Runtime for Each Steps

- Runtime increases with #cameras



Challenge of BA

- Efficiency is the main challenge of BA
- Keyframe or pose graph simplification cannot completely solve this problem
- Two scenarios
  - Large scale SfM
  - Realtime SLAM

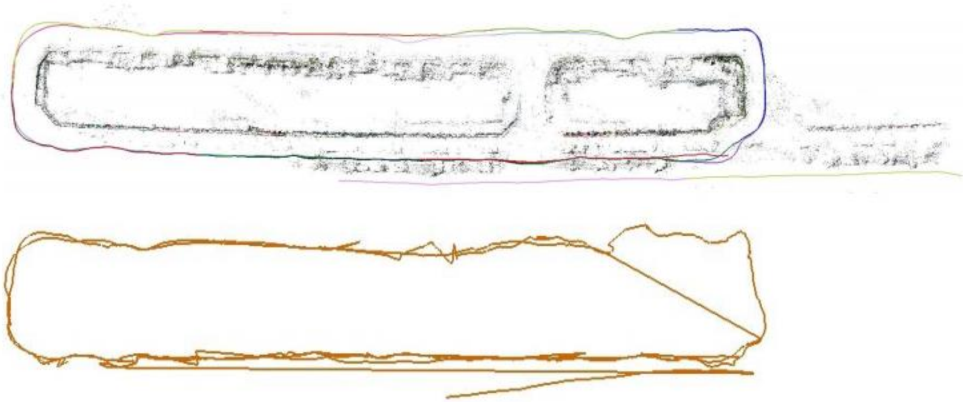
Outline

- Theories in BA
- BA for large scale SfM
- BA for realtime SLAM

Outline

- Theories in BA
- BA for large scale SfM
- BA for realtime SLAM

Challenges for Large-scale SfM



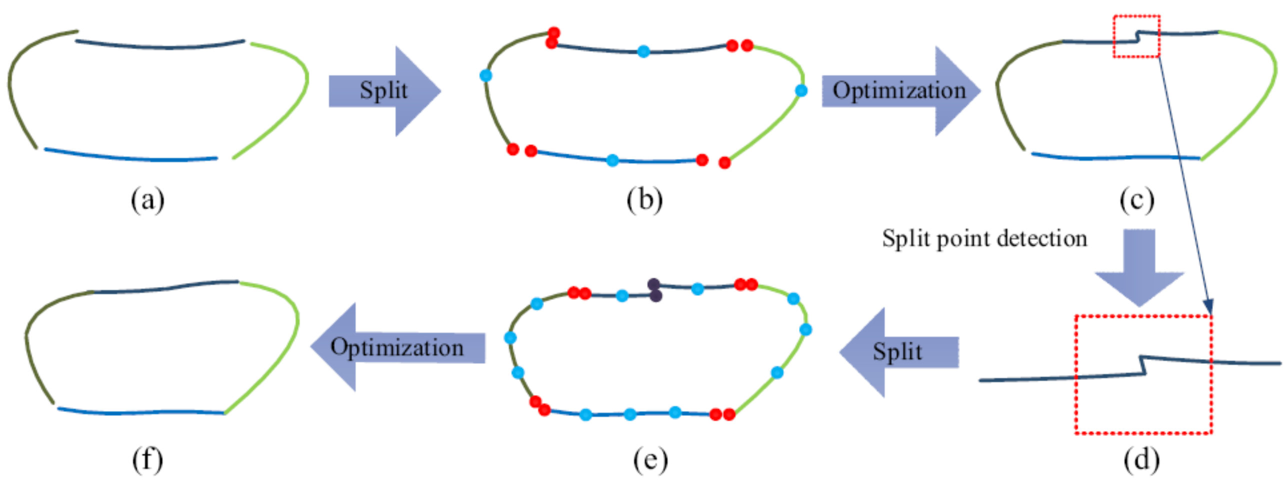
- Global BA
  - Huge #variables
  - Memory limit
  - Time-consuming
- Iterative local BA
  - Large error is difficult to be propagated to whole scene
  - Easily stuck in local optimum
- Pose graph optimization
  - Approximation of BA
  - May not sufficiently minimize the error
- Solutions
  - Hierarchical BA
  - Distributed BA

Segment-based Hierarchical BA

Zhang G, Liu H, Dong Z, et al. Efficient non-consecutive feature tracking for robust structure-from-motion[J]. IEEE Transactions on Image Processing, 2016, 25(12): 5957-5970.

Segment-based Hierarchical BA

- Observations
  - Incremental SfM results in high local accuracy, but low global accuracy
  - The DoF is unnecessarily large by traditional BA
- Solution
  - Split a long sequence to multiple short sub-sequences
  - 7-DoF similarity transformation for each sub-sequence
  - Only optimize overlapping points
  - Hierarchically align sub-sequences



Split Point Detection

- The split point should be at the place where the relative pose error is large, which is unknown in advance
- Naïve solution
  - large re-projection error
  - cannot reliably reflect the relative pose error
- Our solution
  - Revisit the normal equation

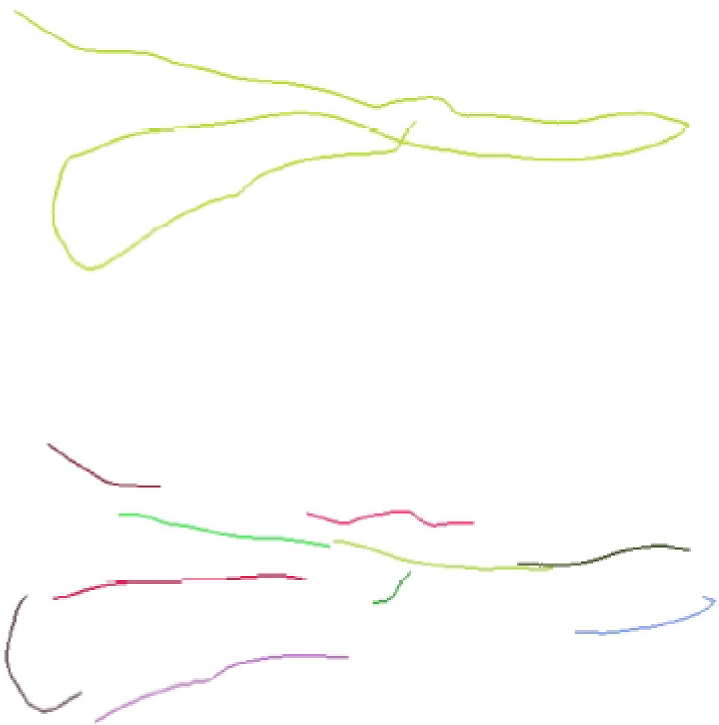
$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \delta_C \\ \delta_X \end{pmatrix} = -\begin{pmatrix} u \\ v \end{pmatrix}$$

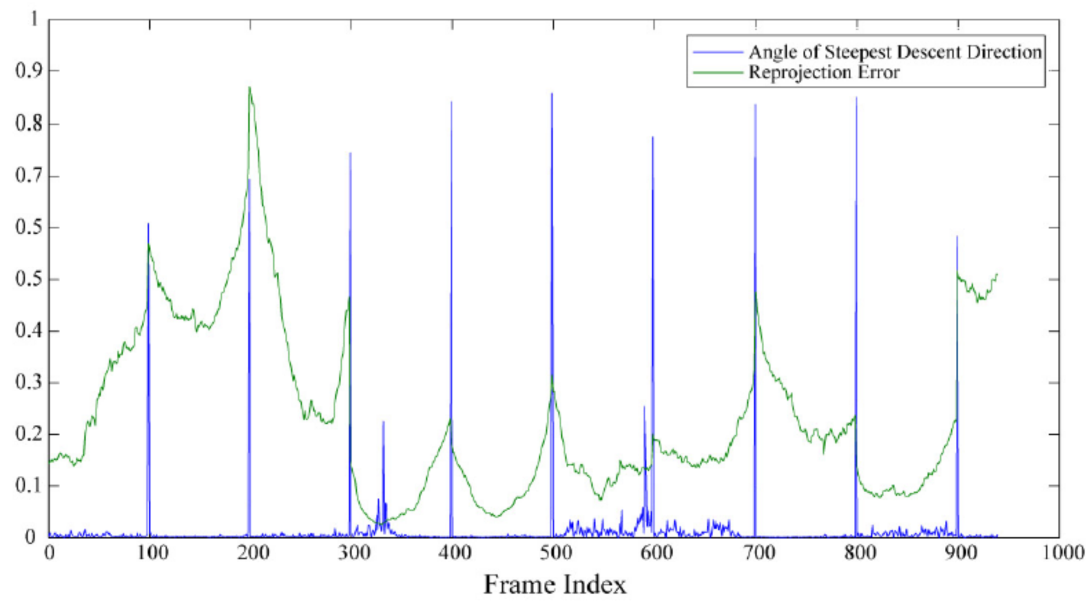
$$u_i = \sum_j A_{ij}^T \varepsilon_{ij}$$

  - ε_{ij} in i-th frame will be best minimized along u_i
  - The inconsistency between i-th and $(i+1)$-th frame

$$E(i, i+1) = \arccos\left(\frac{u_i^T u_{i+1}}{\|u_i\| \|u_{i+1}\|}\right)$$

Split Point Detection



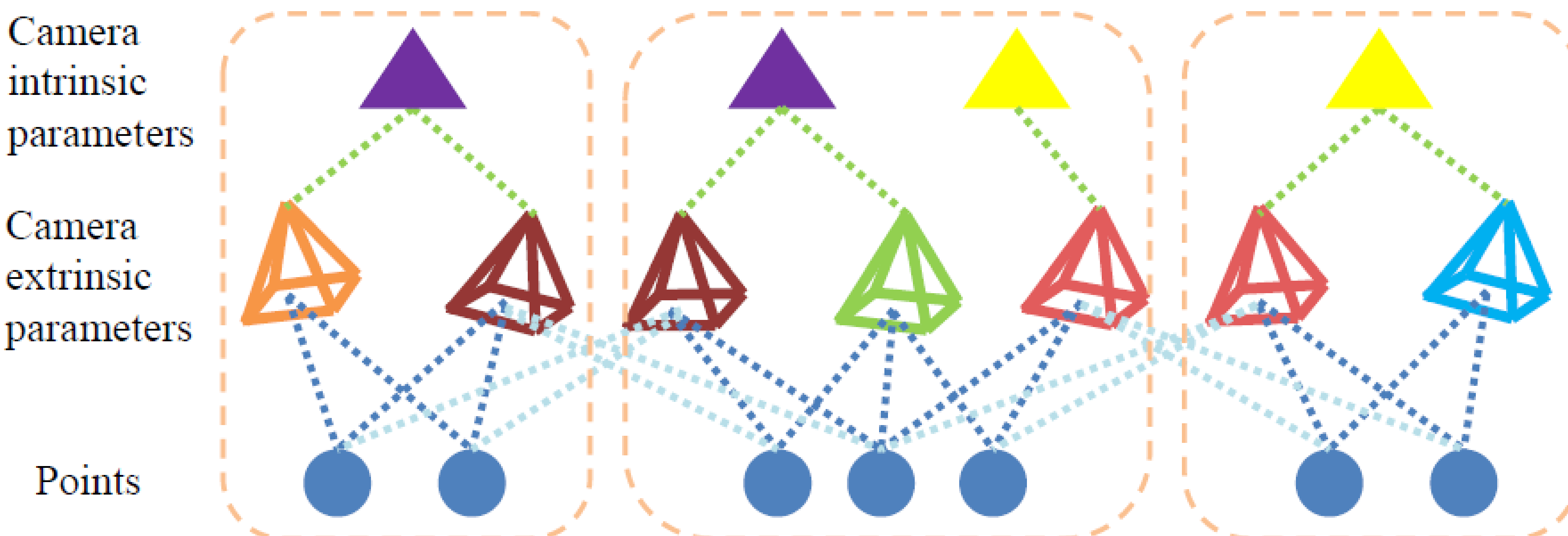


Distributed BA by Global Camera Consensus

Zhang R, Zhu S, Fang T, et al. Distributed very large scale bundle adjustment by global camera consensus[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 29-38.

Split Cameras or Points

- Split cameras
  - Broadcast overlapping points, huge overhead
- Split points
  - Broadcast overlapping cameras, called camera consensus



ADMM for Constrained Optimization

- Constrained optimization

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{w} \end{array}$$

- The ADMM algorithm

$$\begin{aligned} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = {} & f(\mathbf{x}) + g(\mathbf{z}) \\ & + \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{w}) \\ & + \frac{\rho}{2}||\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{w}||_2^2 \end{aligned}$$

$$\begin{aligned} \mathbf{x}^{t+1} &= \arg\min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^t, \mathbf{y}^t) \\ \mathbf{z}^{t+1} &= \arg\min_{\mathbf{z}} L_\rho(\mathbf{x}^{t+1}, \mathbf{z}, \mathbf{y}^t) \\ \mathbf{y}^{t+1} &= \mathbf{y}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{w}) \end{aligned}$$

ADMM for Distributed BA

- Constrained optimization

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) + g(\mathbf{z}) \\ &\text{subject to} && \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{w} \end{aligned}$$

- The ADMM algorithm

$$\begin{aligned} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) &+ g(\mathbf{z}) \\ &+ \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{w}) \\ &+ \frac{\rho}{2}||\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{w}||_2^2 \end{aligned}$$

$$\begin{aligned} \mathbf{x}^{t+1} &= \arg\min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^t, \mathbf{y}^t) \\ \mathbf{z}^{t+1} &= \arg\min_{\mathbf{z}} L_\rho(\mathbf{x}^{t+1}, \mathbf{z}, \mathbf{y}^t) \\ \mathbf{y}^{t+1} &= \mathbf{y}^t + \rho(\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{z}^{t+1} - \mathbf{w}) \end{aligned}$$

- Distributed BA

$$\text{minimize} \sum_{i=1}^{n} f_i(\mathbf{x}_i)$$

$$\text{subject to } \mathbf{x}_i = \mathbf{z}, i = 1, \ldots, n$$

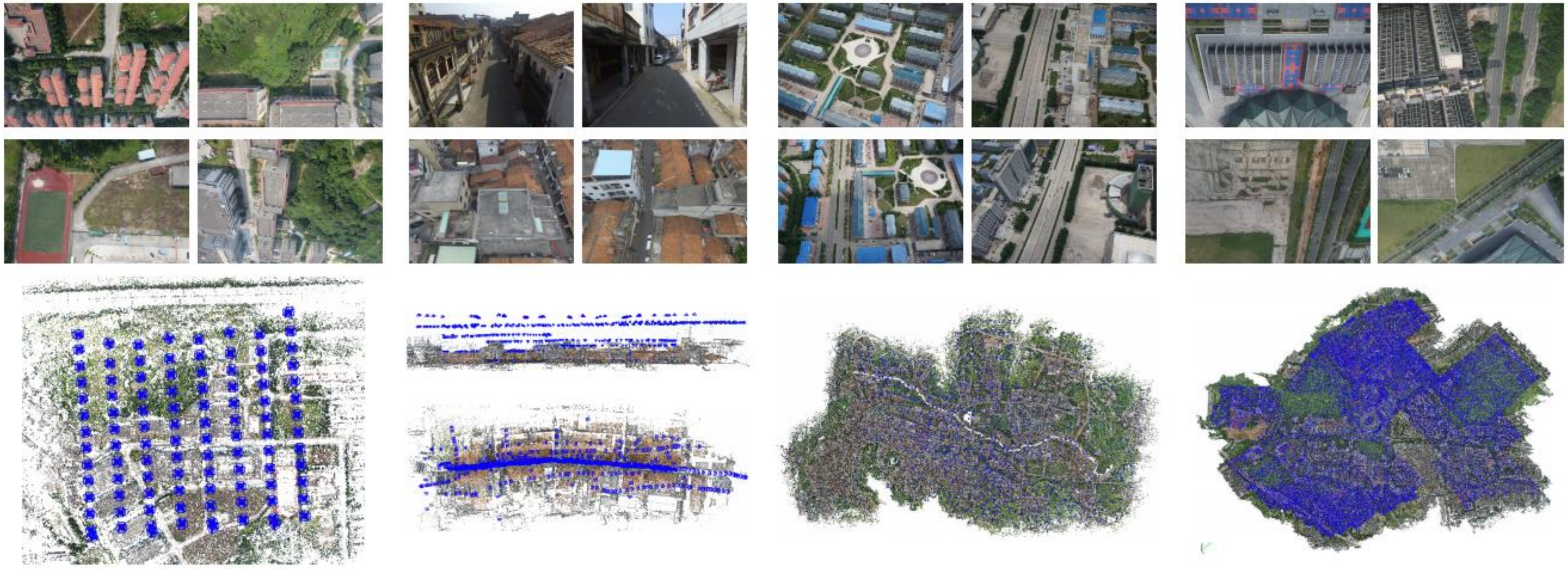
- Applying ADMM

$$\begin{aligned} \mathbf{x}_i^{t+1} = \arg\min_{\mathbf{x}_i} \Big(f_i(\mathbf{x}_i) &+ \left(\mathbf{y}_i^t\right)^T (\mathbf{x}_i - \mathbf{z}^t) \\ &+ \frac{\rho}{2}||\mathbf{x}_i - \mathbf{z}^t||_2^2 \Big) \end{aligned}$$

$$\mathbf{z}^{t+1} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^{t+1}$$

$$\mathbf{y}_i^{t+1} = \mathbf{y}_i^t + \rho(\mathbf{x}_i^{t+1} - \mathbf{z}^{t+1}), i = 1, \ldots, n$$

Large-scale SfM Results



Buildings Street Town City

Outline

- Theories in BA
- BA for large scale SfM
- BA for realtime SLAM

Outline

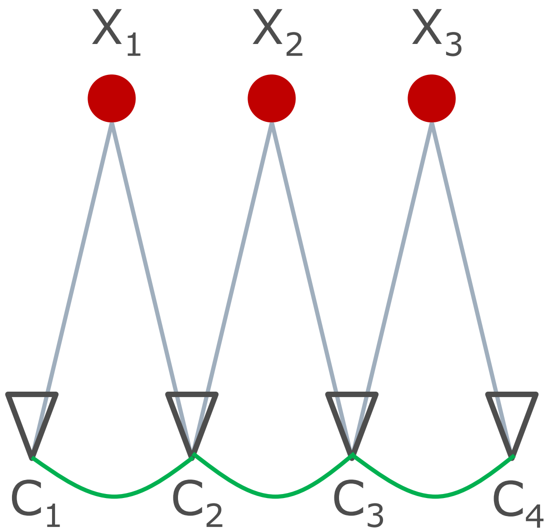
- Theories in BA
- BA for large scale SfM
- BA for realtime SLAM

Significance of BA Efficiency to SLAM

- Higher efficiency of BA means
  - Lower hardware requirement & power consumption
  - Longer sliding window to improve accuracy & robustness
  - Faster map expansion, better robustness

Batch VS Incremental BA

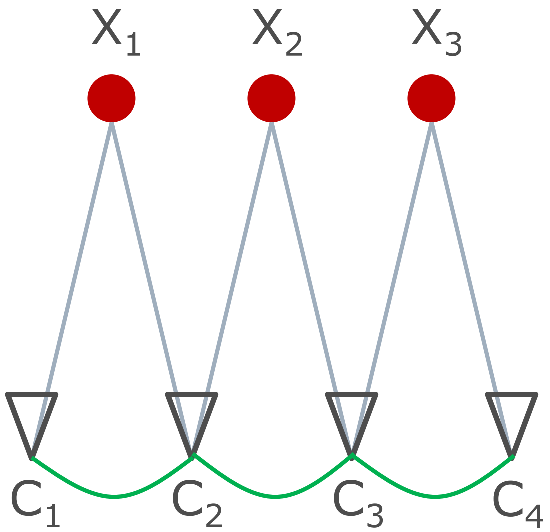
Batch BA



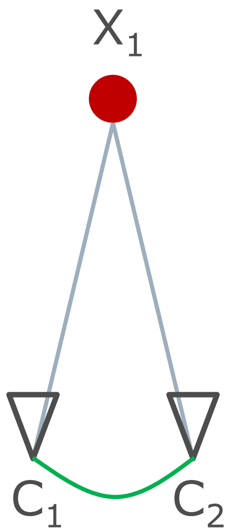
Incremental BA

Batch VS Incremental BA

Batch BA

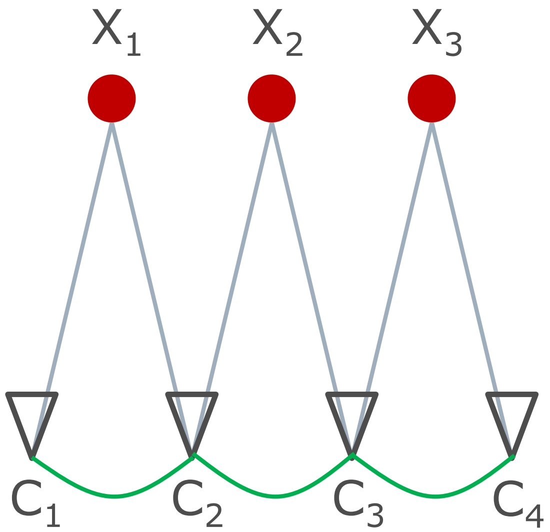


Incremental BA

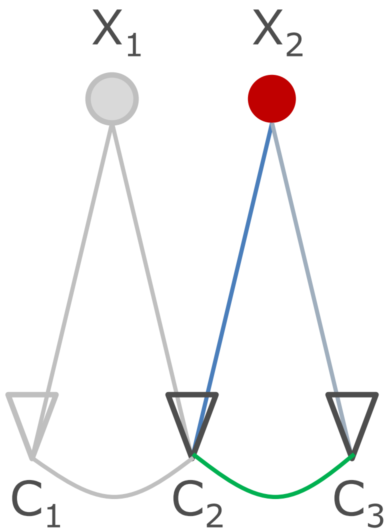


Batch VS Incremental BA

Batch BA

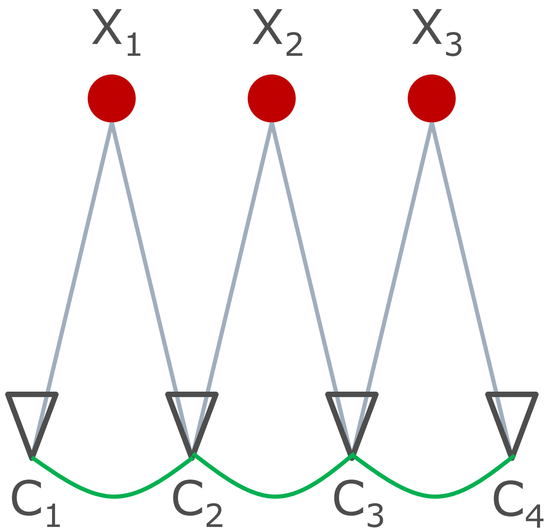


Incremental BA

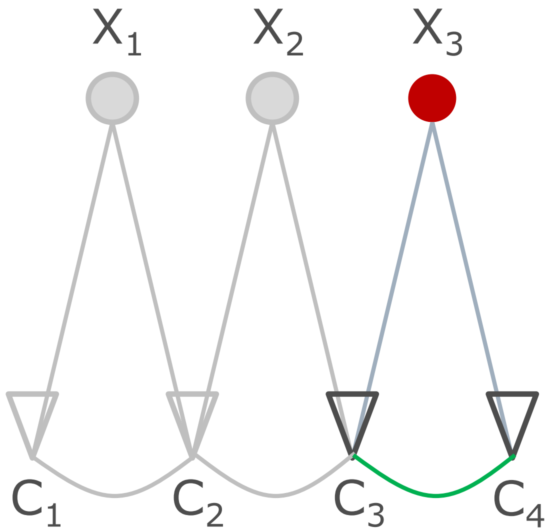


Batch VS Incremental BA

Batch BA



Incremental BA



Representative Methods of Incremental BA

- iSAM/iSAM2
  - Kaess M, Ranganathan A, Dellaert F. iSAM: Incremental smoothing and mapping[J]. IEEE Transactions on Robotics, 2008, 24(6): 1365-1378.
  - Kaess M, Johannsson H, Roberts R, et al. iSAM2: Incremental smoothing and mapping using the Bayes tree[J]. The International Journal of Robotics Research, 2012, 31(2): 216-235.
  - https://bitbucket.org/gtborg/gtsam/
- ICE-BA
  - Liu H, Chen M, Zhang G, et al. Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 1974-1982.
  - https://github.com/baidu/ICE-BA
- SLAM++
  - Ila V, Polok L, Solony M, et al. Fast incremental bundle adjustment with covariance recovery[C]//2017 International Conference on 3D Vision (3DV). IEEE, 2017: 175-184.
  - https://sourceforge.net/p/slam-plus-plus/wiki/Home/

Incremental BA by ICE-BA

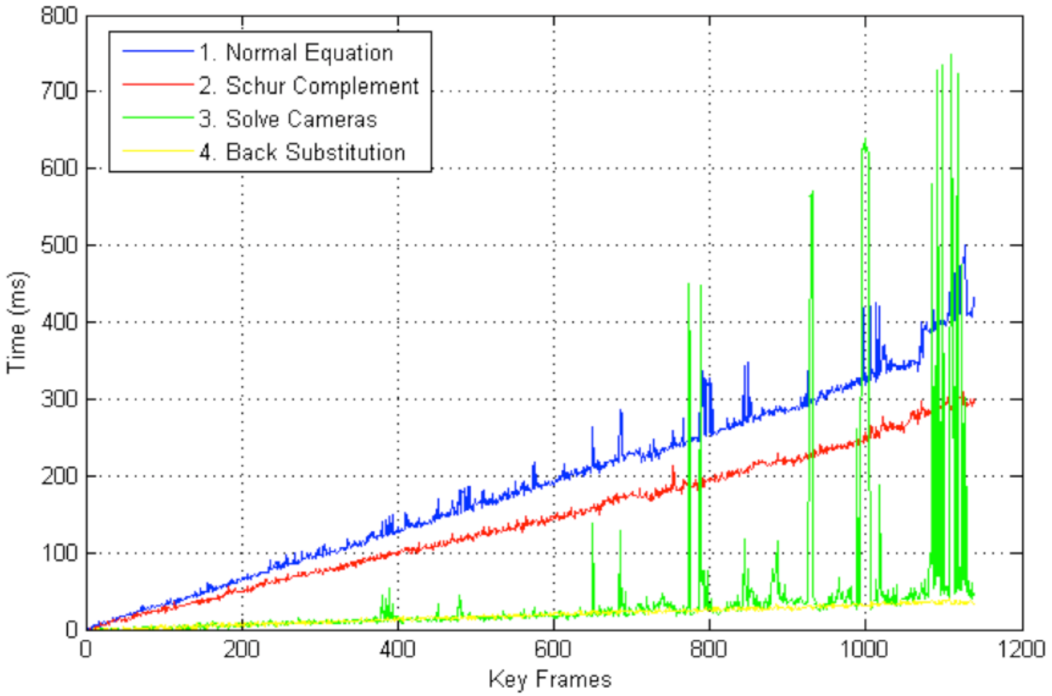
Liu H, Chen M, Zhang G, et al. ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM. CVPR 2018.

Steps of Standard BA

- Steps in one iteration
  1. normal equation
  2. Schur complement
  3. solve cameras
  4. solve points

Observations in Standard BA

- Runtime for steps 1, 2 >> 3, 4
  - #projections >> #cameras

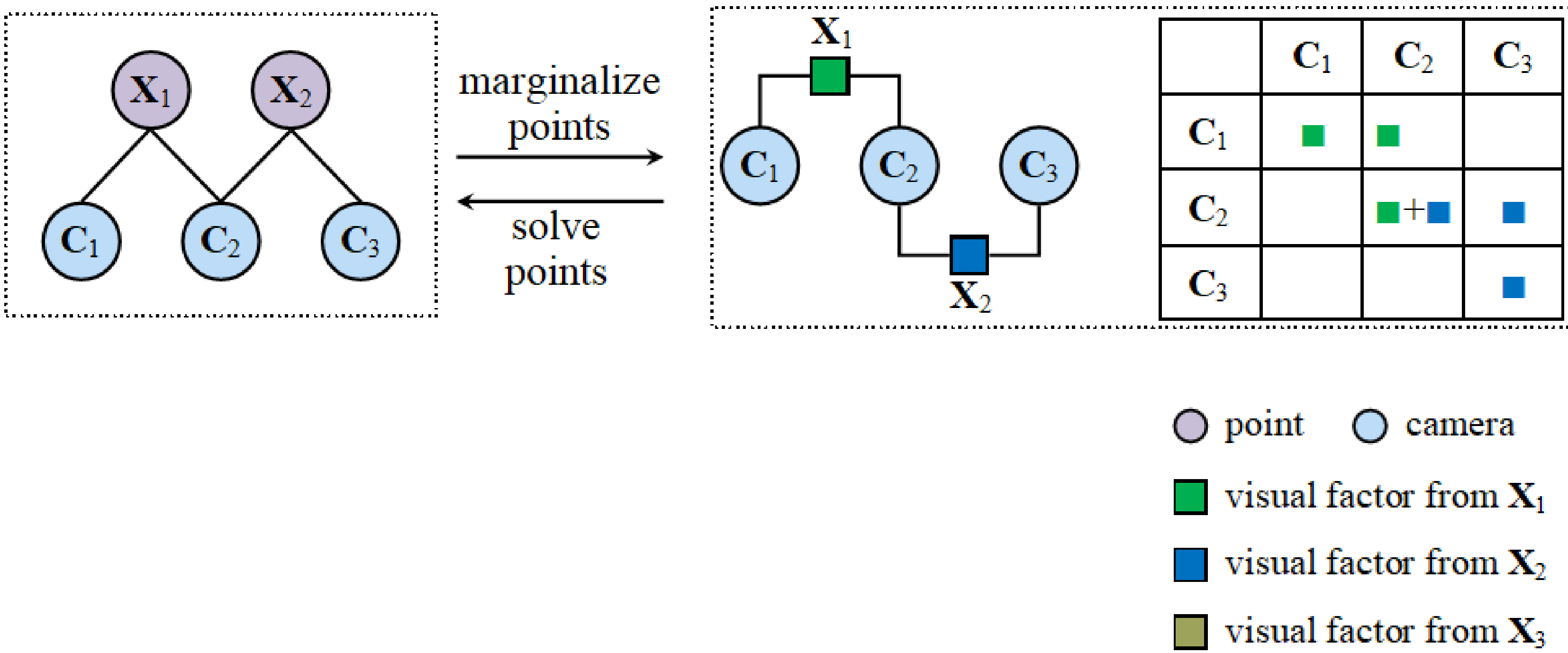


Observations in Standard BA

- Runtime for steps 1, 2 >> 3, 4
- Most cameras and points are nearly unchanged
  - Contribution of most functions nearly unchanged
  - No need to re-compute at each iteration

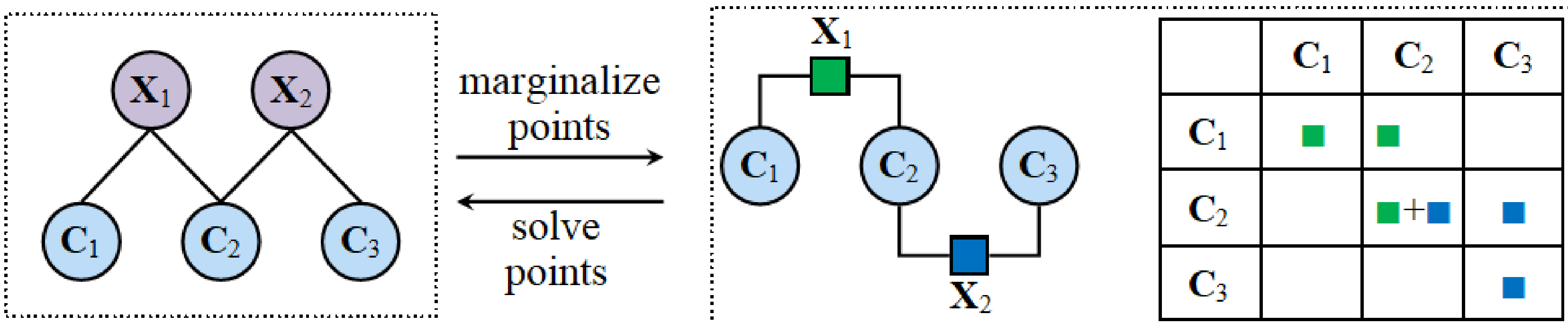
ICE-BA

- Factor graph representation

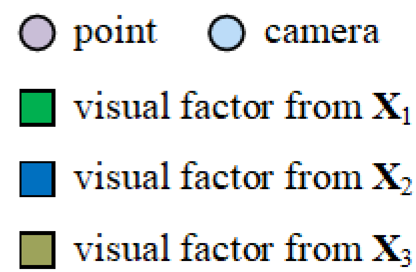


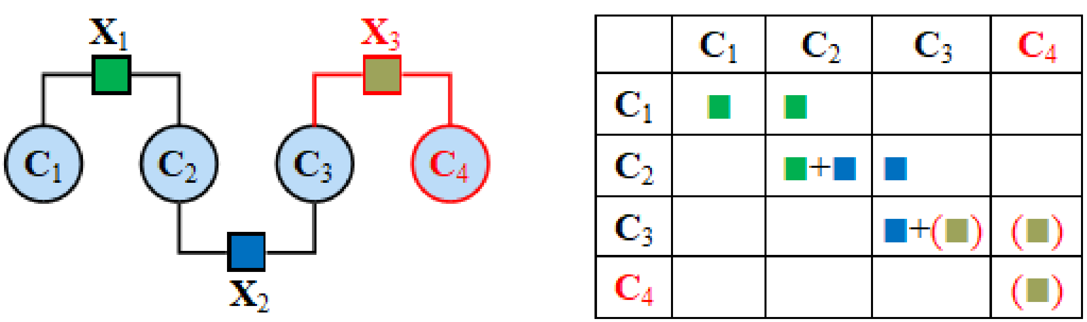
ICE-BA

- Factor graph representation



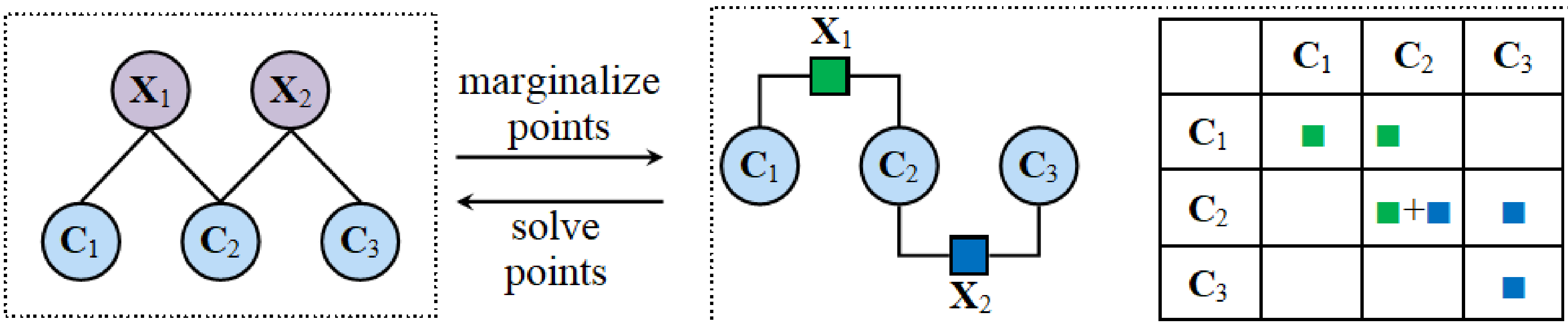
- New cameras or points come



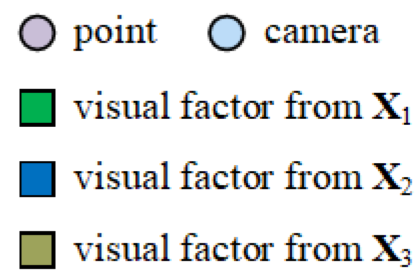


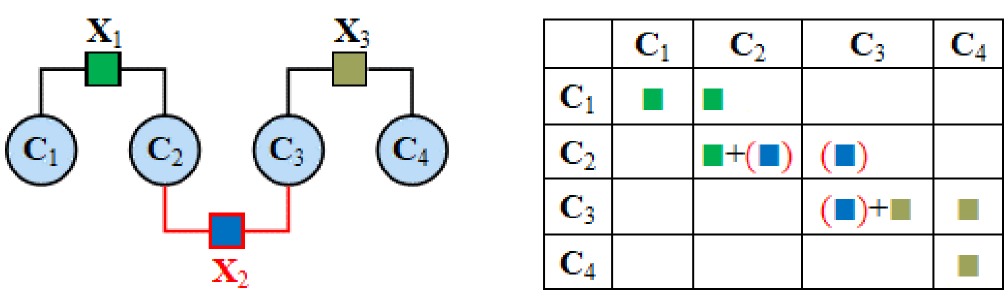
ICE-BA

- Factor graph representation



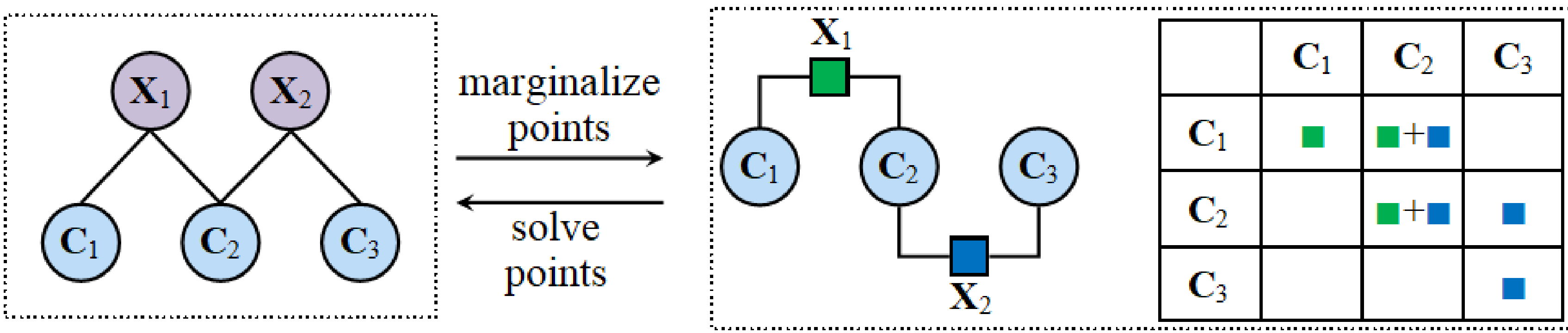
- Points have changed after iteration



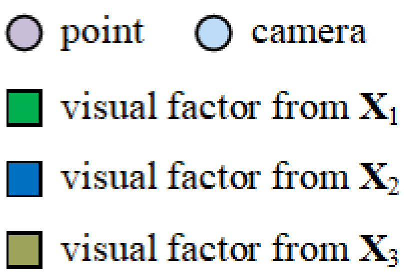


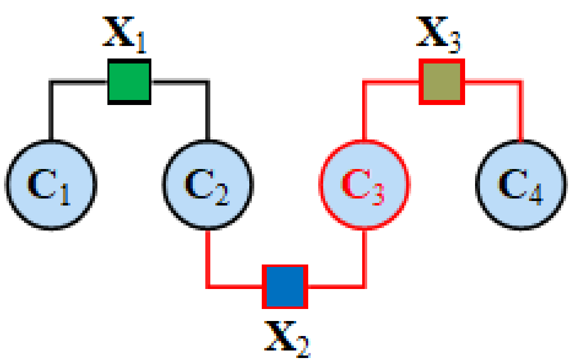
ICE-BA

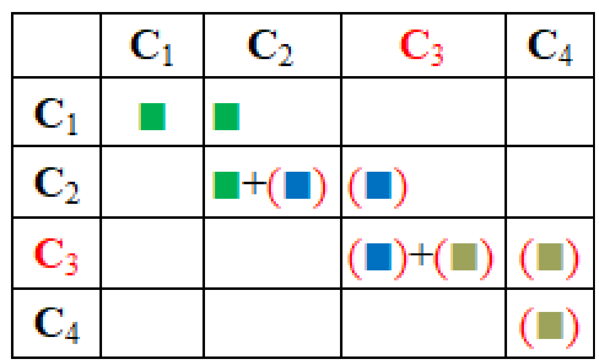
- Factor graph representation



- Cameras have changed after iteration







Step1: Normal Equation

- Batch BA

$\mathbf{U} = \mathbf{0};\ \mathbf{V} = \mathbf{0};\ \mathbf{W} = \mathbf{0};\ \mathbf{u} = \mathbf{0};\ \mathbf{v} = \mathbf{0}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
  Construct linearized equation (11)
  $\mathbf{U}_{ii} += \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{C}_{ij}}$
  $\mathbf{V}_{jj} += \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
  $\mathbf{u}_i += \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{e}_{ij}$
  $\mathbf{v}_j += \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{e}_{ij}$
  $\mathbf{W}_{ij} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
end for

- ICE-BA

for each point j and each camera $i \in \mathcal{V}_j$ that $\mathbf{C}_i$ or $\mathbf{X}_j$ is changed **do**
  Construct linearized equation (11)
  $\mathbf{S}_{ii} -= \mathbf{A}_{ij}^{\mathbf{U}};\ \mathbf{A}_{ij}^{\mathbf{U}} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{C}_{ij}};\ \mathbf{S}_{ii} += \mathbf{A}_{ij}^{\mathbf{U}}$
  $\mathbf{V}_{jj} -= \mathbf{A}_{ij}^{\mathbf{V}};\ \mathbf{A}_{ij}^{\mathbf{V}} = \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}};\ \mathbf{V}_{jj} += \mathbf{A}_{ij}^{\mathbf{V}}$
  $\mathbf{g}_i -= \mathbf{b}_{ij}^{\mathbf{u}};\ \mathbf{b}_{ij}^{\mathbf{u}} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{e}_{ij};\ \mathbf{g}_i += \mathbf{b}_{ij}^{\mathbf{u}}$
  $\mathbf{v}_j -= \mathbf{b}_{ij}^{\mathbf{v}};\ \mathbf{b}_{ij}^{\mathbf{v}} = \mathbf{J}_{\mathbf{X}_{ij}}^\top \mathbf{e}_{ij};\ \mathbf{v}_j += \mathbf{b}_{ij}^{\mathbf{v}}$
  $\mathbf{W}_{ij} = \mathbf{J}_{\mathbf{C}_{ij}}^\top \mathbf{J}_{\mathbf{X}_{ij}}$
  Mark $\mathbf{V}_{jj}$ updated
end for

Step2: Schur Complement

- Batch BA

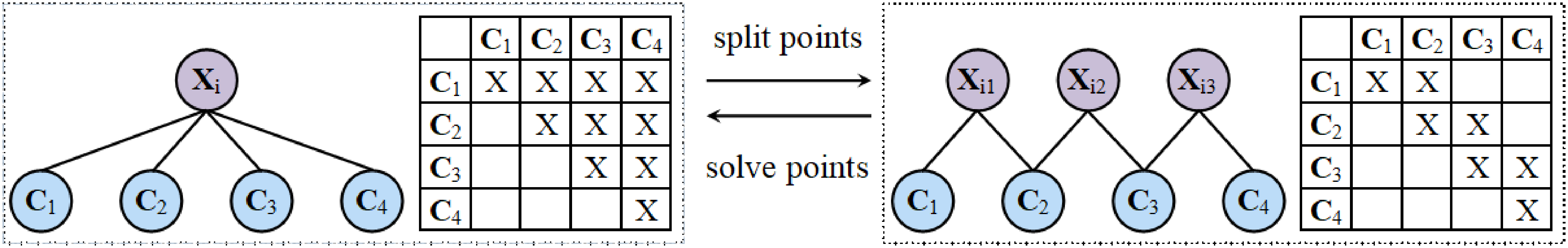
$\mathbf{S} = \mathbf{U}$
for each point j and each camera pair $(i_1, i_2) \in \mathcal{V}_j \times \mathcal{V}_j$ **do**
$\quad \mathbf{S}_{i_1 i_2} -= \mathbf{W}_{i_1 j} \mathbf{V}_{jj}^{-1} \mathbf{W}_{i_2 j}^\top$
end for
$\mathbf{g} = \mathbf{u}$
for each point j and each camera $i \in \mathcal{V}_j$ **do**
$\quad \mathbf{g}_i -= \mathbf{W}_{ij} \mathbf{V}_{jj}^{-1} \mathbf{v}_j$
end for

- ICE-BA

for each point j that $\mathbf{V}_{jj}$ is updated and each camera pair $(i_1, i_2) \in \mathcal{V}_j \times \mathcal{V}_j$ **do**
$\quad \mathbf{S}_{i_1 i_2} += \mathbf{A}^{\mathbf{S}}_{i_1 i_2 j}$
$\quad \mathbf{A}^{\mathbf{S}}_{i_1 i_2 j} = \mathbf{W}_{i_1 j} \mathbf{V}_{jj}^{-1} \mathbf{W}_{i_2 j}^\top$
$\quad \mathbf{S}_{i_1 i_2} -= \mathbf{A}^{\mathbf{S}}_{i_1 i_2 j}$
end for
for each point j that $\mathbf{V}_{jj}$ is updated and each camera $i \in \mathcal{V}_j$ **do**
$\quad \mathbf{g}_i += \mathbf{b}^{\mathbf{g}}_{ij}$; $\mathbf{b}^{\mathbf{g}}_{ij} = \mathbf{W}_{ij} \mathbf{V}_{jj}^{-1} \mathbf{v}_j$; $\mathbf{g}_i -= \mathbf{b}^{\mathbf{g}}_{ij}$
end for

Sub-track Improvement for Local BA

- In LBA, most points may be observed by most frames in the sliding window
  - Dense Schur complement
  - A large portion need to be re-computed
- Split the origin long feature track X_i into several short overlapping sub-tracks $X_{i_1}, X_{i_2}, \cdots$

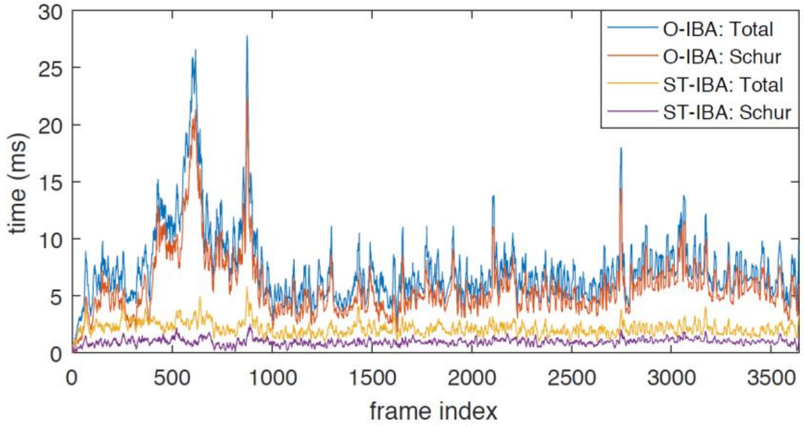


	C1	C2	C3	C4
C1	X	X	X	X
C2		X	X	X
C3			X	X
C4				X

	C1	C2	C3	C4
C1	X	X		
C2		X	X	
C3			X	X
C4				X

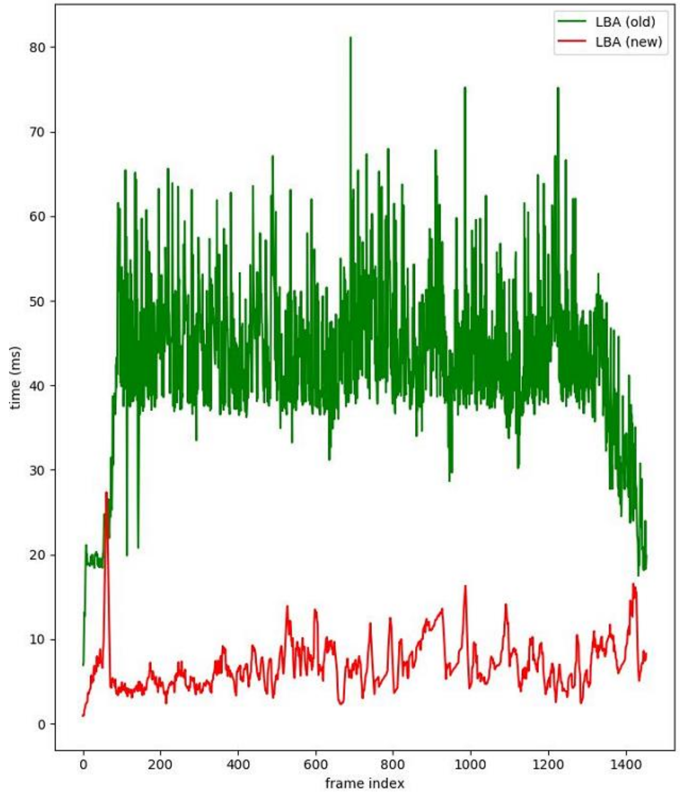
Sub-track Improvement for Local BA

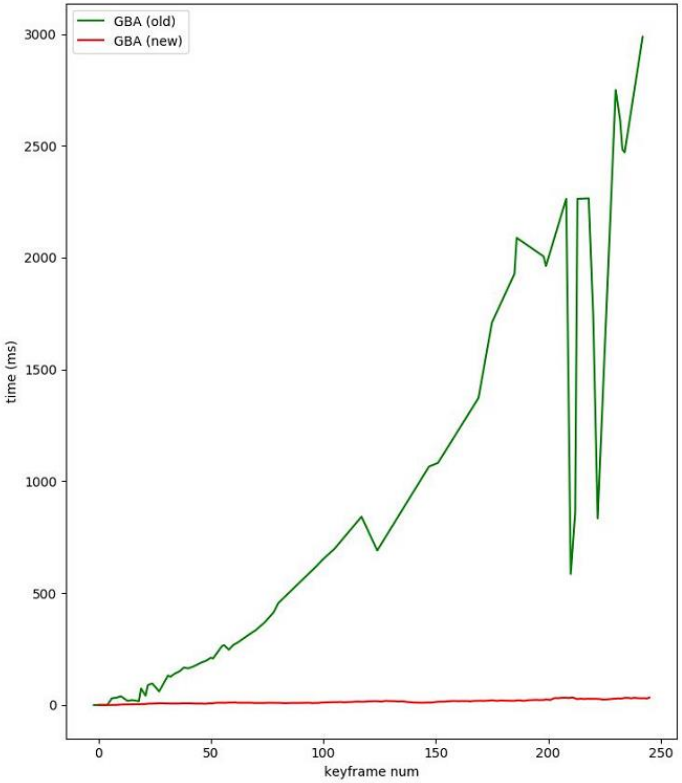
- In LBA, most points may be observed by most frames in the sliding window
  - Dense Schur complement
  - A large portion need to be re-computed
- Split the origin long feature track X_i into several short overlapping sub-tracks $X_{i_1}, X_{i_2}, \cdots$



Efficiency Comparison

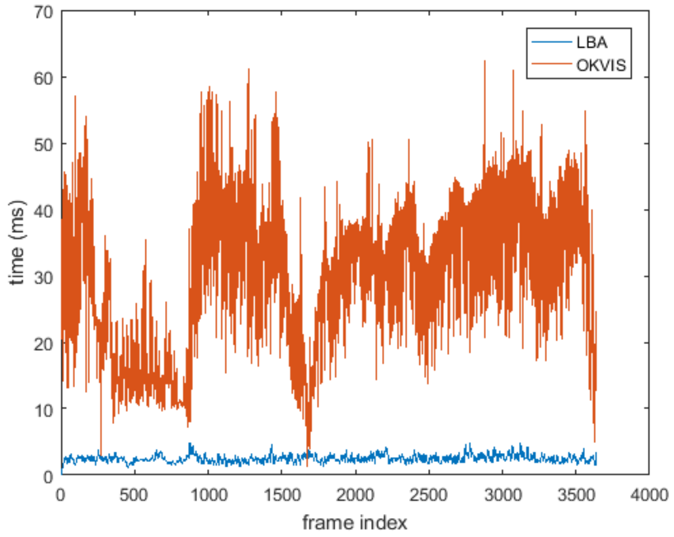
- Local BA (LBA)
  - ICE-BA (50 frames)
  - Ceres (10 frames)
- Global BA (GBA)
  - ICE-BA: $O(1)$
  - Ceres: $O(n^2)$

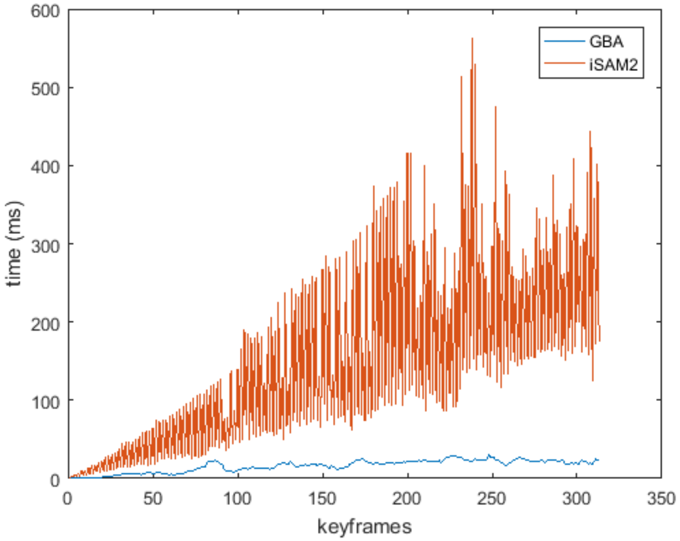




Efficiency Comparison

- Local BA (LBA)
  - ICE-BA (50 frames)
  - OKVIS (8 frames)
- Global BA (GBA)
  - ICE-BA: steady and smooth
  - iSAM2: steep and peaks





Accuracy Comparison

Seq.	Ours w/ loop	Ours w/o loop	OKVIS	SVO	iSAM2
MH_01	0.11	0.09	0.22	**0.06**	0.07
MH_02	0.08	**0.07**	0.16	0.08	0.11
MH_03	**0.05**	0.11	0.12	0.16	0.12
MH_04	**0.13**	0.16	0.18	-	0.16
MH_05	**0.11**	0.27	0.29	0.63	0.25
V1_01	0.07	0.05	**0.03**	0.06	0.07
V1_02	0.08	**0.05**	0.06	0.12	0.08
V1_03	**0.06**	0.11	0.12	0.21	0.12
V2_01	0.06	0.12	**0.05**	0.22	0.10
V2_02	**0.04**	0.09	0.07	0.16	0.13
V2_03	**0.11**	0.17	0.14	-	0.20
Avg	**0.08**	0.12	0.14	0.20	0.13

Open-source Solver & BA

- Bundler: http://www.cs.cornell.edu/~snavely/bundler
- g2o: https://github.com/RainerKuemmerle/g2o
- Ceres Solver: http://ceres-solver.org
- SegmentBA: https://github.com/zju3dv/SegmentBA
- iSAM2: https://bitbucket.org/gtborg/gtsam
- ICE-BA: https://github.com/baidu/ICE-BA
- SLAM++: https://sourceforge.net/p/slam-plus-plus/wiki/Home/

代表性SfM方法

- 增量式SfM
  - 采用逐张图片加入处理的方式
  - 精度高，求解鲁棒，但速度较慢
  - 代表性工作
    - Bundler: http://grail.cs.washington.edu/rome/
    - VisualSFM: http://ccwu.me/vsfm/
    - COLMAP: http://demuc.de/colmap/
    - ACTS: http://www.zjucvg.net/acts/acts.html
- 层次式SfM
  - 先求解局部地图，再进行融合和补充得到完整的重建
  - 显著提高重建的效率
  - 代表性工作
    - ENFT-SFM： https://github.com/zju3dv/ENFT-SfM

代表性SfM方法

- 全局式SfM
  - 直接求解全局的图像外参，然后通过少量的集束调整完成优化
  - 高效，但是容易受到错误匹配的影响
  - 代表性工作
    - OpenMVG: https://github.com/openMVG/openMV
- 混合式SfM
  - 采用全局方法估计图像旋转，然后增量式的求解图像位置
  - 大幅度减少重建时间，错误的匹配关系可以及时修正
  - 代表性工作
    - HSfM: Hybrid Structure-from-Motion(Cui et al.,2017)
- 语义SfM
  - 通过语义信息进行联合优化
  - 三维地图包含语义信息
  - 代表工作
    - Semantic structure from motion (Bao et al.,2011)