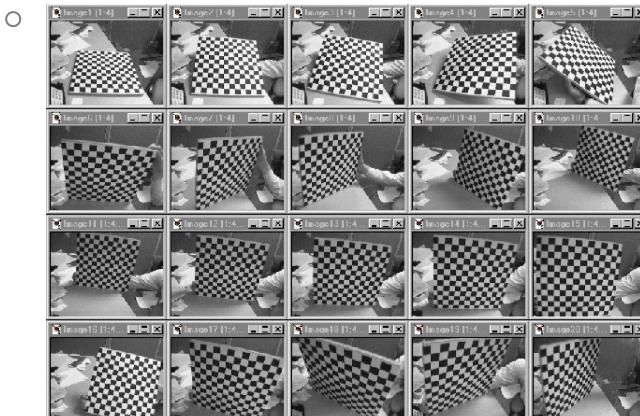


Lecture 3 - Structure from Motion

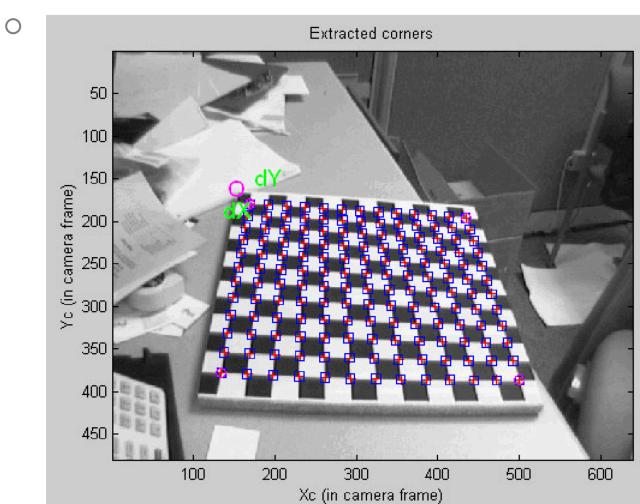
3.1 Preliminaries

3.1.1 Camera Calibration

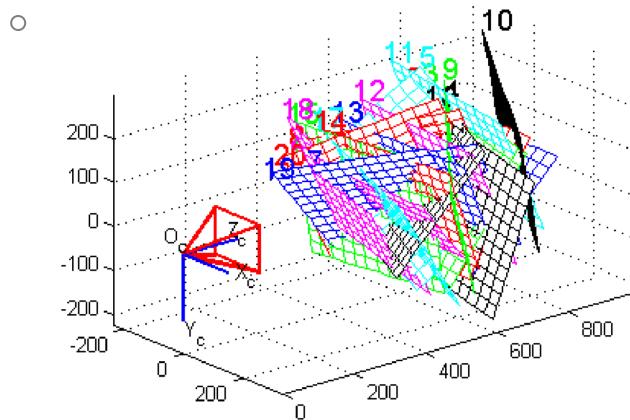
- 相机参数校正 (Camera Calibration)
 - 是通过二维图像重建三维模型的基础
 - 本质上是找到 intrinsic/extrinsic (内部/外部) 参数的过程
 - 常见的，会使用 checkerboard 作为校正 target
- 操作过程
 1. 选择一个 calibration target (比如 checkerboard)
 2. 将其放置于一个平滑的表面，然后以各种不同的角度拍摄 (注意，尽量使每张图中，target 都是完全可见的)



- 3. 检测 target 上的特征 (对于 checkerboard, 例如所有的交点)



- 4. 最后，将相机的外部参数 (extrinsics) 和内部参数 (intrinsics) 结合优化



- extrinsics
 - 相机相对于相机坐标系的位置
- intrinsics
 - 可能 5 个或 10 个，取决于是否使用畸变参数
- initialize
 - 随机初始化 extrinsics 和 intrinsics 可能会导致优化过程陷入局部最优值，因此有一些算法用于初始化
 - **Closed-form solution** initializes all parameters except for distortion parameters
 - **Non-linear optimization** of all parameters by minimizing reprojection errors

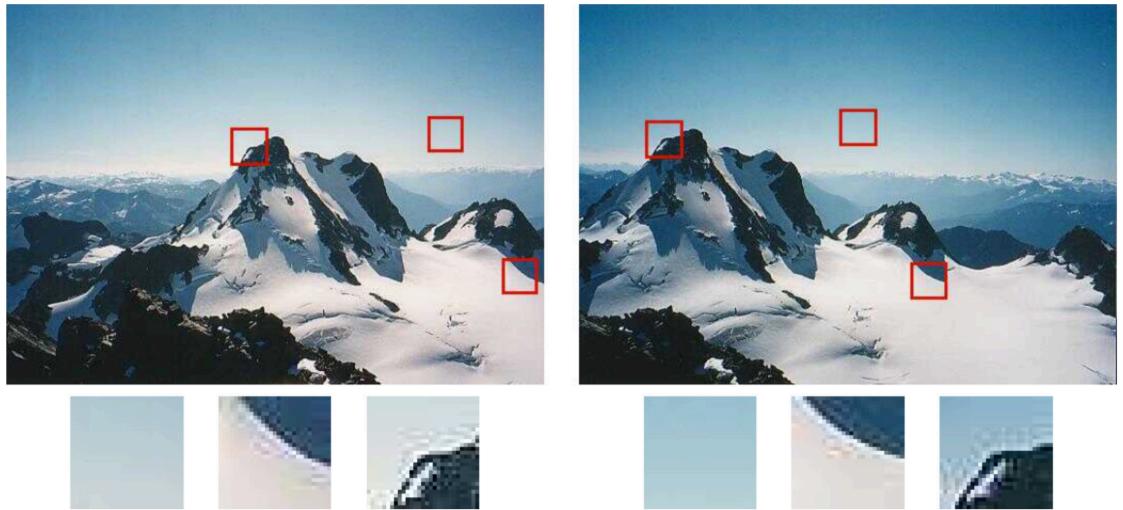
Zhang Z. A flexible new technique for camera calibration[J]. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22(11): 1330-1334.

http://www.vision.caltech.edu/bouguetj/calib_doc/index.html | A good matlab tool box

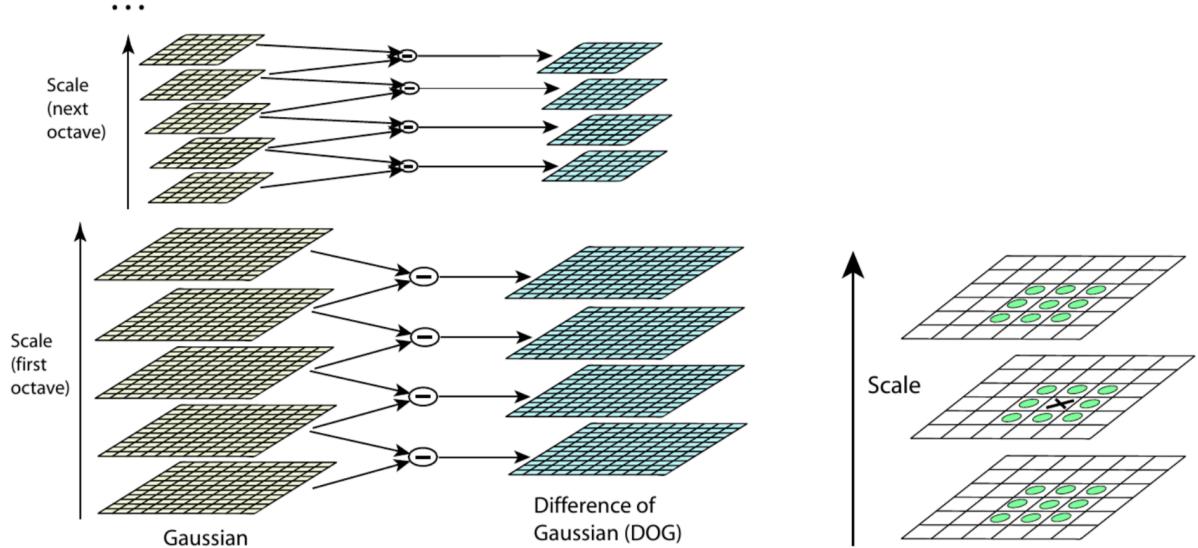
https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html | Relevant support in opencv of python

3.1.2 Feature Detection and Description

- Point Features
 - 对于现实中同一事物的不同视角，我们往往需要找到具有特征的点 (Salient points)，使得我们在不同的图像中可以定位到这些点

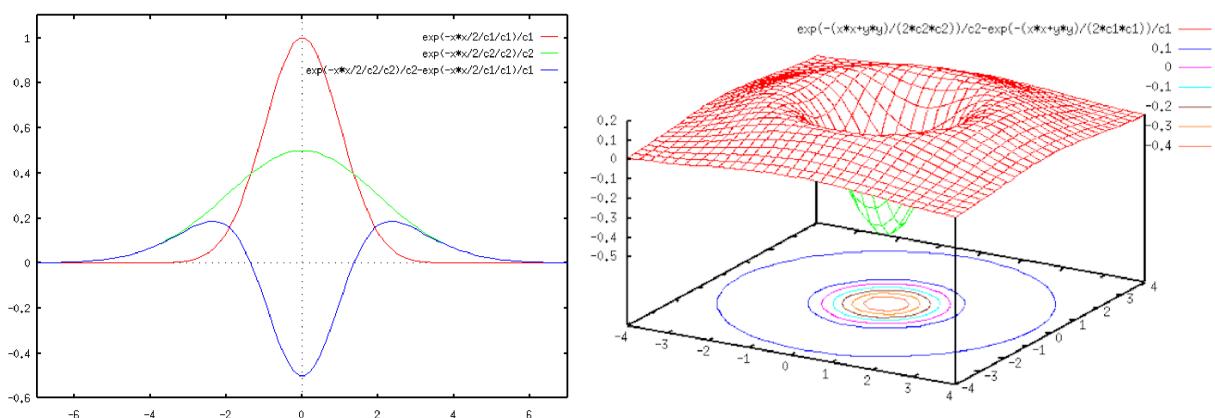


- ■ 例如上图中的三个例子，有些点（如天空中），颜色非常均匀、模糊，这种点似乎到处都是，不好识别其特征；有些点（如山顶尖），具有非常显著的特征，使得我们在别的图像中也可以很容易找到；有些点（如阴影处），黑色与白色之间几乎是一个线性过渡线，这种点在图像中也有很多，尽管它不是很模糊，但是定位难度依旧很大
- 我们希望有一种 algorithm 来实现这一过程，找到所有的 salient point，甚至给出对应的描述（用于匹配不同视角中对应的 salient point）
- 如何定义这些 Salient point 呢？
 - 特征应该具备
 - invariant to perspective effects and illumination （不受透视效果和光照的影响）
 - 相同 salient point 应该在不同位置、不同视角下都具有相同的 feature vector
 - 显而易见，对于 RGB 特征而言，并不具备这些能力——当你旋转、平移、改变光照强度时，这些特征都会变化
- Scale Invariant Feature Transform (SIFT)

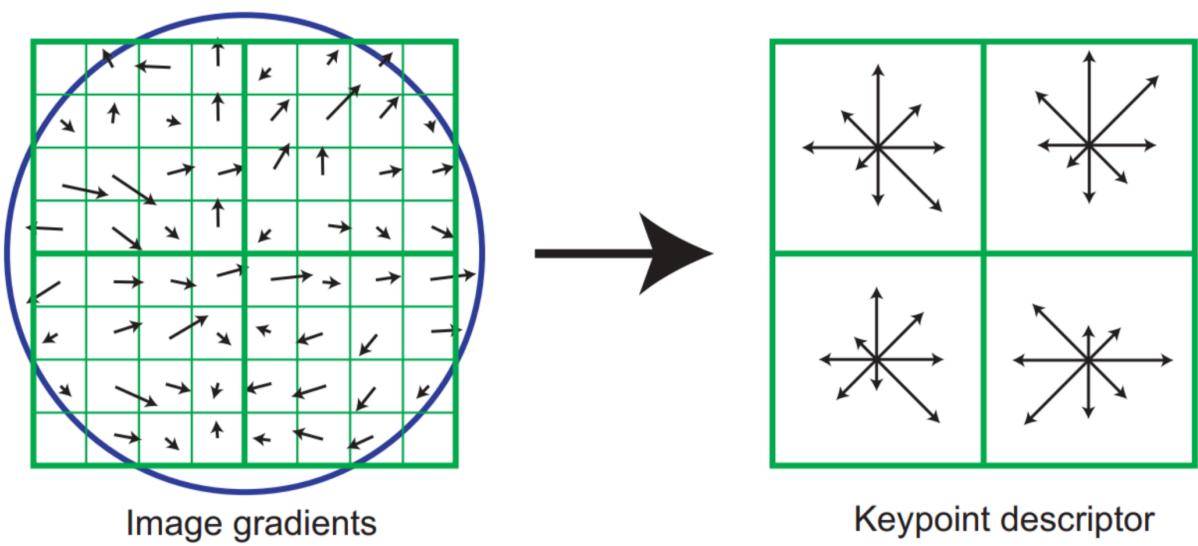
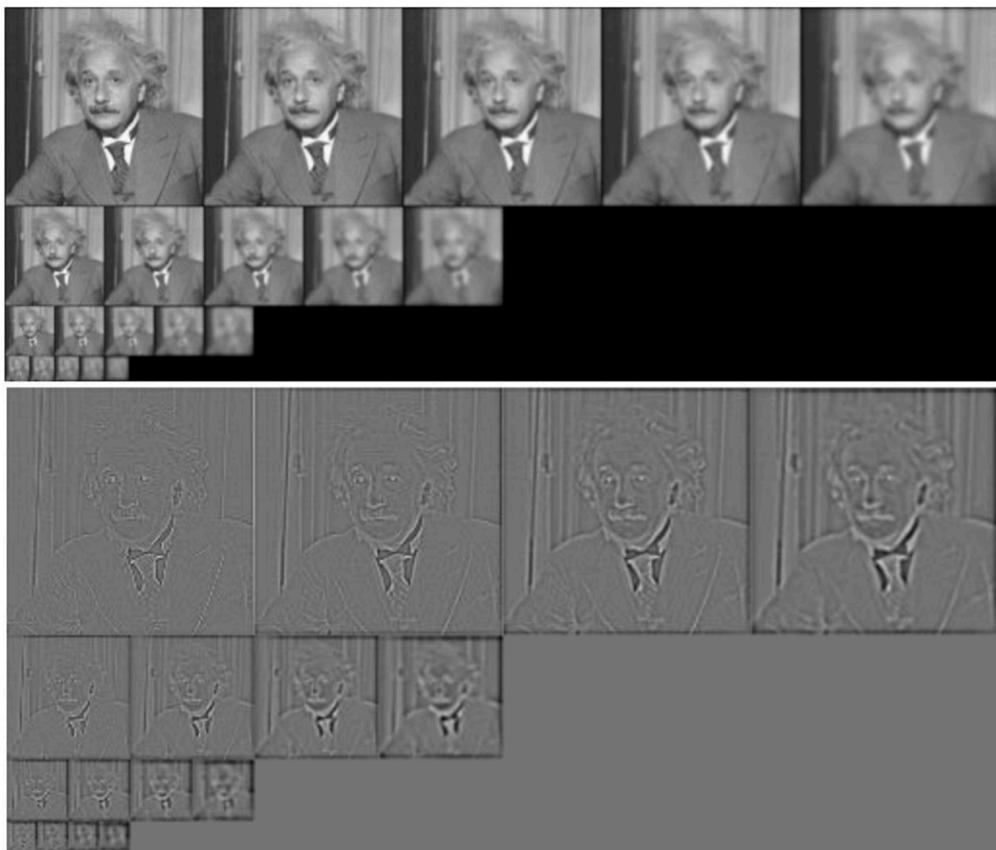


- SIFT 会通过在一张图上迭代式应用 Gaussian 滤波器来构造一个 **scaling space**

1. 具体来说，在原始的图像上不断引用 Gaussian 滤波器，每应用一次它就会模糊一点，直到达到某个提前设定的阈值，然后将图像缩放到下一个层级，继续应用
 2. 每轮迭代的图像之间取差异图像，称为 Difference of Gaussian(DOG)，在这些 DOG 图像中，我们可以看到许多的斑点（blobs），也即 scaling space 中的极值点，这些点便是我们所关注的结果
- 高斯滤波器之间的差异 (DOG) 有什么用？



- 如上图，红色是 a 层，绿色是 b 层，它们之间的 DOG 则是如蓝色线条所示
- 这就是为什么它可以检测斑点 (blobs)
- 下面举一个具体的例子

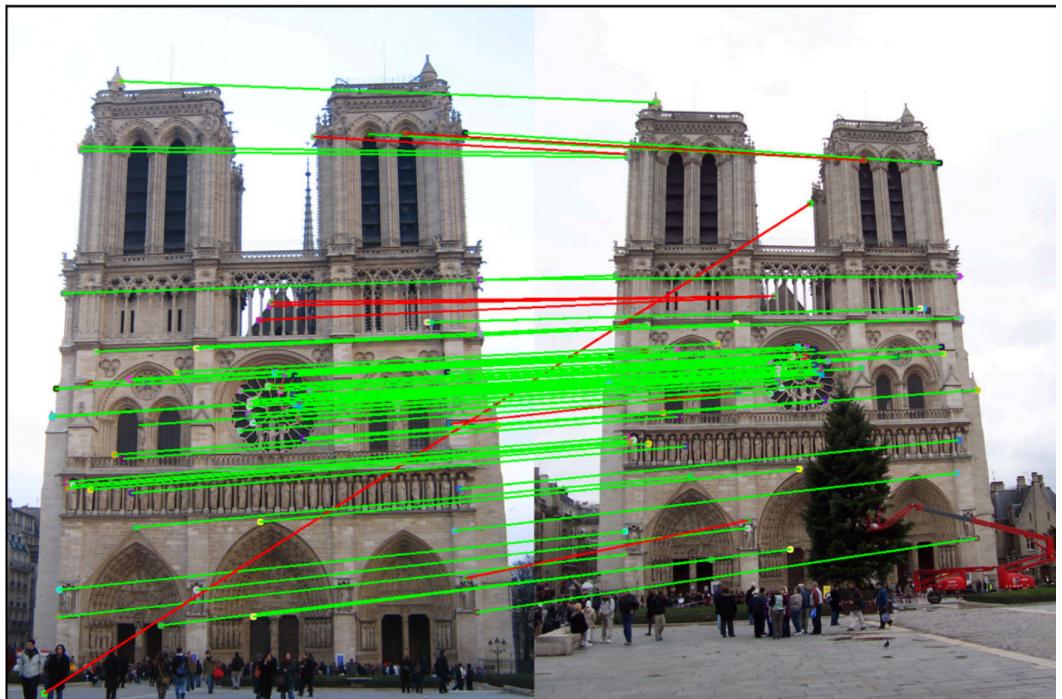


- SIFT 会旋转 descriptor，使其与主导梯度方向对齐
- 对 descriptor 的局部子区域计算梯度直方图
- 将所有的直方图级联在一起，并标准化来构造一个 128D (在本例中) 的特征向量

这里的直方图是由 histogram 直译过来，其实可以理解为图中的箭头

- Summary for Feature and Description

- SIFT 的出现使得 3D 重建领域有了巨大的突破——它具有很强的鲁棒性和不变性，之后又有许多基于 SIFT 的算法被提出，例如 SURF、U-SURF、BRISK、ORB、FAST
- 尽管已经 20 多年过去了，SIFT 至今仍然在被使用
- 特征匹配可以被高效的近邻搜索来替代
- 一些不明确的匹配结果，一般会通过计算最近邻与第二近邻之比来过滤掉
 - 一个很大的比值（例如 >0.8 ）意味着这个匹配可能不是正确的
- 下面给出一个 SIFT 实际应用的例子

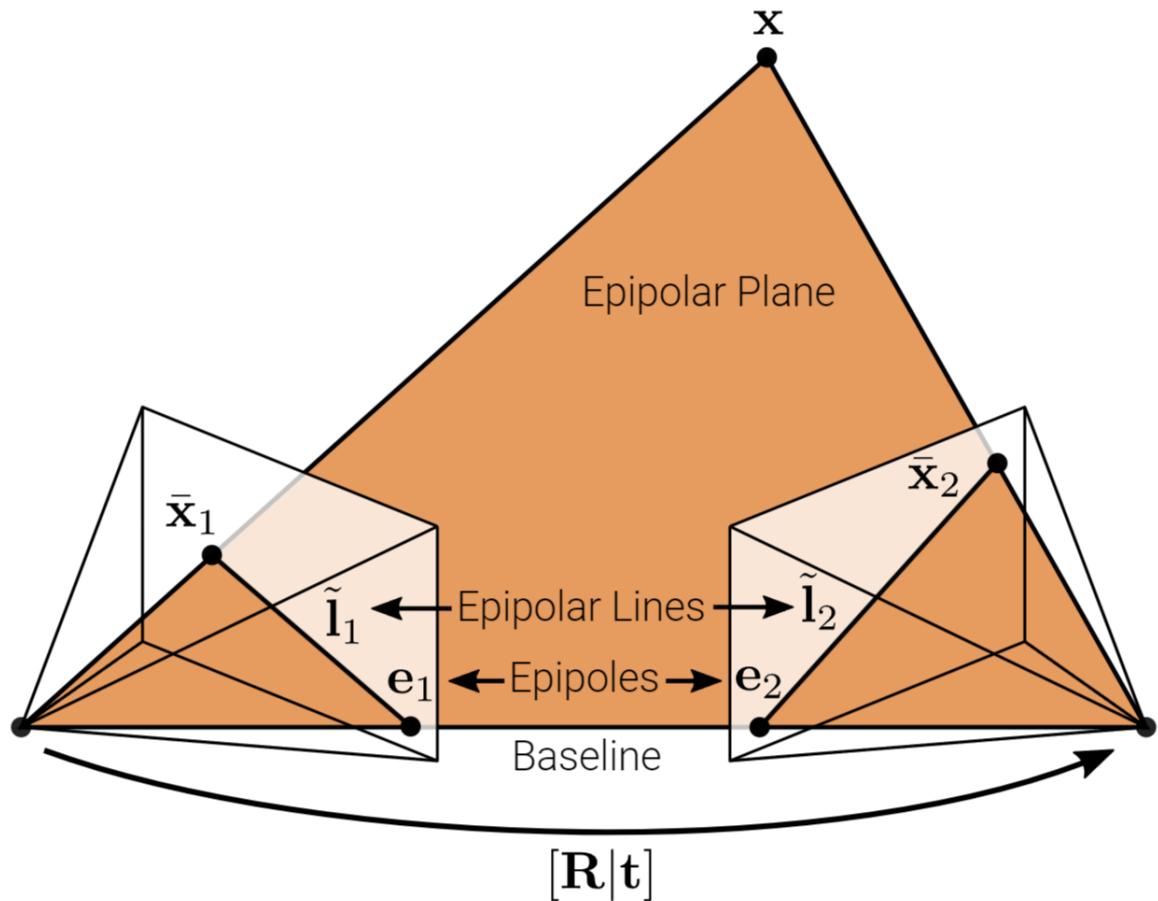


- ○ (Green: correct; Red: wrong)

3.2 Two-frame Structure-from-Motion

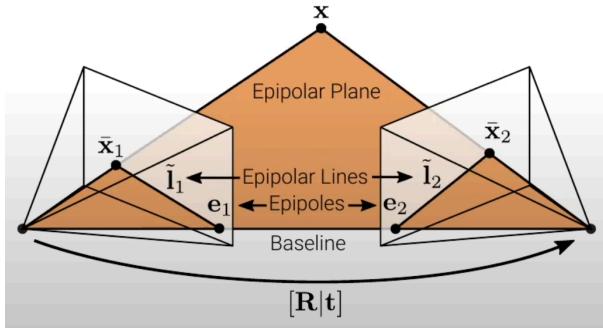
- reconstruct 3D from just two frame 是一件非常令人感到神奇的事

3.2.1 Epipolar Geometry (对极几何)



- 如图所示，我们想要从两个相机 frame 来重建三维模型
- $[R|t]$
- 也即旋转矩阵(Rotation matrix)和平移向量(Translation vector)，用于描述两帧 frame 之间的关系 (理想模型，消除相机误差)
- x 、 \bar{x}_1 、 \bar{x}_2
- x 表示三维模型中的对应点，分别投影在 frame1 和 frame2 上的对应点为 \bar{x}_1 和 \bar{x}_2 (注意，这里为 homogenous vector)
- Epipolar Plane (对极平面)
- 注意到，因为 \bar{x}_1 、 \bar{x}_2 是 x 投影到两个相机成像平面上的，因此两个相机中心点和 x 、 \bar{x}_1 、 \bar{x}_2 是在同一个平面中的，将这个平面定义为 Epipolar Plane
- Baseline
- 两个相机中心点的连线定义为 Baseline
- Epipoles e_1 、 e_2
- 两个 frame 平面与 Baseline 的交线分别为 e_1 、 e_2 ，称为 Epipoles
- 下图展示了，对于相同的 frame 还原的同一个三维模型中的若干点之间，Epipolar Plane、Epipolar Line 是会变的，但是 Baseline 和 e_1 、 e_2 是

始终不变的



- Epipolar Line
 - x_i 与 e_i 的连线称为 Epipolar Line

这种模型几乎适用于所有的 two-frame 重建问题，现在的问题是在实际应用中，我们只知道两个相机参数，如何找到 x 与 \bar{x}_1 和 \bar{x}_2 的关系呢？

- Epipolar Geometry 的代数表示
 - 令 K_i 表示第 i 个相机的 Camera Calibration Matrix (相机参数校准矩阵) ，
 - 令 \tilde{x}_i 表示相机 local ray direction 的点，可知 $\tilde{x}_i = K_i^{-1} \bar{x}_i$

这里解释一下 local ray direction，理解为光线直接射入相机平面上的原始点，也即没有经历过 calibration 的原始点坐标。我们认为 \bar{x}_i 是图像坐标系，也即我们得到的肉眼可见的图像中像素的相对位置； \tilde{x}_i 是归一化相机坐标系，也即没有被相机内参矩阵 K_i 校正过的像素坐标位置

因此，我们只需要将图像坐标系左乘内参矩阵的逆，即可还原出 \tilde{x}_i

\tilde{x}_i 也是 homogenous coordinate，为表示方便，不增加多余的下标

- 可知， \tilde{x}_i 一定和 x_i 成比例，因为它们位于同一条射线 (光线从显示物体的某点射向相机成像平面的某点)
 - 因此，有 $\tilde{x}_i \propto x_i$

- $\tilde{x}_2 \propto x_2 = \mathbf{R}x_1 + \mathbf{t} \propto \mathbf{R}\tilde{x}_1 + s\mathbf{t}$
 - 这个式子将相机 1 和相机 2 的 local ray direction 点与其对应的三维点以比例关系关联起来，尽管我们还不确定这个比例系数到底是多少
 - 由于相机 1 可以通过 $[\mathbf{R}|\mathbf{t}]$ 来变换到相机 2，因此有图中的等式
 - 相机 1 到相机 2 的 local ray direction 点之间可能相差一个缩放系数，因为相机 1 与相机 2 未必距离现实物体的距离相等，因此它们对于现实物体的 scaling 程度也不一样，这里用 s 乘在 \mathbf{t} 上面来表示一个缩放关系
- 左边同时 cross product 一个 \mathbf{t} ，得到
 $-[\mathbf{t}]_{\times} \tilde{x}_2 \propto [\mathbf{t}]_{\times} \mathbf{R}\tilde{x}_1$

由于 $\mathbf{t} \times \mathbf{t} = 0$ ，因此其实 s 的值并不重要

这里 $[\mathbf{t}]_{\times}$ 表示向量 \mathbf{t} 的反对称矩阵，也即将 \mathbf{t} 转换为一个 3×3 的矩阵，并参与矩阵乘法

- 再同时左乘一个 \tilde{x}_2^T
 - $\tilde{x}_2^T [\mathbf{t}]_{\times} \tilde{x}_2 \propto \tilde{x}_2^T [\mathbf{t}]_{\times} \mathbf{R}\tilde{x}_1$
 - 可以发现，等式左侧是一个 triple product (混合积)，混合积表示了三个向量围成的平行六面体的体积，由混合积的性质可知，左侧的式子值恒为 0
 - 而右侧的式子欲与常值 0 成比例，则必须为 0
 - 因此有 $\tilde{x}_2^T [\mathbf{t}]_{\times} \mathbf{R}\tilde{x}_1 = 0$
- 由这个等式可以得到 **Epipolar Constraint** (对极约束)
 - $\tilde{x}_2^T \mathbf{E} \tilde{x}_1 = 0$
 - 其中 $\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$
- 可以看到，由于有 $\tilde{x}_2^T \tilde{l}_2 = 0$ ， \mathbf{E} 可以将 \tilde{x}_1 映射到对应 frame2 的 epipolar line 上
 - $\tilde{l}_2 = \mathbf{E} \tilde{x}_1$
 - 类似的，也有 $\tilde{l}_1 = \mathbf{E}^T \tilde{x}_2$
- 由于 epipole 一定在对应的 epipolar line 上，也即 $\tilde{e}_i^T \tilde{l}_i = 0$
 - 因此， $\tilde{e}_2^T \tilde{l}_2 = \tilde{e}_2^T \mathbf{E} \tilde{x}_1 = 0$ ，对于任意的 \tilde{x}_1 都成立
 - 由于上式对任意的 \tilde{x}_1 都成立，因此 $\tilde{e}_2^T \mathbf{E} = \mathbf{0}$ ，否则上式无法恒成立

left null-space: 我们将所有左乘基础矩阵 \mathbf{E} 结果为 0 向量的向量组成的集合称为左零空间

right null-space: 相应的，还有右零空间

- 因此, $\tilde{e}_2^T \in \text{left null - space}$; $\tilde{e}_1^T \in \text{right null - space}$
-

- **Estimating the Epipolar Geometry**

- 那么我们如何从 image 矩阵中得到 essential 矩阵 $\tilde{\mathbf{E}}$ 呢?

- 答: 可以利用 Epipolar Constraint ($\tilde{x}_2^T \tilde{\mathbf{E}} \tilde{x}_1 = 0$)

- 由于两幅 frame 之间有 N 个对应点, 也即 N 组对应的坐标, 可以代入这个等式来得到 \mathbf{E} 的 9 个元素的多组齐次方程

- \$\$

- \left.

- \begin{aligned}

- $x_{11} + x_{12}e_{12} + x_{13}e_{13}$

- $y_{21} + y_{22}e_{22} + y_{23}e_{23}$

- $x_{31} + y_{13}e_{32} + e_{33}$

- \end{aligned}

- \right.\quad=0

- 由于矩阵 $\tilde{\mathbf{E}}$ 是一个齐次矩阵, 我们需要利用 singular

- SVD 的个人理解

- 首先, $\tilde{\mathbf{E}}$ 是齐次的, 意味着它是按照比例缩放时形式不变的, 也即它没有特定的尺度或单位, 我们只关心它的方向信息

- SVD 过程: $\tilde{\mathbf{E}} = \mathbf{U} \Sigma \mathbf{V}^T$

- 其中 \mathbf{U} 和 \mathbf{V} 是正交矩阵, 它们是 $\tilde{\mathbf{E}}$ 矩阵的主要方向, 不受到矩阵规模大小的影响, 只由奇异值的相对大小决定

- Σ 是一个非负对角矩阵, 对角线上的值称为奇异值, 即使这个矩阵是其次的, 我们仍然可以通过奇异值来理解其内部结构和方向

由于有一些乘积项涉及到两幅图，也有一些项只涉及到一幅图，也即它们是不对称的；因此，当存在有测量误差时，这个计算过程会放大这种不对称的测量噪声

因此，Hartley 提出的一篇开创性的论文 “In Defense of the Eight-Point Algorithm” 提出了一种方法——**normalized 8-point algorithm**

- 通过应用 normalized 8-point algorithm 来将图像特征标准化在二维平面中，使其均值和单位方差(unit variance)为 0
 - 然后再应用 SVD(Singular Value Decomposition)，然后再撤销此归一化处理

在实践中，这样的处理顺序会比直接应用 SVD 得到更加稳定的结果，因为在现实中，得到的结果几乎总是会被噪声污染

- 现在我们从 two-frame 已经恢复出了 $\tilde{\mathbf{E}}$ ，接下来考虑如何恢复 $[\mathbf{R}|\mathbf{t}]$
 - 首先从 $\tilde{\mathbf{E}}$ ，我们可以恢复出平移向量 \mathbf{t} 的方向 $\hat{\mathbf{t}}$
 - $\hat{\mathbf{t}}^T \tilde{\mathbf{E}} = \hat{\mathbf{t}}^T [\mathbf{t}]_\times \mathbf{R} = \mathbf{0}$

这里我们只能得到 \mathbf{t} 的方向向量，因为重建后的模型按照比例缩放、平移，在我们目前的考量中都是合法的重建模型，因此这里的 scaling 比例无法确定

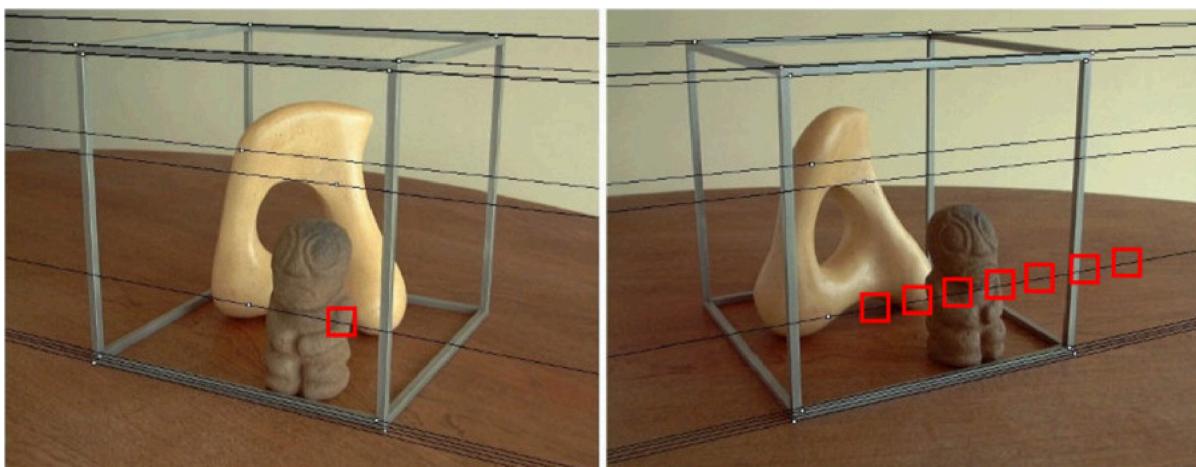
- 因此， $\tilde{\mathbf{E}}$ 是奇异的，并且左奇异向量 $\hat{\mathbf{t}}$ 是与奇异值 0 相关的
 - 在现实中，由于噪声的存在，最小奇异值不会刚好是 0，因此我们一般会选择最小的那个值；
 - 另外两个奇异值大小一般是几乎相等的
- 对于 $\tilde{\mathbf{E}}$ 的奇异值的含义的理解
 - 第 1 个奇异值

- 通常是最大的，表示矩阵的主要方向上的变换尺度，它给出最大信息传播的方向和强度
 - 第 2 个奇异值
 - 通常略小于第 1 个奇异值，表示次要方向上变换尺度
 - 第 3 个奇异值
 - 理论上这个值应该为 0，这个值与平移向量的方向向量 \hat{t} 相关
-

- 旋转矩阵 \mathbf{R} 也是可以计算的，见 Szeliski, Section 11.3 (p. 683)

essential 矩阵 $\tilde{\mathbf{E}}$ 有 5 个 DoF(3 个来自于旋转矩阵 \mathbf{E} ，2 个来自于平移方向 \hat{t})；也即其实我们不需要完整的还原出 8 个 DoF 就可以还原出矩阵 $\tilde{\mathbf{E}}$

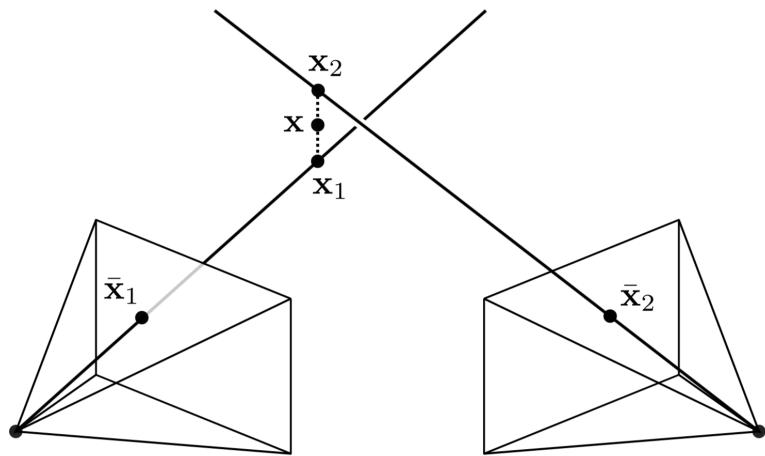
- Fundamental Matrix
 - 如果相机参数 K_i 是位置的，那么我们无法直接使用 local ray directions 点 $\tilde{x}_i = \mathbf{K}_i^{-1}x_i$ ，我们只有 x_i ，则 E 矩阵会写为 \$\$
 \begin{aligned}
 \tilde{x}_2^T \tilde{\mathbf{E}} \tilde{x}_1 &= \bar{x}_2^T \mathbf{K}_2^{-1} \mathbf{E} \mathbf{K}_1^{-1} \bar{x}_1 \\
 &= \bar{x}_2^T \mathbf{F} \bar{x}_1 = 0
 \end{aligned}
 - 其中 $\tilde{\mathbf{F}} = \mathbf{K}_2^{-T} \tilde{\mathbf{E}} \mathbf{K}_1^{-1}$ 被称为 *fundamental matrix* – 与 E 矩阵相同， F 矩阵不如 E 矩阵那么理想，它常用于一些 validation 的场景
 - 当我们有一些除相机参数 K 以外的信息，例如消失点，参数 K 对于所有的 frame 在时间上都是恒定不变的等信息，可以将 perspective reconstruction 提升为一个 metric 级别的 reconstruction



- 上图是一个 Epipolar Geometry 的例子
 - 如图所示可以看到对应的 epipolar lines
 - 左图的一个点可以在右图对应的 epipolar line 上去找对应的点
 - 可以看到，右图中的对应点确实在对应的 epipolar line 上存在

3.2.2 Triangulation

- 我们已经找到了 two-frame 之间的对应关系，但是我们还没有把点的 3 维位置还原出来，那么这恶鬼过程就需要通过 Triangulation (三角测量)
- 如果相机 *intrinsics* 和 *extrinsics* 都已知，如何还原出 3D 几何？



- 给出带有噪声的 2D 图像，并根据之前的对应关系，找到 \bar{x}_1 和 \bar{x}_2 ，并根据相机中心出发，经过该点的射线，找到相交点
- 但是由于噪声的存在，一般两条线不会精准的相交，因此我们需要找到最靠近于两条射线的点
- 令 \tilde{x}_w 表示现实世界中的 3D 坐标点，用等式 $\tilde{x}_i^S = \tilde{\mathbf{P}}_i \tilde{x}_w$ 来表示某 S 平面上的点投影到现实中的点的变换关系
- 由于等式两边都是齐次的，因此它们拥有相同的方向，只是大小可能不同
- 我们可以考虑用叉乘来表示这一点 $\tilde{x}_i^S \times \tilde{\mathbf{P}}_i \tilde{x}_w = \mathbf{0}$
- 如果用 $\tilde{\mathbf{p}}_{ik}^T$ 来表示第 i 个相机的投影矩阵 $\tilde{\mathbf{P}}_i$ 的第 k 行，则可以得出下式
- \$\$

$$\begin{aligned} & \left[\begin{aligned} & \tilde{\mathbf{p}}_{ik}^T \tilde{x}_i^S \\ & \tilde{\mathbf{p}}_{ik}^T \tilde{x}_w \end{aligned} \right] = \mathbf{0} \\ & \tilde{\mathbf{p}}_{ik}^T \tilde{x}_i^S = 0 \\ & \tilde{\mathbf{p}}_{ik}^T \tilde{x}_w = 0 \end{aligned}$$

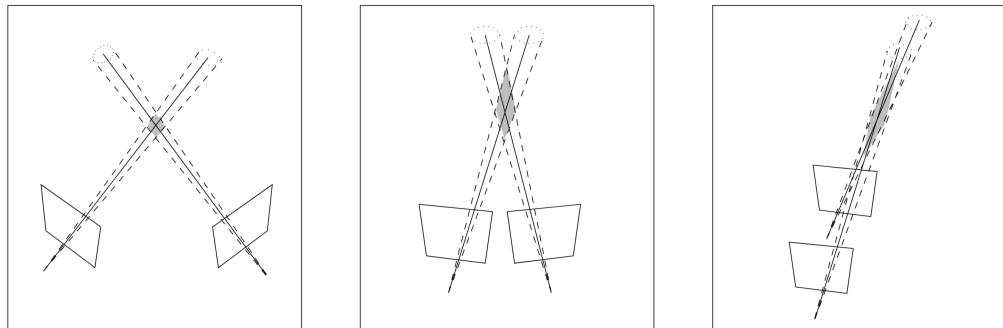
```
\end{aligned}
\right]\tilde{\textbf{x}}_w=\textbf{O}
```

—这里只有 $two-frame$, 也即2个约束, 当然观测的角度 ($frame$) 越多越好

- Reprojection Error Minimization

- 尽管 DLT 在大多数时候都表现得很好, 但是它对于 perspective transformation 并不是 invariant 的, 这意味着它并没有很好的考虑噪声的问题
- 因此, gold standard 算法需要做的就是最小化重投影的误差
 - $\bar{\mathbf{x}}_w^* = \operatorname{argmin}_{\bar{\mathbf{x}}_w} \sum_{i=1}^N \|\bar{\mathbf{x}}_i^s(\bar{\mathbf{x}}_w) - \bar{\mathbf{x}}_i^o\|_2^2 \quad \text{with observation } \bar{\mathbf{x}}_i^o$
 - 根据这个优化关系, 我们可以用一些基于数值梯度的方法, 例如 Levenberg-Marquardt 算法
 - 这个优化关系式, 允许我们根据一些特定的噪声情况, 添加对应的去噪算法

- Triangulation Uncertainty



- ■ 三角测量的效果也取决于相机的位置和姿势
 - Uncertainty
 - 当光线逐渐趋于平行时, 不确定的区域增加 (也即图中灰色区域增加), 更难确定真实点
 - Tradeoff
 - 靠得越近的图像越容易匹配到相同的特征, 但是三角测量环境就会更加困难

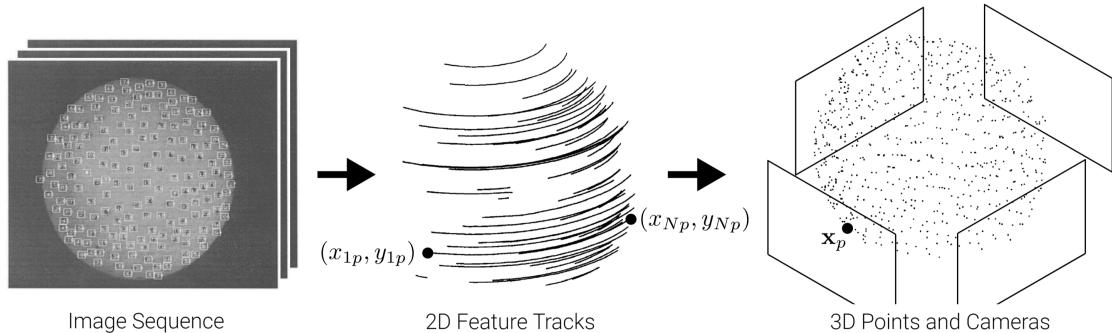
3.3 Factorization

- 目前已经学习了从 $two-frame$ 中构建 construction, 然而更多时候从 2 个以上的 frame 重建模型的效果会更好, 那么如何从多视角重建 3d 模型呢?
 - Kanade 等人给出一种称为 Factorization 的方法, 它是一种古老但是

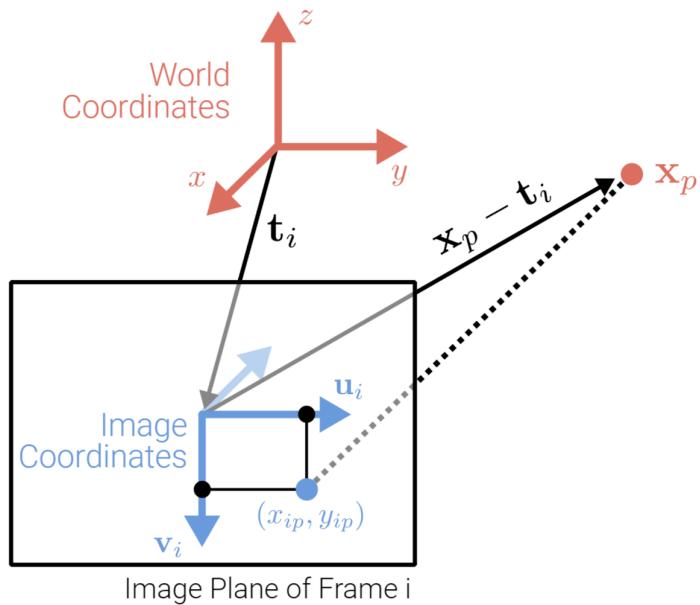
在数学上很优雅的方法

3.3.1 Orthographic Factorization

- 令 $W = \{(x_{ip}, y_{ip}) | i = 1, \dots, N, p = 1, \dots, P\}$ 来表示 N 个 frames, P 个 feature 的 image coordinates (确保每一个 frame 中都包含了所有的 feature)

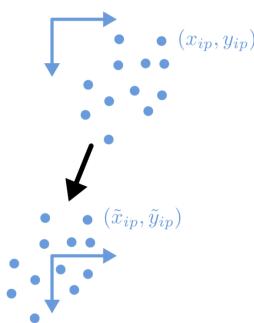


- **Image Sequence**
 - 我们有 N 个 image 组成的 sequence，并且每一副 image 都包含着至少 P 个共同的特征来 detect 和 track
 - **2D Feature Tracks**
 - 利用 LIFT(Learned Invariant Feature Transform) descriptor 或者 Lucas-Kanade(L-K)特征跟踪器，得到 2D 图像特征的轨迹
 - **3D Points and Cameras**
 - 最后我们重建出 3D 模型 (点的集合) 以及自动投影的相机平面
- 在 Orthographic Factorization (正交分解) 方法中，一个 3D 点 \mathbf{x}_p 会被映射到第 i 个 frame 中的像素 (x_{ip}, y_{ip})
 - $x_{ip} = u_i^T(\mathbf{x}_p - \mathbf{t}_i)$
 - $y_{ip} = v_i^T(\mathbf{x}_p - \mathbf{t}_i)$



- 3D 点定义在世界坐标系中，我们计算出世界坐标系和第 i 个 frame 对应的相机坐标系的平移差 t_i ，然后将点 \mathbf{x}_p 转换到相机坐标系中
- 然后我们将其分别映射到 x 、 y 轴上，得到 x_{ip} 和 y_{ip}
 - 由于世界坐标系的定义是人为主观的，为了能够体现 invariant 性质，我们假设世界坐标系的原点位于所有的 3D 点的中心位置，也即理想状态下有 $\frac{1}{P} \sum_{p=1}^P \mathbf{x}_p = 0$
- 现在我们有了 (x_{ip}, y_{ip}) 来表示第 i frame 的第 p 个 feature 的 2D 坐标，将每一个 frame 的坐标系居中到所有特征点中心（也即使所有特征点均值为 0），并收集它们，来产生中心测量矩阵 (centered measurement matrix) $\tilde{\mathbf{W}}$

○



$$\tilde{x}_{ip} = x_{ip} - \frac{1}{P} \sum_{q=1}^P x_{iq}$$

$$\tilde{y}_{ip} = y_{ip} - \frac{1}{P} \sum_{q=1}^P y_{iq}$$

$$\tilde{\mathbf{W}} = \begin{bmatrix} \tilde{x}_{11} & \dots & \tilde{x}_{1P} \\ \vdots & & \vdots \\ \tilde{x}_{N1} & \dots & \tilde{x}_{NP} \\ \tilde{y}_{11} & \dots & \tilde{y}_{1P} \\ \vdots & & \vdots \\ \tilde{y}_{N1} & \dots & \tilde{y}_{NP} \end{bmatrix}$$

- 注意，这里的~符号和前几章都不同！这里的 \tilde{x}_{ip} 和 \tilde{y}_{ip} 分别指的是居中的坐标，并非齐次坐标！，这种用法也只保持在这一小节中
- 根据 $x_{ip} = u_i^T(\mathbf{x}_p - \mathbf{t}_i)$ 居中的 x 坐标 \tilde{x}_{ip} 可以根据下式计算
- \$\$

\begin{aligned}

```

\tilde{x}_{ip} = \frac{1}{P} \sum_{q=1}^P x_{iq}
&= u_i^T (\textbf{x}_p - \textbf{x}_t) + \frac{1}{P} \sum_{q=1}^P (x_{iq} - u_i^T \textbf{x}_t)
&= u_i^T (\textbf{x}_p - \textbf{x}_t) + u_i^T \frac{1}{P} \sum_{q=1}^P (x_{iq} - u_i^T \textbf{x}_t)
&= u_i^T (\textbf{x}_p - \frac{1}{P} \sum_{q=1}^P x_{iq}) + u_i^T \frac{1}{P} \sum_{q=1}^P (x_{iq} - u_i^T \textbf{x}_t)
&= u_i^T (\textbf{x}_p - \tilde{\textbf{w}}_i) + \tilde{\textbf{w}}_i^T \textbf{x}_t

```

—也即，我们可以得到结论 $\tilde{x}_{ip} = u_i^T \mathbf{x}_p$, $\tilde{y}_{ip} = v_i^T \mathbf{x}_P$

- 我们可以将中心测量矩阵 $\tilde{\mathbf{W}}$ 推导出 $\tilde{\mathbf{W}} = \mathbf{R}\mathbf{X}$

- \$\$

```

\mathbf{R} = \begin{bmatrix} 1^T & 2^T & \dots & N^T \end{bmatrix}^T
\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_P \end{bmatrix}

```

> 这里省略 $\tilde{\mathbf{W}}$ 不写了，主要 *latex* 敲起来太麻烦 QWQ 主要是和之前的表达

- 这里， \mathbf{R} 表示相机的 motion(rotation)， \mathbf{X} 表示 3D 场景的结构，其中 $\mathbf{R} \in R^{2N \times 3}$, $\mathbf{X} \in R^{3 \times P}$, 也即 $\tilde{\mathbf{W}} \in R^{2N \times P}$ 。在没有噪声的情况下， $\tilde{\mathbf{W}}$ 的秩最多为 3；当添加噪声后，矩阵 $\tilde{\mathbf{W}}$ 很容易会变为满秩矩阵。因此，我们需要通过消除噪声来检索秩为 3 版本的 $\tilde{\mathbf{W}}$ 矩阵
- 给出现实中观察得到的满秩矩阵 $\tilde{\mathbf{W}}$ 后，我们希望找到对应秩为 3 的版本的矩阵，称为 $\hat{\mathbf{W}}$ ，可以通过 SVD 来计算 $\|\hat{\mathbf{W}} - \tilde{\mathbf{W}}\|_F$ 的最小值
 - SVD: $\hat{\mathbf{W}} = \mathbf{U}\Sigma\mathbf{V}^T$
 - 我们考虑，分解后应该有 3 个最大的奇异值，其余的奇异值应该很小，因为它们只是捕获噪声
 - 计算得到 Σ 矩阵后，对应的也可以计算出 $\hat{\mathbf{R}} = \mathbf{U}\Sigma^{\frac{1}{2}}$ 和 $\hat{\mathbf{X}} = \Sigma^{\frac{1}{2}}\mathbf{V}^T$

Forbenius 范数，用于衡量矩阵的大小或“长度”，定义为

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

可以把 F 范数看作是将矩阵元素展开成一个向量后，计算这个向量的欧几里得范数（向量的长度），也即用一个总体的数值反映了矩阵的大小

- 然而！不幸的是，这种分解并不是唯一的
 - 存在有矩阵 $Q \in R^{3 \times 3}$ ，例如 $\hat{\mathbf{W}} = \mathbf{U}\Sigma\mathbf{V}^T = \hat{\mathbf{R}}\hat{\mathbf{X}} = (\hat{\mathbf{R}}\mathbf{Q})(\mathbf{Q}^{-1}\hat{\mathbf{X}})$

- 这里，不同的 $\hat{\mathbf{R}}$ 和 $\hat{\mathbf{X}}$ 却得到了相同的 $\hat{\mathbf{W}}$
- 那么如何找到正确的（唯一的）那个 Q 矩阵呢？我们可以通过观察 R 矩阵的一些特点
 1. R 矩阵的行向量应该是单位向量
 2. R 矩阵的前半部分应该和后半部分正交
- 基于这两个特点，我们可以给出一些约束条件
 - $\hat{u}_i \mathbf{Q} (\hat{u}_i^T \mathbf{Q})^T = \hat{u}_i^T \mathbf{Q} \mathbf{Q}^T \hat{u}_i = 1$
 - $\hat{v}_i \mathbf{Q} (\hat{v}_i^T \mathbf{Q})^T = \hat{v}_i^T \mathbf{Q} \mathbf{Q}^T \hat{v}_i = 1$
 - $\hat{u}_i \mathbf{Q} (\hat{v}_i^T \mathbf{Q})^T = \hat{u}_i^T \mathbf{Q} \mathbf{Q}^T \hat{v}_i = 0$
- 这些约束条件，我们称为 **Metric Constraints**
 - 这给了我们一系列关于 $\mathbf{Q} \mathbf{Q}^T$ 矩阵的线性方程，我们可以使用 standard Cholesky decomposition 来分解还原出 \mathbf{Q} 矩阵
- Orthographic Factorization 算法过程
 1. 拿到 measurement $\hat{\mathbf{W}}$
 2. 计算 SVD $\hat{\mathbf{W}} = \mathbf{U} \Sigma \mathbf{V}^T$ ，并保留最大的 3 个奇异值
 3. 定义 $\hat{\mathbf{R}} = \mathbf{U} \Sigma^{\frac{1}{2}}$ 和 $\hat{\mathbf{X}} = \Sigma^{\frac{1}{2}} \mathbf{V}^T$
 4. 通过 Metric Constraints，计算 $\mathbf{Q} \mathbf{Q}^T$ ，并利用 Cholesky 分解，计算 \mathbf{Q}
 5. 计算 $\mathbf{R} = \hat{\mathbf{R}} \mathbf{Q}$ 和 $\mathbf{X} = \mathbf{Q}^{-1} \hat{\mathbf{X}}$
 - 算法分析
 - 优点
 - 封闭式的解决方法，计算速度很快，没有局部最小值
(不过这并不是一般情况下的优点，因为有 **中心世界坐标** 的全局假设)
 - 缺点
 - 要求完整的特征检测追踪轨迹，如果有某一帧出现遮挡等情况而失去了对应特征，则无法应用该算法
 - 解决方案 (see T&K, Sec. 5)
 - 可以应用到 features/frames 的子集 (没有问题的部分子集)，并迭代式的传播，从而补全缺失的条目



- ■ 上图是该算法应用的一个例子，它们试图重建了手的模型
 - 这是 1992 年的应用结果

- Future Work

- Tomasi and Kanade 最早提出的分解方法建立在 **orthography** 的假设前提下
- Christy and Horaud (TPAMI, 1996) 则展示了一种初始化 **orthographic** 的重建过程，并以迭代方式纠正透视的效果
- Triggs (CVPR, 1996) 展示了**投影式的分解方法**(projective factorization)，并且迭代式地更新深度
 - 即使结果不是很准确，这些 factorization 方法仍然可以为迭代式技术 (如 bundle adjustment) 提供良好的初始化结果
 - 然而，现代的 Structure from Motion 方法 (例如 COLMAP) 通常会执行 **incremental bundle adjustment** (增量化束带调整)，使用精心选择的双视图重构来进行初始化，并且迭代式的增加新的图像 (相机) 来进行重建

3.4 Bundle Adjustment

开始前先小小吐槽一下，不知道为什么 youtube 上这节课的字幕功能无法正常使用(‘-‘)，

- **Budle Adjustment** (束带调整)
 - 目的：优化重投影的误差 (也即观察到的 feature 和投影在 image plane 上的 3D 点之间的距离)，相关于相机参数、以及 3D 点云
 - 内容
 - 令 $\Pi = \{\pi_i\}$ 表示 N 个相机的 intrinsic 和 extrinsic 参数
 - 令 $\chi_w = \{\mathbf{x}_p^w\}$ ，其中 $\mathbf{x}_p^w \in R^3$ 表示现实中 3D 点的坐标
 - 令 $\chi_s = \{\mathbf{x}_{ip}^s\}$ ，其中 $\mathbf{x}_{ip}^s \in R^2$ ，表示每一个相机 i 的 image plane 上的观察到的特征 p 的坐标
 - Bundle adjustment 优化所有观察数据的重投影的误差，如下式
 - \$\$

$$\begin{aligned} \|\mathbf{p}_i\|, \|\mathbf{c}_{i,w}\| &= \underbrace{\|\mathbf{p}_i, \mathbf{c}_w\|}_{\text{operatorname}{argmin}} \\ \sum_{i=1}^N \sum_{p=1}^{P_i} w_{ip} &\quad \text{||} \quad \text{textrm{bf}}\{\mathbf{x}\}^s\{\mathbf{p}\} - \mathbf{p}_i\{\mathbf{i}\} (\text{textrm{bf}}\{\mathbf{x}\}^w\{\mathbf{p}\}) \quad \text{||}^2 \end{aligned}$$

w_{ip} 表示是否点 p 在 image i 中能够被观察到 (有一些 frame 中)

- Bundle Adjustment 面临的问题

- 初始部分

- bundle adjustment 的 energy landscape 是极度非凸的
 - (因此) 做出好的初始化, 对条件的要求非常苛刻, 为了确保不会陷入局部最优解
 - 由于将全部的 3D 点和相机一次性初始化是非常困难的, 因此 incremental bundle adjustment 从精选的 two-frame 重建开始, 并迭代式地增加新的 3D 点和相机

- 优化部分

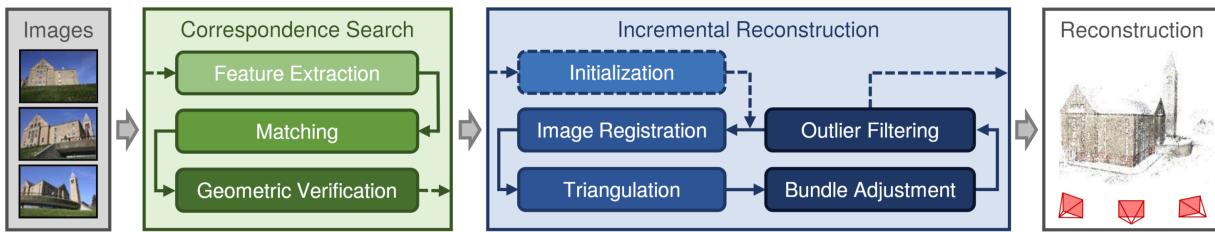
- 当存在数以百万的 features 和数以千计的相机时, 非常大规模的 bundle adjustment 会导致非常庞大的计算量 (参数量立方级别的计算复杂度)
 - 幸运的是, 我们所讨论的问题是 sparse (稀疏) 的 (并非所有的 3D 点都会被任一相机所观察到), 因此一些高效的稀疏实现方案可以被应用 (例如Ceres)

Energy Landscape, 是指能量图或能量表面, 表示优化问题中, 目标函数的值随参数变化的图像

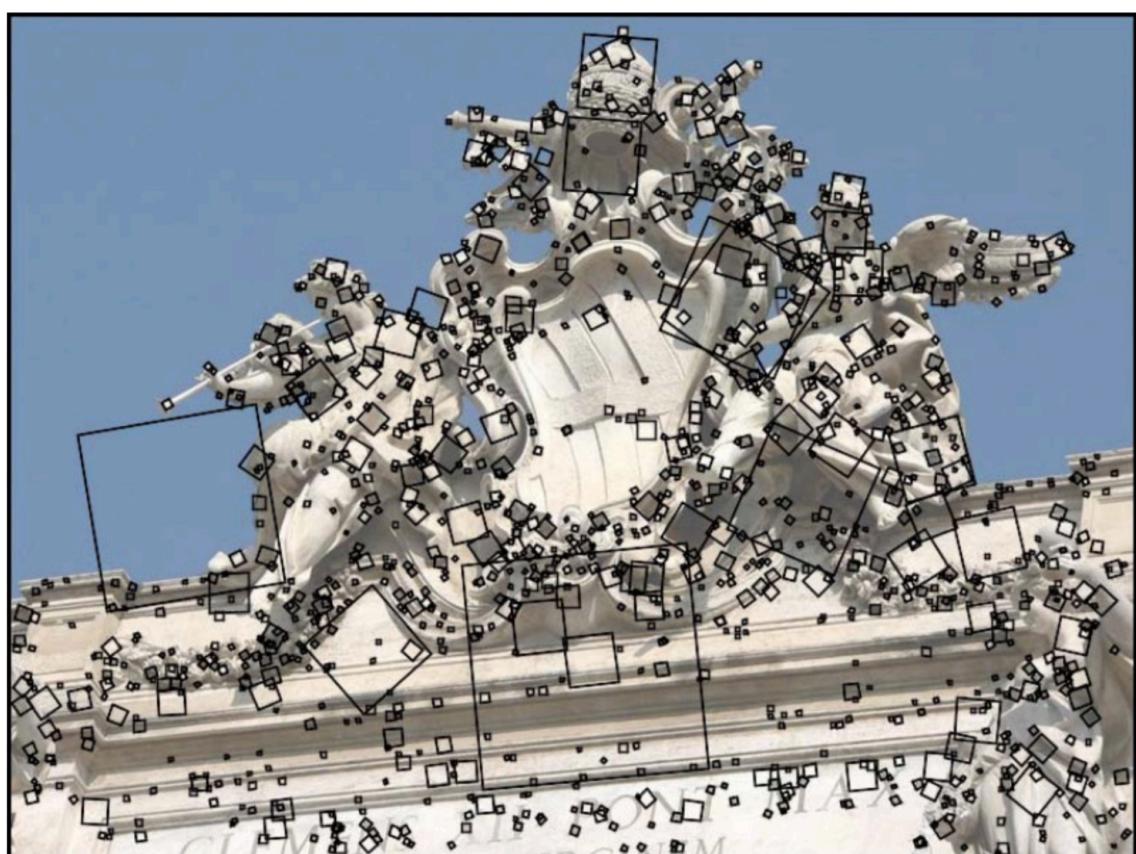
如果能量图是凸(convex)的, 那么它就只有 1 个局部最小值, 优化过程相对简单; 反之, 就意味着存在多个局部最小值, 优化过程会变得很复杂, 容易陷入局部最优解而无法找到全局最优解

Ceres Solver 是一个开源的非线性最小二乘问题 (Non-linear Least Squares, NLS) 求解器, 它支持处理稀疏矩阵, 支持多种优化算法, 适用于不同类型的非线性最小二乘问题

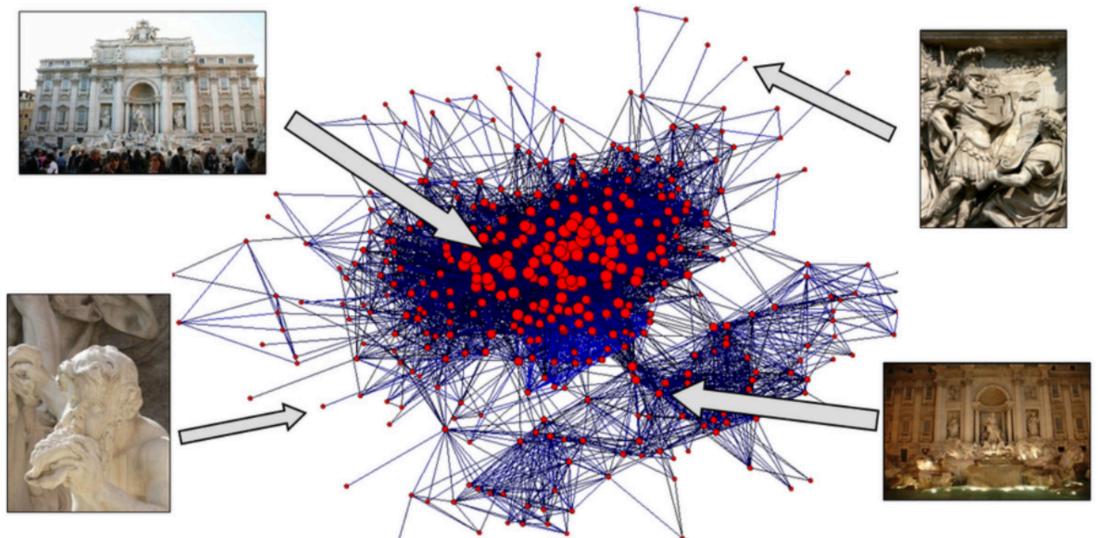
3.4.1 Incremental Structure-from-Motion



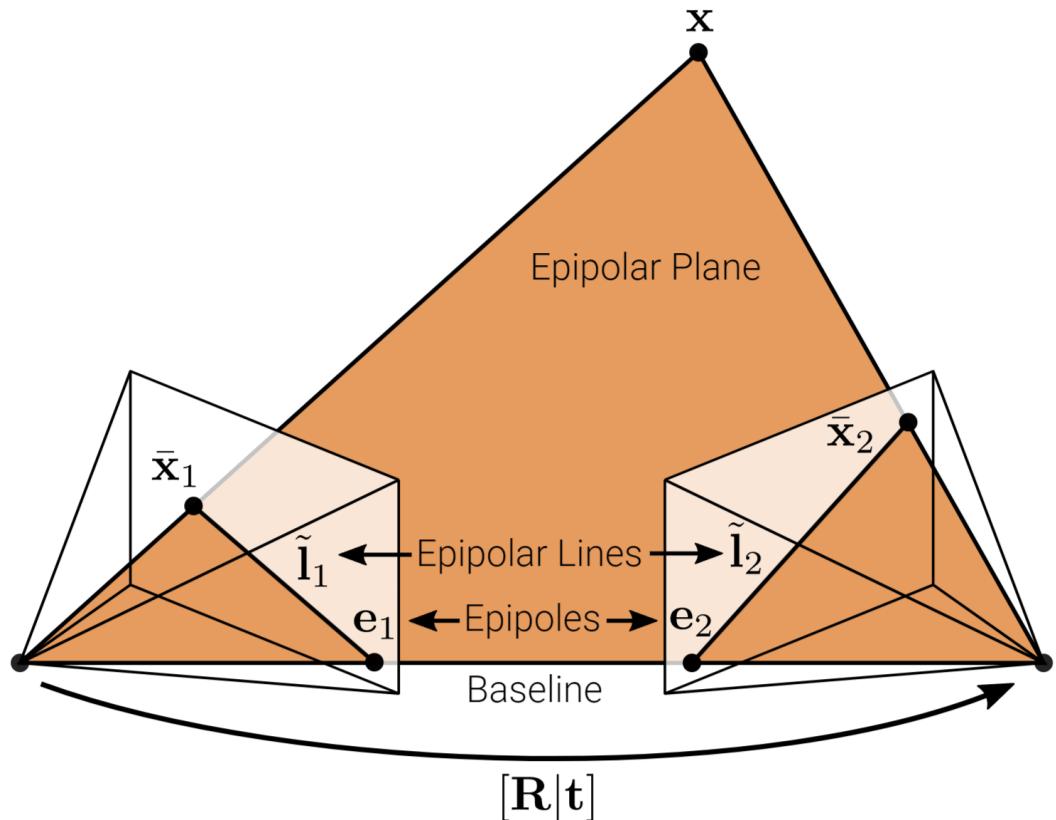
- 上图展示了 Incremental Structure-from-Motion (增量式运动结构) 中最著名的一个方法——COLMAP 的 pipeline
 - 这个 pipeline 实际上可以分为 2 个 stage
 - Correspondence Search
 - 寻找 images 中具有足够鲁棒性的 features
 - 匹配不同 images 中对应的 features
 - 利用 Epipolar geometric 来验证上述结果
 - Incremental Reconstruction
 - 从 two-frame 中进行 reconstruction 的初始化
 - 迭代式增加新的 images
 - 利用 Triangulation 来重构
 - 利用 Bundle Adjustment 来优化
 - Filtering 结果，并重复上述操作，直到迭代结束
 - Detect Features



- ■ 检测 images 中所有可以作为 features 的点
 - 相关 Algorithm
 - SIFT、SURF、BRISK
- Feature Matching & Geometric Verification



- ■ 找到有重叠 feature 部分的 image 对 (也即图中蓝色的线所表示的) , 并且生成对应的 feature 关联关系
- 从图中我们可以看出, 生成的结果出现了多个 cluster, 例如可能会有夜间的图像和白天的图像, 它们之间的特征很难互相匹配
- 在这一步中, 我们需要找到相互关联最密集的两个 frame 作为初始化对象, 因为它们关联的特征最多, 最容易进行重建
- Initialization



- ■ 首先，从一对几乎 feature 全部可见且全部相互关联的 two-frame 进行初始化重建，如 3.2 小节所讲，进行 reconstruction 并构建 Epipolar line 来验证
- Image Registration
 - 找到一个新的，和当前已经重建过的 image 集合有 feature 关联的 image：
 - 令 $\chi = \{\bar{\mathbf{x}}_i^s, \bar{\mathbf{x}}_i^w\}_{i=1}^N$ 为 N 个由投影关系 $\mathbf{x}_i^s = \mathbf{P}\mathbf{x}_i^w$ 关联的 3D 到 2D 的对应关系
 - 由于这些对应关系的向量都是齐次的，他们具有相同的方向、不同的大小。因此，上面的等式 $\mathbf{x}_i^s = \mathbf{P}\mathbf{x}_i^w$ 可以写为 $\mathbf{x}_i^s \times \mathbf{P}\mathbf{x}_i^{ww} = \mathbf{0}$
 - 利用 DLT(Direct Linear Transform) 可以将其写为关于 \mathbf{P} 的线性方程，关于约束系统的解（也即找到 \mathbf{P} 真正的尺度大小）可以由 SVD 给出
 - 假设 $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ ，并且 \mathbf{K} 是一个上三角矩阵，那么 \mathbf{K} 和 \mathbf{R} 都可以利用标准 RQ 分解从 \mathbf{P} 的前 3×3 矩阵中分解得到
 - 如果已知 \mathbf{K} ，我们甚至可以仅从 3 个点来估计 \mathbf{P} 的值 (P3P 算法)
 - 在实践中，通常使用随机采样一致性 (RANSAC) 来去除异常值

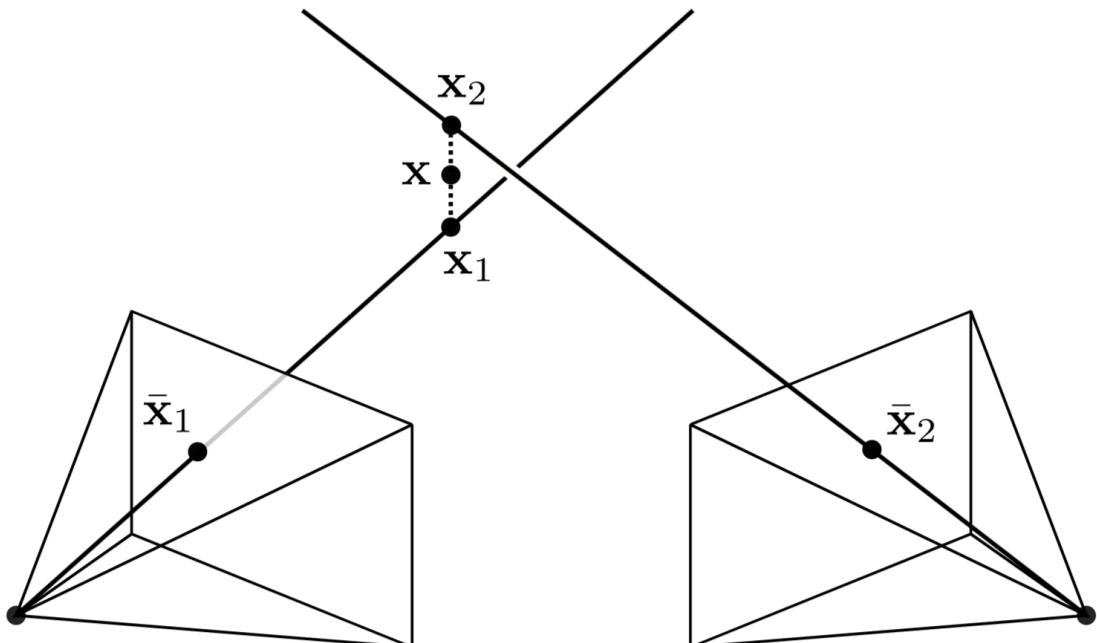
直接线性变换 (DLT, Direct Linear Transform) 是一种用于求解投影矩阵的方法。它通过将一组二维点和对应的三维点表示为线性方程，并求解该方程以获得投影矩阵。

RQ分解是一种矩阵分解方法，将一个矩阵分解为一个上三角矩阵 (R) 和一个正交矩阵 (Q) 的乘积。在计算机视觉中，RQ分解常用于从投影矩阵中提取内参矩阵 (K) 和旋转矩阵 (R)，特别是当内参矩阵是上三角矩阵时。

P3P算法 (Perspective-Three-Point) 用于求解三点透视问题。给定相机中的三个已知点及其在图像平面上的投影，P3P算法可以估计相机的位置和姿态。这在相机标定和定位中非常重要。P3P算法求解的是一组多项式方程，通常会有多个解，需要进一步的判别和筛选。

随机采样一致性 (RANSAC, Random Sample Consensus) 是一种迭代方法，用于从一组数据中估计模型参数。它通过随机选择数据点子集并拟合模型，然后评估所有数据点与模型的吻合程度，从而识别出liers并找到最佳模型参数。RANSAC广泛应用于图像配准、立体匹配和模型拟合等任务中。

- Triangulation



- ■ 给出新加入的 image，新的对应关系可以被三角测量重建出来
■ 在 COLMAP 中，有一种具有很强鲁棒性的三角测量的方法也被应用，从而解决异常值
- Bundle Adjustment & Outlier Filtering
 - 相对于所有的相机和 3D 点，最小化重投影误差如下

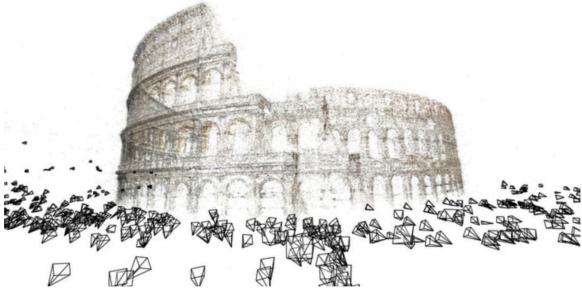
- \$\$

$$\sum_{i=1}^N \sum_{p=1}^{P_i} w_{ip} \| x_{ip} - \pi_i(x_p) \|^2_2$$

- 由于增量式 *Structure from Motion* 只会影响到局部模型，因此 *COLMAP* 方法！

3.4.2 Results and Applications

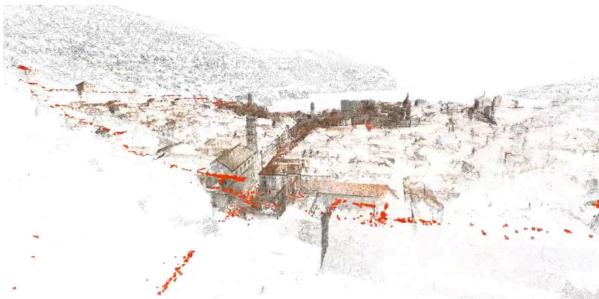
- Building Rome in a Day



- 首个大尺度 SfM: Building Rome in a Day

Agarwal, Snavely, Simon, Seitz and Szeliski: Building Rome in a day.
ICCV, 2009.

- COLMAP SfM



- COLMAP 在之前的基础上，大幅度增加了精确度和鲁棒性

Schonberger and Frahm: Structure-from-Motion Revisited. CVPR, 2016.

- COLMAP MVS



- COLMAP 也有第二个多视点立体阶段，用于获取密集的几何信息

Schonberger, Zheng, Frahm and Marc Pollefeys: Pixelwise View Selection for Unstructured Multi-View Stereo. ECCV, 2016

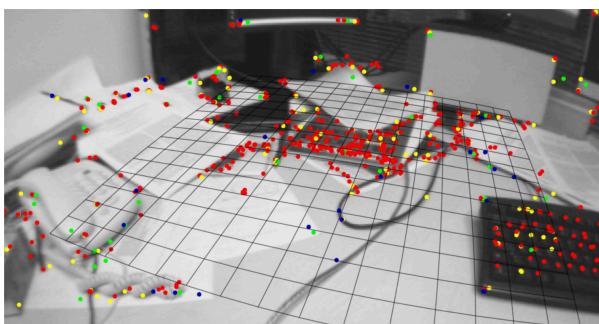
- Photo Tourism



- Photo Tourism / PhotoSynth 允许从三维中检索图片集合

Snavely, Seitz and Szeliski: Photo tourism: exploring photo collections in 3D. SIGGRAPH, 2006.

- Parallel Tracking and Mapping (PTAM)



- PTAM 展示了实时追踪和小的工作空间的映射

Klein and Murray: Parallel Tracking and Mapping for Small AR Workspaces. ISMAR, 2007.

- Match Move



- 2 分 3 秒 的自动相机运动追踪器 Boujou 在 2002 年获得了艾美奖

https://www.youtube.com/watch?v=JRL8_OvLcpw