多视图几何与运动恢复结构

章国锋
浙江大学CAD&CG国家重点实验室

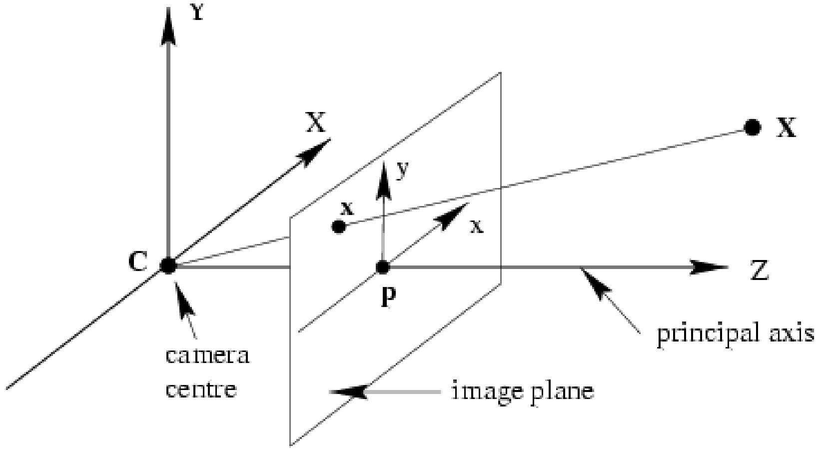
视频场景重建的流程

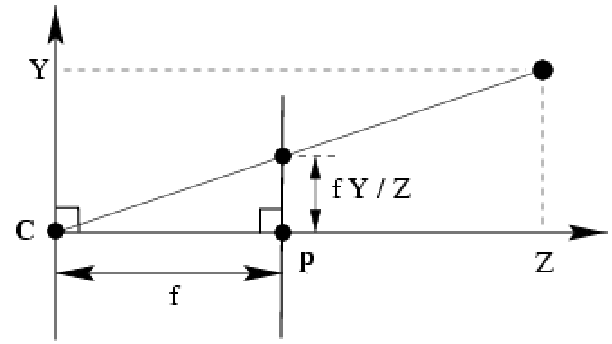
运动恢复结构

深度恢复

三维重建

针孔相机模型





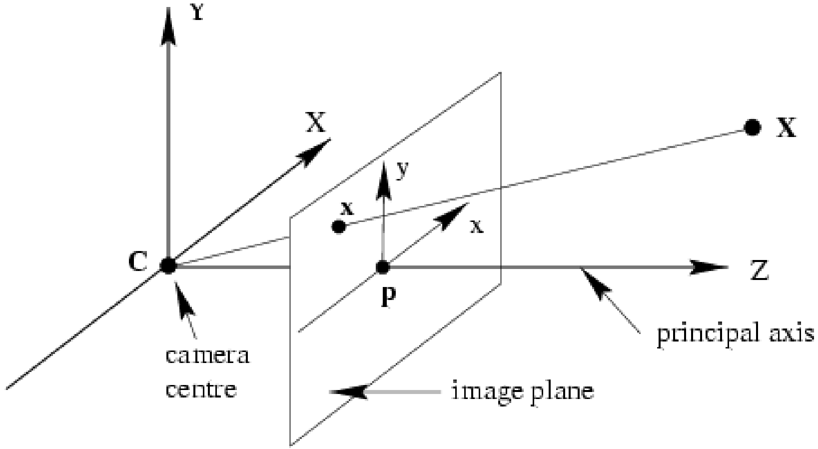
投影方程：
$$x = f\frac{X}{Z}$$
$$y = f\frac{Y}{Z}$$

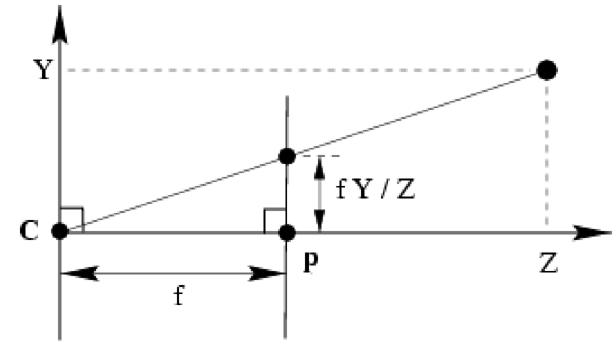
齐次坐标表示：

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

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

针孔相机模型



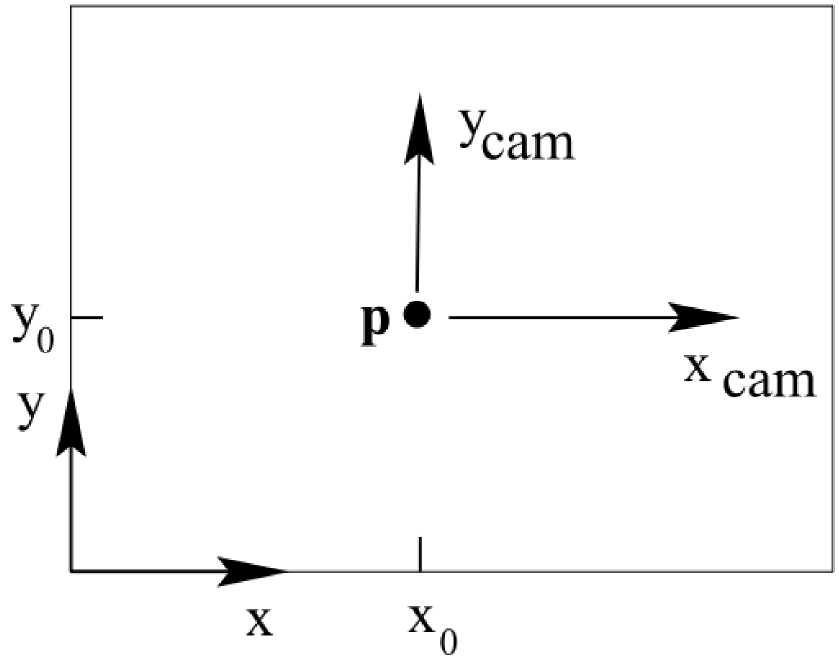


$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

K [R|t]

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

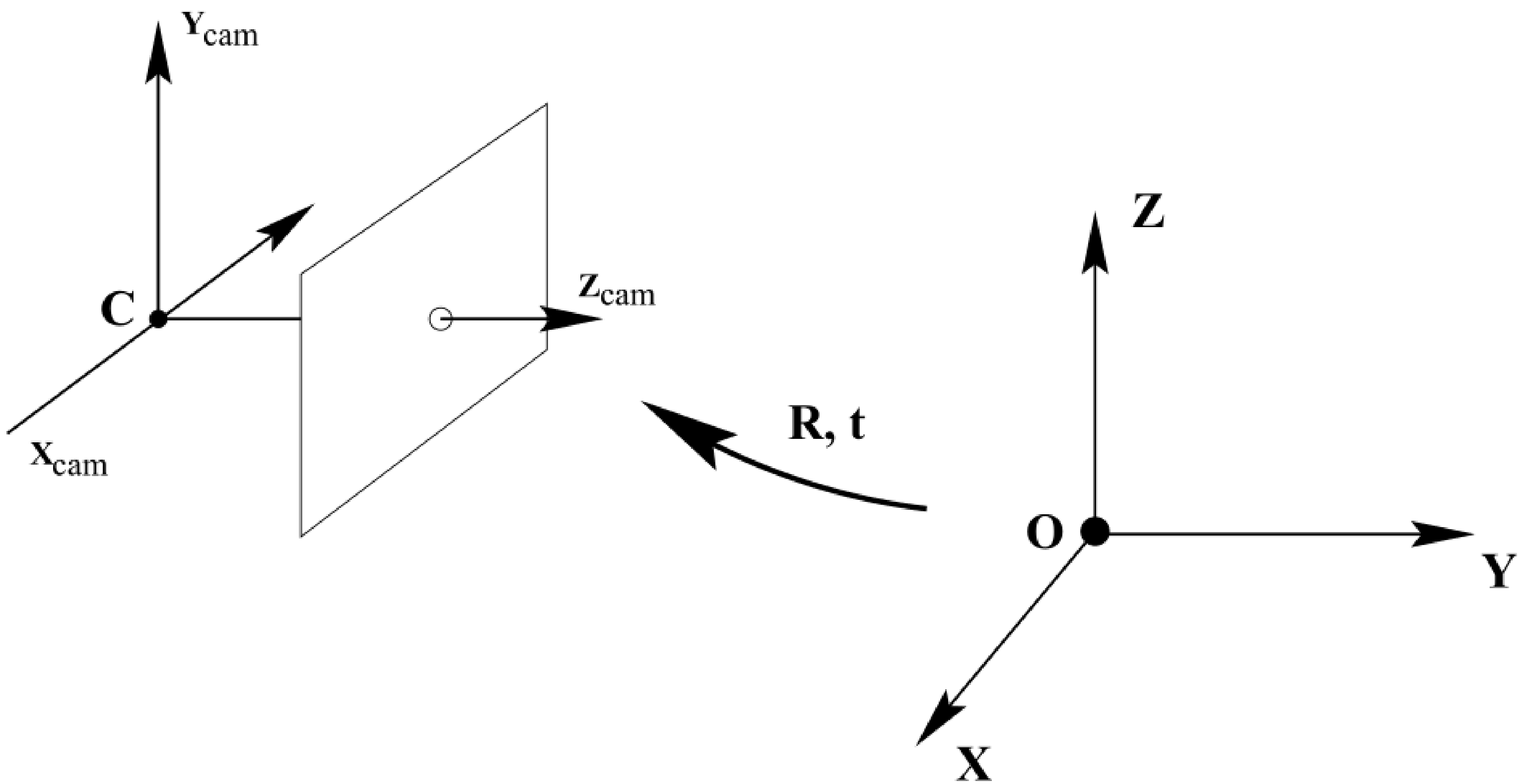
主点的偏移



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

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

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

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

11 DoF (5+3+3)

径向畸变

- 比如鱼眼镜头:

- 数学模型:

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

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

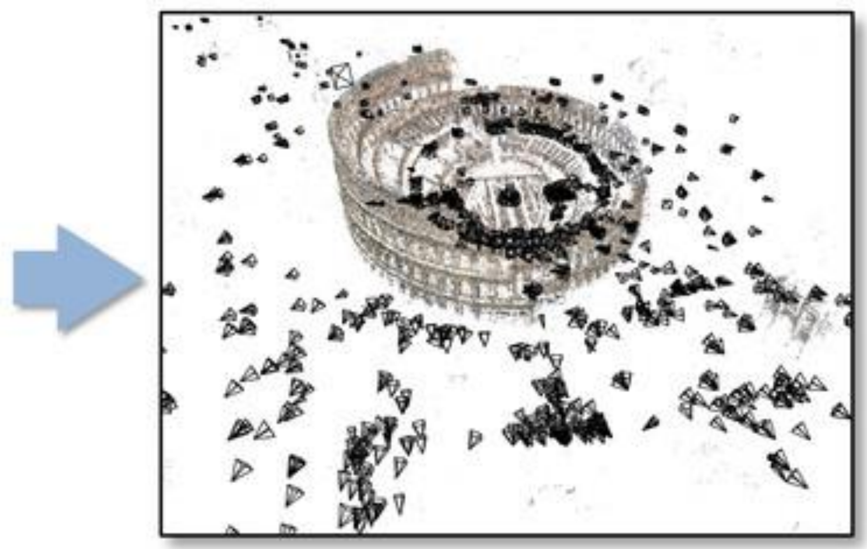
(Marc Pollefeys)

径向畸变矫正例子

(Marc Pollefeys)

Multi-View Geometry

- Structure-from-Motion
  - Automatically recover the camera parameters and 3D structure from multiple images or video sequences.



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

Two-View Geometry

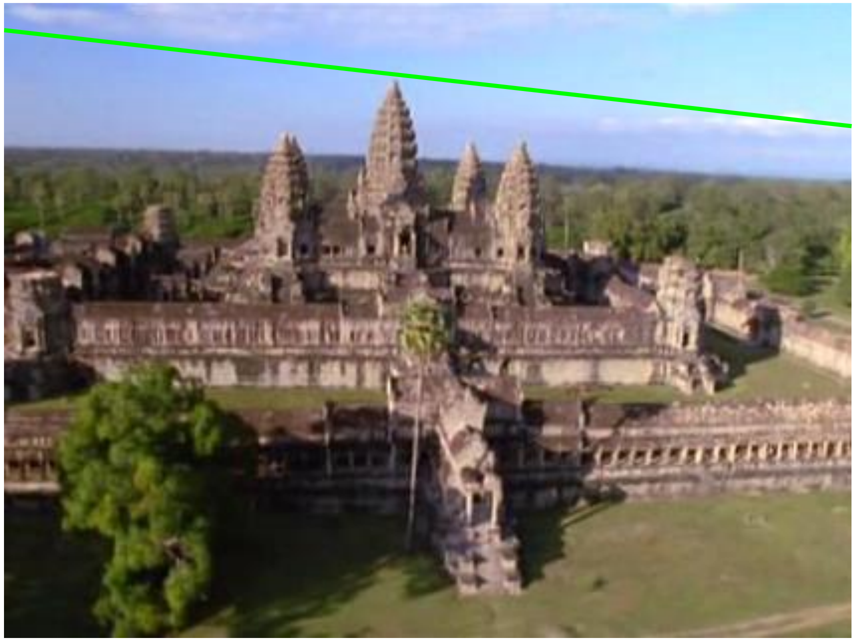
3D???

Two-View Geometry

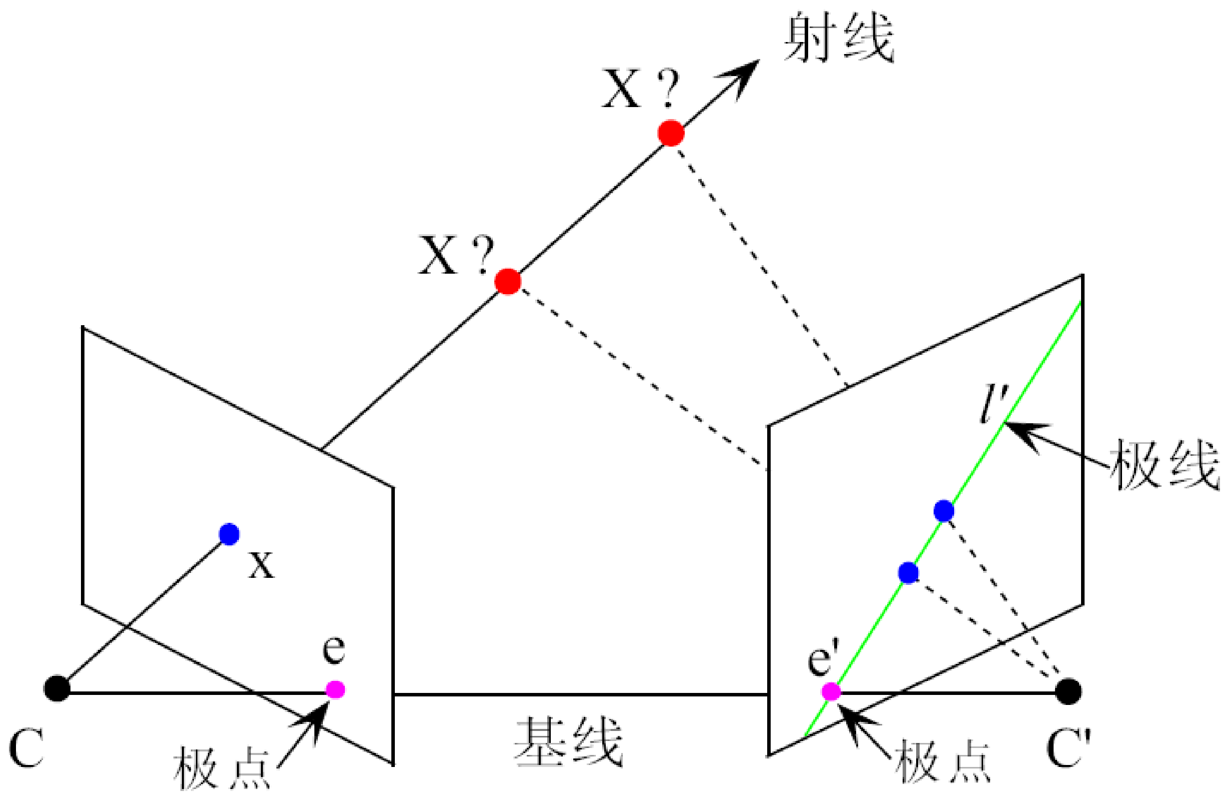
3D???

Two-View Geometry

3D: Epipolar Geometry



极线几何



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

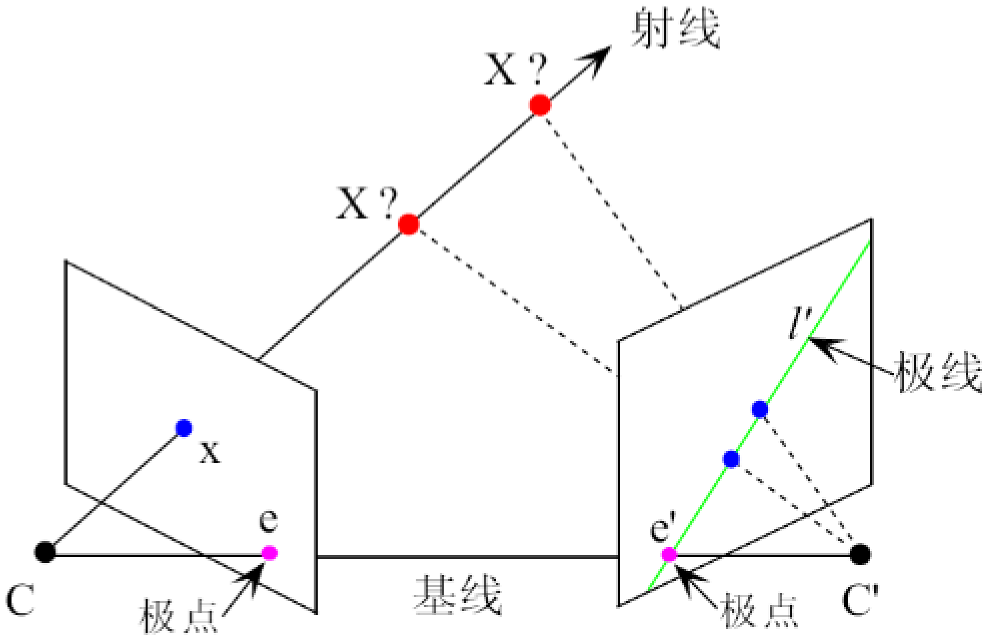
基础矩阵

- 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：

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

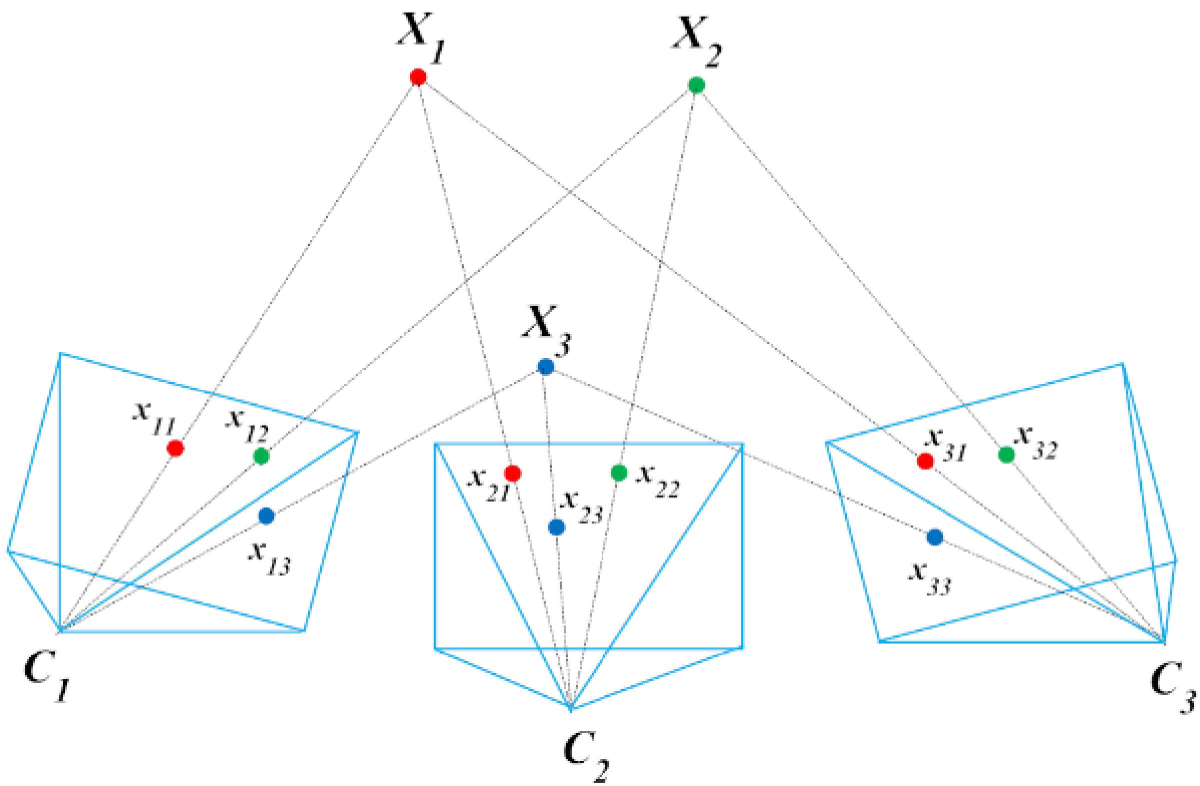
八点法求解基础矩阵

- f 为 9 维向量，若要有解，rank(A) 至多为 8
  - 在 rank(A) = 8 时，f 的方向是唯一的
  - 通过至少 8 对对应点，可恰好得到使 f 方向唯一的 A

- f 为 A 的右零空间的基向量，可用 svd(A) 求得

- 真实数据存在噪音，大于 8 组对应点得到的 A 满秩即 rank(A) = 9
  - 此时同样可计算 (U,Σ,V) = svd(A)
    令 f 为 V 中对应最小奇异值的列向量

多视图几何



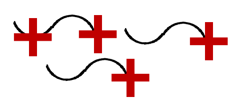
$$\mathbf{x}_{ij} = \pi(\mathbf{P}_i X_j)$$

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

Structure from Motion

- Pipeline
  - Feature Tracking
    - Obtain a set of feature tracks

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



  - Structure from Motion
    - Solve the camera parameters and 3D points of tracks

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

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

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
  - 平滑区域：梯度不明显
  - 边缘区域：梯度明显，方向一致
  - 角落区域：梯度明显，方向不一致
- 方法：
  - 计算像素邻域的梯度二阶矩

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

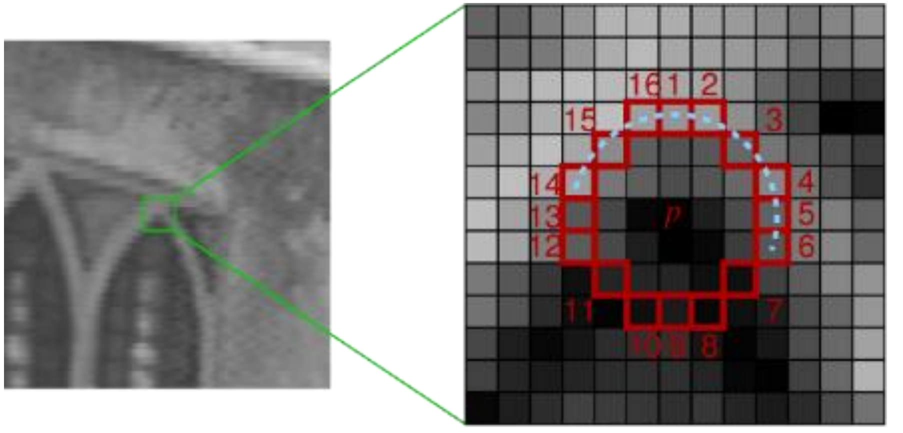
  - 计算上述矩阵的角点响应指标

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

  - 对R进行阈值过滤和非极大值抑制

FAST

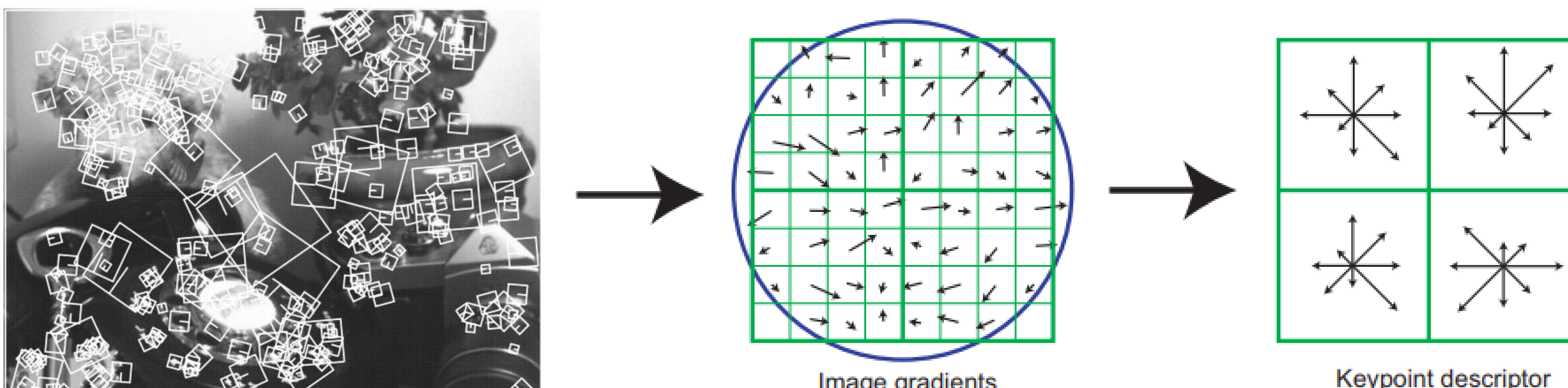
- 通过直接的阈值和判断来加速角点提取
- 考虑中心点周围的16个像素，设中心点亮度为 p
  - 如果有连续 n 个像素亮度都大于 p+t，或者都小于 p-t (如图中的 14~16, 1 ~ 6)
  - 检查 1、5、9、13 四个位置，如果是角点，四个位置中应当有三个满足上面的条件
  - ……
- 速度快，但对噪音不鲁棒



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上寻找极大/极小值来确定特征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。



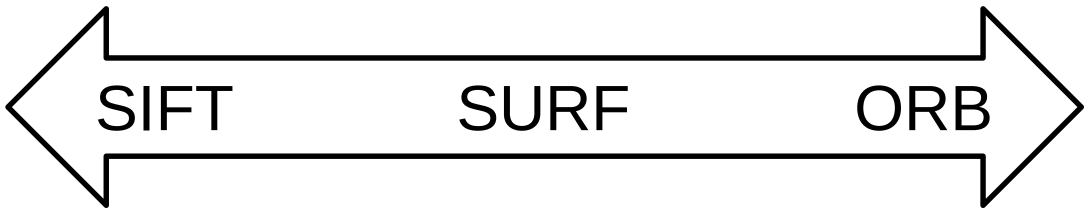
David G. Lowe.Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2): 91-110 (2004).

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
  - 其中SURF性能上接近SIFT
- 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
  - 由于其良好的匹配性能和极快的提取速度也得到了广泛使用。

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

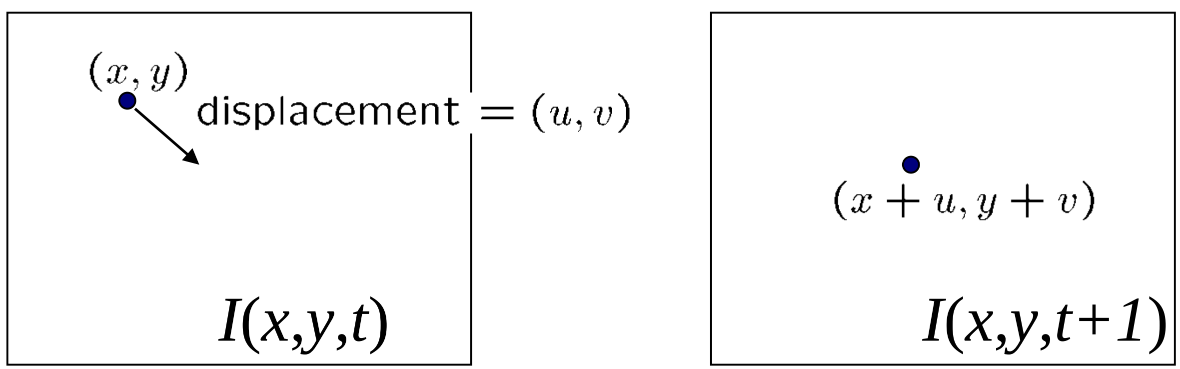
以上三种特征提取算法均在OpenCV中有实现

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：



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

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

$$\nabla I \cdot \begin{pmatrix} u \\ v \end{pmatrix} + I_t = 0 \qquad \text{(单个像素)}$$

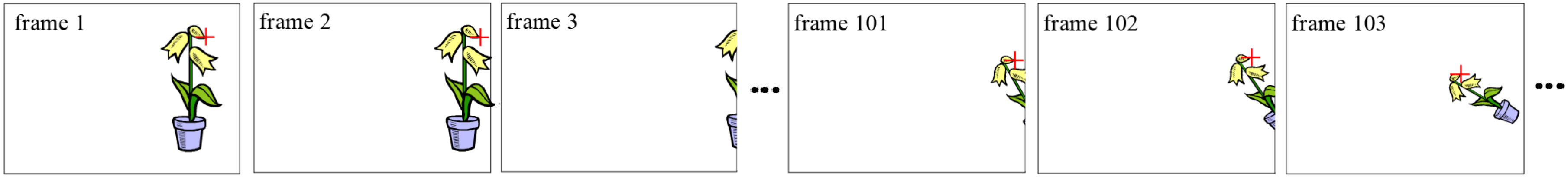
$$\begin{pmatrix} I_x(p_1) & I_y(p_1) \\ \vdots & \vdots \\ I_x(p_n) & I_y(p_n) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} I_t(p_1) \\ \vdots \\ I_t(p_n) \end{pmatrix} = 0 \qquad \text{(邻域窗口)}$$

特征匹配

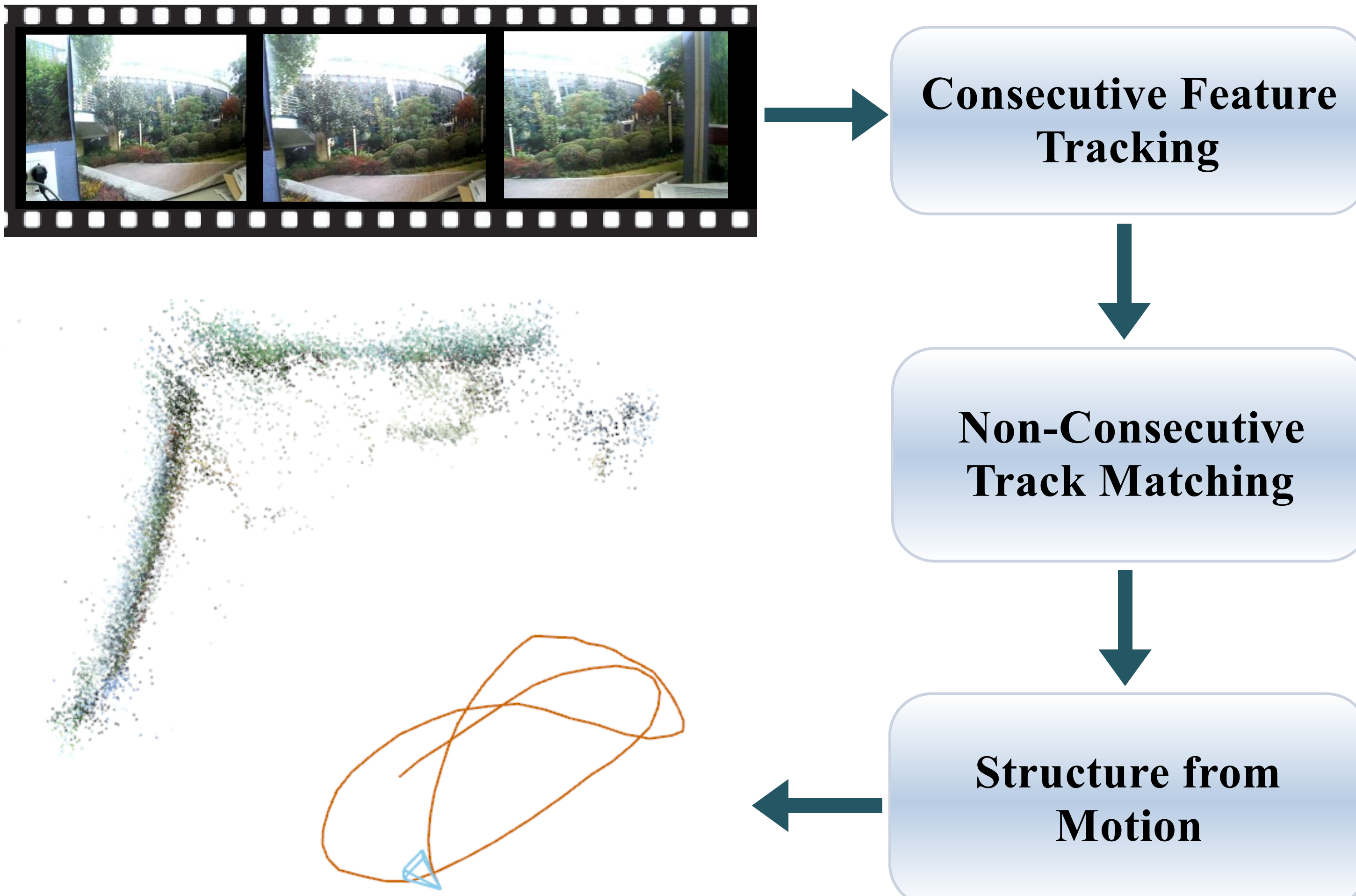
- 大运动情况下的匹配
  - 通过比较特征描述量的距离进行匹配
  - SIFT = 128 维、SURF = 64 维、ORB = 256bits
    - 暴力匹配
    - 快速最近邻匹配
  - OpenCV中提供了相应的匹配算法

Loopback Sequences and Multiple Sequences

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



Non-Consecutive Feature Tracking



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.

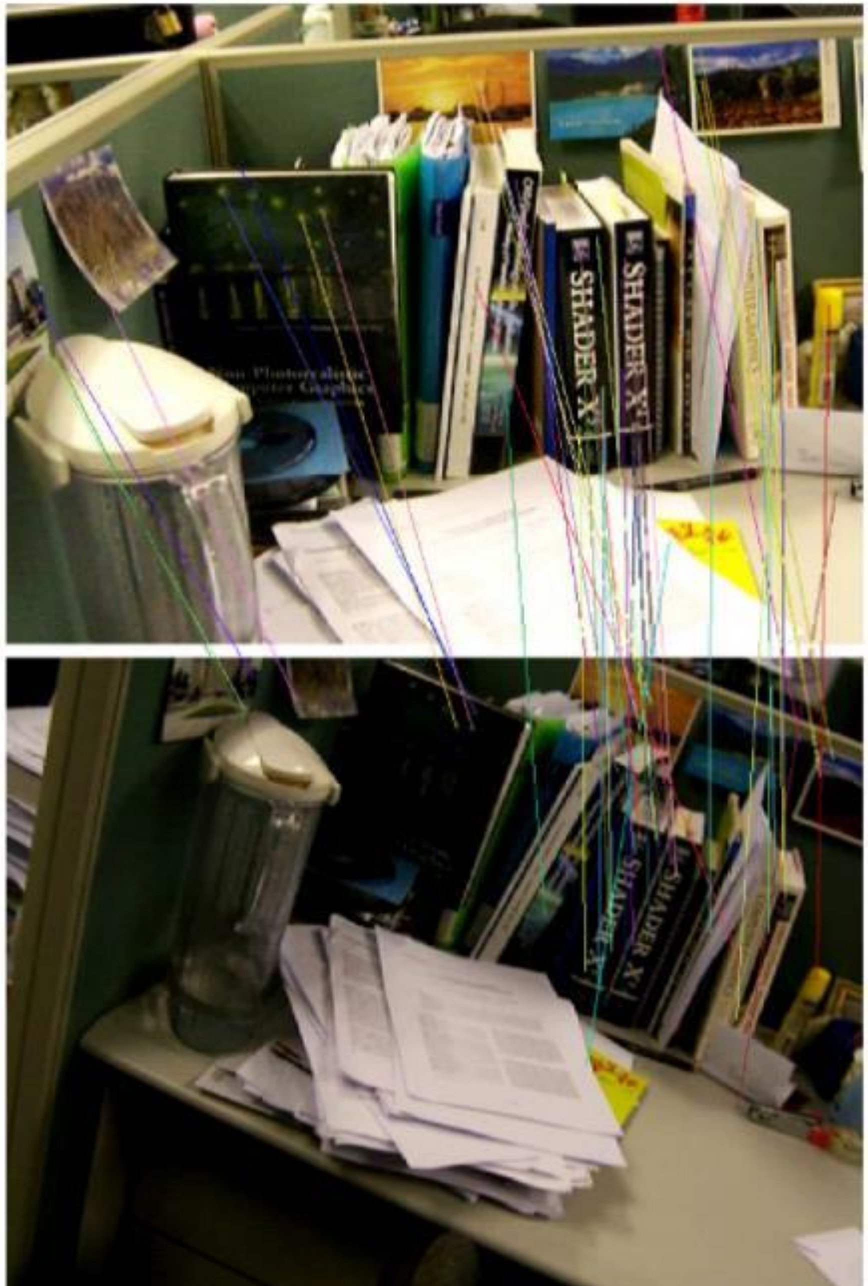
Two-Pass Matching for Consecutive Tracking

- SIFT Feature Extraction

- First-Pass Matching by Descriptor 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

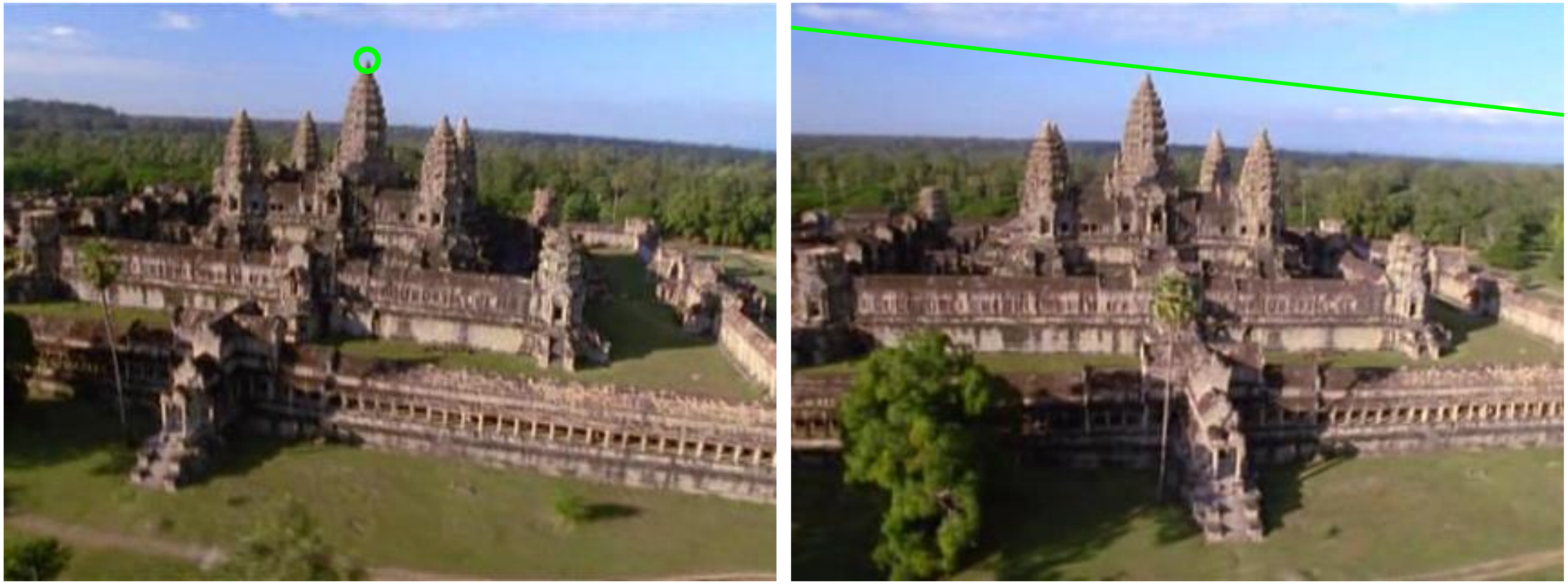

SHADER X
SHADER X

Two-View Geometry

3D???

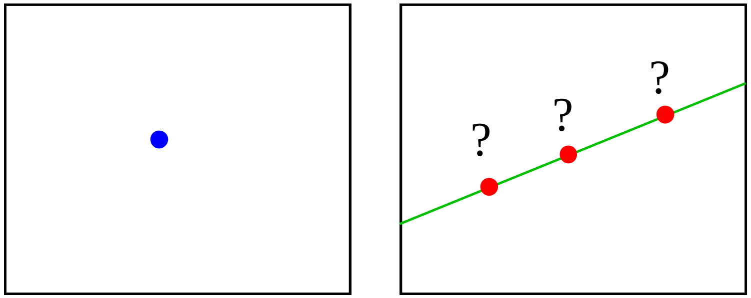
Two-View Geometry

3D: Epipolar Geometry



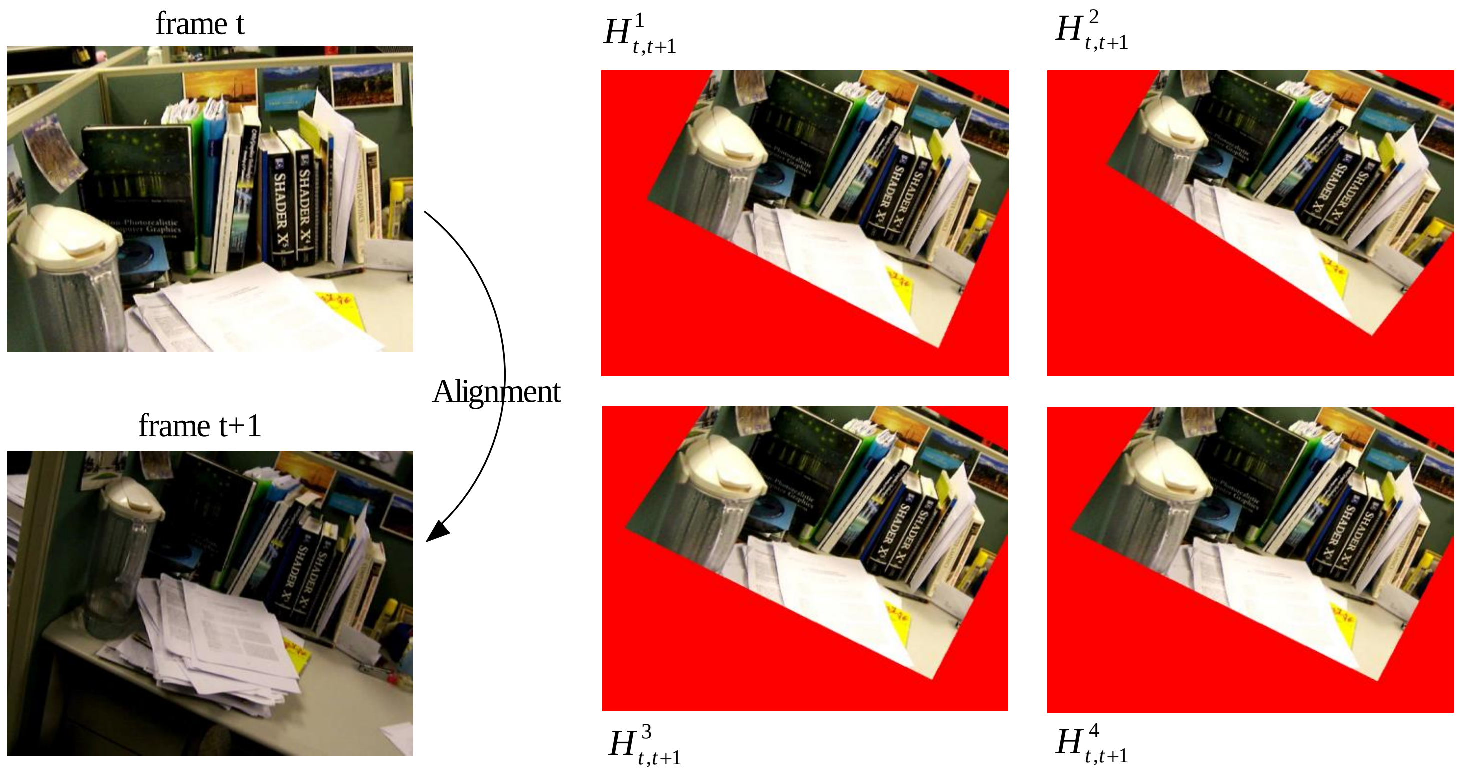
Not enough!

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

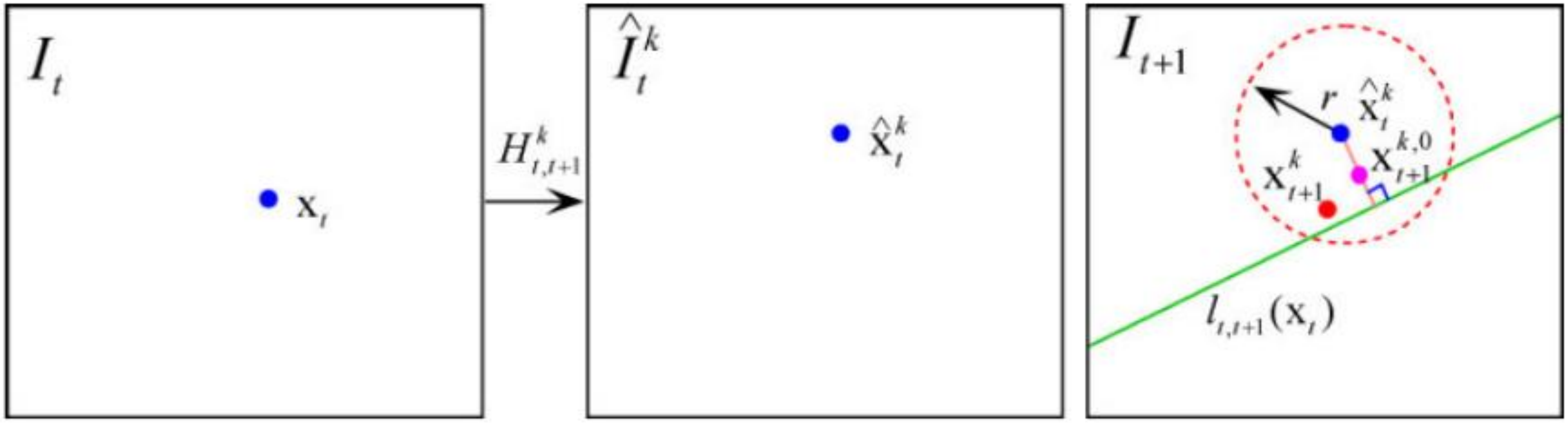


Second-Pass Matching by Planar Motion Segmentation

- Guided matching

$$
\begin{aligned}
S^k_{t,t+1}(\mathbf{x}^k_{t+1}) = \sum_{\mathbf{y}\in W} ||\hat{I}^k_t(\hat{\mathbf{x}}^k_t + \mathbf{y}) - I_{t+1}(\mathbf{x}^k_{t+1} + \mathbf{y})||^2 &+ \\
\lambda_e d(\mathbf{x}^k_{t+1}, l_{t,t+1}(\mathbf{x}_t))^2 + \lambda_h ||\hat{\mathbf{x}}^k_t - \mathbf{x}^k_{t+1}||^2 &
\end{aligned}
$$

Epipolar constraint Homography constraint



Second-Pass Matching with Multi-Homographies

First-Pass Matching
(53 matches)

Direct Searching
(11 matches added)

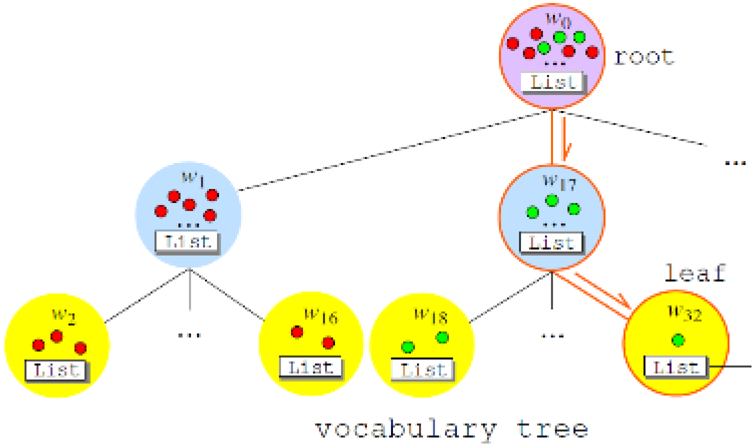
Our Second-Pass Matching
(346 matches added)

Non-Consecutive track matching

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

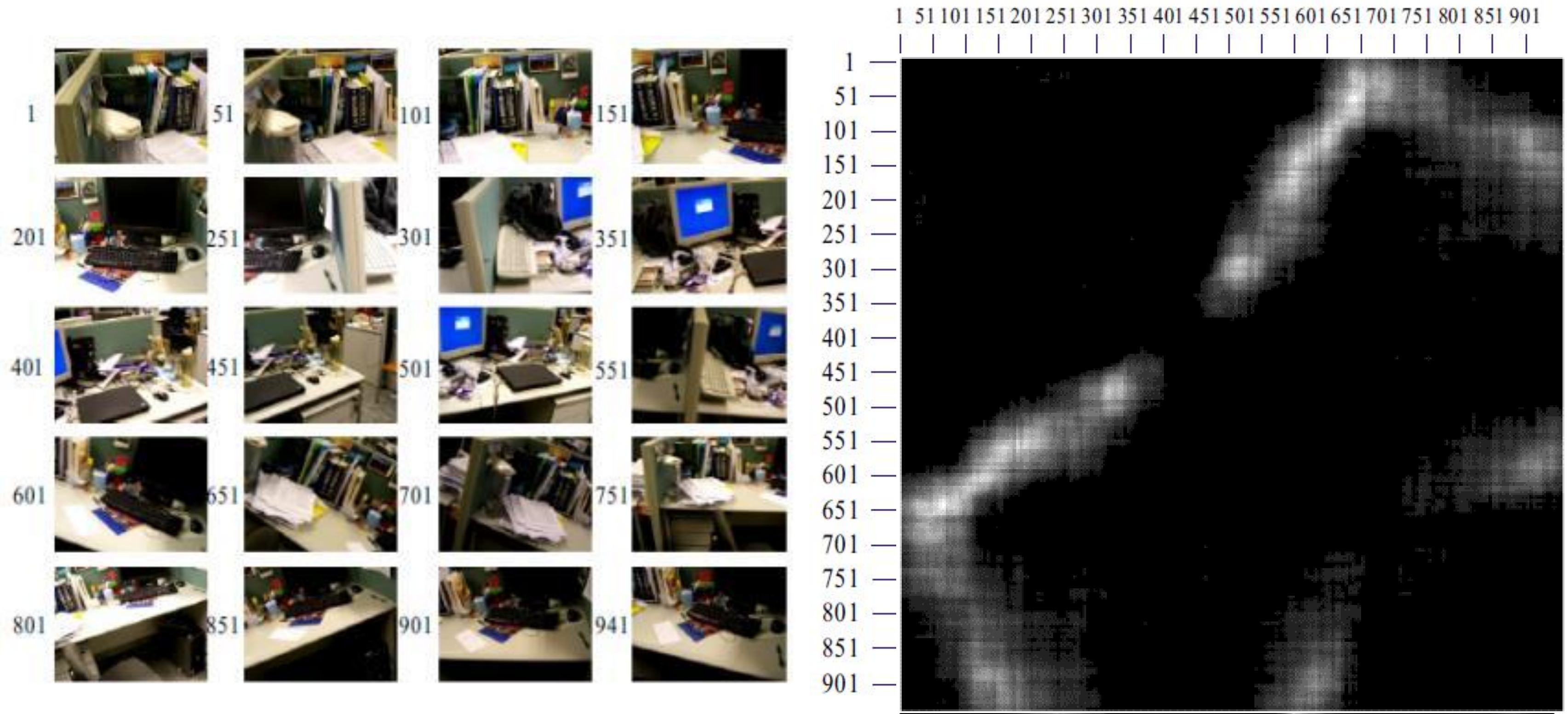
Fast Matching Matrix Estimation

- Each track has a group of description vectors $\mathcal{P}_{\mathcal{X}} = \{\mathbf{p}(\mathbf{x}_t) | t \in f(\mathcal{X})\}$
- Track descriptor $\mathbf{p}(\mathcal{X})$



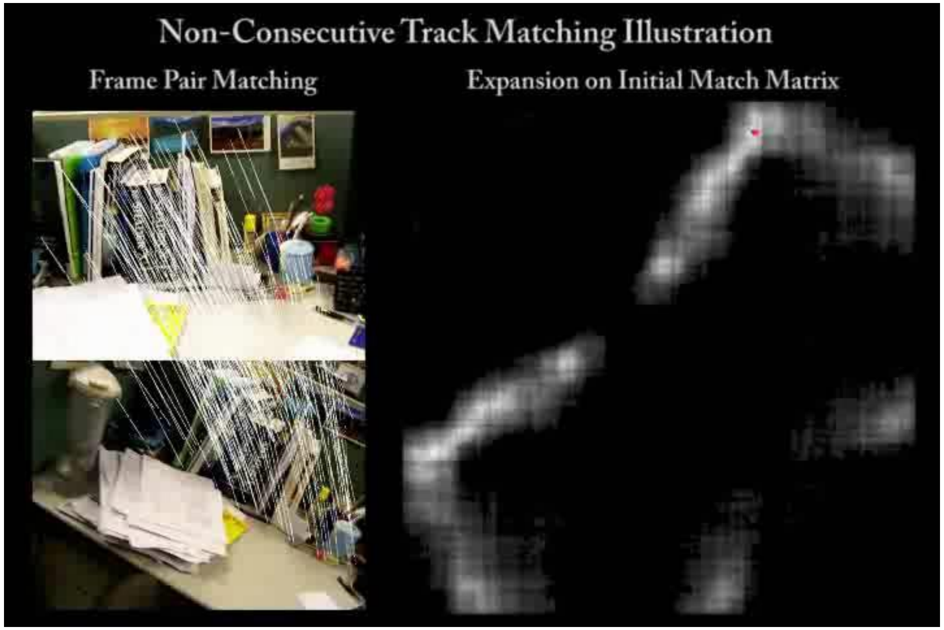
- Use a hierarchical K-means approach to cluster the track descriptors

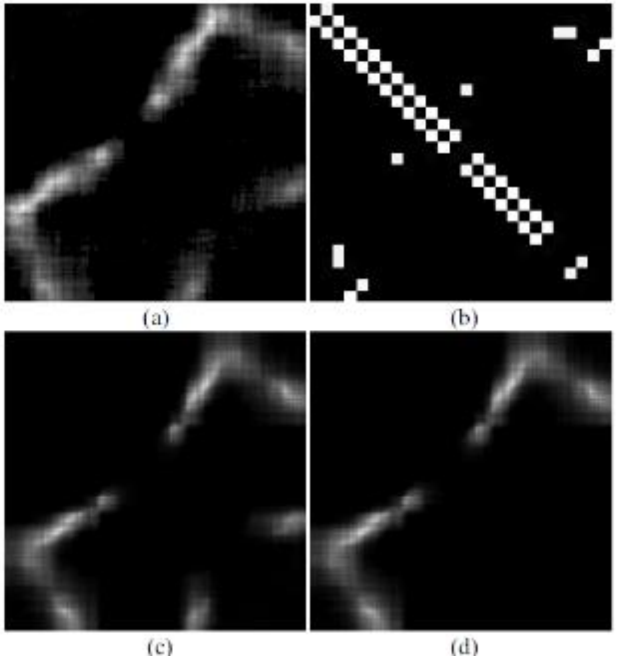
Fast Matching Matrix Estimation



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.





Traditional SfM Framework

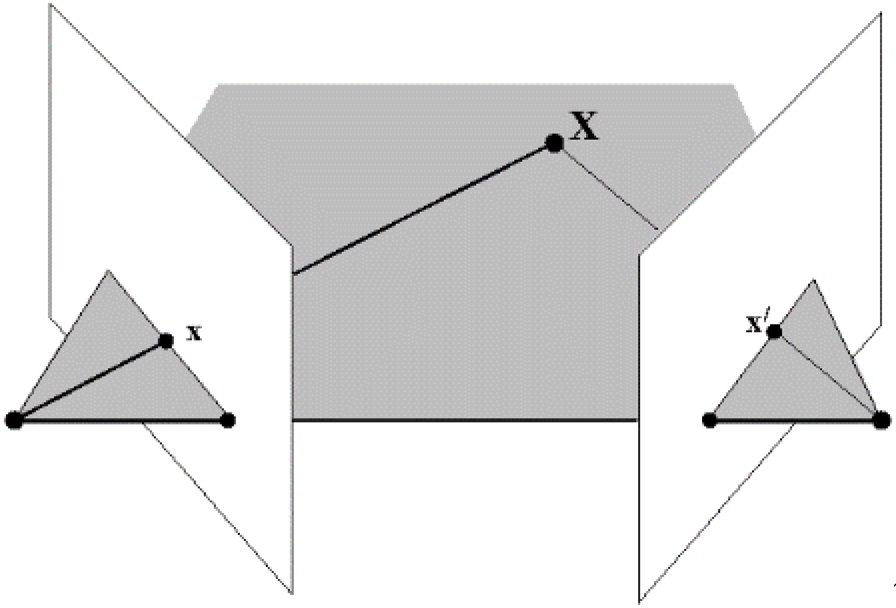
- Feature tracking over whole sequence
- Structure & motion initialization
  - Compute F between two initial images
  - Compute P_1 and P_2
  - Triangulate 3D points of the matched features
- For each additional view
  - Compute the camera pose
  - Refine and extend 3D points
- Self-Calibration
  - Upgrade the projective reconstruction to metric one.
- Refine structure and motion
  - Bundle adjustment

三角化

- 已知*F*, 计算 *P* 和 *P*'

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

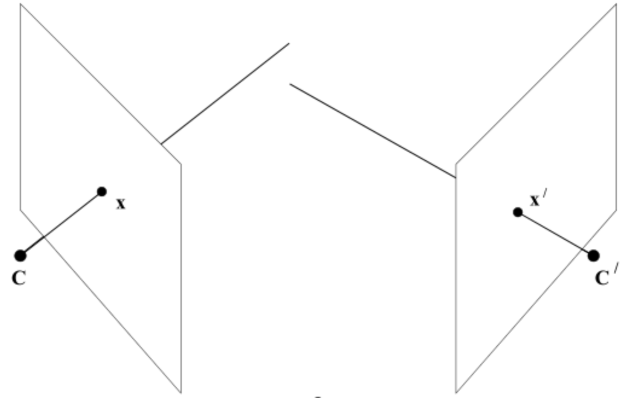
- 已知**x** 和 **x**'
- 计算**X**: $x = \mathbf{P}X$ $x' = \mathbf{P}'X$



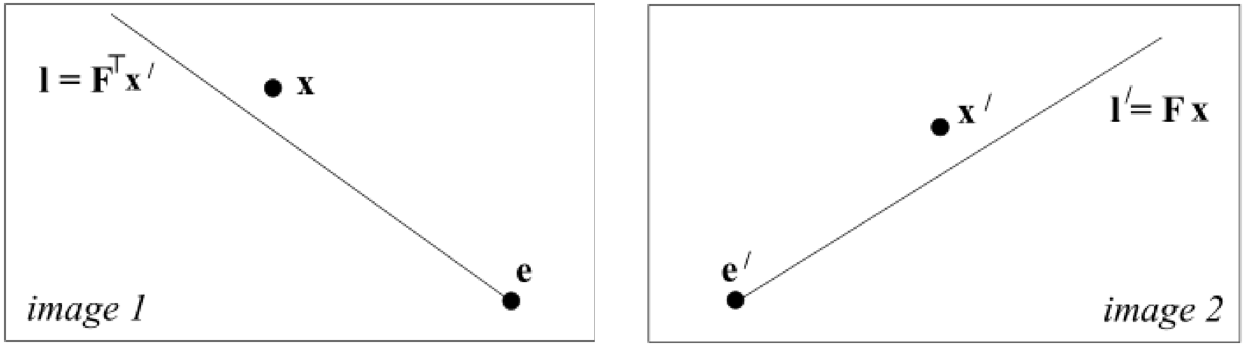
ɿard Hartley and Andrew Zisserman.“Multiple View Geometry in Computer Vision”. Cambridge University Press, Second Edition 2004.

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

  x=**P**X

  x'=**P**'X

  - **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

- Cost function

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

- 用Levenberg-Marquart算法求解

Knowing 3D points, Compute Camera Motion

- 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

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

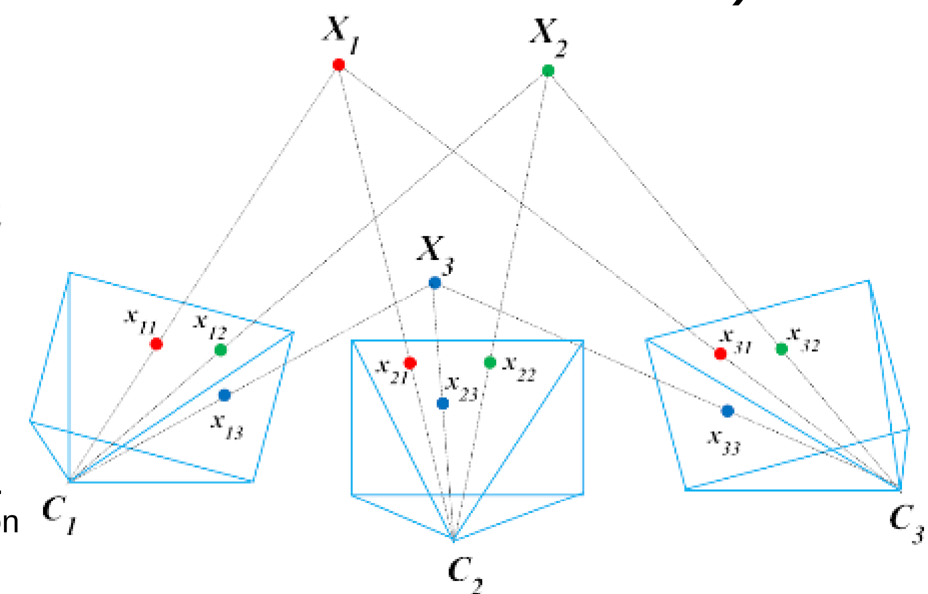
Decompose *M* into *K*, *R* by QR decomposition

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

Bundle Adjustment

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

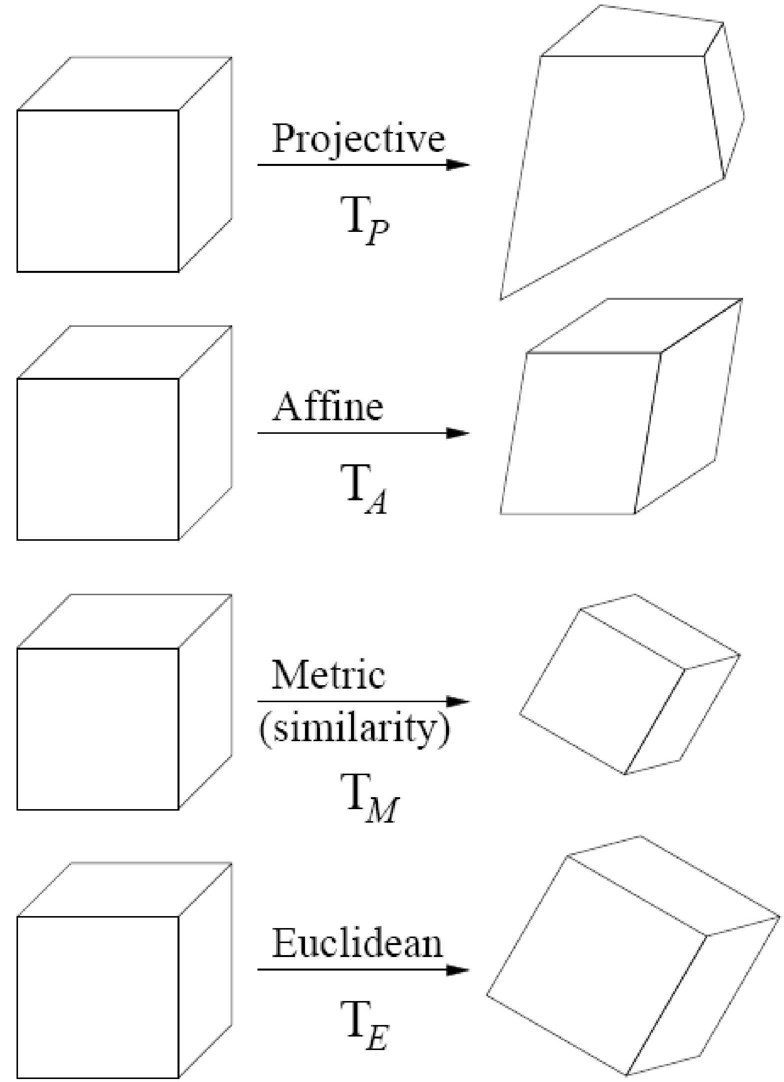
$$\arg\min_{\mathbf{P}_k, \mathrm{X}_i} \sum_{k=1}^{m}\sum_{i=1}^{n} D\left(\mathrm{x}_{ki}, \mathbf{P}_k\left(\mathrm{X}_i\right)\right)^2$$



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.

推荐SfM开源系统

- ENFT-SFM or LS-ACTS
  - http://www.zjucvg.net/ls-acts/ls-acts.html
- OpenMVG
  - https://github.com/openMVG/openMV
- VisualSFM
  - http://ccwu.me/vsfm/